UNIVERSITÉ DE LORRAINE

AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4
Code de la Propriété Intellectuelle. articles L 335.2- L 335.10
http://www.cfcopies.com/V2/leg/leg_droi.php
http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm

Département de formation doctorale en informatique **École doctorale IAEM Lorraine**
UFR STMIA

Preuves par induction dans le calcul des séquents modulo

THÈSE

présentée et soutenue publiquement le 26 Octobre 2007

pour l'obtention du

Doctorat de l'université Henri Poincaré – Nancy 1

(spécialité informatique)

par

Fabrice Nahon

Composition du jury

Président :	Michaël RUSINOWITCH Directeur de Recherche
Rapporteurs :	Thérèse HARDIN Professeur Toby WALSH Professeur
Examinateurs :	Serge AUTEXIER Senior Researcher Adel BOUHOULA Maître de Conférences Adam CICHON Professeur Claude KIRCHNER Directeur de Recherche Hélène KIRCHNER Directeur de Recherche Michaël RUSINOWITCH Directeur de Recherche

Laboratoire Lorrain de Recherche en Informatique et ses Applications — UMR 7503

Mis en page avec la classe thloria.

Remerciements

Il est temps pour moi de remercier tous ceux qui, de près ou de loin, m'ont aidé à réaliser ce travail. Tout d'abord, Claude et Hélène Kirchner, qui ont étroitement encadré ce travail. Je garde un excellent souvenir des entretiens très réguliers et des échanges parfois très animés que nous avons eus ensemble. C'est sans doute grâce à ces nombreuses discussions, où j'ai toujours été écouté et compris, que j'ai pris peu à peu confiance en moi et que j'ai pu commencer à développer mes propres idées. Un long chemin a été parcouru depuis le jour où Claude m'a reçu la première fois dans son bureau et m'a expliqué l'addition dans les entiers de Peano ! En effet, je suis professeur (agrégé) de mathématiques dans l'enseignement secondaire, et il m'a d'abord fallu comprendre une copieuse littérature avant de pouvoir commencer un travail de recherche en informatique. Celui-ci a vraiment débuté quand Claude et Hélène m'ont présenté une ébauche de système de recherche de preuve. Je devais déterminer si on pouvait le corriger pour le rendre correct. J'ai trouvé la question claire et très motivante. C'est en y répondant que d'autres se sont posées au fur et à mesure, et que j'ai finalement écrit cette thèse.

Je tiens ensuite à remercier vivement Thérèse Hardin, mon premier rapporteur, pour l'intérêt qu'elle a immédiatement manifesté pour ce travail quand je l'ai rencontrée pour lui présenter le manuscrit. Ses remarques, aussi bien écrites qu'orales, ont été d'une part une aide précieuse pour rédiger la version finale du document, et contiennent d'autre part de nombreuses pistes pour prolonger la réflexion. Je suis aussi très reconnaissant à Toby Walsh d'avoir accepté, depuis l'Australie, d'être également rapporteur. Son regard sur le manuscrit m'a été très utile pour le préciser et parfois le corriger. J'ai par ailleurs été très honoré que mon jury ait compté autant de chercheurs de renom, dont les ouvrages et articles ont inspiré cette thèse. Leurs nombreuses questions et remarques au cours de la soutenance m'ont en outre bien renseigné sur les nouvelles questions qui se posent à la suite de mes recherches .

Je remercie aussi chaleureusement les membres de l'équipe protheo (notamment Eric Deplagne, dont la thèse est à l'origine de ce travail, et qui m'a également de nombreuses fois "dépanné" lorsque l'ordinateur ne réagissait pas comme je voulais), avec lesquels j'ai travaillé dans une ambiance très détendue, ainsi que Chantal Llorens, qui a beaucoup compté pour les questions d'ordre administratif. Et enfin, je ne dois pas oublier mes proches : mes parents, auxquels je me suis souvent confié et qui qui m'ont beaucoup soutenu tout au long de ce parcours, ainsi que Bettina, ma femme, qui a accepté que je consacre autant de mon temps libre à ce qu'elle appelle *mes formules* !

A Bettina, ma femme, à mes parents, et à mon frère Emmanuel, handicapé.

Table des matières

Introduction générale **xiii**

1 Nécessité de la récurrence . . . xiii

2 Les différents principes de récurrence . . . xvi

3 Récurrence et démonstration automatique . . . xvii

4 Motivations . . . xviii

5 Plan de la thèse . . . xx

Partie I Notions fondamentales **1**

Chapitre 1
Cadre général

1.1 Le paysage syntaxique . . . 6

1.1.1 Mots . . . 6

1.1.2 Termes finis . . . 6

1.1.3 Substitutions . . . 11

1.1.4 Propositions du premier ordre . . . 12

1.2 Déduction . . . 14

1.2.1 Logique du premier ordre . . . 14

1.2.2 Théories . . . 15

1.3 Le paysage sémantique . . . 17

1.3.1 Interprétation d'une signature fonctionnelle . . . 18

1.3.2 Les algèbres de termes . . . 18

1.3.3 Interprétation des symboles relationnels . . . 20

1.4 Satisfaction . . . 20

1.4.1 Conséquence sémantique . . . 20

1.4.2 Conséquence inductive . . . 22

Chapitre 2
Réécriture

2.1 Relation binaire sur un ensemble 26
2.2 Système de réécriture 28
2.2.1 Cas non conditionnel 28
2.2.2 Cas conditionnel 29
2.3 Spécification équationnelle 30
2.3.1 Un système de réécriture particulier 30
2.3.2 Pourquoi orienter les égalités ? 31
2.3.3 Unification 32
2.3.4 Paires critiques 33
2.3.5 La procédure de complétion de Knuth-Bendix 34
2.4 Complétude suffisante 34
2.5 Réécriture modulo un ensemble d'égalités 36
2.5.1 Relation binaire modulo un ensemble d'égalités 36
2.5.2 Réécriture $\mathcal{R}, E$ et complétude suffisante 37

Chapitre 3
Preuves automatiques par récurrence

3.1 Preuves par consistance 42
3.2 Preuves par récurrence explicite 46
3.3 Preuves par récurrence implicite 50
3.4 Récurrence et déduction modulo 53
3.4.1 Introduction 53
3.4.2 Le mécanisme de la déduction modulo 54
3.4.3 Récurrence dans le contexte de la déduction modulo 56
3.5 Descente infinie 58

Partie II Ordre et surréduction : deux outils principaux pour la méthode 61

Chapitre 1
Ordres et quasi-ordres

1.1 Ordre sur les termes et terminaison des systèmes de réécriture 63
1.2 Ordres et quasi-ordres sur les chemins 66
1.3 Ordres et quasi-ordres sur les égalités 67

1.4 Ordres récursifs sur les chemins AC-compatibles 70
1.4.1 Termes variadiques 71
1.4.2 Construction d'un ordre récursif sur les chemins AC-compatible . . . 72

Chapitre 2
Surréduction

2.1 Principe 75
2.2 Surréduction modulo un ensemble d'égalités 76
2.3 Surréduction et complétude suffisante 78
2.3.1 Unificateurs constructeurs 78
2.3.2 Ensembles complets d'unificateurs constructeurs 79
2.3.3 Cas des théories associatives-commutatives 82

Partie III Un système de recherche de preuve par récurrence fondé sur la surréduction 85

Chapitre 1
Une première version

1.1 Le système de recherche de preuve IndNarrow 87
1.2 Un exemple simple 88
1.3 Correction 90
1.4 Correction réfutationnelle 96
1.5 Complétude réfutationnelle 98

Chapitre 2
Optimisation

2.1 Motivation 105
2.2 Règles d'inférence du système de recherche de preuve OptIndNarrow 107
2.3 L'exemple simple revisité 110
2.4 Justification théorique 112
2.4.1 Le système d'inférence DecIndNarrow 112
2.4.2 Labels 113
2.4.3 Correction et complétude réfutationnelle de OptIndNarrow 114
2.5 Exemple 119

Conclusion

Partie IV Extension au cas de la réécriture modulo **125**

Chapitre 1
Un système de recherche de preuve pour la récurrence modulo

1.1 Le système de recherche de preuve IndNarrowMod 127
1.2 Correction 128
1.3 Correction réfutationnelle 131
1.4 Complétude réfutationnelle 133
1.5 Exemple 133

Chapitre 2
Spécialisation au cas des théories associatives-commutatives ou associatives

2.1 Formes plates : étude plus approfondie 136
2.2 Les systèmes de recherche de preuve IndNarrowModAC et IndNarrowModA . . 140
2.3 Deux exemples simples 141
2.4 Correction et complétude réfutationnelle de IndNarrowModAC et IndNarrowModA 144
2.4.1 IndNarrowModAC et IndNarrowModA : deux instances de IndNarrowMod 145
2.5 Autres exemples 148
2.5.1 Exemples AC 148
2.5.2 Exemple A 153

Conclusion générale **155**

Annexes **159**

Annexe A Lemmes du calcul des séquents modulo **159**
A.1 Lemmes logiques 159
A.2 Lemmes ensemblistes 167
A.2.1 Ensembles finis 167
A.2.2 Définition et caractérisation de $\mathcal{T}(\Sigma)$ 168
A.3 Lemmes concernant les formes closes 168
A.4 Lemmes inductifs 171

Annexe B Théorème de compatibilité principale **179**

Annexe C Formes plates **189**

Index **195**

Bibliographie **199**

Résumé

Nous présentons une méthode originale de recherche de preuve par récurrence utilisant la surréduction. Elle a la particularité d'être fondée sur la déduction modulo et d'utiliser la surréduction pour sélectionner à la fois les variables de récurrence et les schémas d'instanciation. Elle donne également la possibilité de traduire directement toute dérivation effectuée avec succès en une preuve dans le calcul des séquents modulo. La correction et la complétude réfutationnelle de la méthode sont démontrées en théorie de la preuve.
Nous étendons ensuite cette première approche aux théories de réécriture équationnelles constituées d'un système de réécriture $\mathcal{R}$ et d'un ensemble E d'égalités. A partir du moment où le système de réécriture équationnel $(\mathcal{R}, E)$ possède de bonnes propriétés de terminaison et de complétude suffisante, et si on suppose également que E préserve les constructeurs, la surréduction au niveau des positions les plus profondes où apparaît un symbole défini s'effectue uniquement à l'aide d'unificateurs qui sont également des substitutions constructeurs. Ceci est particulièrement intéressant dans le cas des théories associatives, ou associatives commutatives, pour lesquelles notre système de recherche de preuve a été raffiné.

Mots clés : déduction modulo, calcul des séquents modulo, récurrence, récurrence noethérienne, récurrence par réécriture, raisonnement équationnel, réécriture de termes, réécriture équationnelle, surréduction équationnelle.

Abstract

We are presenting an original narrowing-based proof search method for inductive theorems. It has the specificity to be grounded on deduction modulo and to rely on narrowing to provide both induction variables and instantiation schemes. It also yields a direct translation from a successful proof search derivation to a proof in the sequent calculus. The method is shown to be correct and refutationally complete in a proof theoretical way.
We are extending this first approach to equational rewrite theories given by a rewrite system $\mathcal{R}$ and a set E of equalities. Whenever the equational rewrite system $(\mathcal{R}, E)$ has good properties of termination, sufficient completeness, and whenever E is constructor preserving, narrowing at defined-innermost positions is performed with unifiers which are constructor substitutions. This is especially interesting for associative and associative-commutative theories for which the general proof search system is refined.

Keywords : deduction modulo, sequent calculus modulo, induction, Noetherian induction, induction by rewriting, equational reasoning, term rewriting, equational rewriting, equational narrowing.

Introduction générale

1 Nécessité de la récurrence

Le premier chapitre de "la science et l'hypothèse" de Henri Poincaré commence ainsi : "la possibilité même de la science mathématique semble une contradiction insoluble. Si cette science n'est déductive qu'en apparence, d'où lui vient cette parfaite rigueur que personne ne songe à mettre en doute ? Si, au contraire, toutes les propositions qu'elle énonce peuvent se tirer les unes des autres par les règles de la logique formelle, comment la mathématique ne se réduit-elle pas à une immense tautologie ?" Dans le second cas, il va jusqu'à ajouter "qu'un esprit assez puissant pourrait d'un seul coup d'oeil en apercevoir toutes les vérités", à la façon d'un joueur d'échec, qui pourrait combiner autant de coups à l'avance qu'il le souhaiterait. Pour répondre à cette question, imaginons que l'on veuille concevoir un algorithme qui calcule la factorielle d'un entier donné en entrée. Sachant que $0! = 1$ et $n! = n(n-1)(n-2)\ldots 1$, le calcul de la factorielle d'un entier peut être représenté par les règles suivantes :

$$fact(0) \rightarrow 1$$
$$fact(n+1) \rightarrow (n+1) * fact(n)$$

La fonction $fact$ ainsi définie est récursive, car $fact(n+1)$ est calculé à partir de $fact(n)$. A partir de cet ensemble de règles, on peut maintenant concevoir l'algorithme suivant :

$fact(n : entier)$
- Si $n = 0$, $fact(n) = 1$
- sinon
  - $m = n$
  - $f = 1$
  - Pour tout $m >= 0$ faire
    - $f = m * f$
    - $m = m - 1$

  $fact(n) = f$

Il faut maintenant prouver que cet algorithme calcule effectivement la factorielle de l'entier donné en entrée. Calculons d'abord $fact(n)$ à l'aide de l'algorithme pour les premières valeurs de n et vérifions que l'on obtient toujours $n!$. L'algorithme donne $fact(0) = 1$, $fact(1) = 1 * fact(0) = 1 * 1 = 1$, $fact(2) = 2 * fact(1) = 2 * 1 * fact(0) = 2 * 1 * 1 = 2$. On a donc bien $fact(0) = 0!$, $fact(1) = 1!$, et $fact(2) = 2!$. Même si ces tests semblent indiquer que notre algorithme est correct, ils ne garantissent pas que celui-ci calcule $n!$ *pour toute* valeur de n. Pour s'assurer que $fact(n) = n!$ pour toute valeur de n, il faudrait effectuer une infinité de vérifications, ce qui reviendrait à "franchir un abîme que la patience de l'analyste, réduit aux seules ressources de la logique formelle, ne parviendra jamais à combler", en reprenant les termes de Poincaré. En effet, Poincaré considère toute vérification de ce type comme "stérile parce-que la conclusion n'est que la traduction des prémisses dans un autre langage". Pour lui,

"la démonstration véritable est féconde au contraire parce-que la conclusion y est en un sens plus générale que les prémisses". Pour montrer que notre algorithme est correct, il est donc nécessaire d'effectuer une démonstration "véritable", autrement dit un raisonnement *par récurrence*. Selon Poincaré, "le caractère essentiel du raisonnement par récurrence c'est qu'il contient, condensés pour ainsi dire en une formule unique, une infinité de syllogismes disposés en cascade". Pour avoir une compréhension intuitive du principe de récurrence, on peut imaginer une rangée de dominos. Lorsqu'on pousse un domino de la rangée, il fait tomber le domino suivant, qui fait tomber celui d'après et ainsi de suite. Pour faire tomber tous les dominos, il suffit donc de pousser le premier. Avec notre exemple, imaginons que montrer l'égalité $fact(n) = n!$ revienne à pousser le $n^{\text{ième}}$ domino de la rangée. Pour montrer l'égalité $fact(n) = n!$ *pour toute valeur de* n, il suffit donc de pousser le premier domino de la rangée, autrement dit de démontrer l'égalité $fact(0) = 0!$. Cela vient du fait que lorsqu'un domino tombe, il fait tomber le domino suivant, ce qui signifie, avec notre analogie, que lorsqu'on a effectué la preuve de l'égalité $fact(n) = n!$ pour un entier n quelconque, on en déduit celle de l'égalité $fact(n+1) = n+1!$. Dans la situation de notre exemple, montrer par récurrence que la valeur $fact(n)$ retournée par notre algorithme est bien $n!$ revient ainsi à prouver les propositions 1 et 2 suivantes :

1. $fact(0) = 0!$
2. Si $fact(n) = n!$, alors $fact(n + 1) = (n + 1)!$

La proposition 1, appelée cas de base, se vérifie au moyen d'un simple calcul. C'est dans la preuve de la proposition 2, appelée cas de récurrence, que l'on fait intervenir le fait que la fonction $fact$ est *récursive* : en effet, $fact(n + 1)$ est exprimé en fonction de $fact(n)$, car $fact(n + 1) = (n + 1) * fact(n)$. Mais on a fait une hypothèse sur $fact(n)$, qui est $fact(n) = n!$, que l' on va pouvoir utiliser dans le calcul de $fact(n + 1)$, en remplaçant $fact(n)$ par $n!$ dans l'égalité $fact(n+1) = (n+1) * fact(n)$. On aura ainsi $fact(n+1) = (n+1) * n!$, et on sera tiré d'affaire, car $(n+1) * n!$ peut encore se mettre sous la forme $(n+1)!$. Pourquoi l'égalité $fact(n) = n!$ nous semble ainsi démontrée "avec une irrésistible évidence" pour toute valeur de n ? Pour Poincaré, c'est parce-que que ce jugement n'est que "l'affirmation de la puissance de l'esprit qui se sait capable de concevoir la répétition indéfinie d'un même acte dès que cet acte est une fois possible". En effet, en mathématiques, tout comme en sciences physiques, des conjectures peuvent être *induites* à partir d'observations. Il est d'ailleurs intéressant de noter que le raisonnement par récurrence est appelé raisonnement *par induction* dans les pays anglo-saxons. Poincaré souligne cette analogie, mais également la différence "essentielle", en écrivant que "l'induction appliquée aux sciences physiques est toujours incertaine, parce-qu'elle repose sur la croyance à un ordre général de l'Univers, ordre qui est en dehors de nous. L'induction mathématique, c'est à dire la démonstration par récurrence, s'impose au contraire nécessairement, parce-qu'elle n'est qu'une affirmation d'une propriété de l'esprit lui-même." Il répond alors à sa question introductive en faisant observer qu'un joueur d'échecs ne peut jamais combiner qu'un nombre fini de coups, et que "s'il applique ses facultés à l'arithmétique, il ne pourra en apercevoir les vérités générales d'une seule intuition directe ; pour parvenir au plus petit théorème, il ne pourra s'affranchir de l'aide du raisonnement par récurrence parce-que c'est un instrument qui permet de passer du fini à l'infini".

Le *principe de récurrence simple* peut ainsi être énoncé comme suit :

"Si une proposition est telle que si elle est vraie du nombre n on peut en déduire qu'elle est vraie du nombre $n + 1$, on en conclut qu'elle est vraie *pour tous les nombres entiers* à partir du moment où on l'aura vérifiée pour $n = 0$." En langage mathématique, ce principe peut s'énoncer comme suit :

$$\forall P\ (P(0) \wedge (\forall n\ P(n) \Rightarrow P(n + 1))) \Rightarrow \forall n\ P(n)$$

ou sous la forme d'une règle :

$$\frac{P(0) \wedge \forall n\ P(n) \Rightarrow P(n+1)}{\forall n\ P(n)}$$

Cette phrase mathématique n'est pas une formule de la logique des prédicats du premier ordre, mais du deuxième ordre : en effet, le premier quantificateur n'est pas appliqué à un individu de la logique du premier ordre, mais à une propriété P d'entiers. Ainsi les objets du principe de récurrence ne sont plus les individus eux-mêmes (les nombres naturels), mais des propriétés sur ces entiers.
Le mathématicien italien G.Peano a introduit une définition des entiers naturels (qui deviendra standard) destinée à contourner les ambiguïtés inhérentes au langage naturel. L'essence de cette définition est un procédé de construction qui est un même temps un procédé de preuve. Son but était de mieux cerner la nature logique du principe de récurrence, c'est à dire la logique du deuxième ordre. Le principe de récurrence de Peano est un mode d'emploi pour la construction d'une échelle de Jacob. Il consiste en un élément de base ("une proposition P est valable pour zéro") et une instruction permettant de poser sur un segment d'échelle déjà existant l'échelon suivant ("si P est valide pour le nombre x, elle l'est aussi pour son successeur $s(x)$"). On allonge l'échelle à volonté (on démontre la validité de la proposition P pour toute valeur de x) en réitérant cette instruction. En langage mathématique, ce principe peut s'énoncer comme suit :

$$\forall P\ (P(0) \wedge (\forall x\ P(x) \Rightarrow P(s(x)))) \Rightarrow \forall x\ P(x)$$

On peut observer que le principe de récurrence de Peano est une reformulation du principe de récurrence simple : il suffit pour s'en apercevoir de construire les entiers naturels à partir des règles

$$0 \leftarrow \emptyset$$
$$s(x) \leftarrow x$$

L'intérêt essentiel de ce principe est qu'on peut le généraliser.
Considérons par exemple les listes d'entiers naturels. Si on pose l'existence d'une liste vide notée nil, on peut construire toute liste d'entiers naturels à partir des règles suivantes :

$$nil \leftarrow \emptyset$$
$$n :: l \leftarrow l$$

On pourra ainsi obtenir la liste vide notée nil, la liste (1) notée $1 :: nil$, la liste $(3, 1)$ notée $3 :: (1 :: nil)$ etc ... Contrairement aux entiers naturels, une liste l n'aura plus un seul, mais une infinité de successeurs : ce seront toutes les listes du type $n :: l$. On pourra alors énoncer un principe de récurrence analogue à celui de Peano, mais qui s'applique au cas de propositions définies sur des listes d'entiers naturels : Il consiste en l' élément de base ("une proposition P est valable pour la liste vide nil") et une instruction ("si P est valide pour la liste l, elle l'est aussi pour tout successeur $n :: l$"). Introduisons par exemple l'opérateur $<>$ défini récursivement par :

$$< nil, l' >= l'$$
$$< n :: l, l' >= n ::< l, l' >$$

Le principe de récurrence sur les listes permet de montrer que $<>$ est associatif, i.e que l'on a $< l, < l', l'' >>=<< l, l' >, l'' >$.
On peut donc se rendre compte que le principe de récurrence de Peano et le principe de récurrence sur les listes énoncé ci-dessus sont des cas particuliers d'un principe plus général

appelé “principe de récurrence structurelle”, dans lequel la récurrence s’effectue sur les règles qui caractérisent la formation des éléments d’un type de donnée. Nous verrons plus loin comment le langage des termes permet de formaliser ce principe.

2 Les différents principes de récurrence

Dans “la science et l’hypothèse”, Poincaré écrit que les mathématiques peuvent “continuer leur marche en avant” grâce au raisonnement par récurrence que l’on retrouve “à chaque pas”, soit sous la forme simple que nous venons de lui donner”(i.e le *principe de récurrence simple* défini au paragraphe 1),“soit sous une forme plus ou moins modifiée”. C’est pour cette raison qu’il le qualifie de “raisonnement mathématique par excellence”. Nous avons déjà mis en évidence au paragraphe 1 quelques unes de ces “formes”, et nous nous proposons ici d’en faire l’inventaire complet. Nous associons à chaque principe de récurrence défini ci-dessous un exemple de proposition qui se démontre à l’aide de ce principe.

- Récurrence simple :
  On peut l’écrire sous la forme de la règle d’inférence suivante :

$$\frac{P(0) \wedge \forall n\ P(n) \Rightarrow P(n+1)}{\forall n\ P(n)}$$

  Exemple 0.1 Pour tout entier naturel $n \geq 3$, $2^n \geq 1 + 2n$
- Récurrence complète.
  Le principe est le suivant :

$$\frac{P(0) \wedge \forall n\ (\forall m\ m < n \Rightarrow P(m)) \Rightarrow P(n)}{\forall n\ P(n)}$$

  Exemple 0.2 Tout entier ≥ 2 se décompose en produit de facteurs premiers.
- Récurrence structurelle.
  On peut définir un principe de récurrence structurelle pour chaque type de donnée. Citons-en deux (nous reprenons les notations et les définitions introduites au paragraphe 1) :
  - Récurrence structurelle sur les entiers de Peano :
    Considérons la définition inductive des entiers de Peano :

$$\begin{array}{l} 0 : Nat \\ s : Nat \rightarrow Nat \end{array}$$

    Cette définition implique le principe de récurrence suivant :

$$\frac{P(0) \wedge \forall x : Nat\ (P(x) \Rightarrow P(s(x)))}{\forall x\ P(x)}$$

    Exemple 0.3 Définissons l’addition sur les entiers de P :

$$\begin{array}{l} 0 + 0 = 0 \\ x + s(y) = s(x + y) \end{array}$$

    Alors, $\forall x : Nat \quad 0 + x = x$

– Récurrence structurelle sur les listes :
Considérons la définition inductive des listes d'entiers :

$$nil : list$$
$$:: : Nat \times list$$

Cette définition implique le principe de récurrence suivant :

$$\frac{P(nil) \wedge \forall l : list\ (P(l) \Rightarrow P(n :: l))}{\forall l : list\ P(l)}$$

Exemple 0.4 Considérons les axiomes suivants :

$$nil <> l = l$$
$$(n :: l_1) <> l_2 = n :: (l_1 <> l_2)$$

Alors, $\forall l_1 : list\ \forall l_2 : list\ \forall l_3 : list\ l_1 <> (l_2 <> l_3) = (l_1 <> l_2) <> l_3$.

– Récurrence sur les ordinaux.
Le principe de récurrence sur les ordinaux est le suivant :

$$\frac{\forall \alpha\ (\forall \beta\ \beta < \alpha \Rightarrow P(\beta)) \Rightarrow P(\alpha)}{\forall \alpha\ P(\alpha)}$$

Supposons que l'on définisse une opération d'addition sur les ordinaux de la façon sivante :

$$\alpha + \emptyset = \alpha$$
$$\alpha + Succ(\beta) = Succ(\alpha + \beta)$$
$$\lambda \in Lim \Rightarrow \alpha + \lambda = \bigcup_{\beta \in \lambda} (\alpha + \beta)$$

Alors, si α et β sont dénombrables, $\alpha + \beta$ l'est également.

– Récurrence noethérienne.
Si $\prec$ est une relation noethérienne sur un ensemble τ, et P est une proposition à une variable libre $x \in \tau$, le *principe de récurrence bien fondé* ou *principe de récurrence noethérien* s'énonce comme suit :

$$\frac{\forall t\ (\forall x\ x \prec t \Rightarrow P(x)) \Rightarrow P(t)}{\forall t\ P(t)}$$

Exemple 0.5 [HUE 80], [NEW 42] Toute relation noethérienne et localement confluente est confluente.

3 Récurrence et démonstration automatique

Avec l'émergence de l'informatique, le principe de récurrence a acquis une importance accrue, car il est très fréquemment utilisé en démonstration automatique. En effet, la récurrence est un outil essentiel pour démontrer des propositions concernant des données ou des fonctions définies récursivement. Elle est par exemple très fréquemment utilisée pour résoudre les problèmes de vérification liés aux protocoles de sécurité des systèmes embarqués.

Globalement, on peut distinguer deux approches. Dans les assistants de preuve comme COQ, ELF,HOL, Isabelle, Larch, NQTHM, PVS, ... l'axiome de récurrence est utilisé de façon explicite : l'utilisateur humain ou une tactique intelligente doit déterminer les hypothèses de récurrence les mieux adaptées à la résolution d'un problème donné. Les démonstrateurs de

théorèmes automatiques, eux, utilisent des méthodes spécifiques pour démontrer des propositions (inductives) de façon autonome. Les plus élaborées d'entre elles sont fondées sur les techniques de réécriture et de saturation. Elles sont appelées respectivement récurrence par réécriture et preuves par consistance. Les systèmes qui utilisent ces méthodes sont Spike, RRL ou INKA. Ces méthodes sont étudiées depuis la fin des années 70 et ont démontré leurs capacités sur beaucoup d'exemples pratiques concernant de simples spécifications algébriques et des plus compliquées comme le tour de cartes de Gilbreath. E.Deplagne [DEP 02] a proposé un cadre théorique qui donne la possibilité à ces deux approches de collaborer pour construire une preuve : les étapes de calcul sont effectuées sans interaction avec l'utilisateur et, dans ce cas, cela correspond à de la récurrence implicite, mais l'utilisateur a la possibilité de contrôler explicitement les étapes de déduction. Cette démarche qui consiste à faire une distinction entre les étapes de calcul et les étapes de déduction dans la construction d'une preuve a été présentée dans [DOW 03a] sous le nom de *déduction modulo* : il n'est donc pas étonnant que l'approche proposée dans [DEP 02] repose sur elle. Cette présentation de la logique du premier ordre ressemble au calcul des séquents classique à la différence près que le raisonnement s'effectue modulo une congruence définie sur les termes et les propositions. Pour pouvoir exprimer l'axiome de récurrence, qui est par essence un axiome du second ordre, E.Deplagne a eu recours à la représentation du premier ordre de la logique d'ordre supérieur élaborée dans [DOW 98]. Ainsi, en utilisant la déduction modulo, passer de la récurrence explicite à la récurrence implicite se fait naturellement et revient à enrichir la congruence avec les hypothèses de récurrence, puis à appliquer les méthodes de raisonnement automatique standard pour simplifier le but à prouver.

4 Motivations

Notre travail a donc d'abord été motivé par le besoin de mieux comprendre la relation entre récurrence explicite et récurrence implicite. Dans ce contexte, nous proposons un mécanisme de recherche de preuve par récurrence du type de celui qui a été décrit à la fin du paragraphe 3. Nous montrons comment l'étape de récurrence peut être réalisée par surréduction au niveau des positions les plus profondes où apparaissent un symbole défini (nous les appelons positions *définie-maximales*) quand la théorie est axiomatisée par un système de réécriture terminant sur les termes clos et suffisamment complet par rapport à un ensemble de constructeurs libres. Cela nous permet de préciser la relation entre la démarche suivie par les prouveurs automatiques de théorèmes fondés sur la réécriture comme Spike ou RRL et celle suivie par les assistants de preuve comme Coq ou PVS. Notre système est composé de règles d'inférences qui sont justifiées par le calcul des séquents modulo. Cela donne la possibilité de construire un terme de preuve pour un assistant, et donc de justifier formellement le résultat obtenu. Ainsi, en partant de la proposition (inductive) à démontrer, le mécanisme construit une preuve dans le calcul des séquents modulo, à partir duquel un terme de preuve peut-être calculé si nécessaire.
Nous avons ensuite optimisé notre système dans le but, d'une part de favoriser les conditions d'application des hypothèses de récurrence, et d'autre part d'aider à empêcher la production de successions infinies de réécriture.
Bien que déjà relativement expressive, notre première approche a été conçue pour des théories que l'on peut définir à l'aide de règles de réécriture. Elle est par conséquent limitée par le fait que des axiomes (comme la commutativité par exemple) ne peuvent pas être orientés sans perdre la propriété de terminaison du système de réécriture sous-jacent. Le système de recherche de preuve développé dans [DEP 02] peut d'ailleurs également échouer quand un axiome ne peut pas être orienté.

La solution consiste alors à utiliser la réécriture équationelle (aussi nommée réécriture modulo) comme explorée par [PET 81] et [JOU 86b], et à étendre notre approche dans le but de réaliser des preuves par récurrence dans des théories contenant de tels axiomes non orientables. On pourrait également comparer cette extension aux techniques utilisées pour effectuer des preuves par récurrence implicite modulo l'associativité-commutativité, comme décrites dans [BER 96b, BER 96a, BER 97]. Cependant, notre méthode est essentiellement différente, et l'exemple suivant est utile pour comparer notre approche et la méthode développée dans [BER 95, BER 97]. Considérons la spécification donnée à la figure 1, et supposons que l'on veuille démontrer la

- **Sortes :** nat ;
- **constructeurs :** $0 : \ \to nat \qquad s : \ nat \to nat$
- **fonctions définies :** $+ : \ nat \times nat \to nat \qquad * : \ nat \times nat \to nat$
- **règles** :

$$\begin{array}{llllll} x+0 & \to x & x*0 & \to 0 & exp(x,0) & \to s(0) \\ x+s(y) & \to s(x+y) & x*s(y) & \to x*y+x & exp(x,s(y)) & \to x*exp(x,y) \end{array}$$

FIG. 1 – **Arithmétique simple**

proposition $\forall x, y, n \ exp(x*y,n) = exp(x,n)*exp(y,n)$, où l'on suppose également que $+$ et $*$ sont associatifs-commutatifs(AC). La méthode développée dans [BER 95, BER 97] est fondée sur les *schémas de récurrence*. Plus précisément, elle détermine un sous-ensemble des variables du but, les *variables de récurrence*, et un ensemble de termes, l'*ensemble test*. Les variables de récurrence sont remplacées par des éléments de l'ensemble test, et cela engendre de nouvelles conjectures qui sont simplifiées par les règles de réécriture de la spécification ou par des instances plus petites de la conjecture initiale (les hypothèses de récurrence). La preuve se termine quand toutes ces conjectures nouvellement engendrées sont simplifiées en égalités triviales ou en théorèmes inductifs déjà connus. Il existe des algorithmes pour déterminer les variables de récurrence et les ensembles tests. Dans l'exemple ci-dessus, les variables de récurrence sont ainsi x, y, et n, et l'ensemble test est $\{0, s(x)\}$. Donc, une *instance test* est $exp(s(x')*s(y'), s(n')) = exp(s(x'), s(n'))*exp(s(y'), s(n'))$. Cette dernière égalité peut être réduite à l'aide de règles de la spécification en $s(x')*s(y')*exp(s(x'+y'+x'*y'), n') = exp(s(x'), n')*exp(s(y'), n')$, qui ne peut cependant *pas* être simplifiée par l'hypothèse de récurrence, et la tentative de recherche de preuve échoue par conséquent. On peut contourner cette difficulté si on restreint l'ensemble des variables de récurrence. C'est pour cette raison que [BER 95, BER 97] a introduit une heuristique dans le but de choisir les *bonnes* variables de récurrence, inspirée par le fonctionnement du système Nqthm-ACL2. En appliquant cette stratégie à l'exemple ci-dessus, seule la variable n est instanciée, et la recherche de preuve s'effectue avec succès. Cependant, la méthode ne reste plus réfutationellement complète lorsqu'on se permet d'utiliser une telle heuristique. Dans notre approche en revanche, l'étape de récurrence est réalisée par surréduction au niveau de positions *définie-maximales*, quand la théorie est axiomatisée par un système de réécriture équationnel suffisamment complet et terminant sur les termes clos. Plus précisément, il suffit de réaliser la surréduction au niveau de l'*une* de ces positions. Dans la situation de notre exemple, ces positions sont celles des sous-termes $x*y$, $exp(x,n)$ et $exp(y,n)$. Maintenant, du fait que $*$ est commutatif, le but reste équivalent si on permute les variables x et y. Il reste donc deux possibilités : surréduire à la position où le symbole $*$ apparaît, où une position où c'est le symbole *exp* qui apparaît. On préfère la seconde, car cela va permettre de créer davantage de réductions ultérieures, et on décide donc d'effectuer la surréduction au niveau du

sous-terme $exp(x, n)$. Après normalisation, on obtient le sous-but trivial $s(0) = s(0)$, et le sous-but $x * y * exp(x * y, n) = x * y * exp(x, n) * exp(y, n)$ auquel on peut appliquer l'hypothèse de récurrence. Il est important de souligner que le fait d'utiliser la stratégie décrite ci-dessus pour choisir la position d'application de la surréduction n'empêche pas notre méthode de rester réfutationellement complète (si on suppose en outre que la spécification possède de *bonnes* propriétés : plus précisément, c'est le cas lorsque, un système de réécriture $\mathcal{R}$ et un ensemble d'égalités E étant donnés, on suppose entre autres que $\mathcal{R}$ est terminant modulo E, que la relation de réécriture $\mathcal{R}, E$ de Peterson et Stickel [PET 81, JOU 86b] est suffisamment complète, et que *E préserve les termes constructeurs* (la confluence n'étant pas requise pour montrer la correction de la méthode). En outre, sous ces conditions, la surréduction au niveau d'une position *définie-maximale* s'effectue à l'aide d'unificateurs qui sont également des substitutions constructeurs. On peut ainsi contourner de sérieuses difficultés, liées à la cardinalité des ensembles complets d'unificateurs. Il devient ainsi possible d'effectuer des preuves par récurrence modulo des théories non finitaires, comme l'associativité par exemple.

5 Plan de la thèse

Ce document est organisé en quatre parties.

- Dans la première partie, nous présentons les notions fondamentales que nous utiliserons tout au long de la thèse.
  - Dans le premier chapitre, nous définissons les concepts de base qui constituent le cadre de notre travail : termes, algèbres, propositions. Nous effectuons en particulier une mise au point sur les notions de conséquence sémantique et de conséquence inductive.
  - Le deuxième chapitre est consacré à la réécriture et aux spécifications équationnelles.
  - Le chapitre qui suit est dédié aux preuves automatiques par récurrence. Nous avons regroupé celles-ci en quatre catégories : les preuves par consistance, les preuves par récurrence explicite, les preuves par récurrence implicite, et les preuves par descente infinie. Nous expliquons également en détail comment le raisonnement par récurrence peut être formalisé en déduction modulo, car c'est sur cette approche que se fonde tout notre travail.
- La seconde partie présente deux outils essentiels à notre méthode, qui sont les ordres et la surréduction.
  - Dans le premier chapitre, nous effectuons des rappels généraux concernant les ordres sur les termes, et notamment les ordres récursifs sur les chemins, que l'on utilisera constamment. Nous définissons également des ordres sur les égalités, et nous démontrons le théorème de compatibilité principale, dont l'importance sera révélée à la partie III.
  - Dans le deuxième chapitre, nous définissons la surréduction et nous rappelons le lemme classique de relevage [HUL 80a, KIR 99] qui fait le lien entre étape de réduction et étape de surréduction, et sur lequel repose la correction de notre système. Nous introduisons en outre le concept clé d'ensemble complet d'unificateurs constructeurs, dont on a déjà souligné l'importance plus haut.
- La troisième partie présente notre système de recherche de preuve par récurrence fondé sur la surréduction.
  - Au premier chapitre, nous définissons les règles d'inférence, et nous étudions un exemple pour illustrer leur fonctionnement. Nous montrons ensuite comment la déduction modulo permet de justifier la correction de ce système, ainsi que sa complétude réfutationnelle sous certaines hypothèses.

  - Dans le chapitre suivant, nous introduisons des modifications dans l'intention, d'une part, de faciliter l'usage de l'hypothèse de récurrence, et d'autre part, de limiter les risques d'application indéfinie des règles d'inférence. Nous commençons ainsi par expliquer nos motivations, puis nous présentons les règles de notre système optimisé dont nous détaillons le fonctionnement dans différentes situations et à l'aide d'un exemple. Nous démontrons ensuite la correction de ce nouveau système, et c'est ici que le théorème de compatibilité principale énoncé à la partie II joue pleinement son rôle.
- Dans la quatrième partie, nous étendons notre système au cas de la réécriture modulo un ensemble E d'égalités.
  - Au premier chapitre, nous définissons des règles d'inférence qui s'appliquent au cas où E est quelconque. Les preuves de correction et de complétude réfutationnelle de ce nouveau système sont largement similaires à celles de la partie III.
  - Le deuxième chapitre est consacré aux cas des théories associatives commutatives ou associatives uniquement. Nous commençons par introduire deux systèmes pour chacune de ces deux théories, et qui travaillent sur les formes plates. Nous illustrons ensuite leur fonctionnement par l'étude de deux exemples. Nous démontrons ensuite que ces deux systèmes sont en fait des instances de celui présenté au premier chapitre, ce qui leur permet d'en hériter toutes les propriétés. Ils ont en outre l'intérêt d'être beaucoup plus pratiques à utiliser. La fin de ce chapitre est dédiée à l'étude de nombreux exemples.

Première partie

Notions fondamentales

Table des figures

1 **Arithmétique simple** . . . xix

1.1 Terme $\bullet(ei(\bullet(ix)e))$ . . . 10
1.2 Le système LK : les règles structurelles . . . 14
1.3 Le système LK : les règles logiques . . . 15
1.4 Σ-morphisme $\hat{\sigma}$ . . . 19
1.5 Interprétation d'une proposition . . . 20
1.6 Factorisation d'un homomorphisme . . . 23
1.7 Factorisation de $i_{\mathcal{H}}$ . . . 23

2.1 Définition 2.31 . . . 38

3.1 Système d'inférence de Comon et Nieuwenhuis . . . 45
3.2 Système d'inférence de Aoto . . . 53
3.3 Le calcul des séquents modulo . . . 55

1.1 Le système de recherche de preuve IndNarrow . . . 89

2.1 OptIndNarrow(première partie) . . . 107
2.2 OptIndNarrow(seconde partie) . . . 108
2.3 Règles spécifiques de DecIndNarrow . . . 113

1.1 Le système de recherche de preuve IndNarrowMod . . . 128
1.2 **liste** . . . 133

2.1 Le système de recherche de preuve IndNarrowModAC . . . 141
2.2 Le système de recherche de preuve IndNarrowModA . . . 142
2.3 **Binôme** . . . 149
2.4 Système d'inférence étendu de Aoto . . . 150

1

Cadre général

Sommaire

1.1 Le paysage syntaxique **6**
1.1.1 Mots 6
1.1.2 Termes finis 6
1.1.3 Substitutions 11
1.1.4 Propositions du premier ordre 12
1.2 Déduction **14**
1.2.1 Logique du premier ordre 14
1.2.2 Théories 15
1.3 Le paysage sémantique **17**
1.3.1 Interprétation d'une signature fonctionnelle 18
1.3.2 Les algèbres de termes 18
1.3.3 Interprétation des symboles relationnels 20
1.4 Satisfaction **20**
1.4.1 Conséquence sémantique 20
1.4.2 Conséquence inductive 22

Nous commençons dans ce chapitre par introduire les objets syntaxiques de base constituant le cadre de notre travail. Lorsqu'on effectue un raisonnement, on travaille toujours à l'intérieur d'une théorie qui exprime les "connaissances communes", que ce soit en mathématiques, en programmation, ou en intelligence artificielle. Un bon moyen pour définir une théorie est de l'engendrer à partir d'un ensemble de formules et d'une relation de déduction. Nous fairons ainsi une mise au point sur la relation de déduction $\vdash$ de la logique classique. Nous effectueront ensuite des rappels sur la notion d'*algèbre*, qui est un formalisme adapté à la description de chacun des objets syntaxiques que l'on aura introduits auparavant. A partir de cette notion d'algèbre, on définit celle de modèle. On peut alors distinguer la relation de *conséquence sémantique* et la relation de *conséquence inductive*. Nous verrons que la relation de *conséquence sémantique* s'identifie avec la relation de déduction $\vdash$ de la logique classique. La relation de *conséquence inductive* est par définition plus faible que celle de *conséquence sémantique*, et nous verrons au chapitre 3 qu'elle est à la base de la recherche automatique de preuve par récurrence. Nous expliquerons finalement l'intérêt des *algèbres quotients* pour caractériser les conséquences inductives. Nous nous sommes inspirés pour faire cette présentation de [LAL 90] et [KIR].

1.1 Le paysage syntaxique

1.1.1 Mots

Les mots sont les objets ordinaires du paysage synataxique de l'informatique, où ils sont connus sous le nom de "chaînes de caractères".

Définition 1.1 Soit A un ensemble, appelé "alphabet", et dont les éléments sont appelés "symboles". Un "mot sur A est une suite finie d'éléments de A. L'ensemble des mots sur A, qui est noté A^*, est donc

$$A^* = \bigcup_{n \geqslant 0} A^n$$

Si $u \in A^n$, $\mid u \mid$ est la *longueur* de u. A^* est muni d'une opération binaire, la *concaténation*, notée par un . :

$$\begin{array}{ccc} A^p \times A^q & \rightarrow & A^{p+q} \\ (u, v) & \rightarrow & u.v \end{array}$$

avec $u.v = (u_1, \ldots, u_p, v_1, \ldots, v_q)$. Cette opération est associative et admet comme élément neutre le mot vide ε, qui est l'unique élément de A^0 : $(A^*, ., \varepsilon)$ est donc un monoïde, appelé *monoïde libre* engendré par A. Ces définitions étant données, il faut introduire une notation pratique pour ces mots. On remarque que si on note simplement a la suite à un élément (a), alors u=$(u_1, \ldots, u_n) \in A^n$ est le résultat de la concaténation des symboles le composant : $u = u_1.u_2.\ldots.u_n$. C'est cette dernière notation qui sera utilisée désormais.

1.1.2 Termes finis

La structure des mots, ou plutôt celle du monoïde libre, est très pauvre : tous les symboles d'un mot sont de même nature. Au contraire, les termes présentés ici sont dotés d'une structure qui différencie les symboles.

Signatures

Dans le cas des expressions arithmétiques, il y a des opérateurs $(+, \times)$, des variables de constantes. La structure de ces expressions tient au caractère fonctionnel des symboles dont chacun requiert un nombre fixe d'arguments, appelé *arité*. Dans un langage de programmation, on pourra rencontrer :

- des symboles de constante : **true**, **256**
- des symboles unaires :**not** _
- des symboles binaires :_ **or** _ , **while** _ **do** _ , _ < _
- des symboles ternaires :**if** _ **then** _ **else** _

(le _ indique la place d'un argument).
L'ensemble de ces symboles constitue la signature du langage.

Définition 1.2 Une signature est un ensemble gradué i.e, un ensemble Σ muni d'une application $ar : \Sigma \rightarrow \mathbb{N}$. Si $f \in \Sigma$ et $ar(f) = n$, on dit que f est d'arité n ou que f est $n - aire$; si $n \geq 1$, f est un *symbole fonctionnel*, et si $n = 0$, c'est un *symbole de constante*.

$\mathcal{T}(\Sigma)$ est la plus petite partie E du monoïde libre Σ^* telle que

1. si $c \in \Sigma$, $ar(c) = 0$, alors $c \in E$
2. si $f \in \Sigma$, $ar(f) = n \geq 1$, et si $t_1, \ldots, t_n \in E$, alors $ft_1 \ldots t_n \in E$

Les éléments de $\mathcal{T}(\Sigma)$ sont appelés des $\Sigma-$*termes finis*, ou des termes du premier ordre, ou plus simplement, des termes.

Exemple 1.1 Soit $\Sigma = \{0, s, +\}$ avec $ar(0) = 0$, $ar(s) = 1$, et $ar(+) = 2$. Alors 0, $s0$, $+00$, $+s00 \in \mathcal{T}(\Sigma)$, mais $+0$, $s00 \notin \mathcal{T}(\Sigma)$.

Nous voulions des objets plus structurés que les mots, mais nous n'avons obtenu que des mots : en fait, la structure est dans la forme même de la définition, pas dans les objets qu'elle introduit. Le paragraphe suivant éclaircit cette observation.

Définitions inductives

Nous avons donné des termes une définition à partir du monoïde libre, ce qui a permis, en passant, de leur octroyer une définition par des mots. $\mathcal{T}(\Sigma)$ étant ainsi défini, on peut valider à l'intérieur de Σ^*, un procédé de preuve par *récurrence sur les termes*, qui n'est autre que celui par récurrence structurelle dont nous avons fait mention dans l'introduction, et que l'on peut maintenant formaliser ici.

Proposition 1.1 Soit P une propriété des mots. Si elle est vraie pour chaque symbole de constante de Σ et si pour chaque $f \in \Sigma$, d'arité n, on sait prouver qu'elle est vraie pour le mot $ft_1 \dots t_n$ si elle est vraie pour les mots $t_1, \dots, t_n$, alors P est vraie pour tous les termes.

Preuve. Soit E la partie de Σ^* formée des mots qui vérifient P. E contient les constantes et les $ft_1 \dots t_n$ dès qu'elle contient $t_1, \dots, t_n$. Par définition de $\mathcal{T}(\Sigma)$, on en déduit $\mathcal{T}(\Sigma) \subseteq E$, i.e, tous les termes clos vérifient P. $\square$

Ce procédé est souvent utilisé pour des définitions, ainsi, la profondeur $|t|$ d'un terme t est défini par induction structurelle par

$$\begin{array}{rcl} |a| & = & 0 \\ |ft_1 \dots t_n| & = & max(|t_1|, \dots, |t_n|) + 1 \end{array}$$

pour tout symbole de constante a et tout symbole f d'arité $n \geq 1$.
Une définition inductive peut aussi être présentée comme un système d'inférence, construction que nous rencontrerons plusieurs fois par la suite. Un *système d'inférence* est la donnée de règles d'inférence portant sur des *jugements* que l'on souhaite prouver. Elle se présente sous la forme

$$\frac{j_1 \dots j_n}{j} \text{ nom}$$

où $j_1, \dots, j_n$ sont des jugements, appelés *prémisses*, j est un jugement appelé *conclusion* ; *nom* est le nom de la règle et l'entier $n \geq 0$ son arité. Elle exprime l'inférence de la conclusion à partir des prémisses. Si $n = 0$, la règle exprime l'assertion de sa conclusion j, qui est appelée *axiome* du système d'inférence ; la barre peut alors être omise.
L'exemple le plus simple de système d'inférence est donné par les règles de formation d'une définition inductive. Dans le cas des termes, on s'intéresse au jugement "terme t" exprimant que t est un terme. La signature Σ détermine les règles suivantes, avec une règle ($n-$aire) par symbole ($n-$aire) :

$$\frac{}{\text{terme } c} \ (c) \qquad \frac{\text{terme } t_1 \dots \text{terme } t_n}{\text{terme } ft_1 \dots t_n} \ (f)$$

où c, $f \in \Sigma$ et $ar(c) = 0$, $ar(f) = n$. Ces règles ont la particularité d'êtres déterministes ; leurs prémisses sont uniquement déterminées par leur conclusion, ce qui est propre aux définitions

par induction structurelle.

Remarque: On notera indifféremment $ft_1 \ldots t_n$ ou $f(t_1, \ldots, t_n)$.

Admettons pour l'instant que le t de "terme t" puisse désigner un mot quelconque de Σ^*. Ces règles servent à prouver qu'un mot est un terme ; la preuve est organisée sous la forme d'un arbre ayant la racine "en bas". Soit le mot $+0s0$; comme 0 est un terme, on en infère d'une part que $s0$ est un terme, et d'autre part avec ce résultat intermédiaire, que $+0s0$ est un terme ; on présente ce raisonnement par un nouvel objet syntaxique appelé *dérivation* :

$$\dfrac{\dfrac{}{\text{terme } 0}\,(0) \quad \dfrac{\dfrac{}{\text{terme } 0}\,(0)}{\text{terme } s0}\,(s)}{\text{terme } +0s0}\,(+)$$

C'est un arbre dont chaque noeud est étiqueté par un jugement et par une règle d'inférence (dans le cas présent, par un terme et par un symbole de la signature). Le jugement figurant à la racine est la *conclusion* de la dérivation. Les règles figurant aux feuilles sont d'arité 0 (ici, l'introduction d'un symbole de constante). Si $t \in \Sigma^*$, alors $t \in \mathcal{T}(\Sigma)$ si et seulement s'il existe une dérivation de ce système d'inférence dont la conclusion est "terme t".

Structures de données

Les termes sont les objets les plus importants du calcul symbolique, et de la programmation fonctionnelle et logique. Leur structure abstraite favorise un style de programmation "algébrique" remarquablement concis et clair. On manipule généralement à la fois plusieurs sortes (ou types) d'objets, par exemple les entiers, les chaînes, les atomes, les listes, ... ; une opération d'arité donnée ne doit pas s'appliquer à n'importe quelle sorte d'objets (qu'est-ce que le successeur d'une liste ?). On va donc restreindre les règles de formation des termes à l'aide de déclarations de types. Un ensemble S (de "sortes") étant donné, nous commençons par définir un ensemble $S-$sorté comme une famille d'ensembles indexés par S. On pourra ainsi attribuer une *sorte* s de S à tout élément de la famille.

Définition 1.3 Un ensemble (de sortes) S étant donné, on appelle *ensemble S-sorté* toute famille $A = (A_s \mid s \in S)$ d'ensembles indexés par S.

Au lieu d'écrire $x \in A_s$, on dit que x est un élément de sorte s, ce que l'on note $x : s$.
On généralise ensuite le concept de signature en la définissant comme un ensemble d'opérateurs agissant sur un ensemble $S-$ sorté. On attribue alors à chaque opérateur f de cette signature un *profil*, qui indique d'une part les sortes des données que celui-ci accepte en entrée et dans quel ordre, et d'autre part la sorte des données qu'il renvoie.

Définition 1.4 Une signature S-sortée est un ensemble Σ dont les éléments sont appelés *symboles d'opération*, auxquels on a adjoint une fonction $profil$, qui associe à chaque élément f de Σ un objet $s_1 \times \ldots \times s_n \to s$, tel que $s_1 \times \ldots \times s_n$ soit un produit fini d'éléments de S, et s soit un élément de S.

Remarque:
- Un élément $f \in \Sigma$ de profil $s_1 \times \ldots \times s_n \to s$ se note $f : s_1 \times \ldots \times s_n \to s$.
- Le produit $s_1 \times \ldots \times s_n$ exprime quelles sortes de données f accepte en entrée et dans quel ordre, et s exprime la sorte des données qu'il renvoie.

- Si $n = 0$, le profil $s_1 \times \ldots \times s_n \to s$ se note simplement s.
- La définition 1.2 est un cas particulier de la définition 1.4 avec un ensemble de sortes S réduit à un seul élément.

Un moyen simple et visuel de représenter une signature est d'énumérer d'abord les sortes après le mot clef *sortes*, puis les symboles d'opérations (déclarés avec leurs profils) après le mot clef *opérations*.

Exemple 1.2 [WEC 92] La signature (mono-sortée) d'un groupe algébrique a la représentation suivante :
- Sortes : $groupe$
- Opérations :
  - $e : groupe$
  - $i : groupe \to groupe$
  - $\bullet : groupe \times groupe \to groupe$

Exemple 1.3 La signature (multi-sortée) des listes d'entiers naturels a la représentation suivante :
- Sortes : nat, $list$
- Opérations :
  - $0 : nat$
  - $s : nat \to nat$
  - $nil : list$
  - $:: : nat \times list \to list$

Variables

Les sont omniprésentes, sous des formes diverses, tant en logique qu'en programmation. En mathématiques, un rôle important des variables est de formuler des équations. On introduit donc un ensemble $\mathcal{X}$ de symboles de variables :

Définition 1.5 Considérons une signature $S-$sortée Σ, un ensemble de variables $\mathcal{X}$ pour Σ est un ensemble $S-$sorté disjoint de Σ, tel que, pour tout $s \in S$, $\mathcal{X}_s$ sera l'ensemble des variables de sorte s.

On étend la fonction d'arité ar à $\Sigma \cup \mathcal{X}$, avec la valeur 0 sur $\mathcal{X}$. Notons que jusqu'à présent rien ne distingue une variable d'une constante.

Formation des termes

Les règles de formation des termes sont données par le système d'inférence suivant, portant sur les jugements 'terme $t : s$', formé d'une règle $n-$aire par symbole d'arité n :

$$\frac{}{\text{terme } c : s}\ (c) \qquad\qquad \frac{}{\text{terme } x : s}\ (x)$$

$$\frac{\text{terme } t_1 : s_1 \ldots \ \text{terme } t_n : s_n}{\text{terme } f t_1 \ldots t_n : s}\ (f)$$

où $c,\ f \in \Sigma\ x \in \mathcal{X}$ et $c : s$, $x : s$, $f : s_1 \ldots s_n \to s$. On dira ainsi que t est un terme de sorte s s'il existe une dérivation fermée par 'terme : s', ce qui revient à introduire la définition suivante :

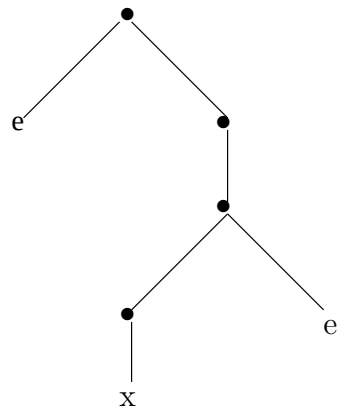

FIG. 1.1 – Terme $\bullet(ei(\bullet(ix)e))$

Définition 1.6 – Pour tout $s \in S$, on appelle ensemble des Σ-termes sur $\mathcal{X}$ de sorte s, et on note $\mathcal{T}(\Sigma, \mathcal{X})_s$, le plus petit ensemble tel que $\mathcal{X}_s \subset \mathcal{T}(\Sigma, \mathcal{X})_s$ et, pour tout $n \in \mathbb{N}$, pour tout $(s_1, \ldots, s_n) \in S^n$, pour tout symbole d'opération $f : s_1 \times \ldots \times s_n \to s \in \Sigma$ et tout $(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{X})_{s_1} \times \ldots \times \mathcal{T}(\Sigma, \mathcal{X})_{s_n}$, $ft_1 \ldots t_n \in \mathcal{T}(\Sigma, \mathcal{X})_s$.

– $\mathcal{T}(\Sigma, \mathcal{X}) = (\mathcal{T}(\Sigma, \mathcal{X})_s \mid s \in S)$

L'ensemble des termes sans variables, appelés *termes clos*, est noté $\mathcal{T}(\Sigma)$. Des termes qui contiennent des variables sont dits *ouverts*. Un terme est *linéaire* si chacune de ses variables n'apparaît qu'une fois.

Exemple 1.4 Dans le cas de l'exemple 1.3, les règles de formation des termes sont :

1. 0, $s0$, $ss0$ sont des termes de sorte *nat*
2. *nil* est un terme de sorte *list*
3. Si n est un terme de sorte *nat*, et si l est un terme de sorte *list*, alors $n :: l$ est un terme de sorte *list*.

Ainsi $ss0 :: nil$ est un terme, mais pas $0 :: s0$.

Termes et arbres

Prenons l'exemple du terme $t = \bullet(ei(\bullet(ix)e))$ construit sur la signature de l'exemple 1.2 (x est une variable de sorte *groupe*). On peut le représenter par l'arbre de la figure 1.1. Numérotons les arcs issus de chaque noeud de gauche à droite, et à partir de 1. On accèdera à un symbole du terme à l'aide du mot obtenu en concaténant les numéros des arcs de la branche menant de la racine à ce symbole : ε, 2.1 accèdent à $\bullet$, 2, 2.1.1 accèdent à i, 1, 2.1.2 accèdent à e. Ces mots sont appelés des *occurences* (ou *positions*) ; ce sont des éléments de A^*, où A est l'ensemble des entiers $\geqslant 1$. On utilisera l'ordre préfixe sur les occurences : $\omega \leqslant \omega'$ exprime que ω est un ancêtre de ω'. La donnée d'une occurence ω du terme détermine un noeud de l'arbre, le symbole qui l'étiquette et le sous-terme (sous-arbre) qui en est issu. Le symbole situé à l'occurence ε est ainsi appelé *symbole de tête*.

Définition 1.7 Pour tout terme t, on définit :

– l'ensemble $\mathcal{D}om(t)$ des occurences de t appelé *domaine* de t ;
– le symbole $t(\omega)$ en ω, pour $\omega \in \mathcal{D}om(t)$;
– le sous-terme $t_{|\omega}$ de t en ω, pour $\omega \in \mathcal{D}om(t)$;

par

1. si $t = c \in \Sigma$ ou $t = x \in \mathcal{X}$, alors $\mathcal{D}om(t) = \{\varepsilon\}$, $t(\varepsilon) = t$, $t_{|\varepsilon} = t$;
2. si $t = ft_1 \dots t_n$, alors :
   (a) $\mathcal{D}om(t) = \{\varepsilon\} \cup \bigcup_{i=1}^{n} i.\mathcal{D}om(t_i)$;
   (b) $t(\varepsilon) = f$ et $t(i.\omega) = t_i(\omega)$;
   (c) $t_{|\varepsilon} = t$ et $t_{|i.\omega} = t_{i|\omega}$.

La *profondeur* $|t|$ du terme t est la longueur de la plus grande chaîne de l'arbre qui représente t et $\mathcal{V}ar(t)$ désigne l'ensemble des variables de t.

Exemple 1.5 Le terme $t = \bullet(ei(\bullet(ix)e)$ construit sur la signature de l'exemple 1.2 a pour profondeur 4 et $\mathcal{V}ar(t) = \{x\}$.

L'opération complémentaire de la construction du sous-terme est la *greffe* (ou *remplacement*), dont voici une définition récursive.

Définition 1.8 Soient $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ et $\omega \in \mathcal{D}om(t)$. Si $s = fs_1 \dots s_n$, avec $n \geqslant 0$, on pose :
– $s[t]_\varepsilon = t$.
– $s[t]_{i.\omega} = fs_1 \dots s_{i-1}s_i[t]_\omega s_{i+1} \dots s_n$.

La notation $s[t]_\omega$ peut également souligner que le terme s contient le terme t comme sous-terme à la position ω. Dans certains cas, on peut simplement écrire $s[t]$.

1.1.3 Substitutions

Pour faire jouer aux variables pleinement leur rôle, on introduit l'opération de substitution.

Définition 1.9 On appelle *substitution* toute application $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$, qui est l'identité presque partout (i.e sauf sur une partie finie de $\mathcal{X}$). L'ensemble $\mathcal{D}om(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$ qui est donc fini, est appelé le *domaine* de σ, et l'ensemble $Im(\sigma) = \{\sigma(x) \mid x \in \mathcal{D}om(\sigma)\}$ est appelé l'*image* de σ.

On fait agir σ sur $\mathcal{T}(\Sigma, \mathcal{X})$ par :

1. $c\sigma = c$ pour $c \in \Sigma$, $ar(c) = 0$.
2. $(ft_1 \dots t_n)\sigma = f(t_1\sigma) \dots (t_n\sigma)$ si $f \in \Sigma$, $ar(f) = n$.
3. $x\sigma = \sigma(x)$ si $x \in \mathcal{X}$.

Cette action étend σ en une application de $\mathcal{T}(\Sigma, \mathcal{X})$ dans lui-même encore notée σ. La clause 3 remplace chaque variable par un terme correspondant et la clause 2 propage cette transformation dans le terme. L'interprétation d'une substitution sur un arbre est simple : $t\sigma$ s'obtient en greffant le terme $\sigma(x_i)$ à chaque occurence de x_i dans t. Dans la suite, on notera $x\sigma$ et non plus $\sigma(x)$ le résultat de l'application de σ à la variable x. On appelle *instance* de t par σ le résultat de l'application de la substitution σ au terme t. Si $\mathcal{D}om(\sigma) = \{x_1, \dots, x_n\}$, tous les x_i étant distincts, et si $x_i\sigma = t_i$, alors σ peut être représentée par l'ensemble des couples variable-terme noté $\{x_1 \mapsto t_1, \dots, x_n \mapsto t_n\}$. Si on pose $\overrightarrow{x} = (x_1, \dots, x_n)$ et $\overrightarrow{t} = (t_1, \dots, t_n)$, on désignera alors par $s\{\overrightarrow{t}/\overrightarrow{x}\}$ l'application de σ à un terme s (dans le cas où $n = 1$, on écrira simplement $s\{t/x\}$).

Exemple 1.6 L'instance du terme $t = g(x, g(a, x))$ par $\sigma = \{x \mapsto f(z, a)\}$ est le terme $t\sigma = g(f(z, a), g(a, f(z, a)))$

Notation: On désigne en général les substitutions par des lettres grecques α, β, γ, σ ...

Définition 1.10 L'ensemble des variables introduites par une substitution σ est appelé son *rang*, et il est défini par :

$$\mathcal{R}an(\sigma) = \bigcup_{x \in \mathcal{D}om(\sigma)} \mathcal{V}ar(x\sigma)$$

Si $\mathcal{R}an(\sigma) = \emptyset$, σ est appelée substitution close. Si $t\sigma$ est une instance de t qui appartient à $\mathcal{T}(\Sigma)$, σ est appelée *instanciation close* de t. La substitution de domaine vide est notée Id : $these.tex, v1.432008/01/1720$: 30 : $01nahonExp$. On désigne par $\sigma_{|V}$ la *restriction* de la substitution σ au sous-ensemble V de $\mathcal{X}$, définie de la façon suivante :

$$\begin{aligned} x\sigma_{|V} &= x\sigma \quad \text{si } x \in V; \\ &= x \quad \text{sinon.} \end{aligned}$$

On note la *composition* des substitutions α et β par la simple juxtaposition $\alpha\beta$ et on la définit par $x\alpha\beta = (x\alpha)\beta$. L'ensemble de toutes les substitutions sur $\mathcal{T}(\Sigma, \mathcal{X})$ est noté $Subst^{\Sigma,\mathcal{X}}$, et l'ensemble des substitutions closes par $Subst^{\Sigma}$.
La composition disjointe de deux substitutions σ et ρ peut aussi être définie quand leurs domaines respectifs sont disjoints ($\mathcal{D}om(\sigma) \cap \mathcal{D}om(\rho) = \emptyset$) :

$$\begin{aligned} x(\sigma + \rho) &= x\sigma \text{ si } x \in \mathcal{D}om(\sigma); \\ &= x\rho \text{ si } x \in \mathcal{D}om(\rho). \end{aligned}$$

Les substitutions *idempotentes* sont importantes car elles possèdent des propriétés utiles qui rendent la définition de concepts plus simples et certaines preuves plus faciles. On verra en particulier que pour l'*unification* on peut se restreindre à des unificateurs idempotents.

Définition 1.11 Une substitution σ est *idempotente* si $\sigma\sigma = \sigma$.

Définition 1.12 Soit σ une substitution $\{x_1 \mapsto y_1, \ldots, x_n \mapsto y_n\}$ avec $y_1, \ldots, y_n$ des variables distinctes. Alors σ est une *permutation* si $\mathcal{R}an(\sigma) = \mathcal{D}om(\sigma)$, et un renommage si $\mathcal{R}an(\sigma) \cap \mathcal{D}om(\sigma) = \emptyset$.

1.1.4 Propositions du premier ordre

Signature

Les langages du premier ordre ont été introduits par Frege en 1879. Leur usage est aujourd'hui largement répandu en mathématiques et en informatique. La signature d'un langage du premier ordre est l'ensemble Σ des symboles de fonction et de relation qui figurent dans ses formules. Plus précisément, on définit une signature du calcul des prédicats comme la donnée de deux signatures disjointes Σ_f et Σ_r, dites respectivement *signature fonctionnelle* et *signature relationnelle*. On notera $\Sigma = \Sigma_f \cup \Sigma_r$. Les éléments de Σ_f (resp Σ_r) sont des symboles de fonction (resp de relation).

Exemple 1.7 La signature du langage de l'arithmétique est $\{0, s, +, \times, <, =\}$ où 0, s, $+$, $\times$ sont des symboles fonctionnels, et $=$, $<$ sont des symboles de relation binaires.

On considère d'abord un ensemble infini dénombrable de variables $\mathcal{X}$ et on forme l'ensemble des termes finis sur la signature Σ_f, que l'on notera de nouveau $\mathcal{T}(\Sigma, \mathcal{X})$. La construction de $\mathcal{T}(\Sigma, \mathcal{X})$ donne un sens fonctionnel aux éléments de Σ_f, il reste à donner un sens relationnel aux éléments de Σ_r.

Formation des propositions

Les *formules atomiques* (ou *atomes*) sont les $Rt_1 \dots t_n$ avec $R \in \Sigma_r$ d'arité n et $t_1, \dots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$: elles expriment une relation entre termes.

Exemple 1.8 $x = x$, et $x > (y \times 0)$ sont des atomes.

On note $At(\Sigma, \mathcal{X})$ l'ensemble des atomes.
Les formules sont construites à partir des atomes à l'aide des connecteurs $\wedge$ ("et"), $\vee$ ("ou"), $\Rightarrow$ ("implique"), $\perp$ ("faux") et du quantificateur universel $\forall$ ("quelque soit") et existentiel $\exists$ ("il existe").

Définition 1.13 L'ensemble $\mathcal{L}(\Sigma, \mathcal{X})$ des formules du calcul des prédicats du premier ordre sur la signature Σ est la plus petite partie E du monoïde libre $(\Sigma \cup \mathcal{X} \cup \{\wedge, \vee, \Rightarrow, \perp, \forall, \exists\})*$ telle que :

1. $At(\Sigma, \mathcal{X}) \subseteq E$;
2. $\perp \in E$, et si $P,\ P' \in E$, $P \wedge P'$, $P \vee P'$, $P \Rightarrow P' \in E$;
3. Si $x \in \mathcal{X}$, $P \in E$, alors $\forall x\ P$ et $\exists x\ P \in E$.

Remarque: On peut également désigner par $\forall (x, y)\ P$ la proposition $\forall x\ \forall y\ P$. Le couple (x, y) est un *vecteur* de variables.

Exemple 1.9 $\forall x\ x = x$, $(x > y \times x)$ sont des formules, mais $x = \perp$, $\forall\ x \times x > 0$ n'en sont pas.

La construction de $\mathcal{L}(\Sigma, \mathcal{X})$ a seulement donné aux connecteurs et quantificateurs un sens fonctionnel : pour l'instant, leur sens logique n'apparaît pas.

Variables libres d'une proposition

Définition 1.14 L'ensemble $\mathcal{V}ar(P)$ des *variables libres* d'une formule est défini par récurrence par :

$$\begin{aligned}
&\mathcal{V}ar(Rt_1 \dots t_n) = \bigcup_{i=1}^{n} \mathcal{V}ar(t_i);\\
&\mathcal{V}ar(\perp) = \emptyset;\\
&\mathcal{V}ar(P \wedge P') = \mathcal{V}ar(P \vee P') = \mathcal{V}ar(P \Rightarrow P') = \mathcal{V}ar(P) \vee \mathcal{V}ar(P');\\
&\mathcal{V}ar(\forall x\ P) = \mathcal{V}ar(\exists x\ P) = \mathcal{V}ar(P) \setminus \{x\}.
\end{aligned}$$

Toute occurence de la variable x est dite *liée* dans $\forall x\ P$ et $\exists x\ P$. Un quantificateur *lie* une variable. Rappelons ici que parmi les conséquances de cette liaison, le nom d'une variable liée n'a pas vraiment de signification : $\forall x\ x > 0$ et $\forall y\ y > 0$ ont le même sens. Comme l'écriture d'une formule ne contient qu'un nombre fini de variables, le fait d'imposer $\mathcal{X}$ infini permet la convention de renommage : la formule obtenue à partir d'une formule P, en remplaçant une variable liée par une nouvelle variable n'apparaissant pas dans P, est considérée comme égale à P.

Exemple 1.10 Dans $\forall x\ \exists y\ x \times y = z$, y peut être renommée en u, mais pas en x ou z.

On peut vérifier que l'ensemble $\mathcal{V}ar(P)$ des *variables libres* de P défini ci-dessus n'est pas modifié par renommage des variables liées de P. Désormais, toute formule P sera considérée au "renommage près" des variables liées (relation d'équivalence implicite) et sera écrite en utilisant comme variables liées des variables choisies hors de $\mathcal{V}ar(P)$. On dira qu'une proposition P est *ouverte* si $\mathcal{V}ar(P) \neq \emptyset$. A partir de maintenant, on utilisera le symbole $=$ *exclusivement* pour représenter l'égalité syntaxique de deux termes, ou l'égalité de deux propositions au renommage près des variables liées.

Clauses

On introduit un nouveau connecteur, la *négation*, comme une abréviation : $\neg P$ est une abréviation de $P \Rightarrow \bot$. Un *littéral* est un atome (littéral positif) ou la négation d'un atome (littéral négatif). Une proposition est *clausale* si elle est de la forme $\forall \overrightarrow{x} P$, où P est une disjonction de littéraux, et $\overrightarrow{x}$ désigne le vecteur des variables libres de P (on les supposera toujours ordonnées linéairement par leur nom). On dira de plus que c'est une *clause de Horn* si elle contient au plus un littéral positif. Les clauses de Horn sont fréquemment utilisées en programmation logique.

1.2 Déduction

1.2.1 Logique du premier ordre

On peut formaliser le raisonnement tel qu'il est pratiqué depuis les Grecs, et en même temps, donner enfin leur sens usuel aux symboles logiques. Plusieurs types de systèmes d'inférence ont été élaborés dans ce but (on a déjà introduit la notion de système d'inférence au paragraphe 1.1.2). L'un d'entre eux est le *calcul des séquents LK* de Gentzen, adapté pour formaliser la logique dite *classique*. L'idée est d'introduire comme jugements des *séquents*, qui sont des listes de symboles de la forme

$$P_1, \ldots, P_m \vdash Q_1, \ldots, Q_n$$

exprimant que de P_1, P_2, ..., et P_m, on peut déduire Q_1, Q_2, ... ou Q_n ; comme cas particuliers

$$\vdash Q \qquad \text{et} \qquad P \vdash$$

expriment respectivement que Q peut être prouvée et que P est réfutable.
Les règles du calcul des séquents respectent une symétrie droite/gauche. Il y a :

- Des règles structurelles ;
- Des règles logiques ;
- l'axiome d'identité ;
- la règle de coupure.

Comme un séquent est une liste (pas un ensemble) de symboles, il faut tenir compte de leur ordre et des répétitions : c'est le rôle des *règles structurelles* de permutation, de contraction et d'affaiblissement. Les règles de permutation et d'affaiblissement sont souvent appliquées de façon implicite. Pour chaque connecteur ou quantificateur, il y a une règle d'introduction à gauche et

$$\frac{\Gamma, P, Q \vdash \Delta}{\Gamma, Q, P \vdash \Delta}\ \pi_l \qquad \frac{\Gamma, P, P \vdash \Delta}{\Gamma, P \vdash \Delta}\ contr_l \qquad \frac{\Gamma \vdash \Delta}{\Gamma, P \vdash \Delta}\ weak_l$$

$$\frac{\Gamma \vdash P, Q, \Delta}{\Gamma \vdash Q, P, \Delta}\ \pi_r \qquad \frac{\Gamma \vdash P, P, \Delta}{\Gamma \vdash P, \Delta}\ contr_r \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash P, \Delta}\ weak_r$$

FIG. 1.2 – Le système LK : les règles structurelles

une règle d'introduction à droite : ce sont les *règles logiques*. On utilise directement la négation $\neg$, au lieu de $\bot$. Une règle d'arité 0 (axiome) est nécessaire : l'*identité*.

$$\frac{}{P \vdash P}\ Ax$$

$$\frac{\Gamma, P, Q \vdash \Delta}{\Gamma, P \wedge Q \vdash \Delta}\ \wedge_l \qquad \frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash Q, \Delta}{\Gamma \vdash P \wedge Q, \Delta}\ \wedge_r$$

$$\frac{\Gamma, P \vdash \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \vee Q \vdash \Delta}\ \vee_l \qquad \frac{\Gamma \vdash P, Q, \Delta}{\Gamma \vdash P \vee Q, \Delta}\ \vee_r$$

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma, Q \vdash \Delta}{\Gamma, P \Rightarrow Q \vdash \Delta}\ \Rightarrow_l \qquad \frac{\Gamma, P \vdash Q, \Delta}{\Gamma \vdash P \Rightarrow Q, \Delta}\ \Rightarrow_r$$

$$\frac{\Gamma \vdash P, \Delta}{\Gamma, \neg P \vdash \Delta}\ \neg_l \qquad \frac{\Gamma, P \vdash \Delta}{\Gamma \vdash \neg P, \Delta}\ \neg_r$$

$$\frac{\Gamma, P\{t/x\} \vdash \Delta}{\Gamma, \forall x\ P \vdash \Delta}\ \forall_l \qquad \frac{\Gamma \vdash P\{y/x\}, \Delta}{\Gamma \vdash \forall x\ P, \Delta}\ \forall_r$$

$$\frac{\Gamma, P\{y/x\} \vdash \Delta}{\Gamma, \exists x\ P \vdash \Delta}\ \exists_l \qquad \frac{\Gamma \vdash P\{t/x\}, \Delta}{\Gamma \vdash \exists x\ P, \Delta}\ \exists_r$$

(Pour les règles $\forall_l$, $\forall_r$, $\exists_l$ et $\exists_r$, on suppose que $x \in \mathcal{V}ar(P)$ et $y \notin \mathcal{V}ar(\Gamma) \cup \mathcal{V}ar(\Delta)$)

Fig. 1.3 – Le système LK : les règles logiques

Exemple 1.11 [LAL 90] Une dérivation du séquent $\vdash ((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$: la tautologie de Pierce.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{P \vdash P}\ Ax}{P \vdash Q, P}\ w_r}{\vdash P \Rightarrow Q, P}\ \Rightarrow_l \qquad \dfrac{}{P \vdash P}\ Ax}{(P \Rightarrow Q) \Rightarrow P \vdash P, P}\ \Rightarrow_l}{(P \Rightarrow Q) \Rightarrow P \vdash P}\ contr_r}{\vdash ((P \Rightarrow Q)) \Rightarrow P) \Rightarrow P}\ \Rightarrow_r$$

La règle de coupure que nous introduisons maintenant permet de faire des démonstrations beaucoup plus intelligentes qu'avec les seules règles logiques.

$$\frac{\Gamma, P \vdash \Delta \qquad \Gamma \vdash P, \Delta}{\Gamma \vdash \Delta}\ cut$$

On remarquera qu'il n'est pas possible de reconstituer P à partir de la conclusion d'une coupure. Ceci a des conséquences significatives dans le domaine de la démonstration automatique : si on cherche à construire une dérivation d'un séquent utilisant des coupures, l'espace de recherche est infini. Cependant, malgré la grande souplesse que la règle de coupure introduit dans les démonstrations, celle-ci n'augmente pas la puissance de LK. Plus précisément, si une dérivation utilise des coupures,on peut la transformer en une dérivation n'utilisant pas de coupure. C'est le *Hauptsatz* (ou théorème d'utilisation des coupures) de Gentzen (1934).

1.2.2 Théories

Définition

Lorsqu'on effectue un raisonnement, on travaille toujours à l'intérieur d'une théorie qui exprime les "connaissances communes", que ce soit en mathématiques, en programmation, ou en intelligence artificielle. Une théorie est ainsi un ensemble de formules d'un langage du premier ordre. Pour le moment, à moins d'énumérer ses formules, un bon moyen pour définir une théorie est de l'engendrer par une relation de déduction $\vdash$. la relation de déduction $\vdash$ que nous choisirons

ici est celle de la logique classique, formalisée par le calcul des séquents de Gentzen présenté plus haut.

Définition 1.15 Etant donnée une partie Γ de $\mathcal{L}(\Sigma, \mathcal{X})$, dont les éléments seront appelés *axiomes* (de la théorie), la *théorie* engendrée par Γ est l'ensemble des propositions $P \in \mathcal{L}(\Sigma, \mathcal{X})$ telles que $\Gamma \vdash P$.

La théorie de l'égalité

On suppose que Σ contient un symbole de relation binaire $\approx$. L'ensemble des axiomes de l'égalité est le suivant :

- $\forall x\ x \approx x$;
- $\forall x\ \forall y\ x \approx y \Rightarrow y \approx x$;
- $\forall x\ \forall y\ \forall z\ x \approx y \wedge y \approx z \Rightarrow x \approx z$;
- $\forall x_1\ \ldots \forall x_n\ \forall y_1\ \ldots \forall y_n\ x_1 \approx y_1 \wedge \ldots \wedge x_n \approx y_n \Rightarrow f x_1 \ldots x_n \approx f y_1 \ldots y_n$;
- $\forall x_1\ \ldots \forall x_n\ \forall y_1\ \ldots \forall y_n\ x_1 \approx y_1 \wedge \ldots \wedge x_n \approx y_n \Rightarrow R x_1 \ldots x_n \Rightarrow R y_1 \ldots y_n$.

pour chaque symbole fonctionnel f, relationnel R de la signature Σ.
Une *clause équationnelle* sera alors une clause qui contiendra $\approx$ pour unique symbole relationnel. Si elle contient exactement un littéral positif et au moins un littéral négatif, on dira que c'est une *égalité conditionnelle*. On représentera une égalité conditionnelle $\neg e_1 \vee \ldots \vee \neg e_m \vee e'$ par $(e_1 \wedge \ldots \wedge e_m) \Rightarrow e'$. e' est alors la *conclusion*, et $(e_1 \wedge \ldots \wedge e_m)$ la *précondition*.

Remarque: $\mathcal{L}(\Sigma, \mathcal{X})$ est un langage du premier ordre car on dispose seulement de variables d'individus. On ne peut pas quantifier sur des symboles de fonction ou de relation, qui sont des "constantes" du langage. Dans un langage du second ordre, il y a en outre une infinité de variables de relation pour chaque arité, ces variables pouvant êtres quantifiées. Cela donne au langage une plus grande puissance d'expression : par exemple, l'égalité y est définissable par la proposition, due à Leibniz :

$$x \approx y \triangleq \forall P\ P(x) \Rightarrow P(y) \tag{1.1}$$

où P est une variable de relation unaire.

L'arithmétique de Peano

C'est l'une des grandes théories étudiée par les logiciens. La signature est la suivante :

- Sortes : *nat*, *bool*.
- Opérations :

$$\begin{array}{llll} 0: \ nat; & s: \ nat \to nat; & +: \ nat \times nat \to nat; & *: \ nat \times nat \to nat; \\ <: \ nat \times nat; & \approx: \ nat \times nat & & \end{array}$$

- Axiomes :

$$\begin{array}{l} \forall x\ \neg(sx \approx 0) \\ \forall x\ \forall y\ sx \approx sy \Rightarrow x \approx y \\ \forall x\ x + 0 \approx x \\ \forall x\ \forall y\ x + s(y) \approx s(x + y) \\ \forall x\ x * 0 \approx 0 \\ \forall x\ \forall y\ x * (sy) \approx x * y + x \\ \forall x\ \neg(x < 0) \\ \forall x\ \forall y\ x < sy \Leftrightarrow x < y \vee x \approx y \\ P(0) \wedge \forall x\ (P(x) \Rightarrow P(sx)) \Rightarrow \forall x\ P(x) \end{array}$$

La dernière ligne est le schéma d'axiomes de récurrence : il y a une infinité dénombrable de tels axiomes, un pour chaque proposition P et chaque variable libre de P. L'arithmétique de Peano n'est pas finiment axiomatisable, d'après un théorème de Ryll-Nardzewski.

Propriétés générales

La *consistance* est la première propriété à exiger d'une théorie

Définition 1.16 une partie Γ de $\mathcal{L}(\Sigma, \mathcal{X})$ (ou la théorie qu'elle engendre) est *consistante* si $\Gamma \nvdash \bot$, i.e, s'il n'existe pas de proposition $P \in \mathcal{L}(\Sigma, \mathcal{X})$ telle que $\Gamma \vdash P$ et $\Gamma \vdash \neg P$. Une théorie est *inconsistante* si elle n'est pas consistante.

La proposition suivante montre que l'inconsistance est directement reliée à la relation de déduction. La preuve est faite dans [LAL 90].

Proposition 1.2

1. $\Gamma \vdash P$ si, et seulement si $\Gamma \cup \{\neg P\}$ est inconsistant.
2. $\Gamma \vdash \neg P$ si, et seulement si $\Gamma \cup \{P\}$ est inconsistant.

La *complétude* est une propriété importante, assez forte, et qu'une "bonne" théorie n'a pas nécessairement.

Définition 1.17 Une théorie d'axiomes Γ est *complète* si, pour toute proposition fermée P, on a $\Gamma \vdash P$ ou bien $\Gamma \vdash \neg P$.

Exemple 1.12 [LAL 90] la théorie des corps n'est pas complète : elle ne permet ni de prouver $1 + 1 = 0$, ni $1 + 1 \neq 0$. Mais la théorie des corps algébriquement clos de caractéristique 0 est complète. L'arithmétique de Peano n'est pas complète : c'est le théorème d'incomplétude de Gödel.

La *décidabilité* est une propriété fondamentale, liée à la notion de calculabilité. Elle est au centre des problèmes théoriques de la démonstration automatique

Définition 1.18 Une théorie d'axiomes Γ est *décidable* si il existe un algorithme qui, une proposition P étant donnée, décide si $\Gamma \vdash P$ ou si $\Gamma \nvdash P$. Une théorie qui n'est pas décidable est dite *indécidable*

Exemple 1.13 L'arithmétique est indécidable. La théorie des corps algébriquement clos de caractéristique 0 est décidable.

1.3 Le paysage sémantique

Nous construisons des schémas d'interprétation de chacun des objets syntaxiques introduits en logique du premier ordre : signatures, variables, termes, propositions.
Une *algèbre multi-sortée* est un formalisme adapté à la description des types de données. Elle contient un ensemble d'opérateurs qui agissent sur ces types et des propriétés algébriques liant ces opérateurs. On obtient ainsi une modélisation abstraite indépendante d'une implantation particulière.

1.3.1 Interprétation d'une signature fonctionnelle

On suppose donné un ensemble S de sortes et une signature $S-$sortée Σ dont la définition a été donnée au paragraphe 1.1.2 (Définition 1.4) , et on interprète un symbole d'arité n comme une fonction de n arguments.

Définition 1.19 Une Σ-algèbre S-sortée $\mathcal{A}$ est constituée d'un ensemble S-sorté A, ainsi que d'une famille d'applications telle que, à chaque symbole de fonction $f : s_1 \times \ldots s_n \to s$ de Σ correspond l'une des applications de la famille $f^{\mathcal{A}} : A_{s_1} \times \ldots \times A_{s_n} \to A_s$. Quand il n'y a pas d'ambigüités, on parle simplement d'algèbre.

Remarque:
- le support de l'algèbre $\mathcal{A}$ est l'ensemble A
- L'algèbre $\mathcal{A} = (A, (f^{\mathcal{A}})_{f\in\Sigma})$ est appelée *interprétation* de Σ

Exemple 1.14 Un singleton $\{a\}$ est le support d'une *algèbre triviale*, chaque symbole de constante étant interprété par a, et les symboles de fonctions par des fonctions constantes de valeur a. Une interprétation triviale identifie tout.

Le mot *homomorphisme* vient du grec et signifie *même structure*. Cela est clairement apparent dans la définition suivante :

Définition 1.20 Soit Σ une signature, $\mathcal{A}$ et $\mathcal{B}$ deux Σ-algèbres. Un Σ-homomorphisme ϕ entre $\mathcal{A}$ et $\mathcal{B}$ est une famille $(\phi_s \mid s \in S)$ d'applications $\phi_s \ : \ A_s \to B_s$, telles que $\phi_s(f^{\mathcal{A}}(a_{s_1}, \ldots, a_{s_n})) = f^{\mathcal{B}}(\phi_{s_1}(a_{s_1}), \ldots, \phi_{s_n}(a_{s_n}))$ pour tout symbole d'opération $f : s_1 \times \ldots \times s_n \to s$ de Σ et tout $(a_{s_1}, \ldots, a_{s_n}) \in A^{s_1} \times \ldots \times A^{s_n}$.

Un homomorphisme bijectif est alors appelé *isomorphisme*.

1.3.2 Les algèbres de termes

Si $\mathcal{X}$ est un ensemble quelconque de variables, l'ensemble $\mathcal{T}(\Sigma, \mathcal{X})$ des termes est muni d'une structure canonique de Σ-algèbre $(\mathcal{T}(\Sigma, \mathcal{X}), \Sigma)$.

Définition 1.21 On associe à chaque symbole d'opération $f : s_1 \times \ldots s_n \to s$ l'opérateur $f^{\mathcal{T}}$ défini par $f^{\mathcal{T}}(t_1, \ldots, t_n) = ft_1 \ldots t_n$ pour tout $(t_1, \ldots, t_n) \in \mathcal{T}(\Sigma, \mathcal{X})_{s_1} \times \ldots \times \mathcal{T}(\Sigma, \mathcal{X})_{s_n}$. L'algèbre de support $\mathcal{T}(\Sigma, \mathcal{X})$ munie des opérateurs définis ci-dessus est appelée algèbre des Σ-*termes* sur $\mathcal{X}$, et on la note $(\mathcal{T}(\Sigma, \mathcal{X}), \Sigma)$.

Remarque: Si $\mathcal{X} = \emptyset$, l'algèbre obtenue est appelée algèbre des Σ-*termes clos*, et on la note $(\mathcal{T}(\Sigma), \Sigma)$ Cette Σ-algèbre est aussi un objet libre au sens de la théorie des catégories (comme il y a des monoïdes libres, des groupes libres, ...).

Définition 1.22 Pour toute Σ-algèbre $\mathcal{A}$, une assignation est une application de $\mathcal{X}$ vers le domaine A de $\mathcal{A}$ qui à un élément $x \in \mathcal{X}$ de sorte s associe un élément $a \in A$ de même sorte s.

Proposition 1.3 Pour toute Σ- algèbre $\mathcal{A}$ et toute assignation $\sigma : \mathcal{X} \to A$, il existe un unique Σ-homomorphisme $\hat{\sigma}$: $\mathcal{T}(\Sigma, \mathcal{X}) \to A$ tel que $\hat{\sigma}\ (x) = \sigma(x)$ pour tout $x \in \mathcal{X}$ (cf Figure 1.4)

$$\begin{tikzcd} \mathcal{X} \arrow[r, "\sigma"] \arrow[d] & A \\ \mathcal{T}(\Sigma, \mathcal{X}) \arrow[ur, dashed, "\hat{\sigma}"'] & \end{tikzcd}$$

FIG. 1.4 – Σ-morphisme $\hat{\sigma}$

$\hat{\sigma}$ est défini récursivement sur les termes :

$$\begin{aligned} &\hat{\sigma}\ (ft_1 \ldots t_n) = f^{\mathcal{A}}(\hat{\sigma}\ (t_1), \ldots, \hat{\sigma}\ (t_n)) \\ &\hat{\sigma}\ (x) = \sigma(x) \end{aligned}$$

On notera $[\![t]\!]^{\mathcal{A},\sigma}$ le résultat de l'application de $\hat{\sigma}$ au terme t, et on étendra à la section suivante cette notation aux propositions et aux théories (dans ce cas, on ne prend en compte que les variables libres).

La proposition 1.3 s'applique en particulier à $(\mathcal{T}(\Sigma), \Sigma)$, l'algèbre des termes clos. Dans ce cas, elle s'écrit plus simplement :

Corollaire 1.1 Pour toute Σ-algèbre $\mathcal{A}$, il existe un unique Σ-homorphisme $i_{\mathcal{A}} : \mathcal{T}(\Sigma) \to A$ qui est défini par :

$$i_{\mathcal{A}}(ft_1 \ldots t_n) = f^{\mathcal{A}}(i_{\mathcal{A}}(t_1), \ldots, i_{\mathcal{A}}(t_n))$$

On dit alors que l'homomorphisme $i_{\mathcal{A}}$ est l'*interprétation* associée à l'algèbre $\mathcal{A}$. A cause de ce corollaire, on dit que $\mathcal{T}(\Sigma)$ est la Σ-*algèbre initiale*. Plus généralement :

Définition 1.23 une algèbre $\mathcal{A}$ est dite *initiale* dans une classe d'algèbre si elle appartient à cette classe, et s'il existe un unique homomorphisme de $\mathcal{A}$ dans toute algèbre de la classe.

Lorsque le Σ-homorphisme $i_{\mathcal{A}}$ défini au corollaire 1.1 est surjectif, on dit que $\mathcal{A}$ est une *algèbre de Herbrand*.

Exemple 1.15 [WEC 92] Soit la signature suivante :
- Sorte : Nat.
- Opérations : $0 : nat \quad s : Nat \to Nat$.

Posons $\mathcal{A}$ l'algèbre de domaine $\mathbb{N}$ définie par $0^{\mathcal{A}} = 0$ et $s^{\mathcal{A}}(n) = (n+1)$ pour tout $n \in \mathbb{N}$. $\mathcal{A}$ est une Σ-algèbre de Herbrand.

Une autre application importante de cette propriété est avec $\mathcal{A} = (\mathcal{T}(\Sigma, \mathcal{X}), \mathcal{X})$. On obtient ainsi un unique homomorphisme $\hat{\sigma}$: $\mathcal{T}(\Sigma, \mathcal{X}) \to \mathcal{T}(\Sigma, \mathcal{X})$ prolongeant $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$. On appelle *domaine* ou *support* de $\sigma : \mathcal{X} \to \mathcal{T}(\Sigma, \mathcal{X})$ (ou de $\hat{\sigma}$), l'ensemble

$$\mathcal{D}om(\sigma) = \{x \in \mathcal{X} \mid \sigma(x) \neq x\}$$

On retrouve ainsi la notion de substitution introduite précedemment : si $\mathcal{D}om(\sigma)$ est fini, σ est une *substitution*.

Remarque:

$[\![Rt_1 \ldots t_n]\!]^{\mathcal{A},\sigma} = \mathbb{T}$ si et seulement si $([\![t_1]\!]^{\mathcal{A},\sigma}, \ldots, [\![t_n]\!]^{\mathcal{A},\sigma}) \in R^{\mathcal{A}}$

$[\![P \vee Q]\!]^{\mathcal{A},\sigma} = [\![P]\!]^{\mathcal{A},\sigma}$ ou $[\![Q]\!]^{\mathcal{A},\sigma}$

$[\![P \wedge Q]\!]^{\mathcal{A},\sigma} = [\![P]\!]^{\mathcal{A},\sigma}$ et $[\![Q]\!]^{\mathcal{A},\sigma}$

$[\![\neg P]\!]^{\mathcal{A},\sigma} =$ non $[\![P]\!]^{\mathcal{A},\sigma}$

$[\![\exists x\ P]\!]^{\mathcal{A},\sigma} = \mathbb{T}$ si et seulement si $\exists \phi\ [\![P]\!]^{\mathcal{A},\phi} = \mathbb{T}$ avec $\phi(y) = \sigma(y)$ si $y \neq x$

$[\![\forall x\ P]\!]^{\mathcal{A},\sigma} = \mathbb{T}$ si et seulement si $\forall \phi\ [\![P]\!]^{\mathcal{A},\phi} = \mathbb{T}$ avec $\phi(y) = \sigma(y)$ si $y \neq x$

FIG. 1.5 – Interprétation d'une proposition

- A partir de maintenant, on désignera par la même lettre une assignation et son prolongement à l'ensemble des termes.
- Comme dans le cas des substitutions, $t\sigma$ désignera l'application d'une assignation σ à un terme t.

1.3.3 Interprétation des symboles relationnels

Passons à la logique du premier ordre. Il reste à interpréter les symboles relationnels de la signature par des relations. On suppose donc que $\Sigma = \Sigma_f \cup \Sigma_r$ est une signature du calcul des prédicats dont la signification a été précisée au paragraphe 1.1.4. On affecte à chaque symbole relationnel un profil de la forme $s_1 \times \ldots \times s_n$. Le produit $s_1 \times \ldots \times s_n$ indique quelles sortes de données ce symbole accepte de mettre en relation et dans quel ordre.

Définition 1.24 Une Σ- algèbre $\mathcal{A}$ est la donnée d'un ensemble A non vide muni, pour chaque symbole de :
- fonction $f : s_1 \times \ldots s_n \to s$, d'une application $f^{\mathcal{A}} : A_{s_1} \times \ldots \times A_{s_n} \to A$;
- relation $R : s_1 \times \ldots s_n$, d'une relation $R^{\mathcal{A}} \subset A_{s_1} \times \ldots \times A_{s_n}$.

1.4 Satisfaction

1.4.1 Conséquence sémantique

Ayant opté des schémas d'interprétation pour tous les objets syntaxiques, nous introduisons la notion de modèle et les propriétés sémantiques de validité et de complétude.

La relation de satisfaction $\models$

Soit $\mathcal{A}$ une Σ-algèbre. Si on note $\{\mathbb{T}, \mathbb{F}\}$ l'ensemble des booléens, une proposition P est interprétée par une application $P^{\mathcal{A}} : A^{\mathcal{X}} \to \{\mathbb{T}, \mathbb{F}\}$, dont la définition par récurrence sur les formules est donnée à la figure 1.5 (on écrit $P^{\mathcal{A},\sigma}$ au lieu de $P^{\mathcal{A}}(\sigma)$) . Dans le cas d'une proposition ouverte P, on dit que $\mathcal{A}$ *satisfait* P si la formule P est vraie dans A en substituant aux variables libres n'importe quels éléments de A. Cela s'écrit formellement :

Définition 1.25 Soit $\mathcal{A}$ une algèbre de support A, et P une proposition. On dit que $\mathcal{A}$ satisfait P, et on note $\mathcal{A} \models P$, si $[\![P]\!]^{\mathcal{A},\sigma} = \mathbb{T}$ pour toute assignation $\sigma \in A^{\mathcal{X}}$.

Pour ce qui concerne les clauses équationnelles, cela peut s'écrire :

Définition 1.26 Soit $\mathcal{A}$ une algèbre et $C = \bigwedge_{i\in\{1,\ldots,m\}} a_i \approx b_i \Rightarrow \bigvee_{j\in\{1,\ldots,n\}} a'_j \approx b'_j$. On dit que $\mathcal{A}$ satisfait C, ou que C est valide dans $\mathcal{A}$, si, et seulement si : pour toute assignation σ, si, pour tout $i \in \{1,\ldots,m\}$, $a_i\sigma = b_i\sigma$, alors, il existe $j \in \{1,\ldots,n\}$, tel que $a'_j\sigma = b'_j\sigma$.

Soit Γ une théorie i.e une partie de $\mathcal{L}(\Sigma,\mathcal{X})$. On dit que $\mathcal{A}$ est un modèle de la théorie Γ, et on note $\mathcal{A} \models \Gamma$ si $\mathcal{A} \models P$ pour tout $P \in \Gamma$.

Notation: L'ensemble des modèles d'une théorie Γ sera noté $Mod(\Sigma,\Gamma)$, ou plus simplement $Mod(\Gamma)$.

Considérons l'exemple suivant :

Exemple 1.16 [COM 01] Soit la spécification suivante :
- Sortes : Nat.
- Opérations : $0 : Nat$; $s : Nat \to Nat$.
- Axiomes : $0 + x \approx x$; $s(x) + y \approx s(x + y)$.

Considérons l'algèbre $\mathcal{A}$, dont le domaine A est l'ensemble à deux éléments $\{0,\ a\}$, et tel que $s^{\mathcal{A}}$, l'interprétation de s dans $\mathcal{A}$, est l'identité $s^{\mathcal{A}}(0) = 0$, $s^{\mathcal{A}}(a) = a$ et $+^{\mathcal{A}}$, l'interprétation de $+$ dans $\mathcal{A}$, soit la projection droite : $u +^{\mathcal{A}} v$ est v. $\mathcal{A}$ est alors un modèle de E.

La relation de satisfaction $\models$, permet de définir une nouvelle relation sur $\mathcal{P}(\mathcal{L}(\Sigma,\mathcal{X})) \times \mathcal{L}(\Sigma,\mathcal{X})$, dite de *conséquence sémantique*.

Définition 1.27 Soit Γ une théorie et P une proposition. On dit que P est une *conséquence sémantique* de Γ, et on note $\Gamma \models P$ si, pour toute Σ-algèbre $\mathcal{A}$ et pour toute assignation $\sigma : \mathcal{X} \to \mathcal{A}$, $\mathcal{A} \models \Gamma\sigma$ entraîne $\mathcal{A} \models P\sigma$.

Exemple 1.17 [COM 01] Reprenons l'exemple 1.16. Nous avons vu que les égalités de E sont satisfaites dans $\mathcal{A}$. Cependant, l'égalité $x + 0 \approx x$ n'est pas valide dans $\mathcal{A}$, puisque $a +^{\mathcal{A}} 0 = 0$. Par conséquent, l'égalité $x + 0 \approx x$ n'est pas une conséquence sémantique de E

Validité et complétude de la déduction

On considère de nouveau la relation de déduction $\vdash$ de la logique classique définie au paragraphe 1.2.1. Soit $\Gamma \subseteq \mathcal{L}(\Sigma,\mathcal{X})$ la *théorie engendrée* par $\Delta \subseteq \mathcal{L}(\Sigma,\mathcal{X})$:

$$\Gamma = \{P \in \mathcal{L}(\Sigma,\mathcal{X}) \mid \Delta \vdash P\}$$

Comme $\Delta \subseteq \Gamma$, il est clair que $Mod(\Gamma) \subseteq Mod(\Delta)$, mais a-t-on l'égalité ? Le théorème de validité énoncé ci-dessous le prouve :

Théorème 1.1 *Si* $\Gamma \vdash P$, *alors* $\Gamma \models P$

Corollaire 1.2 Les modèles d'une théorie sont ceux de ses axiomes.

Le théorème de complétude constitue la réciproque du théorème de validité :

Théorème 1.2 *Si* $\Gamma \models P$, *alors* $\Gamma \vdash P$

Avec ces deux théorèmes, les relations $\vdash$ et $\models$ sont identiques, ce qui permet deux points de vue sur les mêmes problèmes.

1.4.2 Conséquence inductive

La relation de conséquence inductive

Un modèle d'une théorie qui est une algèbre de Herbrand est appelé *modèle de Herbrand* de cette théorie. La restriction aux modèles de Herbrand permet de définir une relation sur $\mathcal{P}(\mathcal{L}(\Sigma, \mathcal{X})) \times \mathcal{L}(\Sigma, \mathcal{X})$, dite de *conséquence inductive*, et qui est plus faible que la relation de conséquence sémantique.

Définition 1.28 Soit Γ une théorie et P une proposition. On dit que P est une *conséquence inductive* de Γ, et on note $\Gamma \models_i P$ si $\mathcal{H}$ satisfait P pour toute modèle de Herbrand $\mathcal{H}$ de Γ

Congruence

Une *congruence* sur une algèbre $\mathcal{A}$ est une relation d'équivalence sur son support A compatible avec les opérations :

Définition 1.29 Une congruence sur une Σ-algèbre $\mathcal{A}$ de domaine A est une relation d'équivalence $\sim$ sur A telle que, pour tout $f : s_1 \times \ldots s_n \rightarrow s \in \Sigma$, pour tout a_1, a'_1, ..., a_n, a'_n dans A :

$$a_1 \sim a'_1 \wedge \ldots \wedge a_n \sim a'_n \Rightarrow f^{\mathcal{A}}(a_1, \ldots, a_n) \sim f^{\mathcal{A}}(a'_1, \ldots, a'_n)$$

Remarque: Une congruence sur les termes est alors une congruence sur l'algèbre $(\mathcal{T}(\Sigma, \mathcal{X}), \Sigma)$.

Exemple 1.18 [LAL 90] Une congruence $\sim$ sur un groupe G est déterminée par la classe de l'élément neutre, qui est un sous-groupe distingué de $H_\sim$ de G ; on a $a \sim b$ si et seulement si $ab^{-1} \in H_\sim$.

Algèbre quotient

Définition 1.30 Si $\sim$ est une congruence sur une Σ-algèbre $\mathcal{A}$ de domaine A, l'ensemble quotient, noté $A/\sim$, est un ensemble muni d'une structure d'algèbre tel que, pour tout $f : s_1 \times \ldots s_n \rightarrow s \in \Sigma$, pour tout $a_1, \ldots, a_n \in A$,

$$f_{\mathcal{A}/\sim}([a_1]_\sim, \ldots, [a_n]_\sim) = [f_{\mathcal{A}}(a_1, \ldots, a_n)]_\sim$$

où, pour tout $i \in \{1, \ldots, n\}$, $[a_i]_\sim$ est la classe d'équivalence de a_i dans A.

Nous allons voir que les algèbres quotients permettent de caractériser les conséquences inductives. Le noyau d'un homomorphisme $\phi : \mathcal{A} \rightarrow \mathcal{B}$ est

$$Ker\ \phi = \{(a, a') \in A^2 \mid \phi(a) = \phi(a')\}$$

Comme en algèbre linéaire, tout homorphisme surjectif $\phi : \mathcal{A} \rightarrow \mathcal{B}$ est factorisable à travers le quotient de son noyau en la surjection naturelle et un isomorphisme $\hat{\phi}$ (figure 1.6) Dans le cas où $\mathcal{H}$ est une algèbre de Herbrand, l'interprétation $i_{\mathcal{H}}$ associée (cf corollaire 1.1) est surjective, et l'application de la factorisation précédente permet de déduire l'existence d'un isomorphisme $\hat{i}_{\mathcal{H}}$ tel que le diagramme de la figure 1.7 commute. En tant que relation binaire sur le support H

$$\begin{tikzcd} \mathcal{A} \arrow[r, "\phi"] \arrow[d] & \mathcal{B} \\ \mathcal{A}/Ker\ \phi \arrow[ur, dashed, "\hat{\phi}"'] & \end{tikzcd}$$

FIG. 1.6 – Factorisation d'un homomorphisme

$$\begin{tikzcd} \mathcal{T}(\Sigma) \arrow[r, "i_{\mathcal{H}}"] \arrow[d] & \mathcal{H} \\ \mathcal{T}(\Sigma)/Ker\ i_{\mathcal{H}} \arrow[ur, dashed, "\hat{i}_{\mathcal{H}}"'] & \end{tikzcd}$$

FIG. 1.7 – Factorisation de $i_{\mathcal{H}}$

de $\mathcal{H}$, $Ker\ i_{\mathcal{H}}$ est une équivalence ; c'est de plus une congruence, puisque si $(h_i, h'_i) \in Ker\ i_{\mathcal{H}}$, c'est que $i_{\mathcal{H}}(h_i) = i_{\mathcal{H}}(h'_i)$, donc

$$\begin{array}{rcl} i_{\mathcal{H}}(f^{\mathcal{H}}(h_1 \dots h_n)) & = & f^{\mathcal{H}}(i_{\mathcal{H}}(h_1), \dots, i_{\mathcal{H}}(h_n)) \\ & = & f^{\mathcal{H}}(i_{\mathcal{H}}(h'_1), \dots, i_{\mathcal{H}}(h'_n)) \\ & = & i_{\mathcal{H}}(f^{\mathcal{H}}(h'_1 \dots h'_n)) \end{array}$$

On peut ainsi énoncer le théorème suivant :

Théorème 1.3 *Pour toute algèbre de Herbrand $\mathcal{H}$, il existe une congruence $\sim$ sur $\mathcal{T}(\Sigma)$ telle que $\mathcal{H}$ soit isomorphe à $\mathcal{T}(\Sigma)/\sim$.*

Remarque: On vient de voir l'intérêt des algèbres définies comme des quotients, cependant, contrairement aux algèbres de termes dont tout élément est facilement représentable en machine, celles-ci ne le sont pas immédiatement. En effet, s'il est commode de représenter une classe d'équivalence par un élément de cette classe, le problème se pose de savoir si deux éléments sont ou non dans la même classe. Ceci n'étant pas toujours décidable, il faut plutôt voir cette construction comme une définition abstraite, qui sera concrétisée dans les cas où l'on dispose en outre d'algorithmes (par exemple de réécriture). On peut ainsi définir récursivement la congruence engendrée par un ensemble d'égalités conditionnelles.

Congruence engendrée par un ensemble d'égalités conditionnelles

Théorème 1.4 *[KAP 84a] Soit E un ensemble d'égalités conditionnelles définies sur une signature Σ. La plus petite congruence engendrée par E sur $\mathcal{T}(\Sigma)$, notée $\sim_E$, se construit récursivement de la manière suivante :*

- *$\sim_0$ est l'égalité syntaxique ;*

– $\sim_n$ *est la plus petite congruence telle que :*

1. $\sim_n \subseteq \sim_{n+1}$ *;*
2. *si* $a_i\sigma \sim_n b_i\sigma$ *pour tout* $\{1, \ldots, m\}$*, alors* $s\sigma \sim_{n+1} t\sigma$*, et cela pour toute équation* $\bigwedge_{i\in\{1,\ldots,m\}} a_i \approx b_i \Rightarrow s \approx t$ *de* E *;*

– $\sim_E = \bigcup_{n\in\mathbb{N}} \sim_n$

Remarque: Si E est un ensemble de simples égalités, $\sim = \Phi_0$, et $\sim$ est notée $=_E$.

Théorème 1.5 *[EHR 85] Si* E *est un ensemble consistant d'égalités conditionnelles, alors* $\mathcal{T}(\Sigma)/\sim_E$ *est initiale dans* $Mod(\Sigma, E)$*. On la désigne alors souvent par* $I(\Sigma, E)$*, pour souligner cette caractéristique.*

Le lemme suivant montre l'intérêt pratique que l'algèbre $I(\Sigma, E)$ peut également avoir :

Lemme 1.1 Si E est un ensemble d'égalités conditionelles et C est une clause positive, C est une conséquence inductive de E ssi $\mathcal{I}(\Sigma, E) \models C$

Exemple 1.19 [COM 01] Reprenons l'exemple 1.17. L'égalité $x + 0 \approx x$ n'est pas une conséquence sémantique de E, mais c'est une conséquence inductive de E, car on a $t + 0 \sim_E t$ pour tout $t \in \mathcal{T}(\Sigma)$, autrement dit on a $\mathcal{I}(\Sigma, E) \models x + 0 \approx x$

On ne doit pas perdre de vue qu'une clause (non positive) peut être satisfaite dans $\mathcal{I}(\Sigma, E)$, sans pour autant être une conséquence inductive de E. Cette différence est illustrée dans [COM 01] par l'exemple suivant :

Exemple 1.20 [COM 01] Reprenons l'exemple 1.16. $\neg(x \approx s(x))$ n'est pas une conséquence inductive de E car le modèle trivial à un élément de E ne satisfait pas $\neg(x \approx s(x))$, mais on a cependant $\mathcal{I}(\Sigma, E) \models \neg(x \approx s(x))$.

Considérons l'exemple suivant, dans lequel la spécification contient une égalité conditionnelle :

Exemple 1.21 [BOU 95] Soit la spécification suivante :

– Sortes :

$$Nat; \quad Bool.$$

– Opérations :

$$0 : Nat; \quad \mathbb{T},\ \mathbb{F} : Bool; \quad s : Nat \to Nat; \quad p : Nat \times Nat \times Nat \to Bool; \quad f : Nat \times Nat \times Nat \to Nat.$$

– Axiomes :

$$p(x, 0, z) \approx \mathbb{T}; \quad p(x, s(y), z) \approx p(x, y, z); \quad p(y, x, z) \approx \mathbb{T} \Rightarrow f(x, y, z) \approx 0.$$

On peut montrer que l'égalité $p(x, y, z) \approx \mathbb{T}$ est une conséquence inductive des axiomes de la spécification.

2

Réécriture

Sommaire

2.1 Relation binaire sur un ensemble 26
2.2 Système de réécriture 28
2.2.1 Cas non conditionnel 28
2.2.2 Cas conditionnel 29
2.3 Spécification équationnelle 30
2.3.1 Un système de réécriture particulier 30
2.3.2 Pourquoi orienter les égalités ? 31
2.3.3 Unification 32
2.3.4 Paires critiques 33
2.3.5 La procédure de complétion de Knuth-Bendix 34
2.4 Complétude suffisante 34
2.5 Réécriture modulo un ensemble d'égalités 36
2.5.1 Relation binaire modulo un ensemble d'égalités 36
2.5.2 Réécriture $\mathcal{R}, E$ et complétude suffisante 37

Dans ce chapitre, nous commençons par rappeler quelques définitions utiles sur les relations binaires, et en particulier sur les relations d'ordre et leurs variantes. Nous effectuons ensuite une mise au point sur la notion de *système de réécriture* et sur celle de *spécification équationnelle*. Les égalités sont symétriques, cela signifie qu'on peut les appliquer dans les deux sens. Nous verrons qu'il peut être pertinent cependant de ne restreindre leur application qu'à un seul sens, afin de les transformer en règles de calcul. Nous effectuerons ensuite quelques rappels sur la notion d'*unification* qui joue un rôle central en informatique, aussi bien pour prouver que pour calculer. Nous parlerons également de la *complétude suffisante*, qui est une propriété importante des systèmes de réécriture, et sur laquelle repose en grande partie la correction des systèmes de recherche de preuves présentés dans les parties III et IV. Nous définirons par ailleurs un sous-ensemble de positions caractéristiques à l'intérieur des termes qui seront utilisées par ces systèmes : les *positions définie-maximales*. La dernière section de ce chapitre est dédiée à la réécriture modulo un ensemble d'égalités. Plusieurs relations de ce type peuvent être définies. L'une d'entre elles est la relation de Peterson et Stickel à partir de laquelle nous proposons une définition d'un *système de réécriture modulo suffisamment complet*. Nous expliquerons alors l'intérêt que les positions *définie-maximales* possèdent également dans le cas de la réécriture modulo. Pour cela, il est d'abord nécessaire de définir une propriété de *conservation des termes constructeurs* sur les ensemble d'égalités.

2.1 Relation binaire sur un ensemble

De nombreuses définitions et propriétés de base des systèmes de réécriture peuvent être énoncées de façon abstraite, via les relations binaires sur les ensembles.

Définition 2.1 Un ensemble T étant donné, on appelle *relation binaire* sur T tout sous-ensemble du produit cartésien $T \times T$.

Remarque: Une relation est souvent notée $\rightarrow$, et on écrit $a \rightarrow b$ pour signifier que deux éléments a et b sont en relation.

On suppose donc que $\rightarrow$ est une relation binaire sur un ensemble T, et on rappelle les définitions suivantes

Définition 2.2
- $\rightarrow$ est réflexive si $\forall t\ t \rightarrow t$.
- $\rightarrow$ est symétrique si $\forall t\ \forall t'\ t \rightarrow t' \Rightarrow t' \rightarrow t$.
- $\rightarrow$ est antisymétrique si $\forall t\ \forall t'\ ((t \rightarrow t') \wedge (t' \rightarrow t)) \Rightarrow t = t'$.
- $\rightarrow$ est transitive si $\forall t\ \forall t'\ \forall t''\ ((t \rightarrow t') \wedge (t' \rightarrow t'')) \Rightarrow t \rightarrow t''$.

Les différentes fermetures de $\rightarrow$ seront désignées de la manière suivante :
- $\xrightarrow{=}$ est la fermeture réflexive.
- $\xrightarrow{+}$ est la fermeture transitive.
- $\xrightarrow{*}$ est la fermeture réflexive transitive.
- $\leftrightarrow$ est la fermeture symétrique.
- $\overset{*}{\leftrightarrow}$ est la fermeture réflexive, symétrique et transitive.

La fermeture réflexive, symétrique et transitive est appelée *relation d'équivalence* ou de *conversion*. On dira alors qu'un élément $t \in T$
- est *réductible* si, et seulement si il existe t' tel que $t \rightarrow t'$;
- est en *forme normale* si, et seulement si il n'est pas réductible ;
- est une *forme normale* de s si, et seulement si $s \xrightarrow{*} t$ et t est en forme normale. On note alors $t{=}s\downarrow$.

Remarque: Si $\rightarrow$ est une relation sur l'ensemble des termes, une forme normale d'une substitution est définie par $x\sigma\downarrow = (x\sigma)\downarrow$ pour tout $x \in \mathcal{D}om(\sigma)$. On dira alors qu'une substitution est une *substitution normalisée* (ou *substitution irréductible*) si elle coïncide avec l'une de ses formes normales.

Les relations binaires sont souvent utilisées pour faire des comparaisons. Nous en profitons pour rappeler ici quelques définitions utiles sur les relations d'ordre et leurs variantes.

Définition 2.3 Une relation binaire sur un ensemble E est
- un *ordre strict* si elle est non réflexive et transitive ;
- un *quasi-ordre* si elle est réflexive et transitive ;
- un *ordre* si elle est réflexive, antisymétrique et transitive ;
- un (quasi-) ordre total si $x \leqslant y$ ou $y \leqslant x$ pour tous $x,\ y \in E$.

Un ordre strict est généralement noté avec le symbole $<$. Si $<$ est un ordre strict, et si on définit $\leqslant$ par $x \leqslant y$ si $x < y$ ou $x = y$, alors $\leqslant$ est un ordre ; si $\leqslant$ est un ordre, alors $<$ définie par $x < y$ si $x \leqslant y$ et $x \neq y$ est un ordre strict ; enfin si $\leqslant$ est un quasi-ordre, $x \gtrless y$ si $x \leqslant y$ et $y \leqslant x$ est une relation d'équivalence, et $x < y$ si $x \leqslant y$ et $y \nleqslant x$ est un ordre strict.

Exemple 2.1 Soit E un ensemble, et E^* le monoïde libre engendré par E. On dit que u est un *préfixe* de v, et on note $u \leqslant v$, s'il existe $u' \in E^*$ tel que $v = uu'$. La relation $\leqslant$ ainsi définie est alors une relation d'ordre.

Une propriété fondamentale des ordres est la bonne fondation.

Définition 2.4 Un (quasi-) ordre $\leqslant$ est un *(quasi-) ordre bien-fondé* (ou *(quasi-) ordre noethérien*), si, et seulement si, il n'existe pas de suites infiniment décroissantes du type $s_0 > s_1 > s_2 > \ldots$.

Nous présentons ci-dessous trois constructions générales de (quasi-)ordres bien fondés.

Définition 2.5 Pour tout entier i tel que $1 \leq i \leq n$, supposons que $\leqslant^i$ soit un (quasi-)ordre bien fondé sur un ensemble E_i. Définissons le (quasi-)ordre $\leqslant_n$ sur le produit cartésien $E_1 \times \ldots \times E_n$ par $(x_1, \ldots, x_n) \leqslant_n (y_1, \ldots, y_n)$ si $x_i \leqslant^i y_i$ pour tout i.

Proposition 2.1 Le quasi-ordre $\leqslant_n$ ainsi défini est bien-fondé si, et seulement si chaque $\leqslant^i$ est bien fondé.

Définition 2.6 Pour tout entier i tel que $1 \leq i \leq n$, supposons que $\leqslant^i$ soit un (quasi-)ordre bien fondé sur un ensemble E_i. Définissons le (quasi-)ordre $\leqslant_{lex}$ sur le produit cartésien $E_1 \times \ldots \times E_n$ par $(x_1, \ldots, x_n) \leqslant_{lex} (y_1, \ldots, y_n)$ si $x_i \leqslant^i y_i$ pour tout i, ou il existe i tel que $x_j = y_j$ (ou $x_j \gtrless^j y_j$ dans le cas d'un quasi-ordre) pour tout $j < i$ et $x_i <^i y_i$.

Proposition 2.2 Le (quasi-)ordre $\leqslant_{lex}$ ainsi défini est bien-fondé si, et seulement si chaque $\leqslant^i$ est bien fondé.

Remarque: Lorsque tous les $\leqslant^i$ sont égaux à un même (quasi-)ordre $\leqslant$ sur un ensemble E, $\leqslant_n$ et $\leqslant_{lex}$ désignent alors respectivement l'*extension cartésienne* et l'*extension lexicographique* de $\leqslant$ au produit cartésien E^n.

Un *multi-ensemble* est un ensemble avec occurences multiples de ses éléments. Plus précisément, si E est un ensemble, un multi-ensemble sur E est une application $M : E \to \mathbb{N}$; $M(x)$ est alors le nombre d'occurences de x dans M, pour $x \in E$. Les éléments d'un multi-ensemble M sont les éléments x de E tels que $M(x) \neq 0$. On utilise aussi une notation ensembliste pour les multi-ensembles : on écrira par exemple $M = \{\{a, a, a, c\}\}$, au lieu de $M(a) = 3$, $M(c) = 1$, et $M(x) = 0$ si $x \neq a, c$. L'ensemble des multi-ensembles finis sur E est noté $\mathcal{M}\ (E)$.

Définition 2.7 Soit $\leqslant$ un quasi-ordre sur E ; l'extension $\leqslant_{mul}$ de $\leqslant$ à $\mathcal{M}(E)$ est définie par $M \leqslant_{mul} M'$ si M est obtenu à partir de M' en remplaçant une occurence d'un élément x de M' par un multi-ensemble d'éléments tous plus petits (ou équivalents) à x pour le (quasi-)ordre $\leqslant$.

De façon plus opérationnelle, les règles suivantes décrivent l'extension multi-ensemble stricte $<_{mul}$:

$$\begin{array}{lccc}
\text{Bigger} & M \cup \{t\} >_{mul} N \cup \{s\} & \twoheadrightarrow & M \cup \{t\} >_{mul} N \\
 & & & \text{si } t > s \\
\text{Erase} & M \cup \{t\} >_{mul} N \cup \{t\} & \twoheadrightarrow & M >_{mul} N \\
\text{Success} & M \cup \{t\} >_{mul} \emptyset & \twoheadrightarrow & \mathbb{T}
\end{array}$$

Proposition 2.3 [DER 79b] Si $\leqslant$ est bien-fondé, $\leqslant_{mul}$ est également bien-fondé.

Exemple 2.2 $\{\{3, 3, 1, 2\}\} >_{mul} \{\{3, 2, 2, 2, 2\}\}$

Un des problèmes opérationnels essentiels du calcul, tel qu'il est pratiqué à travers la programmation, est celui de la terminaison.

Définition 2.8 La relation $\rightarrow$ est terminante (ou fortement normalisante, ou noethérienne, ou bien fondée) si toute séquence de réductions est finie.

Remarque: On dit qu'elle est faiblement terminante (ou faiblement normalisante) si tout terme admet une forme normale.

Exemple 2.3 La relation $t \rightarrow t'$ est terminante, mais pas $t \rightarrow t$.

La confluence est une propriété fondamentale de toute modélisation du calcul comme processus de réduction : elle signifie l'indépendance du résultat du calcul vis à vis du chemin de réduction suivi.

Définition 2.9 La relation binaire $\rightarrow$ sur T est *confluente* en t si, pour tout $t_1,\ t_2$, si $t \xrightarrow{*} t_1$ et $t \xrightarrow{*} t_2$, il existe t' tel que $t_1 \xrightarrow{*} t'$ et $t_2 \xrightarrow{*} t'$

La confluence a comme conséquence que la relation de conversion s'identifie à la réduction à un même terme : c'est la propriété de *Church-Rosser*

Proposition 2.4 Si la relation $\rightarrow$ est confluente, alors $x \overset{*}{\leftrightarrow} y$ équivaut à l'existence de z tel que $x \xrightarrow{*} z$ et $y \xrightarrow{*} z$.

Une relation terminante et confluente est alors appelée *convergente*.

Définition 2.10 La relation binaire $\rightarrow$ sur T est *convergente* si elle est confluente et terminante.

2.2 Système de réécriture

2.2.1 Cas non conditionnel

Contrairement au $\lambda-$calcul, où l'on travaille avec une seule règle, la $\beta-$réduction, les systèmes de réécriture présentent une grande variété de formes.

Définition 2.11 Un *système de réécriture non conditionnel* est la donnée d'une signature Σ et d'un ensemble $\mathcal{R}$ de couples de termes $(l, r) \in \mathcal{T}(\Sigma, \mathcal{X}) \times \mathcal{T}(\Sigma, \mathcal{X})$ tels que $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$ et $l \notin \mathcal{X}$

Chaque couple, aussi appelé *règle* et noté $l \rightarrow r$, décrit une règle de calcul générique. La définition d'une relation de réduction à partir d'un système de réécriture peut être construite par règles d'inférence :

- $$\frac{}{l\sigma \rightarrow r\sigma} \qquad \text{si } (l, r) \in \mathcal{R} \text{ et } \sigma \in Subst^{\Sigma,\mathcal{X}}$$

- $$\frac{t_i \rightarrow t'_i}{ft_1 \ldots t_i \ldots t_n \rightarrow ft_1 \ldots t'_i \ldots t_n} \qquad \text{si } f \in \Sigma \text{ et } 1 \leqslant i \leqslant n$$

ou directement par greffe :

Définition 2.12 Un terme s se réécrit (ou se réduit) en un terme t par application de la règle $l \to r \in \mathcal{R}$ à l'occurence $\omega \in \mathcal{D}om(s)$ si il existe une substitution σ telle que $s_{|\omega} = l\sigma$ et $t = s[r\sigma]_\omega$. Cette relation est notée $s \to_{[l\to r,\omega,\sigma]} t$ ou simplement $s \to_{\mathcal{R}} t$, ou encore $s \to t$ si aucune confusion n'est possible. Le terme $l\sigma$ est appelé *redex* (de l'anglais "reducible expression"), et $r\sigma$ est son *contracté*. σ est alors appelée "filtre" de l vers $s_{|\omega}$ et on dit que s est *$\mathcal{R}$-réductible* à la position ω.

Remarque: Dans la suite, si aucune confusion n'est possible, le symbole $\mathcal{R}$ désignera également la relation $\to_{\mathcal{R}}$.

Exemple 2.4 Considérons le système de réécriture suivant :

$$\begin{array}{ll} 0 + x \to x & x * 0 \to 0 \\ x + s(y) \to s(x + y) & x * s(y) \to (x * y) + x \end{array}$$

On a $(s(s0)) * (s(s0)) \to^*_{\mathcal{R}} s(s(s(s0)))$.

On attribue alors à un système de réécriture $\mathcal{R}$ les propriétés de la relation $\to_{\mathcal{R}}$ qu'il engendre. On dit ainsi qu'un système de réécriture $\mathcal{R}$ est *noethérien* si toute séquence de réductions termine. Un système contenant une règle permutative du genre $x+y \to y+x$ ne peut être noethérien. La preuve de noethérianité d'un système est toujours un problème difficile, lié à la notion d'ordre bien fondé :

Théorème 2.1 *La noethérianité des systèmes formés d'au moins deux règles est indécidable.*

2.2.2 Cas conditionnel

Appliquer une égalité conditionnelle sur un terme t revient à remplacer l'instance du membre gauche dans t par l'instance correspondante du membre droit, à condition que la *préconditition* soit vérifiée. Cependant, l'évaluation de cette précondition peut mener à des dérivations infinies, comme le montre l'exemple suivant, tiré de [BER 97]

Exemple 2.5 Soit l'égalité

$$p(x) > 0 \approx \mathbb{T} \Rightarrow x > 0 \approx \mathbb{T}$$

où p désigne la fonction prédecesseur sur les entiers. L'évaluation de l'égalité $0 > 0 \approx \mathbb{T}$ donne une infinité de dérivations :

$$\begin{array}{lll} 0 > 0 \approx \mathbb{T} & \text{si} & -1 > 0 \approx \mathbb{T} \\ -1 > 0 \approx \mathbb{T} & \text{si} & -2 > 0 \approx \mathbb{T} \\ \vdots & & \end{array}$$

Pour éviter ce type de dérivation cyclique, on peut imposer des restrictions, et on définit ainsi des *règles de réécriture conditionnelles* :

Définition 2.13 Soit $<$ un ordre bien fondé. Une égalité conditionnelle $\bigwedge_{i=1}^{n} a_i \approx b_i \Rightarrow s \approx t$ est dite *règle de réécriture conditionnelle*, notée par $\bigwedge_{i=1}^{n} a_i \approx b_i \Rightarrow s \to t$, si toute variable de $\bigwedge_{i=1}^{n} a_i \approx b_i$ et de t est une variable de s, et pour toute substitution σ, on a $\{s\sigma\} >_{mul} \{t\sigma, a_1\sigma, b_1\sigma, \ldots, a_n\sigma, b_n\sigma\}$. Le terme s est appelé *membre gauche* de la règle.

La relation $s\sigma > t\sigma$ assure la terminaison de la réduction non conditionnelle, tandis que $\{s\sigma\} >_{mul} \{a_1\sigma, b_1\sigma, \ldots, a_n\sigma, b_n\sigma\}$ garantit que les termes apparaissant dans l'évaluation récursive de la précondition diminuent au fur et à mesure de la taille, assurant ainsi la terminaison (l'idée d'établir un ordre entre le membre gauche et la précondition de la règle a été proposée par [KAP 84b], [KAP 87a]).

Remarque: Les variables du membre gauche d'une règle conditionnelle sont implicitement universellement quantifiées. Les variables libres sont donc les variables de la précondition qui n'apparaissent pas dans le membre gauche.

Exemple 2.6 Dans une règle du type $x = f(y) \Rightarrow g(x) \rightarrow c$, la variable y est libre.

Un système de réécriture conditionnel $\mathcal{R}$ est un ensemble de règles conditionnelles.

2.3 Spécification équationnelle

2.3.1 Un système de réécriture particulier

De façon générale, une *spécification* consiste en une signature et un ensemble de propositions, souvent appelées *axiomes* de la spécification. Nous nous intéresserons ici au cas où ces propositions sont des égalités.

Définition 2.14 Une spécification équationnelle est un couple (Σ, E), où Σ est une signature, et E un ensemble d'égalités.

Remarque: Si E contient au moins une égalité conditionnelle, on parle de *spécification équationnelle conditionnelle.*

La représentation des signatures peut être élargie de telle sorte que l'on puisse représenter les spécifications équationnelles de la même manière en utilisant le mot clef "Axiomes" et en énumérant toutes les égalités de E sans les quantificateurs.

Exemple 2.7 Une spécification pour les piles est donnée ci-dessous :
- Sortes : alphabet; pile.
- Opérations :

$$a_1, \ldots, a_n : \text{ alphabet}; \quad Nil : \text{ pile}; \quad pop : \text{ pile} \rightarrow \text{pile}; \quad push : \text{ alphabet} \times \text{pile} \rightarrow \text{pile}.$$

- Axiomes :

$$pop(Nil) \approx Nil; \quad pop(push(x, s)) \approx s.$$

Dans les spécifications équationnelles, les égalités occupent la même position que les règles dans un système de réécriture. Les égalités sont symétriques, et cela est naturel d'un point de vue algébrique. On peut transformer des égalités en règles de calcul si on les oriente.

Exemple 2.8 Les règles de l'exemple 2.4 sont obtenues en orientant les égalités ci-dessous :

$$\begin{array}{ll} 0 + x \approx x & x * 0 \approx 0 \\ x + s(y) \approx s(x + y) & x * s(y) \approx (x * y) + x \end{array}$$

Dans l'exemple 2.8, il est facile d'orienter *correctement* les égalités, du fait de leur caractère récursif. Cependant, il n'est pas toujours facile de savoir à priori dans quel sens orienter les égalités. Et dans de nombreux cas, il n'existe aucun moyen d'orienter les égalités de façon à obtenir un système de réécriture efficace pour effectuer des calculs : c'est le cas par exemple si l'une de ces égalités est $x + y \approx y + x$, qui exprime la commutativité de l'addition.
Si on compare la définition 2.14 avec celle d'un système de réécriture (Définition 2.11), la seule différence est que l'on dispose d'égalités à la place de règles. La différence essentielle avec une règle de réécriture est bien-sûr que l'on n'attache pas d'importance à l'ordre des termes dans une égalité. La définition suivante permet d'effectuer quelques transitions naturelles entre les spécifications équationnelles et les systèmes de réécriture.

Définition 2.15 On considère une spécification équationnelle (Σ, E).

- Deux termes s et t étant donnés, on dit que s *se réduit* en t par E lorsqu'il existe une égalité $e_1 \approx e_2 \in E$, une position $\omega \in \mathcal{D}om(s)$, et une substitution σ, telles que $s_{|\omega} = e_1\sigma$ et $t = s[e_2\sigma]_\omega$. On écrit $s \leftrightarrow_E t$ si s se réduit en t par E, car t se réduit également en s par E, étant donné que l'on peut aussi dire que E contient l'égalité $e_2 \approx e_1$.
- Une séquence de réductions est appelée *conversion* par rapport à la spécification équationnelle E. On écrit $s =_E t$ s'il existe une conversion de s vers t dans E. La relation de conversion $=_E$ est clairement une relation d'équivalence.

On étend la relation $=_E$ aux substitutions de la façon suivante :

Notation: Deux substitutions σ, μ et un ensemble fini $V \subset \mathcal{X}$ de variables étant donnés, l'équivalence $\sigma =_E \mu[V]$ signifiera que l'on a $x\sigma =_E x\mu$ pour tout $x \in V$.

La relation de conversion joue un rôle central dans le *raisonnement équationnel*. Nous entendons de façon intuitive par *raisonnement équationnel* la façon dont les égalités sont utilisées et manipulées en mathématiques. Chaque étape d'un tel raisonnement s'effectue en appliquant une substitution à une égalité faisant partie de l'ensemble des hypothèses, ou qui a déjè été démontrée, et à son utilisation dans un contexte adapté. C'est exactement ce qui se passe lorsque l'on effectue une conversion. Ceci est une conséquence du théorème suivant :

Théorème 2.2

$$s =_E t \Leftrightarrow E \vdash s \approx t$$

avec $\vdash$ la relation de la logique classique définie au paragraphe 1.2.1.

Remarque: On observera que seuls les quatre premiers axiomes de la théorie de l'égalité sont utilisés pour établir une relation du type $E \vdash s \approx t$

2.3.2 Pourquoi orienter les égalités ?

Dans quel cas un système de réécriture est-il plus avantageux qu'une spécification équationnelle ? Un ensemble d'égalités E déterminant une relation d'équivalence $=_E$ sur les termes, le problème $E \vdash s \approx t$ se ramène à savoir si s et t appartiennent à une même classe d'équivalence. On est dans une situation favorable s'il y a un représentant canonique de chaque classe, et si on sait associer à tout élément le représentant canonique de sa classe : l'équivalence se ramène à l'identité des représentants canoniques. Par exemple, pour décider si deux fractions sont équivalentes, il suffit de les simplifier au maximum puis de tester l'égalité des numérateurs et des dénominateurs.

C'est précisément ce que permet un système de réécriture, à condition qu'il ait la propriété de normalisation faible (tout terme admet une forme normale), qu'il soit confluent (d'où l'unicité de la forme normale), et que l'on dispose d'une stratégie effective pour atteindre la forme normale d'un terme. Cette dernière condition est satisfaite en particulier si le système est noethérien.

Définition 2.16 Un système de réécriture noethérien et confluent est appelé *convergent*

Pour utiliser un système de réécriture convergent $\mathcal{R}$ dans les preuves équationnelles, les propriétés suivantes doivent être vérifiées :

- $\mathcal{R}$ est *valide* relativement à E si $s \overset{*}{\leftrightarrow}_{\mathcal{R}} t$ entraîne $s =_E t$;
- $\mathcal{R}$ est *complet* relativement à E si $s =_E t$ entraîne l'égalité syntaxique des formes normales $s \downarrow_{\mathcal{R}}$ et $t \downarrow_{\mathcal{R}}$;
- $\mathcal{R}$ est *canonique* relativement à E s'il est convergent, valide et complet relativement à E.

2.3.3 Unification

L'unification joue un rôle central, aussi bien pour prouver que pour calculer. Elle est devenu un sujet de recherche important avec les travaux de Plotkin (1972) et Huet (1976). L'importance pratique de l'unification est due au fait que c'est une opération élémentaire.

Unification syntaxique

Définition 2.17 Soient Σ une signature , $s,\ t \in \mathcal{T}(\Sigma, \mathcal{X})$, et $\mathcal{A}$ une Σ-algèbre de support A. On appelle *solution* de l'équation $s \approx t$ dans $\mathcal{A}$, toute assignation $\sigma : \mathcal{V}ar(s) \cup \mathcal{V}ar(t) \to A$ telle que $[\![s]\!]^{\mathcal{A},\sigma} = [\![t]\!]^{\mathcal{A},\sigma}$.

Résoudre une équation $s \approx t$, c'est chercher ses solutions, ce qui revient à trouver dans A des valeurs pour les variables de s et de t qui satisfassent cette équation. Selon l'algèbre $\mathcal{A}$, la résolution d'équations prend des noms différents. Nous nous intéresserons pour le moment à l'*unification finie*, qui est la résolution d'équations dans l'algèbre $(\mathcal{T}(\Sigma, \mathcal{X}), \Sigma)$, et uniquement au cas où l'on cherche à résoudre une seule équation.

Définition 2.18 Soient s et t deux termes.

- Soit σ une substitution. On dit que σ est un *unificateur* de s et de t si $s\sigma = t\sigma$.
- On dit que s et t sont *unifiables* si s et t possèdent au moins un unificateur.

On peut alors montrer que si s et t sont unifiables, ils possèdent un unificateur minimal au sens d'un quasi-ordre que nous définissons ci-dessous.

Définition 2.19 Soit $\leqslant$ la relation définie sur l'ensemble des substitutions par

$$\mu \leqslant \sigma \Leftrightarrow \exists \rho\ \sigma = \mu\rho$$

Remarque:

- Si $\mu \leqslant \sigma$, on dit que μ est *plus générale* que σ.
- On peut restreindre la relation $\leqslant$ à un sous-ensemble $V \subset \mathcal{X}$, on écrit alors $\sigma \leqslant \mu\ \ [V]$.

Proposition 2.5 La relation $\leqslant$ définie ci-dessus est un quasi-ordre sur l'ensemble des substitutions.

Théorème 2.3 *Si s et t sont unifiables, ils possèdent un unificateur minimal au sens du quasi-ordre $\leqslant$. On l'appelle* unificateur plus général *de s et de t, et on le note $mgu(s,t)$*

Remarque: L'unificateur le plus général est unique à renommage près.

Unification équationnelle

Les définitions et premiers résultats énoncés dans le cas de l'unification syntaxique peuvent se décliner au cas de l'unification équationnelle.

Définition 2.20 Soient E un ensemble d'égalités, et s et t deux termes.
- Soit σ une substitution. On dit que σ est un *E-unificateur* de s et de t si $s\sigma =_E t\sigma$.
- On dit que s et t sont *E-unifiables* si s et t possèdent au moins un E-unificateur.

Notation: L'ensemble constitué des E-unificateurs de deux termes s et t sera désigné par $Unif_E(s,t)$

Exemple 2.9 [ECH 05] Soit C la théorie de la commutativité définie par l'axiome $f(x,y) \approx f(y,x)$. Alors les termes $f(x,a)$ et $f(y,b)$ sont C-unifiables (il suffit de considérer la substitution $\{x \to b; y \to a\}$) mais ne sont pas $\emptyset$-unifiables.

Définition 2.21 On note $\leqslant_E$ la relation définie sur l'ensemble des substitutions par

$$\mu \leqslant_E \sigma \Leftrightarrow \exists \rho \ \sigma =_E \mu\rho$$

Remarque:
- Si $\mu \leqslant_E \sigma$, on dit que μ est *plus générale* que σ modulo E.
- On peut restreindre la relation $\leqslant_E$ à un sous-ensemble $V \subset \mathcal{X}$, , on écrit alors $\sigma \leqslant_E \mu \ \ [V]$.

Proposition 2.6 La relation $\leqslant_E$ définie ci-dessus est un quasi-ordre sur l'ensemble des substitutions.

Définition 2.22 Un ensemble d'égalités E, et deux termes $s,\ t \in \mathcal{T}(\Sigma, \mathcal{X})$ étant donnés, un *ensemble complet de E-unificateurs* de s et de t est un ensemble noté $CSU_E(s,t)$ tel que :

1. $CSU_E(s,t) \subseteq Unif_E(s,t)$;
2. $\forall \mu \ \exists \sigma \ \sigma \leqslant_E \mu \ \ [\mathcal{V}ar(s) \cup \mathcal{V}ar(t)]$.

Si, de plus, $CSU_E(s,t)$ est tel que :

$$\forall \sigma \forall \mu (\mu \in CSU_E(s,t) \wedge \sigma \leqslant_E \mu \ \ [\mathcal{V}ar(s) \cup \mathcal{V}ar(t)]) \Rightarrow \sigma = \mu,$$

on dit que $CSU_E(s,t)$ est *minimal*.

Remarque:
- Dans le cas où E est la théorie vide (unification syntaxique), tout couple de termes possède un CSU qui a pour seul élément l'unificateur le plus général de ces deux termes.
- Dans une théorie de type *finitaire*, tout couple de termes s et t possède un CSU minimal de cardinalité finie (nous donnons plus loin (exemple 2.1 de la partie II) deux exemples de théories non finitaires).

2.3.4 Paires critiques

Le critère de confluence de Knuth-Bendix est une application importante de l'unification à l'étude des systèmes de réécriture.

Définition 2.23 Soient $l \to r$ et $l' \to r'$ deux règles de réécriture séparées (i.e leurs ensembles de variables sont disjoints) d'un système $\mathcal{R}$, et $\omega \in \mathcal{D}om(l)$. Le couple $(r\sigma, (l[r']_\omega)\sigma)$ est une paire critique de $\mathcal{R}$, quand σ est l'unificateur le plus général de $l_{|\omega}$ et l'.

Exemple 2.10 [LAL 90] La règle d'associativité $(xy)z \to x(yz)$ se superpose à elle-même à l'occurence 1, xy et $(x'y')z'$ étant unifiables, donnant les deux réductions

$$((x'y')z')z \to (x'y')(z'z)$$
$$((x'y')z')z \to (x'(y'z'))z$$

donc la paire critique $((x'y')(z'z), (x'(y'z'))z)$

L'existence de paires critiques est un signe d'ambigüité du système de réécriture.

Théorème 2.4 *[KNU 70],[HUE 80] Considérons un système de réécriture $\mathcal{R}$. Si $s_1 \leftarrow s \to s_2$, alors :*

- *soit il existe des réductions $s_1 \overset{*}{\to} s' \overset{*}{\leftarrow} s_2$,*
- *soit il existe une paire critique (t_1, t_2) de $\mathcal{R}$ et une substitution μ tels que l'on puisse écrire $s_i = s_i[t_i\mu]$ pour $i = 1, 2$*

Ce théorème est utilisé dans la procédure de complétion de Knuth Bendix pour normaliser des preuves équationnelles en *complétant* un système de réécriture. Nous donnons une description très rapide de cette procédure au paragraphe suivant.

2.3.5 La procédure de complétion de Knuth-Bendix

Etant donné E, on cherche $\mathcal{R}$ canonique relativement à E.

Exemple 2.11 Les axiomes de la théorie des groupes sont :

$$\begin{array}{rcl} e.x & \approx & x \\ x^{-1}.x & \approx & e \\ (x.y).z & \approx & x.(y.z) \end{array}$$

Un système de réécriture canonique pour cette théorie est formé des dix règles suivantes :

$$\begin{array}{rcl} e.x & \to & x \\ x^{-1}.x & \to & e \\ (x.y).z & \to & x.(y.z) \\ x^{-1}.(x.y) & \to & y \\ (x^{-1})^{-1} & \to & x \end{array} \qquad \begin{array}{rcl} x.e & \to & x \\ x.x^{-1} & \to & e \\ e^{-1} & \to & e \\ x.(x^{-1}.y) & \to & y \\ (x.y)^{-1} & \to & y^{-1}.x^{-1} \end{array}$$

Il est facile de construire un système valide, simplement en orientant les égalités de E en règles, mais un tel système sera rarement complet.
La procédure de complétion de Knuth-Bendix tente de transformer un ensemble d'égalités E_0 en un système de réécriture $\mathcal{R}$ qui est canonique relativement à E. Comme elle construit progressivement $\mathcal{R}$, elle opère sur des couples $(E, \mathcal{R})$ à partir de $(E_0, \emptyset)$, et elle réussit si elle parvient à $(\emptyset, \mathcal{R}_\infty)$. A chaque étape, $\mathcal{R}$ est valide relativement à E et est noethérien. Tous les $\mathcal{R}$ sont en outre inclus dans un même ordre bien fondé. La procédure est détaillée dans [WEC 92].

2.4 Complétude suffisante

On suppose que la signature Σ est la réunion disjointe d'un sous-ensemble $\mathcal{C}$ de symboles constructeurs et d'un sous-ensemble $\mathcal{D}$ de symboles définis. On suppose également que les constructeurs sont *libres*, i.e qu'ils n'apparaissent pas en tête des membres gauches des règles.
La complétude suffisante [GUT 75] est une propriété fondamentale : elle exprime que chaque terme clos peut se réécrire en un terme constructeur, i.e. un terme construit uniquement à partir de symboles de $\mathcal{C}$.

Définition 2.24 Le système de réécriture $\mathcal{R}$ est suffisamment complet si, et seulement si, pour tout $s \in \mathcal{T}(\Sigma)$, il existe $t \in \mathcal{T}(\mathcal{C})$ tel que $s \rightarrow^*_{\mathcal{R}} t$

De façon intuitive, les termes constructeurs sont les valeurs, et les autres symboles de la signature (appelés symboles de fonctions définies), représentent les fonctions définies sur ces valeurs par $\mathcal{R}$. Dire que $\mathcal{R}$ est suffisamment complet signifie que ces fonctions sont complètement définies. Les procédures de test de complétude suffisante sont fondées sur une définition introduisant la notion de complétude suffisante des symboles de fonctions définies :

Définition 2.25 Un symbole de fonction $f \in \mathcal{D}$ est suffisamment complet pour $\mathcal{R}$ si, et seulement si, pour tout $s_1, \ldots, s_n \in \mathcal{T}(\mathcal{C})$, il existe $t \in \mathcal{T}(\mathcal{C})$ tel que $fs_1 \ldots s_n \rightarrow^+_{\mathcal{R}} t$.

Proposition 2.7 Un système de réécriture $\mathcal{R}$ est suffisamment complet si, et seulement si, chaque symbole $f \in \mathcal{D}$ est suffisamment complet pour $\mathcal{R}$

Preuve. Par induction sur le nombre d'occurences de symboles de $\mathcal{D}$ dans un terme clos donné. □

Nous verrons que la complétude suffisante a des liens étroits avec le raisonnement par récurrence, et plus précisément avec la *réductibilité inductive*, la propriété qui stipule que toutes les instances closes (instances sans variables) d'un terme donné sont réductibles par un système de réécriture donné [KAP 87b], [KAP 91].

Définition 2.26 On dit que t est $\mathcal{R}$-inductivement réductible à la position $\omega \in \mathcal{D}om(t)$ si $t_{|\omega}\alpha$ est $\mathcal{R}$-réductible (cf définition 2.12) pour toute substitution close normalisée α.

La complétude suffisante n'est pas décidable en général [GUT 78a]. Elle peut cependant le devenir dans certains cas [GUT 78b], [HUE 82], [DER 83], [KOU 85], [COM 86], [LAZ 90], [KAP 91]. Il s'est avéré que les arbres automates avec contraintes sont de bons outils pour tester la complétude suffisante d'un système de réécriture [COM 97].
Pour les systèmes de réécriture terminants sur les termes clos et suffisamment complets, on peut facilement repérer des positions inductivement réductibles.

Définition 2.27 Pour tout $t \in \mathcal{T}(\Sigma, \mathcal{X})$, on dit qu'une position ω dans t est *définie-maximale*, et on note $\omega \in \mathcal{DM}(t)$, si $t(\omega) \in \mathcal{D}$ et $t(\omega') \in \mathcal{C} \cup \mathcal{X}$ dès que $\omega < \omega'$.

Considérons par exemple les entiers naturels de Peano, 0 et s sont des symboles constructeurs, $+$ est un symbole défini, et, dans $s((0+0)+s(0+s(x)))$, l'occurence 1.2.1 est *définie-maximale* mais 1 ne l'est pas.

Lemme 2.1 Pour tout système de réécriture $\mathcal{R}$ terminant sur les termes clos et suffisamment complet, pour tout terme t, et pour toute position $\omega \in \mathcal{D}om(t)$, si ω est *définie-maximale* dans $\mathcal{D}om(t)$, alors, t est inductivement réductible à la position ω.

Preuve. Un terme t étant donné, supposons que ω soit *définie-maximale* dans t et posons $f = t(\omega)$. Alors il existe des termes constructeurs $t_1, \ldots, t_n$ et $f \notin \mathcal{C}$, tels que $t_{|\omega} = ft_1 \ldots t_n$. Soit α une substitution close normalisée des variables de $t_{|\omega}$. Puisque $f \notin \mathcal{C}$, $(t_{|\omega})\alpha$ n'est pas un terme constructeur, et, du fait que $\mathcal{R}$ est suffisamment complet, $(t_{|\omega})\alpha$ n'est pas une forme normale close. Par conséquent, on peut trouver un filtre du membre gauche d'une règle $l \rightarrow r$ de $\mathcal{R}$ vers un sous-terme de $(t_{|\omega})\alpha$. Donc
$l(\epsilon)$ apparaît dans $(t_{|\omega})\alpha$ et, si on suppose $f \neq l(\epsilon)$, on peut distinguer deux cas.

- Cas 1 : si $l(\epsilon)$ apparaît dans un t_i,alors $l(\epsilon) \in C$, mais c'est impossible, car on a supposé tous les constructeurs libres.
- Cas 2 : si il existe une variable x de $t_{|\omega}$ telle que $l(\epsilon)$ apparaisse dans $x\alpha$, alors $l(\epsilon) \in \mathcal{C}$, car $x\alpha$ est une forme normale close, du fait que α est une substitution close irréductible, et, par définition de la suffisante complétude, $x\alpha$ est un terme constructeur clos. Cela entraîne donc la même contradiction qu'auparavant.

Par conséquent, $f = l(\epsilon)$, d'où le théorème. $\square$

2.5 Réécriture modulo un ensemble d'égalités

L'algorithme de complétion de Knuth Bendix a été étendu pour traiter le cas des théories dans lesquelles certains axiomes ne peuvent pas être orientés. Les extensions proposées se placent dans le cas où la théorie est un *système de réécriture de termes modulo* défini sur une signature Σ, i.e un ensemble d'axiomes contenant un premier sous-ensemble $\mathcal{R}$ contenant des règles de réécriture, et un second sous-ensemble E contenant des égalités non orientables : on notera $(\Sigma\ ,\ \mathcal{R}\ ,E)$ un tel système (ou plus simplement $(\mathcal{R}\ ,E)$. Une première approche de Lankford et Ballantyne [LAN 79] traite le cas où les classes de E-équivalences contiennent un nombre fini d'éléments, ce qui est le cas si E contient des axiomes commutatifs, ou plus généralement permutatifs. Le cas où il existe des classes de E-équivalences infinies a été étudié par Peterson et Stickel [PET 81] , qui se restreint toutefois au cas où E est un ensemble d'axiomes linéaires, et pour lequel il existe un algorithme fini et complet d'unification. La synthèse de ces deux approches a permis de réaliser une procédure de complétion associative commutative efficace .

2.5.1 Relation binaire modulo un ensemble d'égalités

Définition 2.28 On définit la relation $\rightarrow_{\mathcal{R}/E}$, ou plus simplement $\mathcal{R}/E$, par

$$s \rightarrow_{\mathcal{R}/E} t \Leftrightarrow \exists s'\ \exists t'\ s =_E s' \rightarrow_{\mathcal{R}} t' =_E t$$

Remarque: Si E est la théorie vide, la relation $\mathcal{R}/E$ est simplement la relation $\mathcal{R}$, que l'on appelle parfois *réécriture syntaxique*.

La propriété de E-terminaison est utile pour savoir si deux termes sont équivalents pour la relation $\mathcal{R}/E$.

Définition 2.29 La relation $\mathcal{R}$ est *terminante modulo* E, ou E-*terminante* si, et seulement si la relation $\mathcal{R}/E$ est noethérienne, i.e s'il n'existe pas de séquences de la forme $t_0 =_E t'_0 \rightarrow_{\mathcal{R}} t_1 =_E t'_1 \ldots t_n =_E t'_n \rightarrow_{\mathcal{R}} t_{n+1}$

Remarque: On dira que la relation $\mathcal{R}$ est *terminante modulo* E sur les termes clos (ou E-*terminante* sur les termes clos) si, et seulement si la relation $\mathcal{R}/E$ est noethérienne sur $\mathcal{T}(\Sigma)$.

Si la relation $\mathcal{R}$ est *terminante modulo* E, il suffira donc de déterminer deux formes $\mathcal{R}/E$-normales E-équivalentes de deux termes pour savoir qu'ils sont $\mathcal{R}/E$-équivalents.
La relation $\mathcal{R}/E$ n'est cependant pas complètement satisfaisante d'un point de vue opérationnel pour les raisons suivantes :

- même si $\mathcal{R}$ est fini et si la E-équivalence est décidable, la $\mathcal{R}/E$-réductibilité peut cependant être indécidable si les classes de E-équivalences sont infinies, et reste largement inapplicable s'il est nécessaire de parcourir des classes d'équivalences contenant de trop nombreux éléments.

– la relation $\rightarrow_{\mathcal{R}/E}$ peut facilement ne pas terminer. Par exemple, si $\mathcal{R}$ est non vide et E contient l'axiome d'idempotence $x + x \approx x$, on a alors la suite infinie de réductions :

$$l =_E l + l \rightarrow_{\mathcal{R}} r + l =_E r + (l + l) \rightarrow_{\mathcal{R}} r + (r + l) \ldots$$

Un procédé général pour contourner ces difficultés est de calculer à l'aide d'une autre relation $\mathcal{R}^E$ telle que

$$\mathcal{R} \subseteq \mathcal{R}^E \subseteq \mathcal{R}/E$$

(se référer à [JOU 86b]) pour plus de détails.)

On peut définir pour la relation $\mathcal{R}^E$ une propriété de Church-Rosser et de confluence. J-P.Jouannaud et H.Kirchner [JOU 86b] définissent également une nouvelle propriété appelée *cohérence* qui établit un pont entre la relation $\mathcal{R}^E$ et la relation $\mathcal{R}/E$ en permettant d'identifier les $\mathcal{R}^E-$formes normales avec les $\mathcal{R}/E-$formes normales.

Définition 2.30 On dit qu'un couple (t_1, t_2) est $\mathcal{R}^E$-*confluent modulo* E, et on note $t_1 \downarrow_{\mathcal{R}^E} t_2$ si, et seulement si $\exists t'_1\ \exists t'_2\ t_1 \overset{*}{\rightarrow}_{\mathcal{R}^E} t'_1$, $t_2 \overset{*}{\rightarrow}_{\mathcal{R}^E} t'_2$ et $t'_1 =_E t'_2$

Définition 2.31 (cf Figure 2.1)

– $\mathcal{R}$ est $\mathcal{R}^E$-*Church-Rosser modulo* E si

$$\forall t_1\ \forall t_2\ t_1 \overset{*}{\leftrightarrow}_{\mathcal{R}/E} t_2 \Rightarrow t_1 \downarrow_{\mathcal{R}^E} t_2;$$

– $\mathcal{R}^E$ est *confluente modulo* E si

$$\forall t\ \forall t'\ \forall t''\ ((t \overset{*}{\rightarrow}_{\mathcal{R}^E} t') \wedge (t \overset{*}{\rightarrow}_{\mathcal{R}^E} t'')) \Rightarrow t' \downarrow_{\mathcal{R}^E} t'';$$

– $\mathcal{R}^E$ est *localement confluente modulo* E avec $\mathcal{R}$ si

$$\forall t\ \forall t'\ \forall t''\ ((t \rightarrow_{\mathcal{R}^E} t') \wedge (t \rightarrow_{\mathcal{R}} t'')) \Rightarrow t' \downarrow_{\mathcal{R}^E} t'';$$

– $\mathcal{R}^E$ est *cohérente modulo* E si

$$\forall t\ \forall t'\ \forall t''\ ((t \overset{+}{\rightarrow}_{\mathcal{R}^E} t') \wedge (t =_E t'')) \Rightarrow \exists t''_1\ ((t'' \rightarrow_{\mathcal{R}^E} t''_1) \wedge (t' \downarrow_{\mathcal{R}^E} t''_1));$$

– $\mathcal{R}^E$ est *localement cohérente modulo* E si

$$\forall t\ \forall t'\ \forall t''\ ((t \rightarrow_{\mathcal{R}^E} t') \wedge (t \leftrightarrow_E t'')) \Rightarrow \exists t''_1\ ((t'' \rightarrow_{\mathcal{R}^E} t''_1) \wedge (t' \downarrow_{\mathcal{R}^E} t''_1)).$$

2.5.2 Réécriture $\mathcal{R}, E$ et complétude suffisante

La relation de réécriture de Peterson et Stickel $\mathcal{R}, E$ est l'instance de $\mathcal{R}^E$ la plus couramment utilisée pour la réécriture dans les classes d'équivalences.

Définition 2.32 On définit la relation $\mathcal{R}, E$ par $s \rightarrow_{\mathcal{R},E} t$ si il existe :

– Une règle $l \rightarrow r \in \mathcal{R}$;
– Une position $\omega \in \mathcal{D}om(t)$;
– Une substitution σ appelée *filtre modulo* E de l vers $t_{|\omega}$;

telles que $t_{|\omega} =_E l\sigma$ et $t' = t[r\sigma]_\omega$.

(t_1, t_2) $\mathcal{R}^E$-confluent modulo$E$

$$\begin{array}{ccc} t_1 & & t_2 \\ * \downarrow_{\mathcal{R}^E} & & * \downarrow_{\mathcal{R}^E} \\ t'_1 & =_E & t'_2 \end{array}$$

$\mathcal{R}$ $\mathcal{R}^E$ Church Rosser modulo E

$$\begin{array}{ccc} t_1 & \overset{*}{\leftrightarrow}_{\mathcal{R}/E} & t_2 \\ * \downarrow_{\mathcal{R}^E} & & * \downarrow_{\mathcal{R}^E} \\ t'_1 & =_E & t'_2 \end{array}$$

$\mathcal{R}^E$confluente modulo E

$$\begin{array}{cccc} t & \overset{*}{\rightarrow}_{\mathcal{R}^E} t' & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u' \\ * \downarrow_{\mathcal{R}^E} & & & \|_E \\ t'' & & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u'' \end{array}$$

$\mathcal{R}^E$ cohérente modulo E

$$\begin{array}{cccc} t & \overset{+}{\rightarrow}_{\mathcal{R}^E} t' & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u' \\ \|_E & & & \|_E \\ t'' & \rightarrow_{\mathcal{R}^E} t''_1 & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u'' \end{array}$$

$\mathcal{R}^E$ localement confluente modulo E avec $\mathcal{R}$

$$\begin{array}{cccc} t & \rightarrow_{\mathcal{R}^E} t' & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u' \\ \downarrow_{\mathcal{R}} & & & \|_E \\ t'' & & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u'' \end{array}$$

$\mathcal{R}^E$ localement cohérente modulo E

$$\begin{array}{cccc} t & \rightarrow_{\mathcal{R}^E} t' & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u' \\ \updownarrow_E & & & \|_E \\ t'' & \rightarrow_{\mathcal{R}^E} t''_1 & \overset{*}{\rightarrow}_{\mathcal{R}^E} & u'' \end{array}$$

FIG. 2.1 – Définition 2.31

Remarque: On peut noter de façon plus précise $s \rightarrow_{[l \rightarrow r, \omega, \sigma]\mathcal{R},E} \ t$.

Si on suppose de nouveau que la signature Σ est la réunion disjointe d'un sous-ensemble $\mathcal{C}$ de symboles constructeurs et d'un sous-ensemble $\mathcal{D}$ de symboles définis, on peut décliner les définitions 2.24, 2.25, la proposition 2.7, et la définition 2.26 au cas de la $\mathcal{R}, E$-réécriture.

Définition 2.33 Un système de réécriture $\mathcal{R}$ et un ensemble d'égalités E étant donnés, on dit que le système de réécriture modulo $(\mathcal{R}, E)$ est suffisamment complet si, pour tout $s \in \mathcal{T}(\Sigma)$, il existe $t \in \mathcal{T}(\mathcal{C})$, tel que $s \overset{*}{\rightarrow}_{\mathcal{R},E} t$.

Définition 2.34 Un système de réécriture $\mathcal{R}$, un ensemble d'égalités E, et un symbole de fonction $f \in \mathcal{D}$ étant donnés, on dit que f est $\mathcal{R}, E$-suffisamment complet si, pour tout $t_1, \ldots t_n \in \mathcal{T}(\mathcal{C})$, il existe $t \in \mathcal{T}(\mathcal{C})$, tel que $ft_1, \ldots, t_n \overset{+}{\rightarrow}_{\mathcal{R},E} t$

Proposition 2.8 Un système de réécriture $\mathcal{R}$ et un ensemble d'égalités E étant donnés, $(\mathcal{R}, E)$ est suffisamment complet si, et seulement si, chaque symbole défini $f \in \mathcal{D}$ est $\mathcal{R}, E$-suffisamment complet.

Définition 2.35 Un système de réécriture $\mathcal{R}$, un ensemble d'égalités E, et un terme $t \in \mathcal{T}(\Sigma, \mathcal{X})$ étant donnés, une position $\omega \in \mathcal{D}om(t)$ est inductivement réductible par rapport à $\mathcal{R}, E$, si toutes les instances closes de $t_{|\omega}$ sont $\mathcal{R}, E$-reductibles.

Exemple 2.12 [COM 00] Considérons le système de réécriture modulo $(\Sigma, \mathcal{R}, E)$ défini par :
- Sortes : S
- Opérations : $a : S$; $b : S$; $f : S \rightarrow S$; $g : S \rightarrow S$.
- $\mathcal{R}$: $f(a) \rightarrow a$; $f(b) \rightarrow b$; $f(g(b)) \rightarrow b$.
- E : $g(a) \approx a$; $g(g(x)) \approx g(x)$.

alors, la position ε dans $f(g(x))$ est inductivement réductible par rapport à la relation $\mathcal{R}, E$.

Nous allons voir que les positions *définie-maximales* peuvent être de nouveau des positions $\mathcal{R}, E$ inductivement réductibles. Pour cela, il nous faut étudier d'un peu plus près la répartition des termes dans des classes de E-équivalences. Commençons par introduire quelques définitions.

Définition 2.36 Un ensemble d'égalités E est régulier si, pour toute égalité $e_1 \approx e_2 \in E$, $\mathcal{V}ar(e_1) = \mathcal{V}ar(e_2)$.

La notion de système de réécriture *structuré* apparaît dans [BER 97]. Nous proposons ici une définition analogue concernant les ensembles d'égalités :

Définition 2.37 Un ensemble d'égalités E est *structuré* si, pour toute égalité $e_1 \approx e_2 \in E$, si $e_1 \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, alors $e_2 \in \mathcal{T}(\mathcal{C}, \mathcal{X})$.

La définition 2.38 ci-dessous est une généralisation de la définition 2.37.

Définition 2.38 On dit qu'un ensemble d'égalités E préserve les constructeurs si tout terme E-équivalent à un terme constructeur est également un terme constructeur.

Si les deux membres de chaque égalité de E ont le même ensemble de variables, il suffira de vérifier que E est structuré pour s'assurer qu'il préserve les constructeurs.

Proposition 2.9 Un ensemble d'égalités E préserve les constructeurs si E est régulier et structuré.

Preuve. Il suffit de montrer que :

$$\forall s\ \forall t\ (s \in \mathcal{T}(\mathcal{C}, \mathcal{X}) \wedge s \leftrightarrow_E t) \Rightarrow t \in \mathcal{T}(\mathcal{C}, \mathcal{X})$$

Supposons donc que s et t soient deux termes tels que $s \in \mathcal{T}(\mathcal{C}, \mathcal{X})$ et $s \leftrightarrow_E t$.
Il existe alors une égalité $e_1 \approx e_2 \in E$, une position $\omega \in \mathcal{D}om(s)$, et une substitution σ, telles que $s_{|\omega} = e_1\sigma$ et $t = s[e_2\sigma]_\omega$. Or, $s \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, donc $e_1 \in \mathcal{T}(\mathcal{C}, \mathcal{X})$. Et comme on a supposé E structuré,

$$e_2 \in \mathcal{T}(\mathcal{C}, \mathcal{X}) \tag{2.1}$$

Soit $x \in \mathcal{V}ar(e_2)$. On a alors $x \in \mathcal{V}ar(e_1)$ car on a supposé $\mathcal{V}ar(e_1) = \mathcal{V}ar(e_2)$, donc $x\sigma \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, car $e_1\sigma \triangleleft s$ et $s \in \mathcal{T}(\mathcal{C}, \mathcal{X})$. Mais comme x a été choisi arbitrairement dans $\mathcal{V}ar(e_2)$, on a donc :

$$x\sigma \in \mathcal{T}(\mathcal{C}, \mathcal{X}) \text{ pour tout } x \in \mathcal{V}ar(e_2) \tag{2.2}$$

De (2.1) et (2.2) on tire $e_2\sigma \in \mathcal{T}(\mathcal{C}, \mathcal{X})$. Mais comme $s \in \mathcal{T}(\mathcal{C}, \mathcal{X})$ et $t = s[e_2\sigma]_\omega$, cela implique $t \in \mathcal{T}(\mathcal{C}, \mathcal{X})$ □

Nous pouvons maintenant identifier une situation dans laquelle les positions *définie-maximales* sont inductivement réductibles pour la relation de réécriture $\mathcal{R}, E$:

Théorème 2.5 *Etant donné un système de réécriture modulo $(\mathcal{R}, E)$ tel que :*

1. *$\mathcal{R}$ est E-terminant sur les termes clos ;*
2. *$(\mathcal{R}, E)$ est suffisamment complet ;*
3. *E préserve les constructeurs ;*

alors, pour tout terme t et pour toute position $\omega \in \mathcal{D}om(t)$, si ω est définie-maximale *dans $\mathcal{D}om(t)$, t est $\mathcal{R}, E$-inductivement réductible à la position ω.*

Preuve. Un terme t étant donné, supposons que ω soit *définie-maximale* dans t, et posons $f = t(\omega)$. Il existe alors des termes constructeurs $t_1, \ldots, t_n$ tels que $t_{|\omega} = f t_1 \ldots t_n$. Considérons une substitution close α $\mathcal{R}, E$-normalisée. $(t_{|\omega})\alpha$ n'est pas un terme constructeur, car $f \notin \mathcal{C}$, et, par définition de la $\mathcal{R}, E$-complétude suffisante, $(t_{|\omega})\alpha$ n'est pas une forme $\mathcal{R}, E$-normale close. Par conséquent, il existe une position $\omega' \in \mathcal{D}om((t_{|\omega})\alpha)$, et une règle de réécriture $l \to r \in \mathcal{R}$, telles que :

$$((t_{|\omega})\alpha)_{|\omega'} =_E l\nu \tag{2.3}$$

On peut maintenant distinguer deux cas :

Cas 1 : $\omega' \neq \varepsilon$:

alors, $((t_{|\omega})\alpha)_{|\omega'} \in T(\mathcal{C})$, car α est close $\mathcal{R}, E$-normalisée, $\omega \in \mathcal{DM}(t)$, et $\mathcal{R}, E$ est suffisamment complet. Mais (2.3) contredit l'hypothèse selon laquelle E préserve les constructeurs.

Cas 2 : $\omega' = \varepsilon$:

alors, en remplaçant dans (2.3), on obtient $(t_{|\omega})\alpha =_E l\nu$, donc, $t\alpha$ est $\mathcal{R}, E$- réductible à la position ω, et le lemme est démontré, car α a été choisie arbitrairement. $\square$

3

Preuves automatiques par récurrence

Sommaire

3.1 Preuves par consistance **42**
3.2 Preuves par récurrence explicite **46**
3.3 Preuves par récurrence implicite **50**
3.4 Récurrence et déduction modulo **53**
3.4.1 Introduction 53
3.4.2 Le mécanisme de la déduction modulo 54
3.4.3 Récurrence dans le contexte de la déduction modulo 56
3.5 Descente infinie **58**

La récurrence est utilisée pour démontrer des propositions définies sur des structures de données particulières, comme les entiers naturels, les listes, les tableaux, etc...Nous avons déjà observé qu'il existe des propositions très simples qui ne sont pourtant pas des conséquences sémantiques des axiomes définissant les opérateurs sur ces structures. Dans l'exemple 1.16 du paragraphe 1.4.1, nous avons ainsi mis en évidence une algèbre qui est un modèle de la spécification définissant l'addition sur les entiers naturels, mais nous avons observé, dans l'exemple 1.17, que cette algèbre ne satisfait pourtant pas l'égalité élémentaire $x + 0 \approx x$. Cet echec vient du fait que le support de l'algèbre considérée contient un élément (en l'occurence a) qui ne peut pas être construit à partir des symboles 0 et s. Quand, dans l'exemple 1.19, on a considéré l'algèbre $I(\Sigma, E)$, l'intention était de définir l'addition sur les *entiers naturels*, autrement dit les termes construits à partir des symboles 0 et s. Une telle structure est appelée *structure libre* engendrée par $\{0, s\}$, et c'est également un modèle de Herbrand minimal des axiomes.
La question est maintenant la suivante : comment peut-on prouver des propositions valides dans les modèles de Herbrand minimaux ? Une des approches possibles est de faire un raisonnement *par récurrence*. Si l'on interprète les axiomes dans l'algèbre $I(\Sigma, E)$, il n'est pas difficile de prouver la neutralité à droite de 0 par rapport à l'opération d'addition. On utilise le schéma d'axiomes de récurrence de la théorie de Peano : on a $0 + 0 \approx 0$ d'après la première égalité et, si $s^n(0) + 0 \approx s^n(0)$, alors $s^{n+1}(0) + 0 \approx s(s^n(0) + 0)$ d'après la deuxième égalité, et l'hypothèse de récurrence permet d'écrire $s(s^n(0) + 0) \approx s^{n+1}(0)$ et de terminer ainsi la preuve.
L'objectif de ce chapitre est dans un premier temps de décrire les différentes méthodes qui ont été élaborées pour tenter d'automatiser le raisonnement par récurrence.
Globalement, celles-ci peuvent êtres regroupées en trois catégories : les preuves par consistance, les preuves par récurrence explicites, et les preuves par récurrence implicites. Dans les preuves par consistance, on ajoute la conjecture à prouver à l'ensemble des axiomes. A l'origine, la

procédure de complétion de Knuth-Bendix était alors appliquée. Si la procédure s'effectuait avec succès, la conjecture à prouver était une conséquence inductive. Nous verrons comment H.Comon et R.Nieuwenhuis ont généralisé cette méthode par la suite. Contrairement aux preuves par consistance, les méthodes de recherche de preuve par récurrence implicite et explicite utilisent un *schéma de récurrence*. Plus précisément, elles fournissent un ensemble de termes choisis pour se substituer à la *variable de récurrence*. Les instances ainsi produites sont de nouvelles conjectures qui peuvent être simplifiées par des instances strictement inférieures de la conjecture originale (l'hypothèse de récurrence), relativement à un ordre bien-fondé. La preuve se termine quand toutes les conjectures nouvellement engendrées sont simplifiées en des propositions dont on a déjà démontré qu'elles étaient des conséquences inductives. Dans le cas des preuves par récurrence implicite, le schéma de récurrence n'est pas donné à priori. Une autre différence est que la récurrence explicite utilise un ordre sémantique, alors que la récurrence implicite un ordre syntaxique.
Après la description de ces différentes méthodes, nous expliquons comment le calcul des séquents modulo (ou déduction modulo) peut lui aussi être utilisé pour effectuer une preuve par récurrence. Celui-ci apparaîtra alors comme un cadre théorique permettant d'effectuer une synthèse entre les deux approches implicites et explicites, et c'est sur lui que se fonde notre approche. Nous parlerons pour finir succintement de *descente infinie*, qui est un principe équivalent à celui de récurrence noethérien, et qui a également été formalisé à l'aide du calcul des séquents.

3.1 Preuves par consistance

Nous nous sommes largement inspirés de [COM 01] pour ce paragraphe.

La complétion est un processus qui consiste à orienter les égalités conformément à un ordre de réduction donné et à déterminer leurs conséquences. Plus précisément, une procédure de complétion prend en entrée un ensemble d'égalités E et un ordre de réduction $<$, puis applique l'une des règles données dans [DER 01] (Chapitre 9 p 571) et que nous résumons ici :

- **Deduce** qui ajoute une paire critique
- **Delete**, **Collapse**, **Simplify**, qui simplifient les règles et les égalités.
- **Orient** qui oriente les égalités conformément à l'ordre donné.

La démarche est donc d'appliquer cette procédure à un ensemble d'égalités constitué de l'ensemble E des axiomes de la spécification auquel on a adjoint la conjecture à prouver. On construit ainsi une dérivation qui se termine si un système de réécriture convergent équivalent au système d'égalités initial a été obtenu ou si un "témoin d'incohérence" a été détecté. Dans le premier cas, la conjecture est un théorème inductif, mais pas dans le second. Cette dérivation peut également se terminer si une égalité non orientable a été détectée, ou encore ne jamais s'interrompre, et on ne peut alors rien conclure. Les premières preuves par consistance utilisent la procédure de complétion de Knuth Bendix [KNU 70]. Ainsi, D. Musser [MUS 80a] et J. Goguen [GOG 80] ont proposé la même année[MUS 80a] la méthode suivante : on suppose que E est un système de réécriture convergent et on considère un prédicat d'égalité eq complètement défini et qui vérifie $eq(s,t) \approx true$ si, et seulement si $s =_E t$. La méthode utilise la procédure de complétion de Knuth Bendix et le témoin d'incohérence sera une égalité du type $true \approx false$. Cette méthode est néanmoins restrictive, car il existe des situations où un tel prédicat eq ne peut être défini (cf [COM 01]) pour plus de détails).
Huet et Hullot ont proposé en 1982 [HUE 82] une deuxième variante à cette méthode : E est ici un système de réécriture convergent et suffisamment complet par rapport à un ensemble

de constructeurs libres. La méthode utilise la procédure de complétion de Knuth Bendix et un témoin d'incohérence sera une égalité du type $f(s_1, \ldots, s_m) \approx g(t_1, \ldots, t_n)$, avec f et g deux symboles constructeurs distincts. Comon mentionne cependant que les conditions imposées à E dans ces deux méthodes sont équivalentes [COM 01], et donc que la deuxième n'est pas moins restrictive que la première. Un progrès significatif a été réalisé par Jouannaud et Kounalis en 1986 [JOU 86a] : E est ici donné par un système de réécriture fini et convergent sur les termes clos. La méthode utilise la procédure de complétion de Knuth Bendix, et un témoin d'incohérence sera fourni par une égalité du type $s \approx t$ avec $s > t$ et s un terme possédant au moins une instance close irréductible, i.e qui n'est pas inductivement-réductible. D.Plaisted a montré le premier que la réductibilité inductive est une propriété décidable, et de nombreux algorithmes ont été proposés depuis. La méthode de Jouannaud et Kounalis a encore néanmoins des faiblesses :

1. un ensemble d'axiomes n'est pas toujours équivalent à un système de réécriture convergent sur les termes clos ;
2. la complétion peut ne jamais s'arrêter ;
3. la méthode ne peut pas prouver des conjectures non orientables, comme la commutativité de l'addition, car la procédure de complétion de Knuth Bendix échoue dans ce cas.

Les points 1 et 3 peuvent être améliorés, le point 2 est néanmoins incontournable : en effet, le premier théorème d'incomplétude de Gödel a pour conséquence qu'il existera toujours des conséquences inductives que l'on ne pourra jamais démontrer. Il s'est avéré par ailleurs nécessaire d'élaborer des stratégies, car les procédures de preuve mentionnées plus haut divergent trop souvent, comme dans l'exemple simple suivant :

Exemple 3.1 On considère le système de réécriture :

$$\begin{array}{l} x + 0 \to x \\ x + s(y) \to s(x + y) \end{array}$$

et on suppose que l'on veuille montrer l'associativité de $+$. Si on oriente l'égalité $(x + y) + z \approx x + (y + z)$ de gauche à droite, on obtient une infinité de paires critiques du type $s^n(x + y) + z \approx x + (s^n(y) + z)$.

Fribourg [FRI 86] a remarqué alors qu'il suffit de faire des superpositions entre les règles de la spécification initiale et les égalités engendrées par la procédure. Il a ensuite proposé la première méthode complète par réfutation (i.e une dérivation se termine uniquement si la conjecture à prouver est ou non un théorème inductif), mais la première méthode complète par réfutation *qui n'exige pas que E soit donné par un système de réécriture suffisamment complet par rapport à un ensemble de constructeurs libres* a été proposée par Bachmair [BAC 88]. Les hypothèses sur E et les témoins d'incohérence sont les mêmes que dans la méthode de Jouannaud et Kounalis ainsi que les égalités du type $s \approx t$, avec s et t incomparables, non réductibles sur les termes clos, et différents syntaxiquement. Mais la stratégie n'est pas la même : on ne calcule pas toutes les paires critiques, mais seules les égalités obtenues par superposition d'un axiome de E qui a été orienté sur une conjecture, et il n'est plus nécessaire d'orienter les conjectures. Si une égalité est orientable, seul le membre maximal est surréduit, alors que si elle n'est pas orientable, les deux membres seront surréduits.
Comon et Nieuwenhuis [COM 01] ont ensuite montré que toutes ces méthodes sont des instances d'un même formalisme plus général, et qui peut également s'appliquer aux clauses. A un ensemble E de clauses, il associe un ensemble de formules universellement quantifiées du premier ordre $\mathcal{A}$ qu'il nomme I-axiomatisation de E. Un ensemble $\mathcal{C}$ de clauses étant donné, la consistance

de $E \cup \mathcal{C} \cup \mathcal{A}$ implique nécessairement que $\mathcal{C}$ est valide dans le modèle de Herbrand minimal de E. Le raisonnement par récurrence se ramène ainsi à vérifier qu'un ensemble de formules du premier ordre est consistant. Nous allons donner ci-dessous la définition formelle du concept clé de I-axiomatisation et donner un aperçu du fonctionnement de cette méthode.

Définition 3.1 Une signature Σ et un ensemble E d'égalités étant donnés, une I-axiomatisation de $I(\Sigma, E)$ est un ensemble récursif $\mathcal{A}$ de formules universellement quantifiées tel que :

1. $\mathcal{I}(\Sigma, E) \models \mathcal{A}$
2. Pour tout modèle de Herbrand $\mathcal{M}$ de E, $\mathcal{M} \models \mathcal{A}$ implique $\mathcal{M}$ isomorphe à $\mathcal{I}(\Sigma, E)$.

Comon et Nieuwenhuis proposent ensuite un système d'inférence (Figure 3.1) qui est une généralisation de la procédure de complétion de Knuth Bendix et qui s'applique également aux clauses de Horn. Il fonctionne de la manière suivante : on fixe un ordre de réduction $<$ (voir définition 1.3 de la partie II) défini sur l'ensemble des termes et qui s'étend aux clauses, et on suppose donné un ensemble E de clauses. On peut alors lui associer un système de réécriture $\mathcal{R}$ convergent sur les termes clos, et défini par $s \to t \in \mathcal{R}$ si, et seulement si $E \models s \approx t$ et t est minimal (pour $<$) dans l'ensemble $\{u \mid u \neq s,\ E \models s \approx u\}$. On suppose également que $\mathcal{A}$ est une I-axiomatisation *normale* de $\mathcal{I}(\Sigma, E)$, dont on donne ci-dessous la définition :

Définition 3.2 Une I-axiomatisation $\mathcal{A}$ de $\mathcal{I}(\Sigma, E)$ est *normale* si, pour tous termes clos $s_1, \ldots, s_n$ et tout symbole de prédicat P,

1. $s_1 \downarrow_{\mathcal{R}} \neq s_2 \downarrow_{\mathcal{R}}$ implique $\mathcal{A} \models \neg s_1 \downarrow_{\mathcal{R}} \approx s_2 \downarrow_{\mathcal{R}}$;
2. $E \not\models P(s_1 \downarrow_{\mathcal{R}}, \ldots, s_n \downarrow_{\mathcal{R}})$ implique $\mathcal{A} \models \neg P(s_1 \downarrow_{\mathcal{R}}, \ldots, s_n \downarrow_{\mathcal{R}})$.

Exemple 3.2 [COM 01] Considérons la spécification donnée dans l'exemple 1.16. Posons :

$$\mathcal{A} = \begin{cases} \neg s(x) \approx 0 \\ s(x) \approx s(y) \Rightarrow x \approx y \end{cases}$$

$\mathcal{A}$ est une I-axiomatisation de $I(\Sigma, E)$.

On part alors d'un ensemble $\mathcal{C}_0$ de conjectures à prouver, et on construit une dérivation $\mathcal{C}_0,\ \mathcal{C}_1,\ \mathcal{C}_n, \ldots$ qui est telle que chaque $\mathcal{C}_{i+1}$ est obtenu à partir de $\mathcal{C}_i$ en ajoutant une clause P_i telle que :

- P_i est obtenue à partir de $\mathcal{C}_i$ et du système d'inférence
- P_i n'est pas redondante, i.e elle possède une instance close qui n'est pas une conséquence de E, $\mathcal{A}$, et d'éléments plus petits de $\mathcal{C}_i$ relativement à l'ordre $<$.

Si à l'étape n ce n'est plus possible, on dit que l'ensemble $\mathcal{C} = \mathcal{C}_n$ est saturé. Le résultat remarquable est le suivant : Il suffit que $\mathcal{C}$ soit consistant avec $\mathcal{A}$ pour que $I(\Sigma, E) \models \mathcal{C}$. La preuve s'effectue en deux étapes :

1. on montre que si $E \cup \mathcal{C} \cup \mathcal{A}$ est consistant, alors $\mathcal{I}(\Sigma, E) \models \mathcal{C}$;
2. on montre ensuite que si $E \cup \mathcal{C} \cup \mathcal{A}$ est inconsistant, alors $\mathcal{C} \cup \mathcal{A}$ est également inconsistant.

Preuve. (schéma)

1. Si $E \cup \mathcal{C} \cup \mathcal{A}$ est consistant, cet ensemble possède au moins un modèle de Herbrand minimal, car c'est un ensemble de clauses universellement quantifiées, et par 2. de la définition 3.1, ce ne peut être que $\mathcal{I}(\Sigma, E)$.

– Superposition de conjectures :

$$\frac{D \vee l \approx r \quad C \vee L[s]}{D\sigma \vee C\sigma \vee L[r]\sigma}$$

si
– $\sigma = mgu(l, s)$,
– s n'est pas une variable,
– $r\sigma \not\geq l\sigma$, $D\sigma \not\geq l\sigma$,
– $D \vee l \approx r \in E$,
– $C \vee L[s] \in \mathcal{C}$.

– Résolution d'égalités :

$$\frac{C \vee \neg s \approx t}{C\sigma}$$

si
– $\sigma = mgu(s, t)$,
– $C \vee \neg s \approx t \in \mathcal{C}$.

– Résolution binaire ordonnée :

$$\frac{D \vee P(s_1, \ldots, s_n) \quad C \vee \neg P(t_1, \ldots, t_n)}{D\sigma \vee C\sigma}$$

si
– $\sigma = mgu(P(s_1, \ldots, s_n), P(t_1, \ldots, t_n))$,
– $D \vee P(s_1, \ldots, s_n) \in E$,
– $C \vee \neg P(t_1, \ldots, t_n) \in \mathcal{C}$,
– $P(s_1, \ldots, s_n)\sigma \not\leq D\sigma$.

– Factorisation ordonnée

$$\frac{C \vee L \vee L'}{C\sigma \vee L\sigma}$$

si
– $\sigma = mgu(L, L')$,
– $L\sigma$ est maximal dans $(C \vee L \vee L')\sigma$,
– $C \vee L \vee L' \in \mathcal{C}$.

FIG. 3.1 – Système d'inférence de Comon et Nieuwenhuis

2. Si $E \cup \mathcal{C} \cup \mathcal{A}$ est inconsistant, on peut alors déterminer un élément $c \in \mathcal{C}$ et une substitution close σ des variables de c tels que $c\sigma \cup \mathcal{C} \cup \mathcal{A}$ soit inconsistant (sinon $\mathcal{I}(\Sigma, E)$ serait un modèle de toutes ces formules, donc de $E \cup \mathcal{C} \cup \mathcal{A}$, car c'est un modèle de Herbrand). On considère alors un élément minimal parmi ces clauses $c\sigma$, et sachant que $\mathcal{A}$ est une I- axiomatisation *normale*, on peut démontrer que tout littéral L de $c\sigma$ est inconsistant avec $\mathcal{A}$.

□

Comon vérifie ensuite que toutes les méthodes de recherche de preuve par consistance précédentes ne sont finalement que des instances de sa méthode, il suffit pour cela de trouver la I-axiomatisation normale adéquate :

– Approche de Musser [MUS 80a] et Goguen [GOG 80] :

$$\mathcal{A} = \mathcal{D} \cup \{\neg s \approx t \Leftrightarrow eq(s,t) \approx false\}$$

avec $\mathcal{D}$ le système d'égalités définissant le prédicat eq

– Approche de Huet et Hullot [HUE 82] :

$$\begin{aligned}\mathcal{A} = \quad & \{f(x_1,\ldots,x_n) \approx f(y_1,\ldots,y_n) \Rightarrow x_1 \approx y_1 \wedge \ldots \wedge x_n \approx y_n \mid f \in \mathcal{C}\} \\ & \cup\{\neg f(x_1,\ldots,x_n) \approx g(y_1,\ldots,y_m) \mid f,g \in \mathcal{C}, f \neq g\}\end{aligned}$$

avec $\mathcal{C}$ un ensemble de constructeurs libres.

– Approche de Jouannaud et Kounalis [JOU 86a] :

$$\mathcal{A} = \{\neg s \approx t \mid s,t \in \mathcal{T}(\Sigma), s > t, \neg Red(s)\}$$

avec Red le symbole de prédicat de réductibilité inductive.

– Approche de Bachmair [BAC 88] :
même axiomatisation, mais la stratégie d'application des règles est différente.

La stratégie peut être affinée. Il est ainsi possible de restreindre le calcul des paires critiques au niveau de positions inductivement réductibles (ou encore nommées inductivement complètes). Plus généralement, on peut restreindre le calcul des paires critiques au niveau d'un ensemble *inductivement complet* de positions :

Définition 3.3 Un ensemble $\{\omega_1,\ldots,\omega_n\}$ est un ensemble *inductivement complet* de positions d'un terme t si, pour toute substitution close irréductible σ, l'un des sous-termes de $t\sigma$ à la position $\omega_1,\ \omega_2,\ldots,\ \omega_n$ est réductible.

Une position inductivement réductible, donc à fortiori une position *définie maximale* dans le cas d'un système de réécriture terminant sur les termes clos et suffisamment complet, peut ainsi être considérée comme un ensemble inductivement complet de positions réduit à un seul élément.

3.2 Preuves par récurrence explicite

Nous nous sommes largement inspirés de [BUN 99] pour écrire cette section.
On suppose pour simplifier que P est une proposition contenant une seule variable libre x de type τ, et que $\prec$ est une relation noethérienne définie sur l'ensemble des objets de type τ. Une preuve par récurrence explicite de la proposition P se déroule alors selon les étapes suivantes :

1. On considère le principe de récurrence noethérien que l'on peut exprimer sous la forme d'une règle d'inférence :

$$\frac{\forall x : \tau\ (\forall y : \tau\ y \prec x \Rightarrow P(y)) \Rightarrow P(x)}{\forall x : \tau\ P(x)}$$

2. On instancie P, $\prec$ et τ par des valeurs appropriées à la situation, par exemple :

$$\frac{\forall x : Nat\ (\forall y : Nat\ y \prec x \Rightarrow 0 + y \approx y) \Rightarrow 0 + x \approx x}{\forall x : Nat\ 0 + x \approx x}$$

3. On détermine une condition suffisante CS pour que la proposition située dans le membre supérieur de la règle soit réalisée et qui soit plus commode à vérifier (et cela se ramène souvent à faire un raisonnement par cas). Dans l'exemple précédent, cela peut être :

$$(0 + 0 \approx 0) \wedge \forall x : Nat\ (0 + x \approx x \Rightarrow 0 + s(x) \approx s(x)) \tag{3.1}$$

En effet, supposons (3.1) Prenons maintenant $x : Nat$, et supposons :

$$\forall y : Nat\ y \prec x \Rightarrow 0 + y \approx y \tag{3.2}$$

Si $x = 0$, on a $0 + x \approx x$ d'après (3.1). Sinon, il existe y tel que $x = s(y)$. Or $0 + y \approx y$, car $y \prec x$, et l'hypothèse (3.2). On en déduit alors, d'après (3.1), l'égalité $0 + s(y) \approx s(y)$, soit $0 + x \approx x$. Et comme x a été choisi arbitrairement, on a donc bien :

$$\forall x : Nat\ (\forall y : Nat\ y \prec x \Rightarrow 0 + y \approx y) \Rightarrow 0 + x \approx x$$

4. On obtient une nouvelle règle d'inférence :

$$\frac{CS}{\forall x : \tau\ P(x)}$$

Dans l'exemple précédent, cela donne :

$$\frac{(0 + 0 \approx 0) \wedge \forall x : Nat\ (0 + x \approx x \Rightarrow 0 + s(x) \approx s(x))}{\forall x : Nat\ 0 + x \approx x}$$

Le raisonnement va s'effectuer à l'aide de la règle ainsi obtenue, que l'on nomme "règle de récurrence", ou "schéma de récurrence" . Comme on peut se l'imaginer, le choix de cette règle de récurrence est primordial pour le bon déroulement de la preuve. D'après le premier théorème de Gödel, il ne sera jamais possible de trouver systématiquement la règle de récurrence la mieux adaptée à la preuve d'une conjecture donnée, et cette étape du raisonnement constitue un **premier point de branchement infini**. Cette restriction est également une conséquence de l'indécidabilité du problème de l'arrêt : Turing a en effet démontré qu'il ne pourra jamais exister d'algorithme capable de déterminer si une relation arbitraire donnée en entrée est noethérienne. On ne peut donc pas imaginer un programme capable d'instancier le principe de récurrence noethérien par toutes les relations noethériennes existantes sur un type donné τ. Il est donc nécessaire d'imaginer des heuristiques pour construire ces règles de récurrence, et celles-ci vont dépendre de la conjecture à prouver : l'heuristique standard, appelée "analyse récursive", utilise les définitions des fonctions récursives qui apparaissent dans la conjecture. Considérons l'exemple de Bundy [BUN 99] :

Exemple 3.3 [BUN 99] On définit le prédicat *even* à l'aide du système de réécriture suivant :

$$\begin{array}{l} even(0) \rightarrow true \\ even(s(0)) \rightarrow false \\ even(s(s(n))) \rightarrow even(n) \end{array}$$

Imaginons que l'on cherche à prouver la conjecture :

$$\forall x\ \forall y\ even(x) \wedge even(y) \rightarrow even(x + y)$$

even est une fonction définie de façon récursive et qui possède en argument la variable universellement quantifiée x. On peut alors choisir x comme variable de récurrence, poser $P(x) = \forall y\ even(x) \wedge even(y) \rightarrow even(x+y)$ et proposer la règle de récurrence :

$$\frac{P(0),\ P(s(0)),\ \forall x\ (P(x) \rightarrow P(s(s(x))))}{\forall x\ P(x)} \tag{3.3}$$

La présence de $even(y)$ dans la conjecture suggère également la même règle de récurrence, mais en posant y comme variable de récurrence. Cependant, le symbole $+$ qui apparaît dans $even(x+y)$ est également défini récursivement (voir Exemple 3.1) et suggère de choisir y comme variable de récurrence, donc de poser $P'(y) = \forall x\ even(x) \wedge even(y) \rightarrow even(x+y)$ et de proposer la règle de récurrence :

$$\frac{P'(0), \forall y\ (P'(y) \rightarrow P'(s(y))}{\forall y\ P'(y)} \tag{3.4}$$

On voit sur cet exemple que de nombreuses règles de récurrence peuvent être proposées pour un même problème, et qu'il faut faire appel à d'autres stratégies pour sélectionner la meilleure. Dans l'exemple ci-dessus, choisir y comme variable de récurrence est davantage pertinent que choisir x, car y apparaît toujours en position d'argument récursif : instancier la variable y dans la conjecture permet ensuite d'appliquer davantage de règles de réécriture, et d'augmenter ainsi le nombre de chances de pouvoir utiliser l'hypothèse de récurrence.
Une fois la règle de récurrence choisie, de nombreuses techniques sont utilisées pour montrer l'étape de récurrence. L'une des plus classique, appelée *fertilisation*, peut être illustrée par l'exemple suivant :

Exemple 3.4 [BUN 99] Considérons la spécification suivante :
- Sortes : τ, $list(\tau)$.
- Opérations : $Nil : list(\tau)$, $:: : \tau \times list(\tau) \rightarrow list(\tau)$, $<> : list(\tau) \times list(\tau) \rightarrow list(\tau)$.
- Axiomes : $(Nil <> l) \approx l$; $((h :: t) <> l) \approx (h :: (t <> l))$.

Les égalités de E sont une définition récursive du prédicat $<>$ définissant la concaténation de deux listes, et on peut alors proposer les règles de réécriture suivantes :

$$\begin{array}{ll} L_1 : & nil <> l \rightarrow l \\ L_2 : & (h :: t) <> l \rightarrow h :: (t <> l) \end{array}$$

On définit également l'égalité conditionnelle :

$$L_3 : \quad ((h \approx h') \wedge (l \approx l')) \Rightarrow (h :: l \approx h' :: l')$$

On se propose maintenant de démontrer par récurrence l'associativité de $<>$:

$$\forall x : list(\tau)\ \forall y : list(\tau)\ \forall z : list(\tau)\ x <> (y <> z) \approx (x <> y) <> z$$

On peut choisir x comme variable de récurrence. On introduit P la proposition définie par $\forall y\ : list(\tau) \forall z\ : list(\tau)\ x <> (y <> z) \approx (x <> y) <> z$ et on propose la règle de récurrence suivante :

$$\frac{P(nil),\ \forall x : list(\tau)\ \forall h : \tau\ P(x) \Rightarrow P(h :: x)}{\forall l\ P(l)}$$

L'étape de récurrence est :

$$\begin{array}{l} \forall x : list(\tau)\ [\forall y : list(\tau)\ \forall z : list(\tau)\ x <> (y <> z) \approx (x <> y) <> z \Rightarrow \\ \qquad\qquad \forall y : list(\tau)\ \forall z : list(\tau)\ ((h :: x) <> (y <> z)) \approx ((h :: x) <> y) <> z)] \end{array}$$

Observons que la quantification sur x porte sur toute l'étape de récurrence, alors que les deux quantifications sur y et sur z sont restreintes à l'hypothèse de récurrence et à sa conclusion. Il est courant alors d'ôter les quantificateurs et de remplacer le symbole d'implication $\Rightarrow$ par le symbole $\vdash$. Dans ce format, l'étape de récurrence devient le séquent :

$$x <> (Y <> Z) \approx (x <> Y) <> Z \vdash ((h :: x) <> (y <> z)) \approx (((h :: x) <> y) <> z)$$

Remarquons que la variable de récurrence, x, est considérée comme une constante arbitraire, qui apparaît à la fois dans la partie gauche et la partie droite du séquent. Les autres variables y et z sont considérées elles aussi comme des constantes arbitraires dans la partie droite, mais, en revanche, comme des variables libres dans la partie gauche. Ces transformations sont le résultat d'une double skolémisation. On peut alors alors effectuer la suite de réductions suivante :

$$\begin{array}{rcll} x <> (Y <> Z) \approx (x <> Y) <> Z & \vdash & ((h :: x) <> (y <> z)) \approx (((h :: x) <> y) <> z) & \\ & \vdash & (h :: (x <> (y <> z))) \approx ((h :: (x <> y)) <> z) & (L_2) \\ & \vdash & (h :: (x <> (y <> z))) \approx (h :: ((x <> y) <> z)) & (L_2) \\ & \vdash & (h \approx h) \wedge (x <> (y <> z)) \approx ((x <> y) <> z) & (L_3) \end{array}$$

On voit maintenant qu'il suffit d'appliquer l'hypothèse de récurrence pour finir la preuve.

Dans l'exemple précédent, l'hypothèse de récurrence a été écrite de façon schématique sous la forme $IH \vdash IC[IH\sigma]$, et on conclut en apliquant la règle d'inférence :

$$\frac{IH \vdash IC[\mathbb{T}]}{IH \vdash IC[IH\sigma]}$$

que l'on appelle *fertilisation* : l'hypothèse de récurrence "fertilise" la conclusion. Parfois, un lemme est nécessaire pour montrer l'étape de récurrence.

Exemple 3.5 [BUN 99] Considérons la fonction rev qui retourne une liste et qui est définie par les règles :

$$\begin{array}{ll} L_4 : & rev(nil) \rightarrow nil \\ L_5 : & rev(h :: t) \rightarrow rev(t) <> (h :: nil) \end{array}$$

Supposons que l'on veuille démontrer la conjecture $\forall l\ rev(rev(l)) \approx l$. L'analyse récursive suggère de choisir l comme variable de récurrence. L'étape de récurrence peut alors s'écrire :

$$\begin{array}{rcll} rev(rev(t)) \approx t & \vdash & (rev(rev(h :: t)) \approx h :: t) & \\ & \vdash & rev(rev(t) <> (h :: nil)) \approx h :: t & (L_5) \end{array}$$

Mais à partir d'ici, plus aucune règle ne s'applique. Une solution est alors d'introduire la règle suivante :

$$L_6 : \quad rev(l <> l') \rightarrow rev(l') <> rev(l)$$

La preuve pourra alors continuer et s'écrira :

$$\begin{array}{rcll} rev(rev(t)) \approx t & \vdash & rev(rev(t) <> (h :: nil)) \approx h :: t & \\ & \vdash & (rev(h :: nil) <> rev(rev(t))) \approx h :: t & (L_6) \\ & \vdash & ((rev(nil) <> (h :: nil)) <> rev(rev(t))) \approx h :: t & (L_5) \\ & \vdash & ((nil <> (h :: nil)) <> rev(rev(t))) \approx h :: t & (L_4) \\ & \vdash & ((h :: nil) <> rev(rev(t))) \approx h :: t & (L_1) \\ & \vdash & (h :: (nil <> rev(rev(t)))) \approx h :: t & (L_2) \\ & \vdash & (h :: rev(rev(t))) \approx h :: t & (L_1) \\ & \vdash & (h \approx h) \wedge (rev(rev(t)) \approx t) & (L_3) \end{array}$$

L'introduction d'un lemme dans un raisonnement est un usage implicite de la règle de coupure du calcul des séquents de Gentzen :

$$\frac{A, \Gamma \vdash \Delta \quad \Gamma \vdash A}{\Gamma \vdash \Delta} \; cut$$

la proposition A représentant le lemme. L'utilisation de cette règle constitue un **second point de branchement infini** : en effet A est à priori arbitraire. Gentzen a montré que cette règle est redondante pour les théories du premier ordre, mais Kreisel a prouvé en revanche qu'elle ne l'était pas pour les théories inductives. Il est également parfois nécessaire de généraliser la conjecture pour pouvoir la démontrer. Il faut alors inventer de nouvelles heuristiques pour décider quand il faut généraliser et quel lemme intermédiaire on peut proposer. Par exemple, l'heuristique appelée Rippling est introduite dans [BUN 93].

3.3 Preuves par récurrence implicite

La *récurrence implicite* (ou *récurrence par réécriture*) part de l'observation suivante : si le système de réécriture est terminant, la relation de réécriture peut être utilisée pour comparer les termes quand on effectue un raisonnement par récurrence. Reddy [RED 90] propose une méthode destinée à prouver par récurrence des égalités dans une spécification E dont les égalités peuvent être orientées de telle sorte que le système de réécriture $\mathcal{R}$ est terminant. Sa méthode repose sur le concept d'ensemble couvrant :

Définition 3.4 Soit $\mathcal{R}$ un système de réécriture terminant. Un ensemble $TS = \{t_i\}_i$ est un ensemble couvrant, si, pour tout terme clos $\mathcal{R}$-irréductible s, il existe i et une substitution σ tels que $s =_{\mathcal{R}} t_i\sigma$

Une substitution σ est alors dite couvrante si chaque variable de son domaine a pour image un élément de TS dont les variables ont été renommées. Il suffit alors de montrer qu'une égalité est vraie lorsque les variables sont instanciées par des substitutions couvrantes pour montrer que c'est une conséquence inductive

Exemple 3.6 [RED 90] Considérons la spécification donnée dans l'exemple 3.1, ainsi que l'égalité $0 + x \approx x$. Les instances couvrantes de la variable x sont $\{x \to 0, x \to s(y)\}$, il s'agit donc de montrer les deux sous-buts $0 + 0 \overset{*}{\leftrightarrow}_{\mathcal{R}} 0$ et $0 + s(y) \overset{*}{\leftrightarrow}_{\mathcal{R}} s(y)$.

1. On a $0 + 0 \to_{\mathcal{R}} 0$
2. $0 + s(y) \to_{\mathcal{R}} s(0 + y) \leftrightarrow_{0+x\approx x} s(y)$

Si on suppose que $<$ est un ordre de simplification (voir définition 1.5 de la partie II) permettant de montrer la terminaison de $\mathcal{R}$, l'égalité $0+x \approx x$ est appliquée à une paire de termes plus petite : $\{0+y, y\} <<_{\mathcal{R}} \{0+s(y), s(y)\}$ car $0+y$ est un sous-terme de $s(0+y)$, et $0+s(y) \to_{\mathcal{R}} s(0+y)$.

Reddy élabore à partir de ce principe un système de recherche de preuve qui comporte trois règles d'inférence : *Expand*, *Simplify* et *Delete*. La procédure va alors construire une dérivation du type $(E_0, H_0), \ldots, (E_n, H_n), \ldots$. E_i correspond à l'ensemble des sous-buts à l'étape i, et H_i à l'ensemble des hypothèses de récurrence. L'application de *Expand* à l'étape i consiste à remplacer une égalité de l'ensemble E_i qui a été orientée par toutes les paires critiques obtenues à l'aide d'une substitution couvrante, et à enrichir l'ensemble H_i de cette égalité qui est devenue une nouvelle hypothèse de récurrence. *Simplify* correspond à la réécriture d'un sous-but à l'aide

d'une règle de $\mathcal{R}$ et de l'hypothèse de récurrence, et *Delete* à la suppression d'une égalité triviale. E_0 est ainsi constitué des conjectures à prouver, et si, à une étape n, on a $E_n = \emptyset$, les conjectures de E_0 sont des théorèmes inductifs. Reddy a montré que son système de recherche de preuve est une restriction de la procédure de complétion de Knuth Bendix (dans le sens où les nouvelles règles produites doivent toutes posséder un membre gauche inductivement réductible). On peut relever trois points faibles dans cette méthode :

1. elle ne s'applique pas aux clauses ;
2. elle ne s'applique pas quand $\mathcal{R}$ est un système de réécriture conditionnel ;
3. elle échoue si une égalité de l'ensemble des sous-buts n'est pas orientable (elle n'est donc pas réfutationnellement complète).

Elle a néanmoins l'avantage de pouvoir s'appliquer si $\mathcal{R}$ n'est pas confluent.
Pour remédier à ces défauts, plusieurs approches ont été proposées. L'une d'entre elles est d'autoriser la réécriture d'un terme à l'aide d'égalités non orientées, à partir du moment où cela fait décroître la clause dans lequel ce terme apparaît pour un certain ordre noethérien. Kounalis et Rusinowitch [KOU 90] ont ainsi élaboré une méthode de recherche de preuve par réécriture qui s'applique aux clauses, qui s'applique au cas où $\mathcal{R}$ est conditionnel, et qui est réfutationnellement complète. La contre-partie est qu'elle exige que $\mathcal{R}$ soit convergent sur l'ensemble des termes clos. Celle-ci repose sur le concept clé *d'ensemble test*. Une définition des ensembles tests pour les théories conditionnelles a été introduite dans [BOU 92a]. Il existe des algorithmes de calcul dans le cas d'une théorie équationnelle, pas dans celui d'une théorie conditionnelle. On dira ici de manière approximative qu'un ensemble test est un ensemble couvrant qui est une description finie du modèle initial. Les ensembles tests peuvent être considérés comme des schémas de récurrence raffinés. Considérons l'exemple suivant :

Exemple 3.7 [BOU 97] Supposons que $\mathcal{R}$ soit donné par le système de réécriture suivant

$$\begin{array}{l} even(0) \to \mathbb{T} \\ even(s(0)) \to \mathbb{F} \\ even(s(s(x))) \to even(x) \\ even(x) \approx \mathbb{T} \Rightarrow odd(x) \to \mathbb{F} \\ even(s(x)) \approx \mathbb{T} \Rightarrow odd(x) \to \mathbb{T} \end{array}$$

et que l'on veuille démontrer la conjecture $even(x) \approx \mathbb{T} \vee odd(x) \approx false$. Un ensemble test est $\{0, s(0), s(s(x)), \mathbb{T},\ \mathbb{F}\}$ et on peut facilement montrer la validité de cette conjecture dans le modèle initial à l'aide de cet ensemble, alors que la démonstration échoue si on considère l'ensemble couvrant $\{0,\ s(x),\ \mathbb{T},\ \mathbb{F}\}$.

Les ensembles tests permettent également de réfuter des conjectures par la construction d'un contre-exemple.

Définition 3.5 Un système de réécriture $\mathcal{R}$ et un ensemble test $\mathcal{S}(\mathcal{R})$ étant donnés, une clause $C = g_1 \approx d_1 \vee \ldots \vee g_n \approx d_n$ est un témoin d'incohérence par rapport à $\mathcal{R}$ si il existe une instance test σ de C telle que, pour tout $1 \leq j \leq n$, $g_j\sigma \neq d_j\sigma$ et il est impossible de trouver un filtre entre un membre gauche d'une règle de $\mathcal{R}$ et un élément maximal de $\{g_j\sigma, d_j\sigma\}$ par rapport à $>$ (on dit qu'ils sont alors fortement irréductibles).

Le résultat fondamental est alors le suivant :

Théorème 3.1 *Si la spécification est donnée par un système convergent sur les termes clos $\mathcal{R}$ et un ensemble de constructeurs libres, et si une clause C est un témoin d'incohérence par rapport à $\mathcal{R}$, alors $\mathcal{C}$ n'est pas une conséquence inductive de $\mathcal{R}$.*

Preuve: (Schéma) Sachant que les éléments maximaux de $\{g_j\sigma, d_j\sigma\}$ sont fortement irréductibles, on peut montrer qu'il en existe une τ-instance close irréductible et qui soit injective. On peut alors facilement vérifier que les éléments maximaux de $\{g_j\sigma\tau, d_j\sigma\tau\}$ sont irréductibles et distincts, et que par conséquent l'on ne peut pas avoir une équivalence du type $g_j\sigma\tau \overset{*}{\leftrightarrow}_{\mathcal{R}} d_j\sigma\tau$ $\square$

L'exemple suivant illustre l'application de ce théorème.

Exemple 3.8 [BOU 95] Considérons le système de réécriture $\mathcal{R}$ suivant :

$$\begin{array}{l} union(Nil, Nil) \to Nil \\ union(x :: l, Nil) \to x :: l \\ union(Nil, x :: l) \to x :: l \\ union(x :: l, y :: l') \to x :: (y :: union(l, l')) \end{array}$$

et supposons que l'on cherche à montrer la conjecture $union(l, l') \approx union(l', l)$. $\mathcal{R}$ est convergent, et un ensemble test est $\{Nil, x :: l, 0, s(x)\}$. La conjecture a donc une instance test $union(x :: l, y :: l') \approx union(y :: l', x :: l)$, qui se simplifie en $x :: (y :: union(l, l')) \approx y :: (x :: union(l, l'))$, dont l'une des instances tests est $x :: (y :: union(Nil, Nil)) \approx y :: (x :: union(Nil, Nil))$, qui se simplifie en $x :: (y :: Nil) \approx y :: (x :: Nil)$, mais cette dernière conjecture est un témoin d'incohérence, ce qui implique que la conjecture $union(l, l') \approx union(l', l)$ n'est pas valide dans le modèle initial.

A partir de cette notion d'ensemble test, Bouhoula et Rusinowitch [BOU 95] ont proposé une procédure de recherche de preuve, dont on va expliquer brièvement la démarche : dans toute conjecture C à démontrer on identifie un ensemble particulier de positions, dont les éléments sont appelés positions de récurrence. Cet ensemble est constitué en général des positions situées au niveau d'arguments récursifs de symboles définis de C, et peut être considéré comme une extension de l'ensemble des positions inductivement complètes proposé par Fribourg. Ces variables sont par la suite instanciées par des éléments de l'ensemble test, puis simplifiées par les axiomes, les hypothèses de récurrence, et éventuellement d'autres conjectures. La méthode est capable de manipuler des égalités non orientables, et elle est réfutationnellement complète si le système de réécriture est convergent sur les termes clos et suffisamment complet par rapport à un ensemble de constructeurs libres.
Une autre approche pour pouvoir démontrer des conjectures non orientables par récurrence implicite a été d'utiliser la réécriture modulo. Aoto [AOT 06] a ainsi proposé un système d'inférence qui est une généralisation de celui de Reddy. Etant donné une signature Σ, qui est la réunion disjointe d'un sous-ensemble $\mathcal{C}$ de symboles constructeurs et d'un sous-ensemble $\mathcal{D}$ de symboles définis, il nomme "basiques" les termes de la forme $ft_1 \ldots t_n$, avec $f \in \mathcal{D}$, et $t_i \in \mathcal{T}(\mathcal{C}, \mathcal{X})$. Il introduit alors les notations suivantes. Trois termes s, t, u étant donnés,

- $B(s)$ désigne l'ensemble des sous-termes basiques de s ;
- $Expd_u(s, t) = \{C[r]\sigma \approx t\sigma \mid s = C[u],\ \sigma = mgu(u, l),\ l \to r \in \mathcal{R},\ l \text{ basique }\}$;
- $Expd_{u,v}(s, t) = \bigcup\{Expd_{v\sigma}(t\sigma, s') \mid s' \approx t\sigma \in Expd_u(s, t)\}$.

En posant $\leqslant$ un quasi-ordre défini sur l'ensemble des termes dont $\gtrless$ désignera l'équivalence associée, le système d'inférence est présenté à la figure 3.2. La correction de cette méthode repose sur le théorème suivant :

Théorème 3.2 *Soit $\mathcal{R}$ un système de réécriture suffisamment complet, E un ensemble d'égalités, $\leqslant$ un quasi-ordre de réduction tel que $\mathcal{R} \subseteq \leqslant$. Si il existe des ensembles H, G, tels que $< E, \emptyset, \emptyset > \overset{*}{\to} < \emptyset, H, G >$, alors les égalités de E sont des théorèmes inductifs de $\mathcal{R}$.*

– Simplify :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E \cup \{s' \approx t\}, H, G >} \; s \rightarrow_{(\mathcal{R} \cup H)/G} s'$$

– Delete :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E, H, G >} \; s \overset{*}{\leftrightarrow}_G t$$

– Expand :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E \cup Expd_u(s,t), H \cup \{s \rightarrow t\}, G >} \; u \in B(s),\ s > t$$

– Expand2 :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E \cup Expd2_{u,v}(s,t), H, G \cup \{s \approx t\} >} \; u \in B(s),\ v \in B(t),\ s \gtrless t$$

FIG. 3.2 – Système d'inférence de Aoto

Avec ce système, la preuve de la commutativité de l'addition dans l'arithmétique de Peano (voir exemple 3.1) s'effectue de la façon suivante :

$$\begin{array}{ll}
 & < \{x + y \approx y + x\}, \emptyset, \emptyset\} > \\
\rightarrow_{E2} & \left\langle \begin{array}{l} \{\ 0 \approx 0, s(x_1) \approx s(x_1 + 0), s(x_2 + s(y_2)) \approx s(y_2 + s(x_2))\ \} \\ \emptyset, \{x + y \approx y + x\} \end{array} \right\rangle \\
\rightarrow_D \circ \rightarrow_S \circ \rightarrow_D & \left\langle \begin{array}{l} \{\ s(x_2 + s(y_2)) \approx s(y_2 + s(x_2))\ \} \\ \emptyset, \{x + y \approx y + x\} \end{array} \right\rangle \\
\rightarrow_S \circ \rightarrow_S \circ \rightarrow_D & \langle\ \emptyset, \emptyset, \{x + y \approx y + x\}\ \rangle
\end{array}$$

3.4 Récurrence et déduction modulo

3.4.1 Introduction

La déduction modulo permet de distinguer la partie calcul d'un raisonnement de sa partie déduction : le calcul ne nécessite qu'une exécution aveugle, alors que la déduction impose de faire des choix. La déduction modulo repose sur le fait que le calcul peut être défini par l'utilisation d'une congruence décidable sur les propositions. De telles congruences décidables peuvent souvent être définies par des systèmes de réécriture confluents et terminants (pour montrer que deux termes sont congruents, il suffit de vérifier qu'ils possèdent la même forme normale). La réécriture pourra alors être utilisée de façon déterministe pour tester la congruence de deux termes.

Exemple 3.9 [DOW 03b] (Réécriture appliquée à des termes). On considère deux formalismes :

– Premier formalisme :

$$\forall x\ \forall y\ \forall z\ (x + y) + z \approx x + (y + z)$$
$$\forall x\ x \approx x$$

– Deuxième formalisme : $(x + y) + z \rightarrow x + (y + z)$

et on considère la proposition C suivante :
$((a + b) + ((c + d) + e) \approx a + ((b + c) + (d + e))$.

Démontrer que C n'est pas un théorème dans le premier formalisme n'est pas simple, alors qu'elle se fait automatiquement dans le second.

Une théorie utilisateur Th_u étant données, notre but est donc d'effectuer une preuve d'une proposition P à l'aide du principe de récurrence noethérien noté $NoethInd$, c'est à dire de trouver une dérivation du séquent :$NoethInd, Th_u \vdash P$. Au cours d'une preuve, $NoethInd$ n'est instanciée que par un nombre fini de propositions : ce sont les lemmes intermédiaires démontrés par récurrence. Le principe de récurrence noethérien étant par essence une proposition du second ordre, c'est en fait un séquent de la logique d'ordre supérieur. Puisque nous voulons utiliser largement les concepts et techniques de réécriture du premier ordre, et également considérer les théories du premier ordre, nous avons besoin d'une présentation du premier ordre de la logique d'ordre supérieur. Pour cela, on utilise $HOL_{\lambda\sigma}$, introduite dans [DOW 01], qui est fondée sur la déduction modulo [DOW 03b], et qui semble particulièrement appropriée à notre étude pour les raisons déjà mentionnées plus haut. Dans la sous-section suivante, nous n'allons pas expliquer en détail l'approche complète, mais seulement présenter la démarche. Le lecteur intéressé peut se référer à [DEP 02] et à [DEP 04] pour des explications complémentaires.

3.4.2 Le mécanisme de la déduction modulo

En déduction modulo, les termes, mais aussi les propositions, peuvent être identifiées modulo une congruence. Nous utiliserons une congruence qui peut typiquement être définie par des égalités conditionnelles, et qui tient compte du contexte pour évaluer les conditions. En outre, puisque l'application de cette congruence doit être contrôlée étroitement, un concept approprié de symbole protecteur est utilisé, cf [DEP 02] : en pratique, il est interdit de faire usage de la congruence en-dessous d'un symbole protecteur. En déduction modulo, les notions de terme et de proposition viennent de la logique du premier ordre multi sortée. On considère des théories décrites par un ensemble d'axiomes Γ et une congruence, notée $\sim$, définie sur les termes et les propositions. Cette congruence prend trois arguments : les deux objets à comparer et un ensemble d'axiomes Γ appelé contexte local. Quand on veut souligner cela, on note la congruence $\sim^\Gamma$. Les règles de déduction du calcul des séquents modulo prennent cette équivalence en compte. Par exemple, la règle de conjonction droite n'est pas écrite comme d'habitude :

$$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}$$

mais est formulée de la manière suivante :

$$\frac{\Gamma \vdash_\sim A, \Delta \quad \Gamma \vdash_\sim B, \Delta}{\Gamma \vdash_\sim D, \Delta} \text{ si } D \sim^\Gamma A \wedge B.$$

Nous rappelons Figure 3.3, la définition du *calcul des séquents modulo*. Dans ces règles, Γ et Δ sont des multi-ensembles finis de propositions, et P et Q désignent des propositions. Remplacer la variable x par le term u dans Q se note $Q\{u/x\}$. Quand la congruence $\sim$ est simplement l'identité, le calcul des séquents modulo se ramène au calcul des séquents habituels [GIR 89]. Dans ce cas, les séquents sont écrits comme d'habitude à l'aide du symbole $\vdash$.

Si une proposition admet une preuve en calcul des séquents modulo, elle sera décidable à partir du moment où $\sim^\Gamma$ l'est également, du fait que l'on peut vérifier que les conditions d'application de chaque règle sont satisfaites, et que l'on fournit les informations nécessaires dans les règles faisant intervenir les quantificateurs. Lorsque $\sim^\Gamma$ n'est pas décidable, il est encore possible d'utiliser des instances pour lesquelles on peut vérifier ces conditions, typiquement par

$$\frac{}{\Gamma, P \vdash_\sim Q}\text{axiom} \;\; \text{if } P \sim^\Gamma Q \qquad \frac{\Gamma, P \vdash_\sim \Delta \quad \Gamma \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim \Delta}\text{cut} \;\; \text{if } P \sim^\Gamma Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}\text{contr-l} \;\; \text{if } (\mathbf{A}) \qquad \frac{\Gamma \vdash_\sim Q_1, Q_2, \Delta}{\Gamma \vdash_\sim P, \Delta}\text{contr-r} \;\; \text{if } (\mathbf{A})$$

$$\frac{\Gamma \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}\text{weak-l} \qquad \frac{\Gamma \vdash_\sim \Delta}{\Gamma \vdash_\sim P, \Delta}\text{weak-r}$$

$$\frac{\Gamma, P, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\wedge\text{-l} \;\; \text{if } R \sim^\Gamma (P \wedge Q) \qquad \frac{\Gamma \vdash_\sim P, \Delta \quad \Gamma \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\wedge\text{-r} \;\; \text{if } R \sim^\Gamma (P \wedge Q)$$

$$\frac{\Gamma, P \vdash_\sim \Delta \quad \Gamma, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\vee\text{-l} \;\; \text{if } (\mathbf{B}) \qquad \frac{\Gamma \vdash_\sim P, Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\vee\text{-r} \;\; \text{if } (\mathbf{B})$$

$$\frac{\Gamma \vdash_\sim P, \Delta \quad \Gamma, Q \vdash_\sim \Delta}{\Gamma, R \vdash_\sim \Delta}\Rightarrow\text{-l} \;\; \text{if } (\mathbf{C}) \qquad \frac{\Gamma, P \vdash_\sim Q, \Delta}{\Gamma \vdash_\sim R, \Delta}\Rightarrow\text{-r} \;\; \text{if } (\mathbf{C})$$

$$\frac{\Gamma \vdash_\sim P, \Delta}{\Gamma, R \vdash_\sim \Delta}\neg\text{-l} \;\; \text{if } R \sim^\Gamma \neg P \qquad \frac{\Gamma, P \vdash_\sim \Delta}{\Gamma \vdash_\sim R, \Delta}\neg\text{-r} \;\; \text{if } R \sim^\Gamma \neg P$$

$$\frac{}{\Gamma, P \vdash_\sim \Delta}\bot\text{-l} \;\; \text{if } P \sim^\Gamma \bot$$

$$\frac{\Gamma, Q\{t/x\} \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}(Q, x, t)\ \forall\text{-l} \;\; \text{if } (\mathbf{D}) \qquad \frac{\Gamma \vdash_\sim Q\{y/x\}, \Delta}{\Gamma \vdash_\sim P, \Delta}(Q, x, y)\ \forall\text{-r} \;\; \text{if } (\mathbf{E})$$

$$\frac{\Gamma, Q\{y/x\} \vdash_\sim \Delta}{\Gamma, P \vdash_\sim \Delta}(Q, x, y)\ \exists\text{-l} \;\; \text{if } (\mathbf{F}) \qquad \frac{\Gamma \vdash_\sim Q\{t/x\}, \Delta}{\Gamma \vdash_\sim P, \Delta}(Q, x, t)\ \exists\text{-r} \;\; \text{if } (\mathbf{G})$$

$\mathbf{A} = P \sim^\Gamma Q_1 \sim^\Gamma Q_2$, $\mathbf{B} = R \sim^\Gamma (P \vee Q)$ $\mathbf{C} = R \sim^\Gamma (P \Rightarrow Q)$, $\mathbf{D} = P \sim^\Gamma \forall x\ Q$, $\mathbf{E} = P \sim^\Gamma \forall x\ Q, y$ fresh variable, $\mathbf{F} = P \sim^\Gamma \exists x\ Q, y$ fresh variable, $\mathbf{G} = P \sim^\Gamma \exists x\ Q$

FIG. 3.3 – Le calcul des séquents modulo

le biais d'une approche par contraintes [HUE 72, KIR 90] Nous sommes maintenant en mesure d'introduire la notion fondamentale de compatibilité : on dit qu'une théorie (i.e un ensemble de propositions) $\mathcal{T}$ est compatible avec une congruence $\sim$ si :

$$\mathcal{T}, \Gamma \vdash \Delta \text{ si et seulement si } \Gamma \vdash_\sim \Delta.$$

Comme démontré dans [DEP 02, DEP 04], cette propriété est modulaire : si $\mathcal{T}_1$ est compatible avec une congruence C_1 et $\mathcal{T}_2$ est compatible avec C_2, alors $\mathcal{T}_1 \cup \mathcal{T}_2$ est compatible avec $C_1 \cup C_2$. Grâce à l'équivalence ci-dessus, on peut internaliser les propositions à l'intérieur de la congruence, et on appelle cette opération "push". Il est également possible de les remonter au niveau de la logique, et on appelle cette opération "pop". En outre, grâce à la modularité, cela peut être effectué dynamiquement au cours de la preuve. Cette dualité entre calcul et déduction est représentée de façon très appropriée par cette propriété de compatibilité. Dans [DOW 03b], l'internalisation est effectuée de manière statique et utilisée pour identifier les étapes de calcul à l'intérieur d'un raisonnement. Nous nous proposons ici d'effectuer l'internalisation de façon dynamique, et de l'utiliser pour construire des règles pour la récurrence par réécriture et déterminer la stratégie adéquate pour la récurrence noethérienne. Dans ce qui suit, on considère des congruences engendrées par des systèmes de réécriture de classes conditionnels notés $\mathcal{RE}$, et composées de règles de réécriture (conditionnelles) et d'axiomes (conditionnels) sur les termes. Dans le cas des règles et des axiomes conditionnels, les variables présentes dans le membre droit doivent apparaître également dans le membre gauche. Nous rappelons ici que $\approx$ est un symbole de relation binaire qui satisfait les axiomes de l'égalité (la notation classique $=$ n'étant utilisée uniquement que pour représenter l'égalité syntaxique). Dans ce cas, on associe à tout système de réécriture

de classe conditionnel $\mathcal{RE}$ la théorie notée $Th_{\mathcal{RE}}$, et définie de la manière suivante : pour toute règle de réécriture conditionnelle $(c \Rightarrow l \rightarrow r)$ ou égalité conditionnelle $(c \Rightarrow l \approx r)$ de $\mathcal{RE}$, $Th_{\mathcal{RE}}$ contient la proposition :

– $\forall \overrightarrow{x} (c \Rightarrow (l \approx r))$ avec l et r des termes,

et où toutes les variables de l, dont le vecteur est noté $\overrightarrow{x}$, sont universellement quantifiées. Il est démontré dans [DEP 02] que $Th_{\mathcal{RE}}$ est compatible avec la congruence engendrée par $\mathcal{RE}$ (voir de même [DOW 99] et [DOW 03b]). Cela nous permet de faire librement usage des opérations “push” et ”pop”.

3.4.3 Récurrence dans le contexte de la déduction modulo

Un raisonnement par récurrence ne pourra jamais se réduire à un simple calcul, il comporte toujours des étapes de déduction. On a vu que la déduction modulo est une présentation de la logique du premier ordre qui fait une distinction claire entre la partie calcul d'un raisonnement et sa partie déduction. Ce formalisme est ainsi apparu comme particulièrement adapté pour représenter un raisonnement par récurrence. Cependant, le principe de récurrence noethérien est une proposition du second ordre. Il est donc nécessaire d'utiliser une présentation du premier ordre de la logique du second ordre, et c'est $HOL_{\lambda\sigma}$ qui a été choisie, car il exite une correspondance biunivoque entre les preuves réalisées dans la présentation standard de la logique d'ordre supérieur (HOL_{λ}) et celles réalisées en $HOL_{\lambda\sigma}$, à partir du moment où l'axiome d'extensionnalité n'est pas utilisé dans les deux cas. Elle utilise également le fait que les substitutions explicites simplifient les algorithmes et accélèrent l'implémentation. Sachant que pour appliquer une hypothèse de récurrence il est nécessaire de vérifier une condition, il a fallu également étendre le formalisme de la déduction modulo pour le rendre apte à effectuer des raisonnements à l'aide d'axiomes conditionnels. La méthode va être expliquée en détail ci-dessous.

Dans le contexte de la déduction modulo, les hypothèses de récurrence provenant de buts équationnels peuvent être (dynamiquement) internalisées dans la congruence. En procédant ainsi, la puissance de calcul de la déduction modulo est utilisée pour réaliser la récurrence par réécriture, de la même façon que dans des systèmes comme Spike [BOU 92b] ou RRL [KAP 95]. L'interêt de ces approches est de permettre l'application de règles de réécriture présentes dans la théorie à toute position du but courant, aussi bien que l'application d'hypothèses de récurrence et de conjectures non encore démontrées, à partir du moment où la formule appliquée est plus petite pour l'ordre noethérien utilisé pour la récurrence que le but courant. Quand l'ordre contient la relation induite par un système de réécriture terminant, une formule plus petite est obtenue à partir du moment où une étape de réécriture a été réalisée. En outre, dans Spike par exemple, le choix des variables de récurrence et des schémas d'instantiation est effectué à l'aide de positions de récurrence pré-calculées et d'objets appelés ensembles tests. Dans l'approche décrite ci-dessous, nous montrons comment utiliser la surréduction pour effectuer automatiquement et complètement ces choix.

Une propriété P et une relation R définie sur une sorte τ étant données, le principe de récurrence noethérien $NoethInd(P, R, \tau)$ est défini de la manière suivante :

$$\forall x\, ((x \in \tau \wedge \forall y\, ((y \in \tau \wedge R(x, y)) \Rightarrow P(y))) \Rightarrow P(x)) \Rightarrow \forall x\, (x \in \tau \Rightarrow P(x))$$

et on écrit $Noeth(R, \tau)$ pour affirmer que R est une relation noethérienne sur τ. Le lecteur doit garder présent à l'esprit que ceci est une formule de $HOL_{\lambda\sigma}$: c'est donc une proposition du premier ordre dénotée par une formule d'ordre supérieur (celle-ci est définie explicitement dans

[DEP 02, DEP 04]). Prouver que P est une proposition inductive dans une théorie utilisateur Th_u, ce qui se note $Th_u \models_{Ind} P$, revient à dériver le séquent :

$$\forall R \forall \tau\, (Noeth(R,\tau) \Rightarrow \forall P\, NoethInd(P,R,\tau)), Th_u \vdash P.$$

Bien sûr, pour finir la preuve, il faut également fournir la preuve de $Noeth(R,\tau)$. Pour avoir une meilleure intuition, imaginons un but équationnel Q de la forme $\forall x\, (x \in \tau \Rightarrow t_1(x) \approx t_2(x))$, l'ensemble du problème est formalisé dans $HOL_{\lambda\sigma}$. Le reste de cette sous-section donne les étapes principales qui sont détaillées dans [DEP 02]. On part du séquent :

$$\begin{array}{l} \forall R \forall \tau\, (Noeth(R,\tau) \Rightarrow \forall P\, NoethInd(P,R,\tau)), Th_u \\ \vdash \\ \forall x\, (x \in \tau \Rightarrow t_1(x) \approx t_2(x)) \end{array}$$

Dans la suite, on notera NI la proposition :

$$\forall R \forall \tau\, (Noeth(R,\tau) \Rightarrow \forall P\, NoethInd(P,R,\tau))$$

En choisissant une relation spécifique R (écrite $\prec$) et un type encore noté τ, on obtient :

$$Noeth(\prec,\tau) \Rightarrow \forall P\, NoethInd(P,\prec,\tau)), Th_u \vdash \forall x\, (x \in \tau \Rightarrow t_1(x) \approx t_2(x)).$$

De là, par la règle $\Rightarrow$-l du calcul des séquents modulo, on obtient d'une part le séquent $Th_u \vdash Noeth(\prec,\tau)$ qui correspond à la preuve du fait que $\prec$ est effectivement noethérienne, et d'autre part le séquent

$$\forall P\, NoethInd(P,\prec,\tau)), Th_u \vdash \forall x\, (x \in \tau \Rightarrow t_1(x) \approx t_2(x))$$

qui correspond à l'utilisation du principe de récurrence pour prouver notre but. On instancie P par l'égalité à prouver et on obtient :

$$\begin{array}{l} \forall x\, ((x \in \tau \wedge \forall \underline{x}\, ((\underline{x} \in \tau \wedge \underline{x} \prec x) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \Rightarrow t_1(x) \approx t_2(x)) \\ \Rightarrow \forall x\, (x \in \tau \Rightarrow t_1(x) \approx t_2(x)), Th_u \vdash \forall x\, (x \in \tau \Rightarrow t_1(x) \approx t_2(x)) \end{array}$$

où on a renommé y en $\underline{x}$ pour souligner le fait que $\underline{x}$ est une instance plus petite de x. Quelques étapes simples du calcul des séquents plus tard, on obtient :

$$Th_u \vdash \forall x\, ((x \in \tau \wedge \forall \underline{x}\, ((\underline{x} \in \tau \wedge \underline{x} \prec x) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \Rightarrow t_1(x) \approx t_2(x))$$

On instancie alors x par une variable fraîche que l'on appelle X pour souligner ce statut, et on obtient :

$$Th_u \vdash (X \in \tau \wedge \forall \underline{x}\, ((\underline{x} \in \tau \wedge \underline{x} \prec X) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \Rightarrow t_1(X) \approx t_2(X).$$

Les règles $\Rightarrow$-r et $\wedge$-l du calcul des séquents mènent à la découverte de l'hypothèse de récurrence :

$$Th_u, X \in \tau, \forall \underline{x}\, ((\underline{x} \in \tau \wedge \underline{x} \prec X) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}))) \vdash t_1(X) \approx t_2(X).$$

En utilisant ce que l'on a déjà vu concernant les théories compatibles, cette hypothèse peut maintenant être internalisée sous la forme d'une égalité conditionnelle que l'on note en général

$$\mathcal{RE}_{ind}(Q,\prec,\tau)(X) : \quad (\underline{x} \in \tau \wedge \underline{x} \prec X) \Rightarrow t_1(\underline{x}) \approx t_2(\underline{x}) \tag{3.5}$$

Il faut remarquer que, du fait de son statut de variable fraîche, X se comporte comme une constante, alors que $\underline{x}$ est universellement quantifiée.

3.5 Descente infinie

Nous venons de voir que la déduction modulo est un formalisme adapté pour représenter le raisonnement par récurrence, et c'est sur lui que repose les systèmes de recherche de preuve que nous présenterons dans les parties III et IV . D'autres approches ont toutefois été élaborées pour utiliser le calcul des séquents dans le but d'effectuer des preuves par récurrence. L'une d'entre elles a notamment été la représentation des preuves par descente infinie.

Dans l'introduction, nous avons formalisé le principe de récurrence noethérien par la règle d'inférence :

$$\frac{\forall t\ (\forall x\ x \prec t \Rightarrow P(x)) \Rightarrow P(t)}{\forall t\ P(t)}$$

Or, cette règle est équivalente à :

$$\frac{\forall t\ \neg P(t) \Rightarrow \exists x\ (x \prec t \wedge \neg P(x))}{\forall t\ P(t)}$$

De façon plus concrète, cette règle signifie que si l'affirmation d'une conjecture entraîne l'existence d'une suite infiniment décroissante pour une certaine relation noethérienne, cela veut dire que cette conjecture est fausse. C'est pour cela qu'elle formalise le principe de *descente infinie*. J. Brotherston [BRO 06] rapporte que ce principe a été précisément formulé par Fermat en 1659, dans une correspondance échangée avec Pierre de Carcavi. Il ajoute cependant que cette technique a été utilisée dès l'Antiquité. La preuve originale d'Euclide que $\sqrt{2}$ est irrationnel en est un bel exemple. Wirth [WIR 04] rapporte même que le premier usage connu de la descente infinie apparaît dans une preuve que $\frac{1}{2}(1+\sqrt{5})$ est irrationnel, due à Hippasos au cinquième siècle av-JC.
On trouve une formalisation des preuves par descente infinie à l'aide du calcul des séquents dans [BRO 06]. Dans [BRO 06], J.Brotherston présente un système constitué des règles du calcul des séquents classique LK de Gentzen, enrichi de règles de récurrence exprimées à l'aide de séquents.

Exemple 3.10 La règle de récurrence associée au prédicat "entier naturel" N dans l'ensemble des entiers de Peano est définie par :

$$\frac{\Gamma, t = 0 \vdash \Delta \qquad \Gamma, t = sx, Nx \vdash \Delta}{\Gamma, Nt \vdash \Delta}\ \text{Case N}$$

L'exemple suivant va permettre de comprendre comment une preuve par descente infinie est formalisée dans [BRO 06].

Exemple 3.11 [BRO 06] Supposons que l'on cherche à démontrer que tout entier de Peano est soit pair, soit impair. On commence par choisir un entier x_0 quelconque, et on peut distinguer deux cas : soit $x_0 = 0$, soit $x_0 = sx_1$. Si $x_0 = 0$, x_0 est pair. Si $x_0 = sx_1$, deux cas se présentent à nouveau : soit $x_1 = 0$ et x_0 est impair, soit $x_1 = sx_2$, et il faut encore distinguer deux cas, etc ... On peut ainsi s'apercevoir que si l'entier de départ x_0 n'était ni pair, ni impair, il faudrait distinguer des cas une infinité de fois, et cela signifierait qu'il existe une suite infiniment décroissante $x_o > x_1 > x_2 > \ldots$ d'entiers naturels. Mais cela contredit le fait que l'ordre $<$ usuel sur les entiers est noethérien. On peut donc affirmer que l'on a prouvé que tout entier naturel est soit pair, soit impair par descente infinie.

Maintenant, si on note respectivement N, E et O les prédicats “entier naturel”, “pair” et “impair”, cette preuve est formalisée dans [BRO 06] par l'arbre infini suivant :

$$\dfrac{\dfrac{\dfrac{}{\vdash E0, O0}\,ER_1}{x_0 = 0 \vdash Ex_0, Ox_0}\,= L \qquad \dfrac{\dfrac{\dfrac{\dfrac{\vdots}{Nx_1 \vdash Ex_1, Ox_1}\,(\text{Case N})}{Nx_1 \vdash Ox_1, Osx_1}\,OR_1}{Nx_1 \vdash Esx_1, Osx_1}\,ER_2}{x_0 = sx_1, Nx_1 \vdash Ex_0, Ox_0}\,= L}{Nx_0 \vdash Ex_0, Ox_0}\,\text{Case N}$$

(On reconnaît la règle “Case N” définie dans l'exemple 3.10, et on peut avoir facilement une compréhension intuitive du sens des autres règles). On peut donc facilement s'apercevoir, que si on continue cette dérivation, on obtient un arbre infini avec exactement une branche infinie. En outre, si on suit cette branche, la règle “Case N” est infiniment utilisée. De façon très informelle, on peut alors dire que cette dérivation infinie est une preuve dans le système de Brotherston, car la règle“Case N”, qui distingue des cas, est infiniment utilisée le long de l'unique branche infinie.

Avant de décrire nos systèmes de recherche de preuve dans les parties III et IV , nous considérons dans la partie II deux outils principaux pour la récurrence, à savoir les ordres et les propriétés de surréduction dans les systèmes de réécriture suffisamment complets.

Deuxième partie

Ordre et surréduction : deux outils principaux pour la méthode

1

Ordres et quasi-ordres

Sommaire

1.1 Ordre sur les termes et terminaison des systèmes de réécriture 63
1.2 Ordres et quasi-ordres sur les chemins 66
1.3 Ordres et quasi-ordres sur les égalités 67
1.4 Ordres récursifs sur les chemins AC-compatibles 70
1.4.1 Termes variadiques 71
1.4.2 Construction d'un ordre récursif sur les chemins AC-compatible . . 72

Dans ce chapitre, nous commençons par rappeler l'utilité des ordres sur les termes pour montrer la terminaison des systèmes de réécriture, puis nous faisons à la section 2 une présentation rapide des ordres récursifs sur les chemins. M.C.F Ferreira [FER 95] a ainsi mis en évidence une classe d'ordres récursifs sur les chemins qui possèdent des propriétés intéressantes pour notre étude, et que l'on décrira précisément. A partir des ordres sur les termes, nous construisons à la section 3 deux ordres sur les égalités. Nous énonçons alors le théorème de compatibilité principale sur lequel repose la correction du système optimisé présenté au chapitre 2 de la partie III. Dans la dernière section, nous commençons par rappeler les définitions et les résultats de base concernant les termes variadiques, puis nous présentons l'ordre récursif sur les chemins AC-compatible défini dans [RUB 02] (ce dernier sera utilisé pour instancier l'hypothèse de récurrence dans les systèmes fondés sur la réécriture modulo des théories associatives, ou associatives commutatives, et que nous présenterons dans la dernière partie).

1.1 Ordre sur les termes et terminaison des systèmes de réécriture

Dans cette section, nous nous inspirons largement d'une présentation qui a été faite dans [LAL 90].

Pour autant qu'un calcul soit compris comme un processus produisant un résultat, le problème de la terminaison est crucial. Nous avons déjà mentionné que le problème de la terminaison est indécidable (théorème 2.1) ; cela n'empêche évidemment pas d'étudier des cas particuliers. Il arrive qu'une règle de réécriture diminue visiblement la taille (le nombre de symboles, la hauteur, ...) des termes. Dans le cas de la règle $-(-x) \to x$, on utilise une fonction $h : \mathcal{T}(\Sigma, \mathcal{X}) \to \mathbb{N}$ (par exemple $h(t) = |t|$) qui vérifie $h(l) > h(r)$ pour toute règle $l \to r$. Pour en

déduire que $h(s) > h(t)$ si $s \to t$, il suffit que h satisfasse les conditions suivantes, qui découlent de la définition de la réécriture :

1. $h(l\sigma) > h(r\sigma)$ pour toute règle $l \to r$ et toute substitution σ
2. $h(s_i) > h(t_i)$ entraîne $h(fs_1 \ldots s_i \ldots s_n) > h(fs_1 \ldots t_i \ldots s_n)$

Cependant, une règle usuelle comme l'associativité

$$(x + y) + z \to x + (y + z)$$

ne peut se traiter en raisonnant globalement sur la taille des termes. L'idée est alors de donner un poids plus important au sous-terme de gauche d'une somme ; on peut ainsi définir récursivement une fonction h en posant

$$h(s + t) = 2h(s) + h(t)$$
$$h(a) = 1$$

où a est une constante (ou une variable).
Par ailleurs, il n'est pas nécessaire que h soit à valeurs dans $\mathbb{N}$; la seule propriété des entiers que l'on utilise est que l'ordre usuel $<$ est bien fondé sur l'ensemble des entiers naturels $\mathbb{N}$: il n'y a pas de suite infinie strictement décroissante d'entiers naturels. On pourra donc chercher une fonction h à valeurs dans un ensemble ordonné $(E, <)$ bien fondé ; la considération de cet ensemble E n'est qu'une commodité, puisqu'on peut aussi bien raisonner directement dans $\mathcal{T}(\Sigma, \mathcal{X})$ avec l'ordre :

$$s <_h t \quad \text{si} \quad h(s) < h(t)$$

Cela conduit donc à chercher un ordre bien fondé $<$ sur l'ensemble des termes qui satisfasse les propriétés correspondant à 1 et 2. Cette condition nécessaire et suffisante constitue le théorème de Manna-Ness (1970) :

Théorème 1.1 *Un système de réécriture est noethérien si et seulement s'il existe un ordre bien fondé $<$ sur l'ensemble des termes tel que*
- *$s < t$ entraîne $C[s] < C[t]$*
- *$r\sigma < l\sigma$ pour toute règle $l \to r \in \mathcal{R}$ et toute substitution σ*

Le théorème 1.1 met en évidence deux caractéristiques des ordres sur les termes appelées *stabilité par contextes* et *stabilité par substitutions* :

Définition 1.1 Un ordre $<$ sur $\mathcal{T}(\Sigma, \mathcal{X})$ est :
- *stable (ou clos) par contextes* si $s < t$ implique $C[s] < C[t]$
- *stable (ou clos) par substitutions* si $s < t$ implique $s\sigma < t\sigma$.

Les ordres stables par contextes qui sont également stables par substitution sont appelés ordres de réécriture .

Définition 1.2 Un ordre sur $\mathcal{T}(\Sigma, \mathcal{X})$ est appelé ordre de réécriture si il est clos par contextes et par substitutions. Un quasi-ordre sur $\mathcal{T}(\Sigma, \mathcal{X})$ est appelé quasi-ordre de réécriture si sa partie stricte et l'équivalence associée sont toutes deux closes par contextes et par substitutions.

Si un (quasi-)ordre de réécriture est de plus bien-fondé, il est appelé *(quasi-)ordre de réduction.*

Définition 1.3 Un ordre (respectivement un quasi-ordre) sur $\mathcal{T}(\Sigma, \mathcal{X})$ est appelé *ordre de réduction* (respectivement *quasi-ordre de réduction*) si c'est un ordre de réécriture bien fondé (respectivement un quasi-ordre de réécriture bien fondé).

Le théorème 1.1 énoncé au paragraphe 1.1 de ce chapitre a pour conséquence immédiate :

Théorème 1.2 *Un système de réécriture $\mathcal{R}$ est terminant s'il existe un ordre de réduction $<$ tel que $r < l$ pour toute règle $l \to r$ de $\mathcal{R}$*

Le théorème précédent décrit ce que l'on entend par prouver la terminaison d'un système de réécriture : il s'agit de trouver un ordre bien fondé à partir duquel on peut construire un ordre de réduction adapté. Il existe essentiellement deux méthodes pour y parvenir : une syntaxique et une autre sémantique. Lorsqu'on utilise la méthode syntaxique, on examine de manière approfondie la structure des termes, dans le but de définir un ordre sur l'ensemble $\mathcal{T}(\Sigma, \mathcal{X})$. Avec la méthode sémantique, il faut interpréter les termes dans une algèbre munie d'une relation d'ordre $<$. Moyennant des conditions sur l'interprétation que nous ne détaillerons pas ici, l'ordre $<$ sur l'algèbre et l'interprétation induisent un ordre de réduction sur $\mathcal{T}(\Sigma, \mathcal{X})$ (nous renvoyons le lecteur à [FER 95] pour plus d'informations). Dans le contexte de la réécriture modulo un ensemble d'égalités, la terminaison a une définition plus générale (voir la définition 2.29 au paragraphe 2.5.1 dans la partie I) mais on peut énoncer un théorème similaire au théorème 1.2. On est amené pour cela à introduire la définition suivante :

Définition 1.4
- Un quasi-ordre de réécriture $\leqslant$ est compatible avec un ensemble d'égalités E si l'équivalence associée $\gtrless$ est telle que $=_E \subseteq \gtrless$;
- un ordre de réécriture $<$ est *compatible* avec un ensemble d'égalités E si $<$ est la partie stricte d'un quasi-ordre $\leqslant$ E-compatible ;
- un quasi-ordre de réécriture $\leqslant$ est compatible avec un système de réécriture de termes modulo $(\mathcal{R}, E)$ si :
  - $\leqslant$ est compatible avec E ;
  - la partie stricte $<$ est telle que $r < l$ si $l \to r \in \mathcal{R}$;
- un ordre de réécriture $<$ est compatible avec un système de réécriture de termes modulo $(\mathcal{R}, E)$ si $<$ est la partie stricte d'un quasi-ordre compatible avec $(\mathcal{R}, E)$.

Remarque: On dira également que E est compatible avec un quasi-ordre $\leqslant$ ou un ordre $<$, ou encore que $(\mathcal{R}, E)$ est compatible avec $\leqslant$ ou $<$.

Proposition 1.1 Un ensemble d'égalités E est *compatible* avec un ordre de réécriture $<$ si $\forall s\ \forall t\ \forall s'\ \forall t'\ s =_E s' < t' =_E t \Rightarrow s < t$

Preuve. Il suffit de poser $\leqslant = < \cup =_E$ □

Pour montrer la terminaison d'un système de réécriture de termes modulo, il suffira alors de montrer qu'il est compatible avec un quasi-ordre de réécriture qui est de plus bien fondé, i.e un quasi-ordre de réduction :

Théorème 1.3 *Un système de réécriture de termes modulo $(\mathcal{R}, E)$ est terminant si il est compatible avec un quasi-ordre de réduction.*

Les théorèmes 1.2 et 1.3 sont cependant très exigeants : ils imposent de mettre en évidence un ordre bien fondé sur l'ensemble des termes alors qu'il suffirait de le savoir bien fondé "le long des réductions"(que l'on ait une suite infinie décroissante de termes n'a aucune importance si cette suite n'est pas une réduction). Il existe une classe d'ordres qui sont nécessairement bien fondés le long des réductions. Ce sont les ordres de simplification introduits par N.Dershowitz.

Définition 1.5 Une relation d'ordre strict $<$ sur $\mathcal{T}(\Sigma, \mathcal{X})$ est un *ordre de simplification* si

1. $<$ est stable par contexte

2. $<$ a la propriété de sous-terme, i.e $t < C[t]$ pour tout terme t et tout contexte $C[\ ]$.

Les théorèmes 1.2 et 1.3 ont alors pour conséquences :

Théorème 1.4 (Dershowitz) *Un système de réécriture $\mathcal{R}$ est noethérien s'il existe un ordre de simplification $<$ tel que $r\sigma < l\sigma$ pour toute règle $l \to r \in \mathcal{R}$ et toute substitution σ.*

Théorème 1.5 *Un système de réécriture de termes modulo $(\mathcal{R}, E)$ est terminant si il est compatible avec un quasi-ordre de simplification.*

1.2 Ordres et quasi-ordres sur les chemins

Nous nous inspirons ici d'une présentation qui a été faite dans [FER 95]. L'idée sous-jacente aux ordres récursifs sur les chemins est la construction d'ordres sur les termes à partir d'un quasi-ordre bien fondé sur la signature Σ (appelé généralement *précédence*). En général, un terme s est plus grand que tout terme construit à partir de termes "plus petits" situés sous un symbole plus petit, pour la précédence, que le symbole de tête de s (les termes sont en particulier plus grands que leurs propres sous-termes). Les ordres récursifs sur les chemins comparent ainsi les symboles de tête des termes à l'aide de la précédence, et, dans le cas où ceux-ci sont égaux ou équivalents, on compare les sous-termes de façon récursive.
Les ordres récursifs sur les chemins ont été introduits par les travaux de Plaisted [PLA 78] et Dershowitz [DER 79a] à la fin des années 70. D'autres ordres ont été proposés depuis lors. Citons à titre d'exemple l'*ordre lexicographique sur les chemins* de Kamin et Levy [KAM 82], l'ordre récursif de décomposition de Jouannaud, Lescanne et Reinig [JOU 82], l'ordre sur les chemins de Kapur, Narendran et Sivakumar [KAP 85].
Une modification qui a systématiquement été ajoutée aux ordres récursifs sur les chemins a été d'associer à chaque symbole de fonction $f \in \Sigma$ un statut $\tau(f)$.

Définition 1.6 A chaque symbole de fonction $f \in \Sigma$, on associe un statut $\tau(f)$. On considère deux cas possibles :
- $\tau(f) = mul$ qui indique que, dans la définition ci-dessous de l'ordre $\leqslant$, les arguments de f seront comparés avec une extension multi-ensemble de $\leqslant$ (cf section 2.1)
- $\tau(f) = lex_\pi$, avec π une permutation de l'ensemble $\{1, \ldots, ar(f)\}$, et qui indique que les arguments de f seront permutés par π, et les séquences obtenues comparées lexicographiquement (dans le cas où π est l'identité, on écrira simplement $\tau(f) = lex$)

A l'aide de cette notion de statut, nous introduisons la définition de l'*ordre récursif sur les chemins* telle qu'elle est apparue dans Steinbach [STE 89]. Nous donnons d'abord la définition d'une *(quasi-)précédence* :

Définition 1.7 Une (quasi-)précédence sur Σ est un (quasi-)ordre partiel sur Σ noté $\trianglelefteq$ (dans le cas d'une quasi-précédence, on notera $\sim$ l'équivalence associée à $\trianglelefteq$).

Il est souvent plus commode d'étendre la (quasi-)précédence à l'ensemble $\mathcal{X}$ des variables en posant $x \trianglelefteq x$ pour tout $x \in \mathcal{X}$.
On imposera également aux statuts de ne pas se mélanger, autrement dit, on fera l'hypothèse suivante :

$$\forall f\ \forall g\ (f \sim g) \Rightarrow (\tau(f) = \tau(g)) \tag{1.1}$$

Nous pouvons maintenant définir la relation $\leqslant_{rpo}$:

Définition 1.8 Deux termes s, t étant donnés, on dira que $s \leqslant_{rpo} t$ si $t = gt_1 \dots t_m$, et

1. soit $s = fs_1 \dots s_n$, pour tout i, $s_i <_{rpo} t$, et :
   (a) soit $f \lhd g$;
   (b) soit $f \sim g$ et $((m = k = 0)$ ou $s_1 \dots s_m \leqslant_{rpo,\tau} t_1 \dots t_n)$;
2. soit $\exists i\ 1 \leq i \leq m\ s \leqslant_{rpo} t_i$.

Pour un quasi-ordre de simplification $\leqslant$ quelconque défini sur l'ensemble des termes $\mathcal{T}(\Sigma, \mathcal{X})$, on peut très bien avoir $s < t$ et $C[s] \gtrless C[t]$. Nous allons voir qu'il est plus pratique d'utiliser un ordre de simplification dont la partie stricte et l'équivalence associée soient à la fois stables par substitutions et par contextes. La proposition ci-dessous, démontrée dans Ferreira [FER 95], permet de s'assurer que la relation $\leqslant_{rpo}$ est un quasi-ordre qui possède ces *bonnes* propriétés.

Proposition 1.2 [FER 95] Pour toute (quasi-)précédence $\trianglelefteq$, la relation $\leqslant_{rpo}$ introduite à la définition 1.8 possède les propriétés suivantes :
- $\leqslant_{rpo}$ est un quasi-ordre ;
- $\leqslant_{rpo}$ a la propriété de sous-terme ;
- $<_{rpo}$ et $\sim_{rpo}$ sont clos par substitution et par contexte ;
- si la (quasi-)précédence $\trianglelefteq$ est totale sur Σ, alors l'extension de $<_{rpo}$ aux classes d'équivalences $\mathcal{T}(\Sigma)/\gtrless$ est un ordre total.

Afin de pouvoir utiliser $\leqslant_{rpo}$ pour effectuer un raisonnement par récurrence, il est encore nécessaire que $\leqslant_{rpo}$ soit bien fondé. Kamin et Levy ont prouvé que c'est le cas à partir du moment où deux symboles de fonction équivalents pour $\trianglelefteq$ et qui ont un statut lexicographique ont la même arité. Cependant, Ferreira [FER 95] a montré qu'il suffit en fait qu'il existe un entier naturel majorant l'arité de tous les symboles de fonction d'une même classe d'équivalence, dès que ceux-ci ont un statut lexicographique, pour s'assurer que $\leqslant_{rpo}$ est bien fondé. Cette dernière condition s'énonce de manière formelle :

$$\forall f\ (((f \in \Sigma) \wedge (\tau(f) = lex_\pi)) \Rightarrow (\exists n\ ((n \geq 0) \wedge (\forall g\ g \sim f \Rightarrow ar(g) \leq n)))) \tag{1.2}$$

On peut ainsi énoncer le théorème suivant :

Théorème 1.6 *Sous les hypothèses (1.1) et (1.2), $\leqslant_{rpo}$ est bien fondé sur $\mathcal{T}(\Sigma, \mathcal{X})$ si, et seulement si $\trianglelefteq$ est bien fondé sur $\Sigma \cup \mathcal{X}$.*

1.3 Ordres et quasi-ordres sur les égalités

Nous rappelons que l'ensemble des positions dans un terme t est noté $\mathcal{D}om(t)$, le sous-terme de t à la position ω est noté $t_{|\omega}$, le symbole à la position ω dans t par $t(\omega)$, et que la notation $t[u]_\omega$ signifie que le terme t contient le sous-terme u à la position ω. On étend ces notations aux égalités $t_1 \approx t_2$ considérées comme des termes avec $\approx$ d'arité 2 pour symbole de tête (ce symbole a été introduit à la sous-section 1.2.2). Le symbole $\approx$ utilisé pour décrire les égalités est considéré dans cette partie comme un symbole de la signature. On suppose qu'aucun axiome ne fait par ailleurs intervenir ce symbole. Cela afin de considérer les égalités comme des termes irréductibles en tête. $\mathcal{V}ar(t)$ désigne l'ensemble des variables (libres) du terme t et $|\mathcal{V}ar(t)|$ son cardinal. On définit $\overrightarrow{\mathcal{V}ar(t)}$ comme le vecteur des variables de t que l'on suppose ordonnées linéairement par leur nom. On étend ces notations aux règles de réécriture et aux égalités. A partir de maintenant, on suppose donné un quasi-ordre de simplification $\leqslant$ sur $\mathcal{T}(\Sigma, \mathcal{X})$, bien fondé, dont la partie stricte

et l'équivalence associée sont stables par contextes et par substitutions (un tel quasi-ordre a été construit à la section précédente). On note $<$ sa partie stricte, $\gtrless$ l'équivalence associée et $[t]$ la classe du terme t pour cette équivalence. Dans le but de comparer des n-uplets de termes, pour tout entier naturel n, on utilisera l'extension standard sur le produit cartésien $\leqslant_n$ de $\leqslant$:

$$\forall \overrightarrow{u}, \overrightarrow{v} \in \mathcal{T}(\Sigma, \mathcal{X})^n \quad \overrightarrow{u} \leqslant_n \overrightarrow{v} \Leftrightarrow (\forall i \ 1 \leq i \leq n \Rightarrow u_i \leqslant v_i)$$

Si on note $<_n$ la partie stricte de ce quasi-ordre, alors $<_n$ est noethérien sur l'ensemble $\mathcal{T}(\Sigma, \mathcal{X})^n$, à partir du moment où $<$ est noethérien. On peut de même étendre naturellement le quasi-ordre $\leqslant$ aux égalités en posant :

$$s \approx t \leq_e s' \approx t' \text{ si } s \leqslant s' \text{ et } t \leqslant t'$$

Notation: On notera $\lneq_e$ la partie stricte de $\leq_e$.

Maintenant, du fait que $\leqslant$ est stable par substitution, on obtient :

Lemme 1.1 $\leq_e$ est stable par substitution.

Cette propriété est très utile, car la stabilité par substitution sera en particulier requise lorsque l'on considèrera la version optimisée de notre système de recherche de preuve présentée au chapitre 2 de la partie III.
Toutefois, afin de pouvoir comparer les égalités de manière plus fine, on aura recours à un autre ordre sur les égalités largement similaire à celui de [DEP 02].

Définition 1.9 Soit C la mesure de complexité suivante sur les égalités :

$$\begin{array}{rcll} C(s \approx t) & = & (\{\{[s]\}\}, \{\{[t]\}\}) & \text{si} \quad [t] < [s] \\ & & (\{\{[t]\}\}, \{\{[s]\}\}) & \text{si} \quad [s] < [t] \\ & & (\{\{[s], [t]\}\}, \emptyset) & \text{sinon.} \end{array}$$

(On rappelle que $[s]$ et $[t]$ désignent respectivement les classes d'équivalence de s et de t pour l'équivalence $\gtrless$ associée au quasi-ordre $\leqslant$).

A l'aide de cette notion de complexité, on définit alors un deuxième quasi-ordre sur les égalités $\leqslant_e$ par

$$s \approx t \leqslant_e s' \approx t' \text{ si } C(s \approx t) \ll_{lex} C(s' \approx t') \text{ ou } (s \gtrless s' \text{ et } t \gtrless t')$$

avec $\ll_{lex}$ l'extension lexicographique de l'extension multiset de $<$.

Notation: On note $<_e$ la partie stricte de $\leqslant_e$.

Remarquons que l'ordre $<_e$ est bien adapté aux égalités, puisqu'il est invariant par symétrie ou égalité : pour tous $t, t', u, u' \in \mathcal{T}(\Sigma, \mathcal{X})$, on a :
$t \approx t' <_e u \approx u'$ si, et seulement si $t' \approx t <_e u \approx u'$ si, et seulement si $t \approx t' <_e u' \approx u$.
Par contre, il n'est pas stable par substitution : par exemple avec la substitution $\sigma = \{x \mapsto x_1, \ y \mapsto x_1, \ z \mapsto z_1\}$, on a :

1. $z \approx x + z <_e y \approx x + z$ car
   $C(z \approx x + z) = (\{\{[x + z]\}\}, \{\{[z]\}\})$ et
   $C(y \approx x + z) = (\{\{[x + z], [y]\}\}, \emptyset)$
2. mais $z\sigma \approx x\sigma + z\sigma \nleqslant_e y\sigma \approx x\sigma + z\sigma$ car
   $C(z\sigma \approx x\sigma + z\sigma) = (\{\{[x_1 + z_1]\}\}, \{\{[z_1]\}\})$ et
   $C(y\sigma \approx x\sigma + z\sigma) = (\{\{[x_1 + z_1]\}\}, \{\{[x_1]\}\})$

Il faut relever la différence entre $\leqslant_e$ et $\leq_e$, le dernier étant inclus dans le premier, comme cela se démontre aisément à l'aide d'une simple analyse par cas. Une étape essentielle dans les preuves par récurrence sera de comparer différentes instances d'une même égalité : c'est l'objet de la proposition suivante.

Lemme 1.2 Pour toute égalité Q avec $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(Q)}$ et $n = |\overrightarrow{x}|$, pour toutes substitutions $\sigma,\ \mu \in Subst^{\mathcal{T}(\Sigma,\mathcal{X})}$, pour tout $t, t' \in \mathcal{T}(\Sigma, \mathcal{X})$:

1. Si $t \leqslant t'$ alors $Q[t]_\omega \leqslant_e Q[t']_\omega$
2. Si $\overrightarrow{x}\sigma \leqslant_n \overrightarrow{x}\mu$ alors $Q\sigma \leqslant_e Q\mu$.
3. Si $Q\sigma <_e Q\mu$ et $\overrightarrow{x}\sigma \leqslant_n \overrightarrow{x}\mu$ alors $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$.

Preuve. 1. Si $\omega = \varepsilon$, le résultat est clair. Sinon, posons i et ω', tels que $\omega = i.\omega'$. Puisque $t \leqslant t'$, et puisque $\leqslant$ est stable par contexte, on a :

$$Q_{|i}[t]_{\omega'} \leqslant Q_{|i}[t']_{\omega'} \tag{1.3}$$

Maintenant, on peut facilement vérifier la proposition suivante (qui est une conséquence du lemme B.25 démontré en annexe) :

$$\forall s\ \forall t\ \forall t'\ t \leqslant t' \Rightarrow s \approx t \leqslant_e s \approx t' \tag{1.4}$$

Et (1.3) et (1.4) ci-dessus, ainsi que les propriétés de symétrie de $\leqslant_e$, conduisent à $Q[t]_\omega \leqslant_e Q[t']_\omega$

2. est obtenu à partir de 1 par une simple récurrence fondée sur le nombre d'occurences des variables x_i dans Q
3. Supposons $\overrightarrow{x}\sigma \geqslant_n \overrightarrow{x}\mu$, alors $\overrightarrow{x}\mu \leqslant_n \overrightarrow{x}\sigma$, par conséquent $Q\mu \leqslant_e Q\sigma$ par 2, et cela contredit l'hypothèse $Q\sigma <_e Q\mu$.

□

En d'autres termes, pour toute égalité Q, pour tout vecteur de variables $\overrightarrow{x}$ de Q dans $\mathcal{X}^n$, et pour toutes substitutions $\sigma, \mu \in Subst^{\mathcal{T}(\Sigma,\mathcal{X})}$, dans le but de prouver la proposition $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$, et dès lors que l'on a $Q\sigma <_e Q\mu$, il suffit de vérifier toutes les inégalités $x_i\sigma \leqslant x_i\mu$ pour toutes les composantes x_i de $\overrightarrow{x}$. L'intérêt de ceci vient du fait que l'inégalité $Q\sigma <_e Q\mu$ peut être *automatiquement* vérifiée dans la plupart des cas.
En effet, le lemme suivant relie l'ordre strict $<_e$ sur les buts avec une relation de réécriture $\rightarrow$. Sous réserve de conditions techniques qui peuvent être vérifiées syntaxiquement, ce résultat assure que $Q\sigma <_e Q\mu$. Il est alors possible dans la plupart des cas d'utiliser une égalité Q pour réduire une instance d'elle même, $Q\mu$, à partir du moment où une étape de réécriture a été appliquée auparavant sur $Q\mu$. Ce lemme est donc une étape significative pour justifier l'utilisation correcte de la réécriture noethérienne comme l'ingrédient principal pour réaliser la récurrence noethérienne.

Théorème 1.7 (Théorème de compatibilité principale) *Etant donnés Q_1, Q_2, Q_3 et Q_4 des égalités, $l \rightarrow r$ une règle de réécriture telle que $l > r$, κ_0 étant soit une règle de réécriture $l_{\kappa_0} \rightarrow r_{\kappa_0}$ ou une égalité $l_{\kappa_0} \approx r_{\kappa_0}$ Considérons l'inégalité : I : $(l_{\kappa_0} \approx r_{\kappa_0})\sigma <_e Q_1$ et supposons :*

1. $Q_1 \rightarrow_{l\rightarrow r,\ j.\omega_j,\ \theta} Q_2$
2. $Q_2 \geq Q_3$

3. $Q_3 \rightarrow_{\kappa_0,\ i.\omega_i,\ \sigma} Q_4$
4. $Q_3 \geqslant_e Q_4$

Alors :

1. *I est satisfaite à partir du moment où $\omega_i \neq \varepsilon$ ou $i = j$;*
2. *si $\omega_i = \varepsilon$ et $i \neq j$:*
   (a) *si $l_{\kappa_0} > r_{\kappa_0}$, alors :*

$$I \Leftrightarrow ((Q_{1|i} \gtrless l_{\kappa_0}\sigma) \wedge (Q_{1|j} < Q_{1|i}) \Rightarrow (Q_{1|j} > r_{\kappa_0}\sigma));$$

   (b) *si $l_{\kappa_0} \gtrless r_{\kappa_0}$, alors :*

$$I \Leftrightarrow ((Q_{1|i} \gtrless l_{\kappa_0}\sigma) \Rightarrow (Q_{1|j} > r_{\kappa_0}\sigma));$$

   (c) *sinon :*

$$\begin{aligned} I \Leftrightarrow (\quad & ((Q_{1|i} \gtrless l_{\kappa_0}\sigma) \wedge ((Q_{1|j} < Q_{1|i}) \vee (l_{\kappa_0}\sigma \gtrless r_{\kappa_0}\sigma)) \wedge (r_{\kappa_0}\sigma \leqslant l_{\kappa_0}\sigma)) \\ & \Rightarrow (Q_{1|j} > r_{\kappa}\sigma)\ ). \end{aligned}$$

Preuve. La preuve de ce résultat fondée sur une analyse par cas est donnée en annexe (théorème B.4) □

Une variante de ce lemme est donnée dans [DEP 02] pour un ordre sur les égalités fondé sur une complexité C utilisant un ordre sur les ensembles à la place d'un ordre sur un multi-ensemble comme ici.

1.4 Ordres récursifs sur les chemins AC-compatibles

De nombreuses tentatives ont été réalisées pour définir une version AC-compatible de l'ordre récursif sur les chemins (en anglais *rpo*) défini par Dershowitz [DER 82], car il est simple, facile à utiliser et à automatiser. A.Rubio et R.Nieuwenhuis [RUB 95] ont été les premiers à définir un ordre récursif sur les chemins AC-total et AC-compatible, sans aucune condition préalable sur le nombre de symboles AC, ou sur la précédence définie sur la signature. Malheureusement, il s'est avéré que cet ordre n'oriente pas la distributivité dans le sens habituel, du fait qu'une transformation est d'abord appliquée sur les termes avant de les comparer. Une autre approche, développée dans [KAP 90] par exemple, a été de définir des ordres récursifs sur les chemins dans des modèles, et d'effectuer alors les comparaisons sur les interprétations des termes. Cependant, la nécessité d'interpréter les termes rend le comportement de ces ordres moins intuitif, à la différence de l'ordre récursif sur les chemins standard, dont la définition totalement syntaxique a contribué largement à le rendre populaire. Dans [RUB 02], A. Rubio a enfin construit le premier ordre récursif sur les chemins AC-compatible, défini de façon *totalement syntaxique* (à l'exception du fait que la comparaison s'effectue sur les formes plates des termes). Nous commençons par présenter au paragraphe 1.4.1 les définitions de base particulières aux théories AC, puis nous reprenons au paragraphe 1.4.2 la construction de A.Rubio. Notre cadre de travail est l'*algèbre des termes variadiques*, telle que définie dans [MAR 93].

1.4.1 Termes variadiques

On suppose donnée une signature Σ contenant un sous-ensemble fini $\mathcal{V}$ dont chaque élément est d'arité 2. On note $=_{AC}$ la congruence engendrée sur $\mathcal{T}(\Sigma, \mathcal{X})$ par les axiomes exprimant l'associativité et la commutativité des symboles de $\mathcal{V}$. L'*algèbre des termes variadiques* est une généralisation de l'algèbre des termes, où les symboles de $\mathcal{V}$ ont alors une arité non fixée.

Définition 1.10 L'*algèbre des termes variadiques* $\mathcal{TV}(\Sigma, \mathcal{X})$ sur la signature Σ et sur l'ensemble des symboles variadiques $\mathcal{V}$ est définie par :
- Si $f \notin \mathcal{V}$, $ar(f) = n \geq 0$, et $t_1, \ldots, t_n \in \mathcal{TV}(\Sigma, \mathcal{X})$, alors $ft_1 \ldots t_n \in \mathcal{TV}(\Sigma, \mathcal{X})$;
- Si $f \in \mathcal{V}$, $ar(f) = n \geq 2$, et $t_1, \ldots, t_n \in \mathcal{TV}(\Sigma, \mathcal{X})$, alors $ft_1 \ldots t_n \in \mathcal{TV}(\Sigma, \mathcal{X})$.

L'ensemble des symboles variadiques coïncide souvent comme ici avec l'ensemble des symboles AC. C'est pourquoi les symboles variadiques sont notés de façon infixe, et habituellement par un symbole d'opération classique comme $+$. Tout terme t de $\mathcal{T}(\Sigma, \mathcal{X})$ peut également être vu comme un terme de $\mathcal{TV}(\Sigma, \mathcal{X})$, et l'opération d'*aplatissement* consiste alors à coder les axiomes AC dans la structure du terme.

Définition 1.11 Pour tout $t \in \mathcal{T}(\Sigma, \mathcal{X})$, l'*aplatissement* de t est le terme variadique $\overline{t} \in \mathcal{TV}(\Sigma, \mathcal{X})$ tel que $\overline{t}$ est la forme normale de t pour le système de réécriture contenant les règles

$$fx_1 \ldots x_n(fy_1 \ldots y_m)z_1 \ldots z_r \to fx_1 \ldots x_n y_1 \ldots y_m z_1 \ldots z_r$$

pour chaque $f \in \mathcal{V}$, $m,\ n,\ r \in \mathbb{N}$ tels que $n \geq 1$ ou $r \geq 1$, et $m \geq 2$.
$\overline{t}$ est alors appelé *terme aplati*.

Exemple 1.1 $\overline{(a+b)+c} = a+b+c$

Remarque: On étend cette définition aux égalités, en considérant le symbole $\approx$ comme non variadique. L'associativité et la commutativité correspondent à la *congruence de permutation* sur les termes aplatis, qui est définie sur l'algèbre des termes variadiques de la façon suivante :

Définition 1.12 Deux termes $s,\ t \in \mathcal{TV}(\Sigma, \mathcal{X})$ étant donnés, $\overline{s} \equiv_p \overline{t}$ si il existe $n \in \mathbb{N}$, $s_1, \ldots, s_n,\ t_1, \ldots, t_n \in \mathcal{T}(\Sigma, \mathcal{X})$, tels que :
- soit il existe $f \notin \mathcal{V}$ et $n \geq 1$ tels que $\overline{s} = f\overline{s_1} \ldots \overline{s_n}$, $\overline{t} = f\overline{t_1} \ldots \overline{t_n}$, et $\forall i\ \overline{s}_i \equiv_p \overline{t_i}$;
- soit $\overline{s} = \overline{s_1} + \ldots + \overline{s_n}$, $\overline{t} = \overline{t_1} + \ldots + \overline{t_n}$, et il existe une permutation τ des indices, telle que $\forall i\ \overline{s_i} \equiv_p \overline{t_{\tau(i)}}$.

Exemple 1.2 $a+b+c \equiv_p b+c+a$

Lemme 1.3 Pour tous $s,\ t \in \mathcal{T}(\Sigma, \mathcal{X})$, $s =_{AC} t$ si, et seulement si $\overline{s} \equiv_p \overline{t}$.

Exemple 1.3 L'équivalence observée dans l'exemple 1.2 entraîne l'équivalence $a + (b+c) =_{AC} (b+c) + a$

Les définitions concernant les ordres sur les termes se déclinent naturellement sur les termes variadiques. La définition d'un ordre de simplification doit cependant être adaptée :

Définition 1.13 Un ordre $<$ sur $\mathcal{TV}(\Sigma, \mathcal{X})$
- satisfait la *propriété d'effacement* si $f \ldots\ \ldots < f \ldots s \ldots$ pour tout symbole $f \in \mathcal{V}$.
- est un ordre de simplification s'il possède les propriétés d'effacement et de sous-terme.

1.4.2 Construction d'un ordre récursif sur les chemins AC-compatible

On reprend les notations du paragraphe précédent, et on suppose que Σ est un ensemble fini de symboles ordonné par une précédence $<_\Sigma$. Rubio commence par introduire une chaîne de termes variadiques obtenue à partir d'un terme variadique.

Définition 1.14 L'*aplatissement de tête* d'un terme s par rapport à un symbole AC f, noté $flat(s, f)$, est une chaîne de termes définie par $flat(s, f) = s_1 \dots s_n$ si $fs_1 \dots s_n$ est la forme normale de s par les règles d'aplatissement appliquées uniquement en position de tête, et $flat(s, f) = s$ si $s(\varepsilon) \neq f$

Exemple 1.4 Imaginons que $+$ et $*$ soient deux symboles AC, et posons $s = ((a * (b * c)) + d) * (e * (f + g))$. La forme normale de s par les règles d'aplatissement appliquées uniquement en position de tête de s est $((a*(b*c))+d)*e*(f+g)$, donc $flat(s, f) = ((a*(b*c))+d).e.(f+g)$

Rubio introduit ensuite un ensemble de termes variadiques obtenu à partir d'un terme possédant en tête un symbole variadique.

Définition 1.15 Soit $s = fs_1 \dots s_n$ avec $f \in \mathcal{V}$:
- l'ensemble $Embsmall(s)$ est défini par :

$$\{fs_1 \dots flat(v_j, f) \dots s_n \mid s_i = hv_1 \dots v_r \wedge h <_\Sigma f \wedge j \in \{1, \dots, r\}\};$$

- le multi-ensemble $Bighead(s)$ est défini par :

$$\{\{s_i \mid 1 \leq i \leq n \wedge f <_\Sigma s_i(\varepsilon)\}\}$$

Exemple 1.5 On se place de nouveau dans la situation de l'exemple 1.4, et on suppose que $+ <_\Sigma *$.

$$Embsmall(s) = \{a * b * c * e * (f + g), d * (e * (f + g))\}$$

Il définit alors un multi-ensemble de termes variadiques obtenu à partir d'un terme possédant en tête un symbole variadique.

Définition 1.16 Soit $s = fs_1 \dots s_n$, avec $f \in \mathcal{V}$. Le multi-ensemble $NoSmallHead(s)$, est défini par $\{s_i \mid 1 \leq i \leq n \wedge s_i(\varepsilon) \not<_\Sigma f\}$

Exemple 1.6 Posons $s = (a + a) + (a * a) + a + (a * a)$, et supposons que $a <_\Sigma + <_\Sigma *$.

$$NoSmallHead(s) = \{\{a + a, a * a, a * a\}\}$$

Il détermine finalement un polynôme du premier degré à plusieurs variables à coefficients entiers positifs obtenu à partir d'un terme variadique.

Définition 1.17 On définit la fonction $\#$ sur l'ensemble des termes variadiques par $\#(fs_1 \dots s_n) = \#_v(s_1) + \dots \#_v(s_n)$ avec
- $\#_v(x) = x$ si x est une variable
- $\#_v(t) = 1$ si t est une constante.

Exemple 1.7 Considérons trois symboles de la signature d'arité respective 4, 2 et 1. Posons $s = f(g(hx))xxy$. On a $\#(s) = 3x + y + 1$

On peut maintenant donner la définition de l'ordre $<$. On affecte à tout symbole non variadique f un statut $\tau(f) \in \{lex, mul\}$ (cf Définition 1.6 du paragraphe 1.2), et à tout terme variadique l'unique statut mul.

Définition 1.18 Soit $>$ la relation binaire définie sur $\mathcal{T}(\Sigma, \mathcal{X})$ par $s = fs_1 \dots s_n > gt_1 \dots t_n = t$ *ssi* :

1. $s_i \geq t$ pour un $i \in \{1, \dots n\}$, ou
2. $f >_\Sigma g$ et $s > t_i$ pour tout $i \in \{1, \dots, m\}$, ou
3. $\tau(f) = \tau(g) = lex$, $(s_1, \dots, s_n) >_{lex} (t_1, \dots, t_m)$, et $s > t_i$ pour tout $i \in \{1, \dots, m\}$, ou
4. $\tau(f) = \tau(g) = mul$ et $\{\{s_1, \dots, s_n\}\} \gg \{\{t_1, \dots, t_m\}\}$, ou
5. $f = g \in \mathcal{V}$ et il existe $s' \in Embsmall(s)$ tel que $s' \geq t$, ou
6. $f = g \in \mathcal{V}$, $s > t'$ pour tout $t' \in Embsmall(t)$, $NoSmallHead(s) \underline{\gg} NoSmallHead(t)$, et
   (a) soit $Bighead(s) \gg Bighead(t)$,
   (b) soit $\#(s) > \#(t)$,
   (c) soit $\#(s) \geq \#(t)$ et $\{\{s_1, \dots, s_n\}\} \gg \{\{t_1, \dots, t_m\}\}$.

Exemple 1.8 [RUB 02] Supposons que $g <_\Sigma f <_\Sigma h$. On a $s = f(g(hx))xxy > f(g(f(hx)y))x$ d'après le cas 6b, car $\#(s) = 2x + y + 1 > x + 1 = \#(t)$ et $NoSmallHead(s) = \{\{x, x, y\}\} \gg \{\{x\}\} = NoSmallHead(t)$ et $s > f(hx)yx = t' \in Embsmall(t)$ en appliquant d'abord le cas 5 et $s' = f(hx)xxy > t'$, d'après le cas 6b, puisque $NoSmallHead(s') = \{\{h(x), x, x, y\}\} \gg \{\{hx, y, x\}\} = NoSmallHead(t')$, $EmbSmall(t') = \emptyset$, et, du fait que x est un entier positif, $\#(s') = 2x + y + 1 > x + y + 1 = \#(t')$.

On peut alors énoncer le théorème suivant :

Théorème 1.8 *[RUB 02] Si $<_\Sigma$ est total, $<$ est un ordre de simplification AC-compatible et stable par substitution.*

Exemple 1.9 Supposons que $+$ et $*$ soient des symboles variadiques, et que l'on ait la précédence $0 <_\Sigma s <_\Sigma + <_\Sigma *$. Soit $\mathcal{R}$ le système de réécriture constitué des règles suivantes :

$$\begin{array}{ll} x + 0 \to x & x * 0 \to 0 \\ x + s(y) \to s(x + y) & x * s(y) \to x * y + x \end{array}$$

Pour toute règle $l \to r$ de $\mathcal{R}$, on a $l > r$. Si on pose $AC(+, *)$ l'ensemble des axiomes exprimant la commutativité et l'associativité des opérateurs $+$ et $*$, on peut déduire à l'aide du théorème 1.5, que le système de réécriture modulo $(\mathcal{R}, AC(+, *))$ est terminant.

2

Surréduction

Sommaire

2.1 Principe **75**
2.2 Surréduction modulo un ensemble d'égalités **76**
2.3 Surréduction et complétude suffisante **78**
2.3.1 Unificateurs constructeurs 78
2.3.2 Ensembles complets d'unificateurs constructeurs 79
2.3.3 Cas des théories associatives-commutatives 82

Dans ce chapitre, nous commençons par définir la surréduction et son extension au cas de la réécriture modulo. Nous rappelons en particulier le lemme classique de relevage [HUL 80a, KIR 99] qui fait le lien entre étape de réduction et étape de surréduction. Ce lemme permet d'en déduire que, si on considère un système de réécriture terminant sur les termes clos et suffisamment complet par rapport à un ensemble de constructeurs libres, et, dans le cas de la réécriture modulo un ensemble d'égalités, si on suppose que ces égalités préservent les termes constructeurs, il est possible de surréduire un terme au niveau de *chacune* de ses positions *définie-maximales*. En outre, la substitution surréductrice correspondante est un unificateur *constructeur* entre le sous-terme situé à cette position et le membre gauche de la règle appliquée. Dans les parties III et IV, nous verrons qu'il nous suffira en fait de ne rechercher qu'un ensemble *générateur* de ces unificateurs. Lorsque l'unification est syntaxique, cet ensemble est vide, ou réduit à l'unificateur le plus général. Dans le cas de la réécriture modulo, nous verrons qu'il existe des situations dans lesquelles cette recherche peut être simplifiée. La détermination d'un tel ensemble est notamment immédiate dans le cas de la réécriture modulo l'associativité-commutativité ou modulo l'associativité seule. Dans ce qui suit, on considère un système de réécriture $\mathcal{R}$ et un ensemble d'égalités E.

2.1 Principe

Le principe de surréduction d'un terme t consiste à appliquer à t une substitution rendant t réductible par une règle, et à appliquer cette règle ; la surréduction d'un terme fermé coïncide avec la réécriture. Cette méthode a été introduite par Slagle [SLA 74], étudiée par Fay [FAY 79], et complètement étudiée par J.M. Hullot [HUL 80a] dans le cas de la réécriture syntaxique. Elle est utilisée pour résoudre des équations, mais elle joue également un rôle fondamental dans le processus de récurrence.

Définition 2.1 Un terme t est *surréduit* en t' à une position ω avec la règle $l \to r \in \mathcal{R}$ et la substitution σ, si σ est l'unificateur le plus général (mgu en abrégé) de l et $t_{|\omega}$, et $t' = (t[r]_\omega)\sigma$. Cette étape de surréduction se note $t \underset{[l\to r,\omega,\sigma]}{\rightsquigarrow} t'$.

Remarque: On dit alors que t est $\mathcal{R}$-*surréductible* à la position ω. Le lemme classique de relevage ci-dessous fait le lien entre étape de réduction et étape de surréduction.

Lemme 2.1 [HUL 80a, KIR 99] Etant donnés deux termes $s_1, t'_1 \in \mathcal{T}(\Sigma, \mathcal{X})$, une substitution $\mathcal{R}$-normalisée α, une règle $l \to r \in \mathcal{R}$, une position $\omega \in \mathcal{D}om(s_1)$, une substitution ν, et un ensemble fini V de variables, tels que $s_1\alpha \to_{[l\to r,\omega,\nu]} t'_1$ et $(\mathcal{V}ar(s_1) \cup \mathcal{D}om(\alpha)) \subseteq V$, il existe deux substitutions $\sigma,\ \mu \in Subst^{\Sigma,\mathcal{X}}$, et un terme $t_1 \in \mathcal{T}(\Sigma, \mathcal{X})$, tels que :

1) $s_1 \rightsquigarrow_{[l\to r,\omega,\sigma]} t_1$;
2) $\sigma\mu = \alpha[V]$;
3) $t_1\mu = t'_1$.

d'où le diagramme :

$$\begin{array}{lll} s_1\alpha & \to_{[l\to r,\omega,\nu]} & t'_1 = t_1\mu \\ \uparrow \alpha & & \uparrow \mu \\ s_1 & \rightsquigarrow_{[l\to r,\omega,\sigma]} & t_1 \end{array}$$

Preuve. On donne au paragraphe suivant une preuve de la version de ce lemme étendue au cas de la réécriture de Peterson et Stickel définie à la sous-section 2.5.2 de la partie I □

La proposition suivante permet de deviner l'importance de la surréduction dans le processus de récurrence.

Proposition 2.1 Pour tout terme t et pour toute position $\mathcal{R}$-inductivement réductible ω de $\mathcal{D}om(t)$, t est $\mathcal{R}$-surréductible à la position ω.

Preuve. C'est une conséquence immédiate du lemme 2.1 et de la définition 2.35 de la partie I (dans la cas où $E = \emptyset$). □

Le corollaire ci-dessous permet alors de comprendre pourquoi les positions *définie-maximales* vont jouer un rôle important dans le système de recherche de preuve fondé sur la surréduction que nous présenterons dans la partie III :

Corollaire 2.1 Si $\mathcal{R}$ est terminant sur les termes clos et suffisamment complet (par rapport à un ensemble de constructeurs libres), alors, pour tout terme t, et pour toute position $\omega \in \mathcal{D}om(t)$, si ω est *définie-maximale* dans $\mathcal{D}om(t)$, t est également $\mathcal{R}$-surréductible à la position ω.

Preuve. C'est une conséquence du lemme 2.1 de la partie de la partie I et de la proposition 2.1 □

2.2 Surréduction modulo un ensemble d'égalités

Les résultats de J.M. Hullot ont été étendus au cas de la $\mathcal{R}, E-$réécriture dans [JOU 83]. Dans ce paragraphe, on considère un système de réécriture modulo $(\mathcal{R}, E)$, comme défini au paragraphe 2.5. La définition de la surréduction au cas de la $\mathcal{R}, E$-réécriture est une simple extension de la définition 2.1 ci-dessus.

Définition 2.2 Un terme t est E-surréduit en t' à la position ω avec la règle $l \to r \in \mathcal{R}$ et la substitution σ, si $\sigma \in CSU_E(t_{|\omega}, l)$, et $t' = (t[r]_\omega)\sigma$. Cette étape de surréduction est notée $t \underset{[l\to r,\omega,\sigma]\mathcal{R},E}{\rightsquigarrow} t'$.

Dans ce contexte, le lemme de relevage se met sous la forme :

Lemme 2.2 Etant donnés deux termes $s_1, t'_1 \in \mathcal{T}(\Sigma, \mathcal{X})$, une substitution $\mathcal{R}, E$-normalisée α, une règle $l \to r \in \mathcal{R}$, une position $\omega \in \mathcal{D}om(s_1)$, une substitution ν, et un ensemble fini V de variables, tels que $s_1\alpha \to_{[l\to r,\omega,\nu]\mathcal{R},E} t'_1$ et $(\mathcal{V}ar(s_1) \cup \mathcal{D}om(\alpha)) \subseteq V$, il existe deux substitutions $\sigma,\ \mu \in Subst^{\Sigma,\mathcal{X}}$, et un terme $t_1 \in \mathcal{T}(\Sigma, \mathcal{X})$, tels que :

1) $s_1 \rightsquigarrow_{[l\to r,\omega,\sigma]\mathcal{R},E} t_1$;
2) $\sigma\mu =_E \alpha[V]$;
3) $t_1\mu =_E t'_1$.

d'où le diagramme :

$$\begin{array}{lll} s_1\alpha & \to_{[l\to r,\omega,\nu]\mathcal{R},E} & t'_1 =_E t_1\mu \\ \uparrow \alpha & & \uparrow \mu \\ s_1 & \rightsquigarrow_{[l\to r,\omega,\sigma]\mathcal{R},E} & t_1 \end{array}$$

Preuve. Par hypothèse, il existe une règle de réécriture $l \to r \in \mathcal{R}$, et une substitution ν, telles que :

$$s_1\alpha = (s_1\alpha)[t]_\omega \quad , \quad t =_E l\nu \quad , \quad t'_1 = t'_1[r\nu]_\omega. \tag{2.1}$$

Remarquons que l'on peut supposer :

$$\mathcal{V}ar(l) \cap V = \emptyset \quad \text{et} \quad \mathcal{D}om(\nu) \subseteq \mathcal{V}ar(l). \tag{2.2}$$

Puisque $u \in \mathcal{D}om(s_1)$, $t = (s_1\alpha)_{|\omega}$, on a :

$$s_1 = s_1[u]_\omega \quad \text{avec} \quad u\alpha = t. \tag{2.3}$$

(2.1) et (2.3) conduisent alors à $u\alpha =_E l\nu$, et, $u \notin \mathcal{X}$, car α est $\mathcal{R}, E$ -normalisée.
Maintenant, il faut se souvenir que $\mathcal{V}ar(s_1) \cap \mathcal{V}ar(l) = \emptyset$, donc, du fait que $s_1 = s_1[u]_\omega$, $\mathcal{V}ar(u) \cap \mathcal{V}ar(l) = \emptyset$. Et, puisque $\mathcal{D}om(\nu) \subseteq \mathcal{V}ar(l)$, d'après (2.2), on obtient $\mathcal{V}ar(u) \cap \mathcal{D}om(\nu) = \emptyset$. Maintenant, d'après (2.2) et les hypothèses, $\mathcal{V}ar(l) \cap \mathcal{D}om(\alpha) = \emptyset$. Si on se souvient que $u\alpha =_E l\nu$, si on pose ρ la substitution définie par $\rho = \nu + \alpha$, on est alors prêt à comprendre les égalités suivantes :

$$u\rho = u\alpha =_E l\nu = l\rho$$

Par conséquent, u et l sont E unifiables. Donc, il existe deux substitutions σ et μ', telles que :

$$\sigma \in CSU_E(u, l) \quad \text{et} \quad \sigma\mu' =_E \rho\ [\mathcal{V}ar(u) \cup \mathcal{V}ar(l)] \tag{2.4}$$

Posons $t_1 = (s_1[r]_\omega)\sigma$, on a :

$$s_1 \underset{[l\to r,\omega,\sigma]\mathcal{R},E}{\rightsquigarrow} t_1$$

Evidemment, $\sigma\mu' =_E \rho\ [\mathcal{V}ar(u)]$, et, du fait que $\rho = \nu + \alpha$ et $\mathcal{V}ar(u) \cap \mathcal{D}om(\nu) = \emptyset$, cela conduit à l'égalité suivante :

$$\sigma\mu' =_E \alpha[\mathcal{V}ar(u)] \tag{2.5}$$

On peut maintenant supposer $\mathcal{R}an(\sigma) \cap V = \emptyset$ et poser $\mu = \mu'_{|Ran(\sigma)} + \alpha_{|V}$. Soit $x \in V$. On peut distinguer deux cas.

– Cas 1 : $x \in \mathcal{D}om(\sigma)$.
  Soit $y \in \mathcal{V}ar(x\sigma)$. On a $y \in \mathcal{R}an(\sigma)$, donc, par définition de μ, $y\mu = y\mu'$. Par conséquent, du fait que y est choisi arbitrairement, on a $x\sigma\mu = x\sigma\mu'$. Observons maintenant que $\mathcal{D}om(\sigma) \subseteq \mathcal{V}ar(u) \cup \mathcal{V}ar(l)$, ainsi, puisque $\mathcal{V}ar(l) \cap V = \emptyset$ (d'après 2.2), $\mathcal{D}om(\sigma) \cap V \subseteq \mathcal{V}ar(u)$. On a donc $x \in \mathcal{V}ar(u)$ car $x \in \mathcal{D}om(\sigma)$ et $x \in V$ par hypothèse, et, en considérant (2.5), cela nous conduit à écrire :

$$x\sigma\mu =_E x\alpha \tag{2.6}$$

– Cas 2 : $x \notin \mathcal{D}om(\sigma)$.
Premièrement, on a :

$$x\sigma\mu = x\mu \tag{2.7}$$

Deuxièmement, sachant que $x \in V$, et par définition de μ, on a :

$$x\mu = x\alpha \tag{2.8}$$

Par conséquent, d'après (2.7) et (2.8), cela entraîne l'égalité $x\sigma\mu = x\alpha$

Ainsi, du fait que x est choisi arbitrairement dans V, on obtient :

$$\sigma\mu =_E \alpha[V] \tag{2.9}$$

Posons maintenant $x \in \mathcal{V}ar(r)$. sachant que $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$, et que l'on peut supposer $\mathcal{V}ar(l) \subseteq \mathcal{D}om(\sigma)$, on a $x \in \mathcal{D}om(\sigma)$. Prenons $y \in \mathcal{V}ar(x\sigma)$, $y \in \mathcal{R}an(\sigma)$, donc, par définition de μ, $y\mu = y\mu'$. Par conséquent, du fait que y a été choisi arbitrairement, on obtient :

$$x\sigma\mu = x\sigma\mu' \tag{2.10}$$

Mais, puisque $\mathcal{V}ar(r) \subseteq \mathcal{V}ar(l)$, et d'après (2.4), cela entraîne $x\sigma\mu =_E x\rho$. Souvenons-nous maintenant que $\mathcal{V}ar(l) \cap \mathcal{D}om(\alpha) = \emptyset$, donc $\mathcal{V}ar(r) \cap \mathcal{D}om(\alpha) = \emptyset$. Cela implique, par définition de ρ, que l'on a $x\rho = x\nu$. Donc, $x\sigma\mu =_E x\nu$ pour tout $x \in \mathcal{V}ar(r)$, et cela entraîne $r\sigma\mu =_E r\nu$. En outre, d'après (2.9), et du fait que $\mathcal{V}ar(s_1) \subseteq V$, on a $s_1\sigma\mu =_E s_1\alpha$. On peut alors écrire :

$$t_1\mu = (s_1[r]_{|\omega})\sigma\mu = (s_1\sigma\mu)[r\sigma\mu]_{|\omega} =_E (s_1\alpha)[r\nu]_{|\omega} = t'_1$$

Autrement dit :

$$t_1\mu =_E t'_1$$

□

On peut ensuite décliner les propositions 2.1 et le corollaire 2.1 au contexte de la $\mathcal{R}, E-$réécriture.

Proposition 2.2 Pour tout terme t et pour toute position $\mathcal{R}, E-$inductivement réductible $\omega \in \mathcal{D}om(t)$, t est surréductible à la position ω.

Preuve. C'est une conséquence immédiate du lemme 2.2 et de la définition 2.35 de la partie I. □

2.3 Surréduction et complétude suffisante

On suppose dans tout ce paragraphe que la signature Σ est la réunion disjointe d'un sous-ensemble $\mathcal{C}$ de symboles constructeurs et d'un sous-ensemble $\mathcal{D}$ de symboles définis.

2.3.1 Unificateurs constructeurs

Le corollaire suivant montre que si on suppose que l'ensemble d'égalités E préserve les constructeurs et que le système de réécriture modulo $(\mathcal{R}, E)$ est suffisamment complet (par rapport à un ensemble de constructeurs libres), la surréduction à des positions *définie-maximales* permet de n'avoir à considérer que les unificateurs dont les images sont des termes constructeurs. Il n'est alors plus nécessaire de déterminer des ensembles complets d'unificateurs, et on peut

ainsi contourner des difficultés importantes comme par exemple le fait que l'unification AC est de complexité doublement exponentielle. Cela rend également possible d'effectuer des preuves par récurrence dans le cas où la théorie E n'est pas finitaire (par exemple dans le cas E est un ensemble d'axiomes associatifs).

Proposition 2.3 Supposons que :

1. $\mathcal{R}$ est E-terminant sur l'ensemble des termes clos ;
2. le système $(\mathcal{R}, E)$ est suffisamment complet ;
3. E préserve les constructeurs ;

alors, pour tout terme t, pour toute position $\omega \in \mathcal{DM}(t)$, pour toute instance close $\mathcal{R}, E$-normalisée α de t, pour tout ensemble V tel que $(\mathcal{V}ar(t) \cup \mathcal{D}om(\alpha)) \subseteq V$, il existe une règle de réécriture $l \to r \in \mathcal{R}$, et deux substitutions σ, μ telles que :

1. $\sigma \in CSU_E(t_{|\omega}, l)$
2. $Im(\sigma) \subset \mathcal{T}(\mathcal{C}, \mathcal{X})$
3. $\sigma\mu =_E \alpha[V]$

Preuve. Posons $t \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega \in \mathcal{DM}(t)$, α une substitution close $\mathcal{R}, E$-normalisée de t , et V un ensemble tel que $(\mathcal{V}ar(t) \cup \mathcal{D}om(\alpha)) \subseteq V$. En considérant les hypothèses, et d'après le lemme 2.2 ci-dessus, on sait qu'il existe une règle de réécriture $l \to r \in \mathcal{R}$, $\sigma \in CSU_E(t_{|\omega}, l)$, et une substitution μ, telles que : $\sigma\mu =_E \alpha[V]$. Par conséquent, il reste à prouver que σ est une substitution constructeur.
Prenons $x \in \mathcal{V}ar(t)$. On a $x\sigma\mu =_E x\alpha$. De plus, sachant que α est une substitution close $\mathcal{R}, E$-normalisée de t, et que le système $(\mathcal{R}, E)$ est suffisamment complet, $x\alpha \in \mathcal{T}(\mathcal{C})$. Des deux résultats ci-dessus, et sachant que E préserve les constructeurs, on déduit $x\sigma \in T(\mathcal{C})$. □

On appellera *unificateurs constructeurs* les unificateurs à valeurs dans l'ensemble des termes constructeurs.

Définition 2.3 Soient $s,\ t \in \mathcal{T}(\Sigma, \mathcal{X})$ et σ une substitution. On dit que σ est un E-unificateur constructeur de s et de t si

1. $s\sigma =_E t\sigma$;
2. $Im(\sigma) \subseteq \mathcal{T}(\mathcal{C}, \mathcal{X})$.

2.3.2 Ensembles complets d'unificateurs constructeurs

La proposition 2.3 donne déjà une idée du rôle important que vont jouer dans le processus de récurrence les unificateurs constructeurs entre les sous-termes des égalités à démontrer et les membres gauches des règles. Ce rôle sera clairement défini dans le système de recherche de preuve par récurrence utilisant la réécriture modulo un ensemble d'égalités que nous présenterons dans la partie IV. Nous serons alors plus particulièrement motivés par la recherche de parties génératrices de ces ensembles. La définition d'une telle partie génératrice, que nous introduisons maintenant, est analogue à celle des ensembles complets d'unificateurs, introduite d'abord par G. Plotkin [PLO 72], suivi de G. Huet [HUE 76] puis de J.-M. Hullot [HUL 80b]. L'unique et importante différence est que l'on se restreint ici aux seuls unificateurs constructeurs.

Définition 2.4 Deux termes $s, t \in \mathcal{T}(\Sigma, \mathcal{X})$ étant donnés, $CSUC_E(s, t)$ est un ensemble complet de E-unificateurs constructeurs de s et de t si :

1. **Correction :** Chaque élément $\sigma \in CSUC_E(s,t)$ est un E-unificateur constructeur de s et de t ;
2. **Complétude :** Pour tout E-unificateur constructeur θ de s et de t, il existe $\sigma \in CSUC_E(s,t)$ et une substitution μ, tels que $\theta =_E \sigma\mu[\mathcal{V}ar(s) \cup \mathcal{V}ar(t)]$;
3. **Domaine :** pour tout $\sigma \in CSUC_E(s,t)$, $\mathcal{R}an(\sigma) \cap \mathcal{D}om(\sigma) = \emptyset$.

Nous allons mettre en évidence une situation où la recherche d'ensembles complets de E-unificateurs constructeurs peut être simplifiée. Il s'agit du cas où l'ensemble E contient soit des égalités dont chaque membre contient des symboles définis uniquement en tête, soit des égalités dont chaque membre contient des symboles définis ailleurs qu'en position de tête : c'est par exemple le cas des théories associatives-commutatives, en faisant l'hypothèse que tous les symboles AC sont également des symboles définis. Nous commençons par introduire un lemme où ces caractéristiques apparaissent.

Lemme 2.3 Supposons que E soit un ensemble d'égalités, et que F et G soient deux sous-ensembles de E, tels que :

1. $$F = \{e_1 \approx e_2 \in E \mid e_1(\varepsilon) \in \mathcal{D},\ e_2(\varepsilon) \in \mathcal{D},\ \text{et } e_i(\omega) \in \mathcal{C} \cup \mathcal{X} \text{ si } \omega \neq \varepsilon \text{ et } i \in \{1,2\}\}$$

   (i.e F est le sous-ensemble des égalités de E qui possèdent un symbole défini uniquement en tête de chaque membre)
2. $$G = \{e_1 \approx e_2 \in E \mid \exists\omega\ \exists\omega'\ \omega \neq \varepsilon,\ \omega' \neq \varepsilon,\ e_1(\omega) \in \mathcal{D} \text{ et } e_2(\omega) \in \mathcal{D}\}$$

   (i.e G est le sous-ensemble des égalités de E qui possèdent un symbole défini dans chaque membre ailleurs qu'en tête)
3. $$E = F \cup G$$
4. $$\mathcal{F} = \{f \in \Sigma \mid \exists e_1\ \exists e_2\ e_1 \approx e_2 \in F,\ f = e_1(\varepsilon) \text{ ou } f = e_2(\varepsilon)\}$$

Alors, pour tout symbole défini f d'arité n, pour tout $t_1, \ldots t_n \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, pour tout $t \in \mathcal{T}(\Sigma, \mathcal{X})$, on a :

1. Si $ft_1 \ldots t_n =_F s$, alors il existe un symbole défini g d'arité m et des termes constructeurs $s_1, \ldots, s_m \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, tels que : $s = gs_1 \ldots s_m$.
2. $$ft_1 \ldots t_n =_E s \Leftrightarrow ft_1 \ldots t_n =_F s.$$
3. $$((f \notin \mathcal{F}) \wedge (ft_1 \ldots t_n =_E t)) \Rightarrow ft_1 \ldots t_n = t$$

Preuve. – Pour faire la preuve de 1. et 2., il suffit de montrer 1' et 2' ci-dessous :

1' si $ft_1 \ldots t_n \leftrightarrow_F s$, alors il existe un symbole défini g d'arité m et des termes constructeurs $s_1, \ldots, s_m \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, tels que $s = gs_1 \ldots s_m$

2'
$$ft_1 \ldots t_n \leftrightarrow_E s \Leftrightarrow ft_1 \ldots t_n \leftrightarrow_F s$$

1' Supposons $ft_1 \dots t_n \leftrightarrow_F t$. Il existe une égalité $e_1 \approx e_2 \in F$ et une substitution σ, telles que $ft_1 \dots t_n = C[e_i\sigma]_\omega$ et $t = C[e_j\sigma]_\omega$. Mais comme, par définition de F, $e_i(\varepsilon)$ est un symbole défini et $t_1, \dots, t_n$ sont des termes constructeurs, C est donc le contexte vide, $Im(\sigma) \subset \mathcal{T}(\mathcal{C}, \mathcal{X})$, $ft_1 \dots t_n = e_i\sigma$, $t = e_j\sigma$, et la preuve est terminée.

2' Supposons $ft_1 \dots t_n \leftrightarrow_E t$. Il existe une égalité $e_1 \approx e_2 \in E$ et une substitution σ, telles que $ft_1 \dots t_n = C[e_i\sigma]_\omega$, et $t = C[e_j\sigma]_\omega$. Or, e_i contient un symbole défini, par définition de F et de G, et $t_1, \dots, t_n$ sont des termes constructeurs. C est donc le contexte vide et $e_i(\omega) \in \mathcal{C} \cup \mathcal{X} \Leftrightarrow \omega \neq \varepsilon$. Et, en considérant les hypothèses 1, 2, 3, cela entraîne $e_1 \approx e_2 \in F$, ce qui termine la preuve.

– Supposons $(f \notin \mathcal{F}) \wedge (ft_1 \dots t_n =_E t)$. Alors, d'après 2., on a $ft_1 \dots t_n =_F t$. Pour prouver 3., il suffit donc de vérifier qu'une étape de dérivation $ft_1 \dots t_n \leftrightarrow_F t'$ ne peut se produire, ce qui est immédiat, car $f \notin \mathcal{F}$ et $t_1, \dots t_n \in \mathcal{T}(\mathcal{C}, \mathcal{X})$

□

Le corollaire suivant montre ainsi que, lorsque les conditions du lemme 2.3 sont réunies, la recherche d'un ensemble complet de E-unificateurs constructeurs peut être considérablement simplifiée. Il distingue même un cas où cet ensemble est vide ou admet l'unificateur le plus général dans la théorie vide pour unique élément.

Théorème 2.1 *On reprend les hypothèses du lemme 2.3. Pour tout symbole défini f d'arité n, pour tous $t_1, \dots t_n \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, pour tout $t \in \mathcal{T}(\Sigma, \mathcal{X})$, on a :*

1. *Si $f \in \mathcal{F}$,*
$$CSUC_E(ft_1 \dots t_n, t) = CSUC_F(ft_1 \dots t_n, t);$$
2. *Si $f \notin \mathcal{F}$,*
$$CSUC_E(ft_1 \dots t_n, t) = CSUC_\emptyset(ft_1 \dots t_n, t);$$

Preuve. Soit σ un E-unificateur de $ft_1 \dots t_n$ et de t.

1. On a $ft_1\sigma \dots t_n\sigma =_E t\sigma$. Or, $t_i\sigma \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, et, d'après le lemme 2.3, on a $ft_1\sigma \dots t_n\sigma =_F t\sigma$, et la preuve est terminée.
2. Si $f \notin \mathcal{F}$, sachant que $ft_1\sigma \dots t_n\sigma =_E t\sigma$, $t_i\sigma \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, et, par le lemme 2.3, on a l'égalité syntaxique $ft_1\sigma \dots t_n\sigma = t\sigma$ et la preuve est terminée.

□

L'exemple suivant aide à comprendre l'intérêt de ce corollaire.

Exemple 2.1 On suppose que $+$ est un symbole défini d'arité 2 et a un symbole de constante.

– Soit $A(+) = \{(x + y) + z \approx x + (y + z)\}$. Plotkin [PLO 72] a montré que $A(+)$ n'est pas de type finitaire. Par exemple, les termes $x + a$ et $a + x$ possèdent une infinité de $A(+)$-unificateurs indépendants $\{x \mapsto a\}$, $\{x \mapsto a+a\}$, $\{x \mapsto a+(a+a)\}$, ... Or, d'après le théorème 2.1, $CSUC_{A(+)}(x + a, a + x) = \{x \mapsto a\}$.
– Soit $E(+) = \{(x + y) + z \approx (y + x) + z\}$. M.Echenim [ECH 05] a montré que $E(+)$ n'est pas finitaire. En particulier, tout ensemble complet d'unificateurs minimal pour $x + y$ et $z + y$ possède nécessairement une infinité d'éléments. Or, d'après le théorème 2.1, $CSUC_{E(+)}(x + y, z + y) = CSUC_\emptyset(x + y, z + y) = \{x \mapsto z\}$.

2.3.3 Cas des théories associatives-commutatives

On se place de nouveau dans le contexte du paragraphe 1.4.1. Nous étudions ici une situation très particulière que nous rencontrerons cependant très souvent dans la partie IV : il s'agit du cas où l'on cherche à déterminer un ensemble complet d'unificateurs constructeurs entre deux termes de la forme fx_1x_2 et ft_1t_2 , avec $f \in \mathcal{V}$ qui est également un symbole défini, $x_1,\ x_2$ des variables, et $t_1,\ t_2$ des termes.
Nous commençons par introduire le lemme suivant :

Lemme 2.4 Supposons que tous les symboles de $\mathcal{V}$ sont des symboles définis, alors, pour tous $t_1, t_2 \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, pour tout $t \in \mathcal{T}(\Sigma, \mathcal{X})$, et pour tout $f \in \mathcal{V}$, $ft_1t_2 =_{AC} t$ si, et seulement si $ft_1t_2 = t$ ou $ft_2t_1 = t$.

Preuve. Il suffit de vérifier que, pour tous $t_1, t_2 \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, pour tout $t \in \mathcal{T}(\Sigma, \mathcal{X})$, et pour tout symbole $f \in \mathcal{V}$, $ft_1t_2 \leftrightarrow_{AC} t$ si, et seulement si $ft_1t_2 = t$ ou $ft_2t_1 = t$. Soit $F = \{fxy \approx fyx \mid f \in \mathcal{V}\}$, $G = \{f(fxy)z \approx fx(fyz) | f \in \mathcal{V}\}$, et $E = F \cup G$. On peut facilement s'apercevoir que les hypothèses 1, 2, 3 du lemme 2.3 sont satisfaites. Soit $f \in \mathcal{V}$, $t_1, t_2 \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, $t \in \mathcal{T}(\Sigma, \mathcal{X})$, et supposons l'équivalence $ft_1t_2 \leftrightarrow_{AC} t$. On a alors, avec ces notations, $ft_1t_2 \leftrightarrow_E t$, donc d'après le lemme 2.3, on en déduit $ft_1t_2 \leftrightarrow_F t$. On a alors clairement $t = ft_2t_1$, et le lemme se démontre par une récurrence facile sur la longueur de la dérivation $ft_1t_2 \overset{*}{\leftrightarrow}_F t$ □

Nous montrons maintenant que dans la situation du lemme 2.3, la recherche d'ensembles complets d'unificateurs peut devenir triviale. Ce corollaire très souvent utilisé implicitement dans la partie IV.

Corollaire 2.2 Supposons que :

1. tous les symboles de $\mathcal{V}$ sont définis ;
2. $f \in \mathcal{V}$, $t,\ t' \in \mathcal{T}(\Sigma, \mathcal{X})$, $t_1,\ t_2 \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, et $x_1,\ x_2 \in \mathcal{X}$ sont tels que :
   (a) $t = ft_1t_2$;
   (b) $t' = fx_1x_2$;
   (c) $\mathcal{V}ar(t) \cap \mathcal{V}ar(t') = \emptyset$;
3. θ, σ_1, σ_2 sont des substitutions telles que :
   (a) θ est un renommage des variables de $\mathcal{V}ar(t)$ par des variables fraîches ;
   (b) $\sigma_1 = \{x_1 \to t_1; x_2 \to t_2\}$;
   (c) $\sigma_2 = \{x_1 \to t_2; x_2 \to t_1\}$;

alors, $CSUC_{AC}(t, t') = \{\sigma_1\theta, \sigma_2\theta\}$

Preuve. Il faut vérifier 1, 2, et 3 de la Définition 2.4.

1. : Clair.
2. : Supposons $\sigma \in UnifC_{AC}(t, t')$. On a $fx_1\sigma x_2\sigma = ft_1\sigma t_2\sigma$. Par conséquent, du fait que $Im(\sigma) \subset \mathcal{T}(\mathcal{C}, \mathcal{X})$, $f \in \Sigma_{AC}$, et d'après le lemme 2.4, on a :

$$\begin{cases} x_1\sigma = t_1\sigma \\ x_2\sigma = t_2\sigma \end{cases} \quad \text{ou} \quad \begin{cases} x_1\sigma = t_2\sigma \\ x_2\sigma = t_1\sigma \end{cases}$$

On peut alors facilement vérifier que, dans le premier cas, $\sigma = (\sigma_1\theta)(\theta^{-1}\sigma)$, et, dans le second cas, $\sigma = (\sigma_2\theta)(\theta^{-1}\sigma)$.

3. : clair

□

L'exemple suivant peut aider à comprendre l'intérêt du corollaire 2.2.

Exemple 2.2 Supposons que + est un symbole d'arité 2, et que u, v sont deux symboles de constante. Posons $AC(+) = \{x + y \approx y + x, (x + y) + z \approx x + (y + z)\}$. Alors, les termes $x + y$ et $u + v$ possèdent un ensemble complet de $AC(+)$-unificateurs qui compte 7 éléments ([KIR]). Or, d'après le corollaire 2.2, $CSUC_{AC(+)}(x + y, u + v) = \{\{x \mapsto u; y \mapsto v\}; \{x \mapsto v; y \mapsto u\}\}$.

Troisième partie

Un système de recherche de preuve par récurrence fondé sur la surréduction

1

Une première version

Sommaire

1.1 Le système de recherche de preuve IndNarrow **87**
1.2 Un exemple simple **88**
1.3 Correction **90**
1.4 Correction réfutationnelle **96**
1.5 Complétude réfutationnelle **98**

Le système de recherche de preuve IndNarrow pour les preuves par récurrence introduit dans cette section est fondé sur la surréduction et la réécriture. La règle principale, intitulée **Induce**, réalise l'étape de récurrence. C'est l'étape clé qui établit un pont entre les approches implicites et explicites pour la récurrence. On démontre également la correction et la complétude réfutationnelle de ce système.

1.1 Le système de recherche de preuve IndNarrow

Les règles exposées dans la Figure 1.1 s'appliquent sur des séquents modulo de la forme $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$, où :

- $\mathcal{RE}_1$ est l'ensemble des règles de réécriture de la spécification utilisateur ;
- Γ_1 contient les axiomes de la spécification utilisateur qui ne sont pas (encore) transformés en règles de réécriture ;
- Γ_2 contient des propositions fondamentales pour le raisonnement équationnel ou par récurrence et que l'on va expliciter ;
- $\mathcal{RE}_2$ est destiné à recevoir des règles de réécriture qui seront produites au fur et à mesure du déroulement de la preuve et qui représentent les hypothèses de récurrence ;
- Q est un but équationnel.

On fait maintenant les hypothèses suivantes :

1. $\mathcal{RE}_1$ ne contient que des règles ou des égalités non conditionnelles ;
2. $\mathcal{RE}_1$ est terminant sur les termes clos et suffisamment complet par rapport à un ensemble de constructeurs libres ;
3. $\leqslant$ est un quasi-ordre récursif sur les chemins induit par une quasi-précédence sur la signature (voir définition 1.8 de la partie II) dont on note $<$ la partie stricte et $\leqslant_n$ l'extension cartésienne de $<$ à $\mathcal{T}(\Sigma)^n$ (voir définition 2.5 de la partie I) ;
4. Γ_2 est initialisé avec :

(a) la proposition NI définie à la sous-section 3.4

$$NI: \ \forall R \ \forall \tau \ Noeth(R, \tau) \Rightarrow \forall P \ NoethInd(P, R, \tau);$$

(b) l'égalité de Leibniz $L(\approx)$:

$$L(\approx): \ x \approx y \Rightarrow (\forall P \ \forall x \ \forall y \ P(x) \Rightarrow P(y));$$

(c) Un prédicat $Noeth(<, \mathcal{T}(\Sigma))$ qui établit que $<$ est noethérien sur $\mathcal{T}(\Sigma)$.

En prenant en compte les vecteurs de variables, nous sommes maintenant en mesure d'instancier l'hypothèse noethérienne de récurrence $\mathcal{RE}_{ind}(Q, \prec, \tau)(X)$ définie à la section 3.4 de la partie I Si $\overrightarrow{x} \in \mathcal{X}^n$ est le vecteur des variables de Q, on a :

$$\mathcal{RE}_{ind}(Q, <_n, \mathcal{T}(\Sigma)^n) = (\overrightarrow{\underline{x}} \in \mathcal{T}(\Sigma)^n) \wedge (\overrightarrow{\underline{x}} <_n \overrightarrow{x}) \Rightarrow Q\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}$$

Pour simplifier les notations, et si il n'y a pas d'ambigüités, on la notera simplement $\mathcal{RE}_{ind}(Q, <)$ (ou encore $\mathcal{RE}_{ind}(Q)$). Pour toute substitution σ, on pose alors :

$$\mathcal{RE}_{ind}(Q, <)\sigma = (\overrightarrow{\underline{x}} \in \mathcal{T}(\Sigma)^n) \wedge (\overrightarrow{\underline{x}} <_n \overrightarrow{x}\sigma) \Rightarrow Q\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}$$

$\mathcal{RE}_2$ reçoit ainsi les hypothèses de récurrence fournies par l'application de la règle **Induce**. Donc $\mathcal{RE}_2$ peut contenir des égalités conditionnelles. Les séquents sont stockés dans une structure de multi-ensemble modélisée par l'opérateur $\bullet$, qui est un opérateur AC sur les séquents, avec $\diamond$ pour élément neutre. La règle principale est **Induce**, du fait qu'elle réalise l'étape de récurrence. Nous verrons qu'elle utilise la surréduction pour choisir à la fois les variables de récurrence et le schéma d'instanciation. La surréduction est appliquée uniquement au niveau de positions *définie-maximales* (voir Définition 2.27 de la partie I) $\mathcal{DM}(Q)$ du but courant Q. Les autres règles effectuent les tâches suivantes : **Trivial** élimine une égalité triviale, **Push** pousse une hypothèse équationnelle de la partie déduction vers la partie calcul, **Orient** oriente une égalité de la partie calcul pour la transformer en règle de réécriture suivant l'ordre sur les termes qui aura été choisi, **Rewrite** (1 ou 2) réécrit à l'aide d'une règle, d'une égalité, ou d'une instance plus petite d'un but précédent. **Push** et **Rewrite** sont dupliquées du fait de la distinction $\Gamma_1/\mathcal{RE}_1$ et $\Gamma_2/\mathcal{RE}_2$.

1.2 Un exemple simple

Afin d'avoir une meilleure compréhension de la façon dont cet ensemble de règles fonctionne, observons la preuve de la commutativité de l'addition dans l'arithmétique de Peano. Si on pose $\Gamma_2 = \{NI, L(\approx), Noeth(<_2, \mathcal{T}(\Sigma)^2)\}$, le but est donc représenté par le séquent suivant :

$\boxed{x + 0 \approx x, x + s(y) \approx s(x + y) | \Gamma_2 \vdash_{\emptyset|\emptyset} X + Y \approx Y + X}$

En appliquant la règle $\mathbf{Push}_1$ deux fois, nous obtenons :

$\boxed{\emptyset | \Gamma_2 \vdash_{x+0\approx x, x+s(y)\approx s(x+y)|\emptyset} X + Y \approx Y + X}$

Puis , l'application de **Orient** deux fois nous donne :

$\boxed{\emptyset | \Gamma_2 \vdash_{x+0\rightarrow x, x+s(y)\rightarrow s(x+y)|\emptyset} X + Y \approx Y + X}$

Il est maintenant possible d'appliquer la règle **Induce**, car $\mathcal{RE}_1 = \{x+0 \rightarrow x, x+s(y) \rightarrow s(x+y)\}$ est terminant sur les termes clos et suffisamment complet. Cela peut être fait à l'occurence 1 ou alors à l'occurence 2 du but. On choisit arbitrairement l'occurence 1, et cela nous conduit à prouver les deux séquents :

Induce $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[t]_\omega \rightarrowtail$

$\bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma = mgu(t,l)}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\sigma \cup \mathcal{RE}_{ind}(Q,<)\sigma} (Q[r]_\omega)\sigma$

si $\omega \in \mathcal{DM}(Q)$

Orient $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \to r\}|\mathcal{RE}_2} Q$

si $\kappa = l \approx r$ ou $\kappa = r \approx l$ et $l > r$

Push$_1$ $\Gamma_1, l \approx r|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \approx r\}|\mathcal{RE}_2} Q$

Push$_2$ $\Gamma_1|\Gamma_2, l \approx r \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{l \approx r\}} Q$

Rewrite$_1$ $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} (Q[l]_\omega)\sigma \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} (Q[r]_\omega)\sigma$

si $\kappa = l \to r$ ou $\kappa = l \approx r$ ou $\kappa = r \approx l$

Rewrite$_2$ $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\kappa\}} (Q[l]_\omega)\sigma \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\kappa\}} (Q[r]_\omega)\sigma$

si $\kappa = l \approx r$ ou $\kappa = r \approx l$ ou

$((\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$ ou $\kappa = \mathcal{RE}_{ind}(r \approx l)\mu)$ et $\overrightarrow{\mathcal{V}ar(l \approx r)}\sigma <_n \overrightarrow{\mathcal{V}ar(l \approx r)}\mu)$

Trivial $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} t \approx t \rightarrowtail \diamond$

Refutation $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail$ Refutation

si aucune autre règle ne peut être appliquée

FIG. 1.1 – Le système de recherche de preuve IndNarrow

$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto X_1;Y\mapsto 0\}} X_1 \approx 0 + X_1$
$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto X_1;Y\mapsto s(Y_1)\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$

On doit maintenant prouver en particulier que 0 est neutre à gauche. La seule règle applicable à ce but est **Induce** une nouvelle fois, et on obtient les deux nouveaux sous-buts :

$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\substack{\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto 0;Y\mapsto 0\} \\ \mathcal{RE}_{ind}(X_1\approx 0+X_1,<)\{X_1\mapsto 0\}}} 0 \approx 0$
$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\substack{\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto s(X_2);Y\mapsto 0\} \\ \mathcal{RE}_{ind}(X_1\approx 0+X_1,<)\{X_1\mapsto s(X_2)\}}} s(X_2) \approx s(0 + X_2)$

Trivial élimine le premier. **Rewrite**$_2$ peut être appliquée au second car $X_2 < s(X_2)$. On obtient :

$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\substack{\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto s(X_2);Y\mapsto 0\} \\ \mathcal{RE}_{ind}(X_1\approx 0+X_1,<)\{X_1\mapsto s(X_2)\}}} s(X_2) \approx s(X_2 + 0)$

Maintenant, l'application de **Rewrite**$_1$ prouve que 0 est neutre à gauche pour l'addition. Le but $X_1 \approx 0 + X_1$ est ainsi démontré. Il nous reste maintenant à prouver :

$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto X_1;Y\mapsto s(Y_1)\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$

On peut appliquer **Induce** à la position 2, ce qui conduit à :

$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\substack{\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto 0;Y\mapsto s(Y_3)\} \\ \mathcal{RE}_{ind}(s(X_1+Y_1)\approx s(Y_1)+X_1,<)\{X_1\mapsto 0;Y_1\mapsto Y_3\}}} s(0 + Y_3) \approx s(Y_3)$
$\emptyset\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\substack{\mathcal{RE}_{ind}(X+Y\approx Y+X,<)\{X\mapsto s(X_3);Y\mapsto s(Y_3)\} \\ \mathcal{RE}_{ind}(s(X_1+Y_1)\approx s(Y_1)+X_1,<)\{X_1\mapsto s(X_3);Y_1\mapsto Y_3\}}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$

Comme précédemment, le but $s(0 + Y_3) \approx s(Y_3)$ est résolu. En réécrivant à l'aide des règles **Rewrite**, on démontre le but restant, ce qui achève la preuve de la conjecture initiale. Il faut noter que, en vertu de la preuve de correction développée à la section suivante, cet exemple peut se traduire par une dérivation du calcul des séquents modulo.

Observons maintenant ce qui arrive lorsque l'on veut effectuer la preuve du même théorème à l'aide du logiciel Spike. Le logiciel commence à déterminer un ensemble test qui contiendra les éléments 0 et $s(x)$. Il distingue ensuite des variables du but, "les variables de récurrence", qui sont les variables qui sont présentes à l'intérieur d'arguments récursifs de symboles définis apparaissant dans ce but : ce sont ici les deux variables du but x et y. La règle **Expand** va alors instancier chacune de ces variables par un élément de l'ensemble test 0 ou $s(x)$ et réduire l'expression obtenue à l'aide de l'une des règles $x + 0 \to x$ ou $x + s(y) \to s(x + y)$. On obtiendra ainsi les quatre sous-buts :

1. $0 \approx 0 + 0$
2. $s(0 + Y) \approx s(Y) + 0$
3. $s(X) \approx 0 + s(X)$
4. $s(s(X) + Y) \approx s(Y) + s(X)$

Le logiciel parviendra à montrer les trois premiers sous-buts. Observons ce qu'il advient du quatrième. Celui-ci est normalisé en $s(s(X) + Y) \approx s(s(Y) + X)$. Il faut alors appliquer de nouveau la règle **Expand**, et on obtient :

1. $s(s(0)) \approx s(0) + s(0)$
2. $s(s(s(X))) \approx s(0) + s(s(X))$
3. $s(s(0) + s(Y)) \approx s(s(s(Y)) + 0)$
4. $s(s(s(s(X)) + Y)) \approx s(s(Y)) + s(s(X))$

Les trois premiers nouveaux sous-buts seront démontrés. Le quatrième se simplifiera en $s(s(s(s(X)) + Y)) \approx s(s(s(s(Y)) + X)))$. On voit ainsi que le logiciel construit la liste infinie de sous-buts :

$s(s(X) + Y) \approx s(Y) + s(X)$
$s(s(s(s(X)) + Y)) \approx s(s(s(s(Y)) + X))$
$s(s(s(s(s(s(X))) + Y))) \approx s(s(s(s(s(s(Y))) + X)))$
$\vdots$

et il y a divergence, car trop de variables ont été instanciées.

1.3 Correction

Démontrer la correction revient à montrer que, pour chaque règle du système de recherche de preuve IndNarrow de la forme :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \bullet_{i\in I}\ \Gamma_1^i|\Gamma_2^i \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i$$

$\Gamma_1|\Gamma_2, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ est dérivable dans le calcul des séquents modulo, à partir du moment où tous les $\Gamma_1^i|\Gamma_2^i, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2^i \cup \{Q^i\})} \in \mathcal{T}(\Sigma)^{n_i} \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i$ le sont. [1] Introduisons d'abord un ensemble de règles de base dont on va se servir dans la preuve de la correction.

[1] Les égalités ou les règles de $\mathcal{RE}_1$ étant non conditionnelles, leurs variables sont universellement quantifiées.

Théorème 1.1 *Les règles suivantes sont dérivables dans le calcul des séquents modulo :*

$\dfrac{\Gamma \vdash_{\mathcal{RE}} P_1 \Rightarrow P_2, \Delta}{\Gamma, P_1 \vdash_{\mathcal{RE}} P_2, \Delta}\ imp$	
$\dfrac{}{\Gamma, x \in V \vdash_{\mathcal{RE}} x\alpha \approx x\beta}\ r(E)$	*avec E ensemble d'égalités tel que* $\alpha =_E \beta[V]$ *et* $E \subseteq \Gamma$
$\dfrac{}{\Gamma, s \approx t, P\{s/x\} \vdash_{\mathcal{RE}} P\{t/x\}}\ r_s$	
$\dfrac{\Gamma, x \in V \vdash_{\mathcal{RE}} x\alpha \approx x\beta \qquad \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha}{\Gamma \vdash_{\mathcal{RE}\beta} P\beta}\ r_e$	*si* $\mathcal{V}ar(P) \cup \mathcal{V}ar(\mathcal{RE}) \subseteq V$.
$\dfrac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P,}{\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha}\ r_\alpha$	*si :* $\alpha \in Subst^{\Sigma}$, $\mathcal{V}ar(\mathcal{RE} \cup \{P\}) \subseteq \mathcal{D}om(\alpha)$, *et* $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE} \cup \{P\})}$.
$\dfrac{\bigwedge_{\alpha \in Subst^{\Sigma} \wedge \mathcal{V}ar(\mathcal{RE} \cup \{P\}) \subseteq \mathcal{D}om(\alpha)} \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P}\ r_{\overrightarrow{x}}$	*si* $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE} \cup \{P\})}$
$\dfrac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE} \cup \mathcal{RE}_{ind}(P,<)} P \quad \Gamma \vdash_{\mathcal{RE}} Noeth(<, \mathcal{T}(\Sigma))}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P}\ r_I$	*si* $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE} \cup \{P\})}$

Preuve. Les preuves sont faites en annexe.

- imp : lemme A.7 ;
- $r(E)$: lemme A.11 ;
- r_s : lemme A.10 ;
- r_e : corollaire A.2 ;
- r_α : lemme A.18 ;
- $r_{\overrightarrow{x}}$: lemme A.19 ;
- r_I : lemme A.22.

□

On est maintenant en mesure de prouver la correction de IndNarrow dans le calcul des séquents modulo en considérant à tour de rôle chaque règle d'inférence de IndNarrow.

Théorème 1.2 *Si :*

1. **Induce** *est appliquée sur*
   $\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q[t]_\omega$
   pour obtenir :
   - $_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma = mgu(t,l)}}\ \Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2 \sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} (Q[r]_\omega)\sigma$ *(rappel :* $\omega \in \mathcal{DM}(Q)$ *) ;*

2. *$<$ est noethérien sur $\mathcal{T}(\Sigma)$;*
3. *chaque séquent* $\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} (Q[r]_\omega)\sigma$ *est dérivable dans le calcul des séquents modulo, avec $\overrightarrow{x}_\sigma \in \mathcal{X}^{n_\sigma}$ le vecteur des variables libres de $\mathcal{RE}_2\sigma \cup \{Q\sigma\}$;*

alors, si $\overrightarrow{x} \in \mathcal{X}^n$ désigne le vecteur des variables libres de $\mathcal{RE}_2 \cup \{Q\}$, le séquent $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q[t]_\omega$ est dérivable dans le calcul des séquents modulo.

Preuve. Premièrement, introduisons les notations suivantes :

$$\begin{array}{lcl} \Gamma & = & \Gamma_1 \cup \Gamma_2; \\ \mathcal{RE} & = & \mathcal{RE}_1 \cup \mathcal{RE}_2; \\ \mathcal{RE}' & = & \mathcal{RE} \cup \{\mathcal{RE}_{ind}(Q,<)\}; \\ \mathcal{RE}\sigma & = & \mathcal{RE}_1 \cup \mathcal{RE}_2\sigma; \\ \mathcal{RE}'\sigma & = & \mathcal{RE}\sigma \cup \{\mathcal{RE}_{ind}(Q,<)\}\sigma. \end{array} \tag{1.1}$$

Soit V un ensemble fini de variables et α une substitution close telles que $\mathcal{V}ar(\mathcal{RE} \cup \{Q\}) \subseteq \mathcal{D}om(\alpha) \subseteq V$. Notons $\alpha \downarrow$ l'une de ses $\mathcal{RE}_1$-formes normales. On sait, d'après le lemme 2.2 de la partie II, qu'il existe une règle $l \to r \in \mathcal{RE}_1$ et une substitution close μ telles que, si $\sigma = mgu(t, l)$, alors :

$$\sigma\mu = (\alpha \downarrow)[V] \tag{1.2}$$

Considérons les dérivations suivantes :

Π_1

$$\dfrac{\dfrac{}{x\alpha \downarrow\approx x\sigma\mu \vdash_{\mathcal{RE}_1} x\alpha \approx x\sigma\mu}\ Ax}{\Gamma, x \in V, x\alpha \downarrow\approx x\sigma\mu \vdash_{\mathcal{RE}} x\alpha \approx x\sigma\mu}\ w + push$$

Π_2

$$\dfrac{\dfrac{}{\Gamma, x \in V \vdash x\alpha \downarrow\approx x\sigma\mu}\ r(\emptyset)\ \text{(d'après (1.2))}}{\Gamma, x \in V \vdash_{\mathcal{RE}} x\alpha \downarrow\approx x\sigma\mu, x\alpha \approx x\sigma\mu}\ w + push$$

Π_3

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma, x \in V \vdash_{\mathcal{RE}} x\alpha \approx x\sigma\mu}\ cut$$

$\Pi_{1,\sigma}$

$$\dfrac{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}'\sigma} (Q[r]_\omega)\sigma\ \text{(supposé)}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}'\sigma} Q\sigma, (Q[r]_\omega)\sigma}\ w$$

$\Pi_{2,\sigma}$

$$\dfrac{\dfrac{}{(Q[r]_\omega)\sigma \vdash_{\mathcal{RE}_1} (Q[l]_{|\omega})\sigma}\ Ax}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma}, (Q[r]_\omega)\sigma \vdash_{\mathcal{RE}'\sigma} (Q[l]_\omega)\sigma}\ w$$

Or $\sigma = mgu(Q_{|\omega}, l)$, donc $(Q[l]_{|\omega})\sigma = Q\sigma$, et $\Pi_{2,\sigma}$ peut donc encore s'écrire :

$\Pi_{2,\sigma}$

$$\dfrac{\dfrac{}{(Q[r]_\omega)\sigma \vdash_{\mathcal{RE}_1} Q\sigma}\ Ax}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma}, (Q[r]_\omega)\sigma \vdash_{\mathcal{RE}'\sigma} Q\sigma}\ w$$

$\Pi_{3,\sigma}$

$$\dfrac{\Pi_{1,\sigma} \qquad \Pi_{2,\sigma}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}'\sigma} Q\sigma}\ cut$$

Si on note $\mathcal{PE}_{ind}(Q,<)$ la proposition canonique associée à $\mathcal{RE}_{ind}(Q,<)$, cela nous conduit vers :

$$\dfrac{\dfrac{\Pi_{3,\sigma}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma}, \mathcal{PE}_{ind}(Q,<)\sigma \vdash_{\mathcal{RE}\sigma} Q\sigma}\ pop}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma} \mathcal{PE}_{ind}(Q,<)\sigma \Rightarrow Q\sigma}\ \Rightarrow -r$$

d'où :

$$\dfrac{\Pi_{4,\sigma}}{\Gamma \vdash_{\mathcal{RE}\sigma\mu} (\mathcal{PE}_{ind}(Q,<) \Rightarrow Q)\sigma\mu}\ r_\mu$$

Or, $\mathcal{V}ar(\mathcal{PE}_{ind}(Q,<) \Rightarrow Q) = \mathcal{V}ar(Q)$, donc $(\mathcal{V}ar(\mathcal{PE}_{ind}(Q,<) \Rightarrow Q) \cup \mathcal{V}ar(\mathcal{RE})) \subseteq V$. et :

$$\dfrac{\Pi_{\sigma,\mu} \quad \Pi_3}{\Gamma \vdash_{\mathcal{RE}\alpha} (\mathcal{PE}_{ind}(Q,<) \Rightarrow Q)\alpha}\ r_e$$

Et, puisque α représente toute substitution close choisie telle que $\mathcal{V}ar(\mathcal{RE} \cup \{Q\}) \subseteq \mathcal{D}om(\alpha)$ et $\overrightarrow{x} = \mathcal{V}ar(\mathcal{RE} \cup \{Q\})$, on peut écrire :

$$\dfrac{\dfrac{\dfrac{\bigwedge_{\alpha \in Subst^\Sigma \wedge \mathcal{V}ar(\mathcal{RE}\cup\{Q\}) \subseteq \mathcal{D}om(\alpha)} \Pi_\alpha}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} \mathcal{PE}_{ind}(Q,<) \Rightarrow Q}\ r_{\overrightarrow{x}}}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, \mathcal{PE}_{ind}(Q,<) \vdash_{\mathcal{RE}} Q}\ imp}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}\cup\mathcal{RE}_{ind}(Q,<)} Q}\ push$$

$$\dfrac{\Pi_{\overrightarrow{x}} \qquad \Gamma \vdash_{\mathcal{RE}} Noeth(<, \mathcal{T}(\Sigma)) \quad \text{(hypothèse 2)}}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} Q}\ r_I$$

ce qui achève la preuve □

Considérons maintenant les inférences **Rewrite**.

Théorème 1.3 *Si :*

1. *une règle* **Rewrite** *est appliquée sur* $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ *pour obtenir* $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q'$ *;*
2. *le séquent* $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{X}^n \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2} Q'$ *admet une preuve dans le calcul des séquents modulo, avec* $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q'\})}$ *;*

alors, le séquent $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{X}^n \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2} Q$ *est dérivable dans le calcul des séquents modulo.* [2]

Preuve. On utilise les mêmes notations que dans le théorème précédent. Par l'hypothèse 1, et par définition de la règle **Rewrite**, il existe une règle de réécriture $\kappa = l \to r$, une égalité $\kappa = l \approx r$, ou une égalité conditionnelle $\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$, une position $\omega \in \mathcal{D}om(Q)$, et une substitution σ telles que :

1. $Q = Q[l\sigma]_{|\omega}$;

[2] Une application d'une règle **Rewrite** ne change pas l'ensemble des variables libres du séquent.

2. $Q' = Q[r\sigma]_{|\omega}$;

3. si $\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$, et $\overrightarrow{y} \in \mathcal{X}^m$ est défini par $\overrightarrow{y} = \overrightarrow{\mathcal{V}ar(l \approx r)}$, alors $\overrightarrow{y}\sigma <_m \overrightarrow{y}\mu$ (car c'est **Rewrite**$_2$ qui est appliquée).

Nous détaillons le cas le plus complexe correspondant à $\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$. Considérons alors les dérivations suivantes :

Π_1 :

$$\frac{\Gamma \vdash_{\mathcal{RE}} \overrightarrow{y}\sigma <_m \overrightarrow{y}\mu \ \ \text{(d'après 3)}}{\Gamma, Q[r\sigma] \vdash_{\mathcal{RE}} \overrightarrow{y}\sigma <_m \overrightarrow{y}\mu, Q[l\sigma]}\ w$$

Π_2 :

$$\frac{\dfrac{}{\Gamma, Q[r\sigma], l\sigma \approx r\sigma \vdash Q[l\sigma]}\ r_s}{\Gamma, Q[r\sigma], l\sigma \approx r\sigma \vdash_{\mathcal{RE}} Q[l\sigma]}\ w + push$$

Π_3 :

$$\frac{\Pi_1 \quad \Pi_2}{\Gamma, Q[r\sigma], \overrightarrow{y}\sigma <_m \overrightarrow{y}\mu \Rightarrow l\sigma \approx r\sigma \vdash_{\mathcal{RE}} Q[l\sigma]}\ \Rightarrow_l$$

Π_4 :

$$\frac{\dfrac{\Pi_3}{\Gamma, Q[r\sigma], \forall \underline{\overrightarrow{y}}\ \ \underline{\overrightarrow{y}} <_m \overrightarrow{y}\mu \Rightarrow (l \approx r)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\} \vdash_{\mathcal{RE}} Q[l\sigma]}\ \forall_l}{\Gamma, Q[r\sigma] \vdash_{\mathcal{RE}\cup\{\kappa\}} Q[l\sigma]}\ push$$

Mais comme $\kappa \in \mathcal{RE}$, on peut écrire :

Π_4 :

$$\frac{\dfrac{\Pi_3}{\Gamma, Q[r\sigma], \forall \underline{\overrightarrow{y}}\ \ \underline{\overrightarrow{y}} <_m \overrightarrow{y}\mu \Rightarrow (l \approx r)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\} \vdash_{\mathcal{RE}} Q[l\sigma]}\ \forall_l}{\Gamma, Q[r\sigma] \vdash_{\mathcal{RE}} Q[l\sigma]}\ push$$

Π_5 :

$$\frac{\Pi_4}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, Q[r\sigma] \vdash_{\mathcal{RE}} Q[l\sigma]}\ w$$

Π_6 :

$$\frac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} Q[r\sigma] \ \text{(supposé)}}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} Q[r\sigma], Q[l\sigma]}\ w$$

Π :

$$\frac{\Pi_5 \quad \Pi_6}{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} Q[l\sigma]}\ cut$$

Et le résultat est démontré. □

Théorème 1.4 *Le système de recherche de preuve* IndNarrow *est correct.*

Preuve. La correction des règles **Induce** et **Rewrite** a été démontrée ci-dessus (théorèmes 1.2 et 1.3). La correction de **Push** est une simple conséquence de la correction du calcul des séquents modulo, et la règle **Orient** n'est qu'une transformation d'écriture. □

Le corollaire suivant signifie que la consistance du but avec toute instance close des hypothèses est préservée lorsqu'on effectue une preuve avec notre système.

Corollaire 1.1 pour chaque règle du système de recherche de preuve IndNarrow de la forme :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \bullet_{i\in I}\ \Gamma_1^i|\Gamma_2^i \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i$$

il existe une substitution close α de $\mathcal{RE}_2$ telle que $\Gamma_1 \cup \Gamma_2, Q \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha} \emptyset$ soit dérivable dans le calcul des séquents modulo, à partir du moment où il existe un i et une substitution close α^i de $\mathcal{RE}_2^i$ tels que $\Gamma_1^i \cup \Gamma_2^i, Q^i \vdash_{\mathcal{RE}_1^i\cup\mathcal{RE}_2^i\alpha^i} \emptyset$ soit dérivable.

Preuve. Nous détaillons uniquement le cas de la règle **Induce**.

Supposons que **Induce** soit appliquée sur $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[t]_\omega$ pour obtenir :

$$\bullet_{\substack{l\to r\in\mathcal{RE}_1\\ \sigma=mgu(t,l)}}\ \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\sigma\cup\{\mathcal{RE}_{ind}(Q,<)\sigma\}} (Q[r]_\omega)\sigma \quad (\text{rappel : } \omega\in\mathcal{DM}(Q))$$

Par hypothèse, il existe une règle $l \to r \in \mathcal{RE}_1$, $\sigma = mgu(t, l)$, et une substitution close μ de $\mathcal{RE}_2\sigma \cup \{Q\sigma\}$, telles que le séquent S suivant soit démontrable :

$$S:\ \Gamma_1 \cup \Gamma_2, (Q[r]_\omega)\sigma \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\sigma\mu\cup\{\mathcal{RE}_{ind}(Q,<)\sigma\mu\}} \emptyset$$

On en déduit

Π_1 :

$$\dfrac{\dfrac{S}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\sigma\mu\cup\{\mathcal{RE}_{ind}(Q,<)\sigma\mu,(Q[r]_\omega)\sigma\}} \emptyset}\ push}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\sigma\mu\cup\{\mathcal{RE}_{ind}(Q,<)\sigma\mu,(Q[r]_\omega)\sigma\mu\}} \emptyset}\ r_\mu$$

Posons $\alpha = \sigma\mu$, et soit V un ensemble fini de variables tel que $\mathcal{D}om(\alpha) \subseteq V$

Π_2 :

$$\dfrac{}{\Gamma_1 \cup \Gamma_2, x \in V \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q,<),Q[r]_\omega\}} x\alpha \approx x\sigma\mu}\ r(\emptyset)$$

Π_3 :

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha\cup\{\mathcal{RE}_{ind}(Q,<)\alpha,(Q[r]_\omega)\alpha\}} \emptyset}\ r_e$$

(car $\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\}) \subseteq \mathcal{D}om(\alpha)$, donc $\mathcal{V}ar(\mathcal{RE}_2) \cup \mathcal{V}ar(\mathcal{RE}_{ind}(Q, <)) \cup \mathcal{V}ar(Q[r]_\omega) \subseteq V$).

Π_4 :

$$\dfrac{\dfrac{\Pi_3}{\Gamma_1 \cup \Gamma_2, (Q[r]_\omega)\alpha \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha\cup\{\mathcal{RE}_{ind}(Q,<)\alpha\}} \emptyset}\ pop}{\Gamma_1 \cup \Gamma_2, Q\alpha, (Q[r]_\omega)\alpha \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha\cup\{\mathcal{RE}_{ind}(Q,<)\alpha\}} \emptyset}\ w$$

Π_5 :

$$\dfrac{Q\alpha \vdash_{\mathcal{RE}_1} (Q[r]_\omega)\alpha\ (\text{car } t\alpha = t\sigma\mu = l\sigma\mu = l\alpha.)}{\Gamma_1 \cup \Gamma_2, Q\alpha \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha\cup\{\mathcal{RE}_{ind}(Q,<)\alpha\}} (Q[r]_\omega)\alpha}\ w$$

Π_6 :

$$\dfrac{\Pi_4 \quad \Pi_5}{\Gamma_1 \cup \Gamma_2, Q\alpha \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha\cup\{\mathcal{RE}_{ind}(Q,<)\alpha\}}}\ cut$$

Π_7 :

$$\dfrac{\dfrac{\Pi_6}{\Gamma_1 \cup \Gamma_2, Q\alpha, \mathcal{PE}_{ind}(Q, <)\alpha \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha} \emptyset}\ pop}{\Gamma_1 \cup \Gamma_2, Q, Q\alpha, \mathcal{PE}_{ind}(Q, <)\alpha \vdash_{\mathcal{RE}_1\cup\mathcal{RE}_2\alpha} \emptyset}\ w$$

Π_8 :

$$\dfrac{\dfrac{\overline{Q \vdash Q}\ Ax}{Q \vdash \forall \overrightarrow{x} Q}\ \forall - r}{Q \vdash \forall \overrightarrow{x} Q, Q\alpha}\ w$$

Π_9 :

$$\dfrac{\dfrac{Q\alpha \vdash Q\alpha}{\forall \overrightarrow{x} Q \vdash Q\alpha}\ \forall_l}{Q, \forall \overrightarrow{x} Q \vdash Q\alpha}\ w$$

Π_{10} :

$$\dfrac{\Pi_8 \quad \Pi_9}{Q \vdash Q\alpha}\ cut$$

Π_{11} :

$$\dfrac{\Pi_{10}}{\Gamma_1 \cup \Gamma_2, Q, \mathcal{PE}_{ind}(Q, <)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha} Q\alpha}\ w$$

Π_{12} :

$$\dfrac{\Pi_7 \quad \Pi_{11}}{\Gamma_1 \cup \Gamma_2, Q, \mathcal{PE}_{ind}(Q, <)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha} \emptyset}\ cut$$

Π_{13} :

$$\dfrac{\dfrac{\dfrac{\dfrac{Q \vdash Q}{Q, \underline{\overrightarrow{x}} <_n \overrightarrow{\mathcal{V}ar(Q)}\alpha \vdash Q}\ w}{Q \vdash \underline{\overrightarrow{x}} <_n \overrightarrow{\mathcal{V}ar(Q)}\alpha \Rightarrow Q}\ \Rightarrow_r}{Q \vdash \mathcal{PE}_{ind}(Q, <)\alpha}\ \forall_r}{\Gamma_1 \cup \Gamma_2, Q \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha} \mathcal{PE}_{ind}(Q, <)\alpha}\ w$$

Π

$$\dfrac{\Pi_9 \quad \Pi_{10}}{\Gamma_1 \cup \Gamma_2, Q, \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha} \emptyset}\ cut$$

□

1.4 Correction réfutationnelle

La correction réfutationnelle revient à montrer que, pour chaque règle du système de recherche de preuve IndNarrow de la forme :

$$\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q \rightarrowtail \bullet_{i \in I}\ \Gamma_1^i | \Gamma_2^i \vdash_{\mathcal{RE}_1^i | \mathcal{RE}_2^i} Q^i$$

tous les $\Gamma_1^i \cup \Gamma_2^i, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2^i \cup \{Q^i\})} \in \mathcal{T}(\Sigma)^{n_i} \vdash_{\mathcal{RE}_1^i | \mathcal{RE}_2^i} Q^i$ sont dérivables dès lors que $\Gamma_1 \cup \Gamma_2, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ l'est. Nous détaillons ici le point le plus délicat, qui est bien-sûr le cas de la règle **Induce**, dans la preuve du théorème suivant.

Théorème 1.5 *Si :*

1. $\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q[t]_\omega \rightarrowtail_{\mathbf{Induce}} \bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma = mgu(t,l)}} \Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2 \sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} Q[r]_\omega \sigma$;
2. *le séquent* $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q[t]_\omega$ *admet une preuve dans le calcul des séquents modulo, avec* $\overrightarrow{x} \in \mathcal{X}^n$ *défini par* $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})}$ *;*

alors, chaque séquent :

$$\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} Q[r]_\omega\sigma$$

est également démontrable dans le calcul des séquents modulo, en posant $\overrightarrow{x}_\sigma \in \mathcal{X}^{n_\sigma}$ *le vecteur des variables libres de* $\mathcal{RE}_2\sigma \cup \{Q\sigma\}$.

Preuve. On reprend les notations (1.1). Soit $\sigma = mgu(t, l)$. Pour toute substitution close μ telle que $\mathcal{V}ar(\mathcal{RE} \cup \{Q\}) \subset \mathcal{D}om(\mu)$, on a :

$\Pi_{\sigma,\mu}$

$$\frac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} Q}{\Gamma \vdash_{\mathcal{RE}\sigma\mu} Q\sigma\mu}\ r_{\sigma\mu}$$

Considérons maintenant les dérivations suivantes :

$\Pi_{1,\sigma}$:

$$\frac{\bigwedge_{\mu \in Subst^\Sigma \wedge \mathcal{V}ar(\mathcal{RE}\sigma \cup \{Q\sigma\}) \subseteq \mathcal{D}om(\mu)} \Pi_{\sigma,\mu}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma} Q\sigma}\ r_{\overrightarrow{x}}$$

$\Pi_{2,\sigma}$:

$$\frac{\Pi_{1,\sigma}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma} Q\sigma, (Q[r]_{|\omega})\sigma}\ w$$

Si on note $Th_{\mathcal{RE}_2\sigma}$ la théorie canonique associée à $\mathcal{RE}_2\sigma$, puisque $\mathcal{RE}\sigma = \mathcal{RE}_1 \cup \mathcal{RE}_2\sigma$, et comme $Q\sigma = (Q[l]_\omega)\sigma$ on peut écrire :

$\Pi_{3,\sigma}$:

$$\frac{\dfrac{\dfrac{}{Q\sigma \vdash_{\mathcal{RE}_1} (Q[r]_\omega)\sigma}\ Ax}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma}, Q\sigma, Th_{\mathcal{RE}_2\sigma} \vdash_{\mathcal{RE}_1} (Q[r]_\omega)\sigma}\ w}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma}, Q\sigma \vdash_{\mathcal{RE}\sigma} (Q[r]_\omega)\sigma}\ push$$

Si on note $\mathcal{PE}_{ind}(Q)$ la proposition canonique associée à $\mathcal{RE}_{ind}(Q, <)$, cela nous conduit à :

$\Pi_{4,\sigma}$:

$$\frac{\dfrac{\dfrac{\Pi_{2,\sigma} \quad \Pi_{3,\sigma}}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma} (Q[r]_\omega)\sigma}\ cut}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma}, \mathcal{PE}_{ind}(Q)\sigma \vdash_{\mathcal{RE}\sigma} ((Q[r]_\omega)\sigma)}\ w}{\Gamma, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}\sigma \cup \{\mathcal{RE}_{ind}(Q)\sigma\}} ((Q[r]_\omega)\sigma)}\ push$$

ce qui achève la preuve. □

Comme corollaire du théorème 1.5, on a :

Théorème 1.6 *Le système de recherche de preuve* IndNarrow *est réfutationnellement correct.*

Preuve. La règle **Induce** a déjà été traitée dans la preuve du théorème 1.5, et la preuve de la correction réfutationnelle des règles **Rewrite** est analogue à celle de leur correction. La correction réfutationnelle des règles **Push** est une conséquence de la correction réfutationnelle de la déduction modulo, et la règle **Orient** n'est qu'une transformation d'écriture. □

1.5 Complétude réfutationnelle

La complétude réfutationnelle est réalisée grâce à la règle **Refutation**, qui s'applique lorsqu'aucune autre règle de IndNarrow ne peut s'appliquer. On suppose également que Γ_2 contient une définition de l'égalité syntaxique $=$, une définition de l'ensemble des termes constructeurs $\mathcal{T}(\mathcal{C},\mathcal{X})$, et que $\approx$ et $=$ coïncident sur l'ensemble des termes constructeurs. Plus précisément, pour tout $f, g \in \Sigma$, Γ_2 contient les propositions ci-dessous (inspirées de [HUE 82]) :

$A(f)$	$fx_1 \ldots x_n = fy_1 \ldots y_n \Leftrightarrow x_1 = y_1 \wedge \ldots \wedge x_n = y_n$	avec f d'arité n
$B(f,g)$	$fx_1 \ldots x_n \neq gy_1 \ldots y_m$	avec f, g d'arités n et m tels que $f \neq g$
$B(f,x)$	$fx_1 \ldots x_n \neq x$	avec f d'arité n
$C(f)$	$fx_1 \ldots x_n \in \mathcal{T}(\mathcal{C},\mathcal{X}) \Leftrightarrow x_1 \in \mathcal{T}(\mathcal{C},\mathcal{X}) \wedge \ldots \wedge x_n \in \mathcal{T}(\mathcal{C},\mathcal{X})$	avec $f \in \mathcal{C} \cup \mathcal{X}$
$D(f)$	$fx_1 \ldots x_n \notin \mathcal{T}(\mathcal{C},\mathcal{X})$	avec $f \notin \mathcal{C} \cup \mathcal{X}$
$S(\approx)$	$(x,y) \in \mathcal{T}(\mathcal{C})^2 \Rightarrow (x \approx y \Rightarrow x = y)$	

On définira Eq et $Cons$ de la façon suivante :

$$\begin{array}{l} Eq = \{A(f), B(f,g), B(f,x) \mid f, g \in \Sigma, x \in \mathcal{X}\} \\ Cons = \{C(f), D(f) \mid f \in \Sigma\} \end{array}$$

Théorème 1.7 *Si :*

$$\Gamma_1{}^0|\Gamma_2{}^0 \vdash_{\mathcal{RE}_1{}^0|\mathcal{RE}_2{}^0} Q^0 \overset{*}{\rightarrowtail} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \textbf{Refutation}$$

alors, en posant $\overrightarrow{x_0} \in \mathcal{X}^{n_0}$ *le vecteur défini par* $\overrightarrow{x_0} = \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2{}^0 \cup \{Q^0\})}$, *et* $\overrightarrow{x} \in \mathcal{X}^n$ *le vecteur défini par* $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})}$ *:*

- *soit le séquent* $\Gamma_1{}^0 \cup \Gamma_2{}^0, \overrightarrow{x_0} \in \mathcal{T}(\Sigma)^{n_0} \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0} Q^0$ *n'admet pas de preuve dans le calcul des séquents modulo ;*
- *soit il existe une substitution close* α^0 *de* $\mathcal{RE}_2^0$ *telle que :*

$$\Gamma_1{}^0 \cup \Gamma_2{}^0, Q^0 \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0 \alpha^0} \emptyset$$

(i.e l'ensemble $\Gamma_1^0 \cup \Gamma_2^0 \cup \mathcal{RE}_1^0 \cup \mathcal{RE}_2^0\alpha^0 \cup \{Q^0\}$ *est inconsistant).*

Preuve. Si Q contenait un symbole défini, il existerait une position *définie-maximale* dans $Dom(Q)$, on pourrait alors appliquer la règle **Induce**, ce qui serait contradictoire. Puisque la règle **Trivial** ne peut pas non plus être appliquée, on a $Q = s \approx t$, avec $s,\ t$ des termes constructeurs qui ne sont pas syntaxiquement égaux (1). Considérons alors les dérivations suivantes :

Π_1 :

$$\dfrac{\dfrac{\dfrac{}{(s,t) \in \mathcal{T}(\mathcal{C})^2 \Rightarrow (s \approx t \Rightarrow s = t) \vdash (s,t) \in \mathcal{T}(\mathcal{C})^2 \Rightarrow (s \approx t \Rightarrow s = t)}\ Ax}{S(\approx) \vdash (s,t) \in \mathcal{T}(\mathcal{C})^2 \Rightarrow (s \approx t \Rightarrow s = t)}\ \forall_l}{S(\approx), (s,t) \in \mathcal{T}(\mathcal{C})^2 \vdash s \approx t \Rightarrow s = t}\ imp$$

Sachant que $s, t \in \mathcal{T}(\mathcal{C},\mathcal{X})$, il existe une dérivation du type suivant :

$$\dfrac{\vdots}{Cons \vdash (s,t) \in \mathcal{T}(\mathcal{C})^2}$$

Π_2 :

$$\dfrac{\dfrac{\vdots}{Cons \vdash (s,t) \in \mathcal{T}(\mathcal{C})^2}}{S(\approx) \vdash (s,t) \in \mathcal{T}(\mathcal{C})^2, s \approx t \Rightarrow s = t}\ w$$

Π_3 :

$$\dfrac{\Pi_1 \quad \Pi_2}{S(\approx) \vdash s \approx t \Rightarrow s = t}\ cut$$

Π_4 :

$$\dfrac{\dfrac{\Pi_3}{S(\approx), s \approx t \vdash s = t}\ imp}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n s \approx t \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} s = t}\ w$$

Supposons que le séquent ${\Gamma_1}^0 \cup {\Gamma_2}^0, \overrightarrow{x_0} \in \mathcal{T}(\Sigma)^{n_0} \vdash_{{\mathcal{RE}_1}^0 \cup {\mathcal{RE}_2}^0} Q^0$ admette une preuve, alors, par correction réfutationnelle de IndNarrow (théorème 1.6) , le séquent $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ serait également démontrable, et on pourrait alors écrire :
Π_5 :

$$\dfrac{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} s \approx t, s = t}\ w$$

Π_6 :

$$\dfrac{\Pi_4 \quad \Pi_5}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} s = t}\ cut$$

Sachant que $s \neq t$, il existe une dérivation du type suivant :

$$\dfrac{\vdots}{Eq \vdash \neg(s = t)}$$

On peut ainsi écrire :
Π_7 :

$$\dfrac{\dfrac{\dfrac{\vdots}{Eq \vdash \neg(s = t)}}{Eq, s = t \vdash}\ \neg_l}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, s = t \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} \emptyset}\ w$$

Π_8 :

$$\dfrac{\Pi_6 \quad \Pi_7}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} \emptyset}\ cut$$

Posons α une substitution close de $\mathcal{RE}_2$, on a :

$$\dfrac{\Pi_8}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha} \emptyset}\ r_\alpha$$

Π :

$$\dfrac{\dfrac{\Pi_8}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha}}\ r_\alpha}{\Gamma_1 \cup \Gamma_2, Q\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha} \emptyset}\ w$$

Or, d'après le corollaire 1.1, cela entraîne l'existence d'une substitution close α^0 de $\mathcal{RE}_2^0$ telle que :

$$\Gamma_1{}^0 \cup \Gamma_2{}^0, Q^0 \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0 \alpha^0} \emptyset$$

□

Définition 1.1 Une dérivation

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \ldots \rightarrowtail \bullet_{i\in I}\ \Gamma_1^i|\Gamma_2^i \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i \rightarrowtail \ldots$$

est dite : *dérivation strictement décroissante* si chaque application d'une règle d'inférence **Rewrite** s'effectue à l'aide d'une règle orientée.

Théorème 1.8 *On reprend les hypothèses du théorème 1.7. Si* $\Gamma_1{}^0 \cup \Gamma_2{}^0, \overrightarrow{x_0} \in \mathcal{T}(\Sigma)^{n_0} \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0} Q^0$ *n'a pas de preuve en calcul des séquents modulo, alors toute dérivation strictement décroissante issue de* $\Gamma_1{}^0|\Gamma_2{}^0 \vdash_{\mathcal{RE}_1{}^0|\mathcal{RE}_2{}^0} Q^0$ *se termine par l'application de la règle* **Refutation**.

Preuve. Supposons que $\Gamma_1{}^0 \cup \Gamma_2{}^0, \overrightarrow{x_0} \in \mathcal{T}(\Sigma)^{n_0} \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0} Q^0$ n'ait pas de preuve en calcul des séquents modulo, et considérons une dérivation strictement décroissante :

$$\Gamma_1{}^0|\Gamma_2{}^0 \vdash_{\mathcal{RE}_1{}^0|\mathcal{RE}_2{}^0} Q^0 \rightarrowtail \ldots \rightarrowtail \bullet_{i\in I}\ \Gamma_1^i|\Gamma_2^i \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i \rightarrowtail \ldots$$

Supposons maintenant que $\Gamma_1{}^0 \cup \Gamma_2{}^0 \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0 \alpha \cup \mathcal{RE}_{ind}(Q^0,<)\alpha} Q^0\alpha$ ait une preuve en calcul des séquents modulo pour toute substitution close α telle que $\mathcal{V}ar(\mathcal{RE}_2{}^0 \cup \{Q^0\}) \subseteq \mathcal{D}om(\alpha)$. En utilisant les règles $r_{\overrightarrow{x}}$ et r_I définies au théorème 1.1 de la section 1.3, on pourrait alors écrire :
Π_1 :

$$\frac{\bigwedge_{\alpha\in Subst^{\Sigma}\wedge \mathcal{V}ar(\mathcal{RE}_2{}^0\cup\{Q^0\})\subseteq\mathcal{D}om(\alpha)} \Gamma_1{}^0 \cup \Gamma_2{}^0 \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0 \alpha \cup \{\mathcal{RE}_{ind}(Q^0,<)\alpha\}} Q^0\alpha}{\Gamma_1{}^0 \cup \Gamma_2{}^0, \overrightarrow{x_0} \in \mathcal{T}(\Sigma)^{n_0} \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0 \cup \{\mathcal{RE}_{ind}(Q^0,<)\}} Q^0}\ r_{\overrightarrow{x}}$$

Π :

$$\frac{\Pi_1 \quad \Gamma_1{}^0 \cup \Gamma_2{}^0 \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0} Noeth(<, \mathcal{T}(\Sigma))}{\Gamma_1{}^0 \cup \Gamma_2{}^0, \overrightarrow{x_0} \in \mathcal{T}(\Sigma)^{n_0} \vdash_{\mathcal{RE}_1{}^0 \cup \mathcal{RE}_2{}^0} Q^0}\ r_I$$

Mais cela contredit l'hypothèse du théorème. L'ensemble :

$\mathcal{E} = \{Q^i\alpha \mid \alpha$ substitution close, $\Gamma_1^i|\Gamma_2^i \vdash_{\mathcal{RE}_1^i|\mathcal{RE}_2^i} Q^i$ sous but et $\Gamma_1^i \cup \Gamma_2^i \vdash_{\mathcal{RE}_1^i \cup \mathcal{RE}_2^i \alpha \cup \{\mathcal{RE}_{ind}(Q^i,<)\alpha\}} Q^i\alpha$ n'a pas de preuve en calcul des séquents modulo$\}$

est alors non vide et, comme $<_2$ est noethérien, il existe un élément minimum pour $<_2$ dans cet ensemble. Posons alors $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ un sous-but, tel que $\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha \cup \{\mathcal{RE}_{ind}(Q,<)\alpha\}} Q\alpha$ n'a pas de preuve en calcul des séquents modulo, et $Q\alpha$ minimum pour $<_2$ dans $\mathcal{E}$. Maintenant, si on suppose que chaque sous-chemin de la dérivation issu de ce sous-but est de longueur finie, celui-ci se termine nécessairement par $\diamond$ ou par **Refutation**. Si ils se terminent tous par $\diamond$, le séquent $\Gamma_1 \cup \Gamma_2,\ \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ admet une preuve en déduction modulo, par correction

de IndNarrow (théorème 1.4 de la section 1.3). On peut alors écrire :
Π :

$$\dfrac{\dfrac{\Gamma_1 \cup \Gamma_2,\ \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q}{\Gamma_1 \cup \Gamma_2,\ \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \cup \mathcal{RE}_{ind}(Q,<)} Q}\ w + push}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \alpha \cup \{\mathcal{RE}_{ind}(Q,<)\alpha\}} Q\alpha}\ r_\alpha$$

et cela entraîne une contradiction.
Sinon, il existe un chemin qui se termine par **Refutation** et le théorème est démontré.

Supposons maintenant que le sous-but $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$ soit situé sur un chemin de longueur infinie. Comme les règles **Push** et **Rewrite** ne peuvent se succéder indéfiniment, on peut supposer qu'une règle d'inférence **Rewrite** ou **Induce** est appliquée au sous-but $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q$.

– Supposons que **Induce** soit appliquée à la position ω dans Q.
  Alors, sachant que $\mathcal{RE}_1$ est suffisamment complet sur les termes clos, et que ω est *définie-maximale* dans Q, le lemme 2.1 de la partie II permet d'en déduire l'existence d'un diagramme commutatif du type :

$$\begin{array}{lll} Q\alpha & \rightarrow_{[l \rightarrow r, \omega, \nu]} & Q\alpha[r\nu]_\omega \\ \uparrow \alpha & & \uparrow \mu \\ Q & \rightsquigarrow_{[l \rightarrow r, \omega, \sigma]} & (Q[r]_\omega)\sigma \end{array}$$

  Et, par définition de **Induce**, on a :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \quad \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} (Q[r]_\omega)\sigma$$

  Or, d'après le lemme 2.1 de la partie II, on peut supposer $\alpha = \sigma\mu[V]$, avec V quelconque tel que $\mathcal{V}ar(Q) \cup \mathcal{D}om(\alpha) \subseteq V$, donc en particulier tel que $\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\}) \subseteq V$. Observons maintenant que $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} (Q[r]_\omega)\sigma$ est un sous-but, et $(Q[r]_\omega)\sigma\mu <_2 Q\alpha$ (d'après le diagramme précédent), donc $(Q[r]_\omega)\sigma\mu \in \mathcal{E}$, et ceci entraîne que le séquent :

$$\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma\mu \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\mu, \mathcal{RE}_{ind}((Q[r]_\omega)\sigma,<)\mu\}} (Q[r]_\omega)\sigma\mu \tag{1.3}$$

  a une preuve dans le calcul des séquents modulo, car $Q\alpha$ est minimal dans $\mathcal{E}$.
  Posons $\overrightarrow{y} \in \overrightarrow{X}^m$ défini par $\overrightarrow{y} = \overrightarrow{\mathcal{V}ar(Q)}$, $\overrightarrow{z} \in \mathcal{X}^p$ défini par $\overrightarrow{z} = \overrightarrow{\mathcal{V}ar(Q\sigma)}$, et τ une substitution close de $\overrightarrow{z}$. Comme $\overrightarrow{z}$ est le vecteur des variables de $\overrightarrow{y}\sigma$, $<$ est un ordre de réduction, et d'après la définition 2.5 de la partie I, on peut écrire :
  $\Pi_{1,\tau}$:

$$\dfrac{\dfrac{\vdots}{\Gamma_1 \cup \Gamma_2, \overrightarrow{z}\tau <_p \overrightarrow{z}\mu \vdash \overrightarrow{y}\sigma\tau <_m \overrightarrow{y}\sigma\mu}}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \overrightarrow{z}\tau < \overrightarrow{z}\mu \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu, Q\sigma\tau}\ w$$

  On peut également écrire :

$$\dfrac{\dfrac{\dfrac{}{\overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ Ax}{(\overrightarrow{y}\sigma\tau \in \mathcal{T}(\Sigma)^m) \wedge (\overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau) \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ w + \wedge - l}{\forall \underline{\overrightarrow{y}} (\underline{\overrightarrow{y}} \in \mathcal{T}(\Sigma)^m) \wedge (\underline{\overrightarrow{y}} < \overrightarrow{y}\sigma\mu \Rightarrow Q\{\underline{\overrightarrow{y}}/\overrightarrow{y}\}) \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ \forall - l$$

ce qui se note :

$$\dfrac{\dfrac{\dfrac{}{\overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ Ax}{(\overrightarrow{y}\sigma\tau \in \mathcal{T}(\Sigma)^m) \wedge (\overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau) \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ w + \wedge - l}{\mathcal{PE}_{ind}(Q,<)\sigma\mu \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ \forall - l$$

Prolongeons maintenant cette dérivation :
$\Pi_{2,\tau}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ Ax}{(\overrightarrow{y}\sigma\tau \in \mathcal{T}(\Sigma)^m) \wedge (\overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau) \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ w + \wedge - l}{\mathcal{PE}_{ind}(Q,<)\sigma\mu \vdash \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \Rightarrow Q\sigma\tau}\ \forall - l}{\mathcal{PE}_{ind}(Q,<)\sigma\mu, \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \vdash Q\sigma\tau}\ imp}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \overrightarrow{z}\tau < \overrightarrow{z}\mu, \overrightarrow{y}\sigma\tau < \overrightarrow{y}\sigma\mu \vdash Q\sigma\tau}\ w$$

$\Pi_{3,\tau}$:

$$\dfrac{\dfrac{\Pi_1 \quad \Pi_2}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \overrightarrow{z}\tau < \overrightarrow{z}\mu \vdash Q\sigma\tau}\ cut}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu \vdash \overrightarrow{z}\tau < \overrightarrow{z}\mu \Rightarrow Q\sigma\tau}\ \Rightarrow -r$$

Π_4 :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\bigwedge_{(\tau \in Subst^{\Sigma}) \wedge (\mathcal{V}ar(Q\sigma) \subseteq \mathcal{D}om(\alpha))} \Pi_{3,\tau}}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \overrightarrow{z} \in \mathcal{T}(\Sigma)^p \vdash \overrightarrow{z} < \overrightarrow{z}\mu \Rightarrow Q\sigma}\ r_{\overrightarrow{y}}}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, ((\overrightarrow{z} \in \mathcal{T}(\Sigma)^p) \wedge (\overrightarrow{z} < \overrightarrow{z}\mu)) \vdash Q\sigma}\ imp + \wedge - l}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu \vdash ((\overrightarrow{z} \in \mathcal{T}(\Sigma)^p) \wedge (\overrightarrow{z} < \overrightarrow{z}\mu)) \Rightarrow Q\sigma}\ \Rightarrow -r}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu \vdash \mathcal{PE}_{ind}(Q\sigma,<)\mu}\ \forall - r}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu} \mathcal{PE}_{ind}(Q\sigma,<)\mu, (Q[r]_\omega)\sigma\mu}\ w + push$$

Π_5 :

$$\dfrac{\dfrac{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\mu, \mathcal{RE}_{ind}((Q[r]_\omega)\sigma,<)\mu\}} (Q[r]_\omega)\sigma\mu \text{ (séquent (1.3))}}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \mathcal{PE}_{ind}((Q[r]_\omega)\sigma,<)\mu \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu} (Q[r]_\omega)\sigma\mu}\ pop}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \mathcal{PE}_{ind}(Q\sigma,<)\mu, \mathcal{PE}_{ind}((Q[r]_\omega)\sigma,<)\mu \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu} (Q[r]_\omega)\sigma\mu}\ w$$

Π_6 :

$$\dfrac{\dfrac{}{\mathcal{PE}_{ind}(Q\sigma,<)\mu \vdash_{\mathcal{RE}_1} \mathcal{PE}_{ind}((Q[r]_\omega)\sigma,<)\mu}\ Ax}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \mathcal{PE}_{ind}(Q\sigma,<)\mu \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu} \mathcal{PE}_{ind}((Q[r]_\omega)\sigma,<)\mu, (Q[r]_\omega)\sigma\mu}\ w + push$$

Π_7 :

$$\dfrac{\Pi_5 \quad \Pi_6}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu, \mathcal{PE}_{ind}(Q\sigma,<)\mu \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu} (Q[r]_\omega)\sigma\mu}\ cut$$

Π_8 :

$$\dfrac{\Pi_4 \quad \Pi_7}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\sigma\mu \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu} (Q[r]_\omega)\sigma\mu}\ cut$$

Π_9 :

$$\dfrac{\dfrac{\Pi_8}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\mu\}} (Q[r]_\omega)\sigma\mu}\ push}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\mu\}} (Q[r]_\omega)\sigma\mu, Q\alpha}\ w$$

Π_{10} :

$$\dfrac{\dfrac{}{(Q[r]_\omega)\sigma\mu \vdash_{\mathcal{RE}_1} Q\alpha}\ Ax}{\Gamma_1 \cup \Gamma_2, (Q[r]_\omega)\sigma\mu \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\mu\}} Q\alpha}\ w + push$$

Π :

$$\dfrac{\Pi_9 \quad \Pi_{10}}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma\mu \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\mu\}} Q\alpha}\ cut$$

Or $\alpha = \sigma\mu[V]$, avec V quelconque tel que $\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\}) \subset V$, donc le séquent obtenu peut aussi s'écrire :

$$\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha \cup \{\mathcal{RE}_{ind}(Q,<)\alpha\}} Q\alpha$$

mais cela contredit la définition de α.

– Supposons qu'une règle **Rewrite** soit appliquée à la position ω dans Q.
Par définition de **Rewrite**, on a :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \quad \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[r\nu]_\omega$$

Observons maintenant que $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[r\nu]_\omega$ est un sous-but, et $(Q[r\nu]_\omega)\alpha <_2 Q\alpha$, donc le séquent :

$$\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha \cup \{\mathcal{RE}_{ind}(Q[r\nu]_\omega,<)\alpha\}} (Q[r\nu]_\omega)\alpha \tag{1.4}$$

a une preuve dans le calcul des séquents modulo, car $Q\alpha$ est minimal dans $\mathcal{E}$.
Considérons maintenant les dérivations suivantes :
Π_1 :

$$\dfrac{\dfrac{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha \cup \{\mathcal{RE}_{ind}(Q[r\nu]_\omega,<)\alpha\}} (Q[r\nu]_\omega)\alpha \text{ (séquent 1.4)}}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q[r\nu]_\omega,<)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha} (Q[r\nu]_\omega)\alpha}\ pop}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\alpha, \mathcal{PE}_{ind}(Q[r\nu]_\omega,<)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha} (Q[r\nu]_\omega)\alpha}\ w$$

Π_2 :

$$\dfrac{\dfrac{}{\mathcal{PE}_{ind}(Q,<)\alpha \vdash_{\mathcal{RE}_1} \mathcal{PE}_{ind}(Q[r\nu]_\omega,<)\alpha}\ Ax}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha} \mathcal{PE}_{ind}(Q[r\nu]_\omega,<)\alpha, (Q[r\nu]_\omega)\alpha}\ w$$

Π_3 :

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma_1 \cup \Gamma_2, \mathcal{PE}_{ind}(Q,<)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha} (Q[r\nu]_\omega)\alpha}\ cut$$

Π_4 :

$$\dfrac{\dfrac{\Pi_3}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha \cup \{\mathcal{RE}_{ind}(Q,<)\alpha\}} (Q[r\nu]_\omega)\alpha}\ push}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha \cup \{\mathcal{RE}_{ind}(Q,<)\alpha\}} (Q[r\nu]_\omega)\alpha, Q\alpha}\ w$$

Π_5 :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{(Q[r\nu]_\omega) \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q}\ Ax}{\overrightarrow{x} \in \mathcal{T}(\Sigma)^n, (Q[r\nu]_\omega) \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q \text{ (avec } \overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})})}\ w}{\overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} (Q[r\nu]_\omega) \Rightarrow Q}\ \Rightarrow -r}{\vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_{2\alpha}} (Q[r\nu]_\omega)\alpha \Rightarrow Q\alpha}\ r\alpha}{(Q[r\nu]_\omega)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_{2\alpha}} Q\alpha}\ imp}{\Gamma_1 \cup \Gamma_2, (Q[r\nu]_\omega)\alpha \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_{2\alpha} \cup \{\mathcal{RE}_{ind}(Q,<)\alpha\}} Q\alpha}\ w + push$$

Π :

$$\dfrac{\Pi_4 \quad \Pi_5}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_{2\alpha} \cup \{\mathcal{RE}_{ind}(Q,<)\alpha\}} Q\alpha}\ cut$$

et cela contredit de nouveau la définition de α.

□

2

Optimisation

Sommaire

2.1 Motivation **105**
2.2 Règles d'inférence du système de recherche de preuve OptInd-Narrow **107**
2.3 L'exemple simple revisité **110**
2.4 Justification théorique **112**
2.4.1 Le système d'inférence DecIndNarrow 112
2.4.2 Labels 113
2.4.3 Correction et complétude réfutationnelle de OptIndNarrow 114
2.5 Exemple **119**

2.1 Motivation

Dans le système de recherche de preuve optimisé OptIndNarrow que nous allons présenter dans ce chapitre, on tire profit du fait que, lorsqu'une réduction a lieu à l'intérieur d'un membre q_i d'un but $q_1 \approx q_2$, celui-ci décroît ou reste équivalent à lui-même par rapport à l'ordre $\leqslant_e$ construit à la section 1.3 de la partie II. Les conséquences principales sont :

1. aider à faire en sorte que les conditions d'ordre soient satisfaites lorsque l'on a besoin d'appliquer l'hypothèse de récurrence ;
2. aider à empêcher la production de successions infinies de réécritures ;
3. donner néanmoins la possibilité de réécrire un but avec une égalité $l \approx r$ qui soit telle que $l \gtrless r$;
4. empêcher de nombreuses applications de règles d'inférence qui ont peu de chance d'aboutir à une preuve.

Le premier point sera justifié de manière formelle à la fin de ce chapitre, les deuxièmes et troisièmes points sont clairs, et l'exemple suivant va permettre d'illustrer le quatrième.
Considérons :

1. une signature $\Sigma = \{0, s, +, *\}$;
2. une précédence totale $\prec$ sur Σ définie par $0 \prec s \prec + \prec *$;
3. une permutation π de $\{1, 2\}$ telle que $\pi(1) = 2$ et $\pi(2) = 1$;

4. une fonction statut τ sur Σ qui envoie $+$ et $*$ sur lex_π ;
5. un quasi-ordre récursif sur les chemins $\leqslant$ induit par $\prec$ et τ (voir définition 1.8 de la partie II) ;
6. deux systèmes de réécriture $\mathcal{RE}_1$ et $\mathcal{RE}_2$ définis par :

$$\mathcal{RE}_1 = \begin{cases} x + 0 \to x \\ x + s(y) \to s(x+y) \\ x * 0 \approx 0 \\ x * s(y) \approx x * y + x \end{cases} \qquad \mathcal{RE}_2 = \begin{cases} (x+y)+z \approx x+(y+z) \\ (x+y)+z \approx (x+z)+y \end{cases}$$

Posons $\Gamma_2 = \{NI, LI(\approx), Noeth(<_2, \mathcal{T}(\Sigma)^2)\}$, et utilisons le système de recherche de preuve IndNarrow pour démontrer le but :

$$\boxed{\emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} (Y+Y)*Z \approx Y*Z+Y*Z}$$

On peut appliquer la règle **Induce**, car $\mathcal{RE}_1$ est terminant sur les termes clos et suffisamment complet. Cela peut être fait aux occurrences 1.1, 2.1 ou 2.2 du but. On choisit arbitrairement l'occurrence 2.1, et cela nous conduit à prouver les deux séquents :

$$\begin{array}{|l|}\hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_1\}} (Y_1+Y_1)*0 \approx 0+Y_1*0 \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} (Y_1+Y_1)*s(Z_1) \approx (Y_1*Z_1+Y_1)+Y_1*s(Z_1) \\ \hline \end{array}$$

avec

$$\begin{array}{|l|}\hline \kappa_1 = \mathcal{RE}_{ind}((Y+Y)*Z \approx Y*Z+Y*Z, <)\{Y \mapsto Y_1; Z \mapsto 0\} \\ \hline \kappa_2 = \mathcal{RE}_{ind}((Y+Y)*Z \approx Y*Z+Y*Z, <)\{Y \mapsto Y_1; Z \mapsto s(Z_1)\} \\ \hline \end{array}$$

Après normalisation à l'aide des règles de $\mathcal{RE}_1$, on obtient les deux nouveaux sous-buts :

$$\begin{array}{|l|}\hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_1\}} 0 \approx 0 \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} (Y_1+Y_1)*Z_1+(Y_1+Y_1) \approx (Y_1*Z_1+Y_1)+(Y_1*Z_1+Y_1) \\ \hline \end{array}$$

Trivial élimine le premier. **Rewrite**$_2$ peut être appliquée sur le second, à l'aide de la règle κ_2, car $(Y_1, Z_1) < (Y_1, s(Z_1))$. On obtient alors :

$$\boxed{\emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} (Y_1*Z_1+Y_1*Z_1)+(Y_1+Y_1) \approx (Y_1*Z_1+Y_1)+(Y_1*Z_1+Y_1)}$$

Rewrite$_2$ peut de nouveau être appliquée sur ce sous-but, mais à l'aide de l'égalité $(x+y)+z \approx x+(y+z)$ cette fois-ci. Cependant, cette dernière peut être utilisée pour réécrire le but de quatre façons différentes :

$$\begin{array}{|ll|}\hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} ((Y_1*Z_1+Y_1*Z_1)+Y_1)+Y_1 \approx (Y_1*Z_1+Y_1)+(Y_1*Z_1+Y_1) & (1) \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} Y_1*Z_1+(Y_1*Z_1+(Y_1+Y_1)) \approx (Y_1*Z_1+Y_1)+(Y_1*Z_1+Y_1) & (2) \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} (Y_1*Z_1+Y_1*Z_1)+(Y_1+Y_1) \approx ((Y_1*Z_1+Y_1)+Y_1*Z_1)+Y_1 & (3) \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} (Y_1*Z_1+Y_1*Z_1)+(Y_1+Y_1) \approx Y_1*Z_1+(Y_1+(Y_1*Z_1+Y_1)) & (4) \\ \hline \end{array}$$

On peut néanmoins s'apercevoir que seules les réécritures vers (1) et (3) engendrent des sous-buts qui sont plus petits pour l'ordre $<$. Choisissons la réécriture vers (1). **Rewrite**$_2$ peut alors être appliquée, à l'aide de la même égalité $(x+y)+z \approx x+(y+z)$. Cependant, on peut encore l'utiliser de deux façons différentes :

$$\begin{array}{|ll|}\hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} ((Y_1*Z_1+Y_1*Z_1)+Y_1)+Y_1 \approx ((Y_1*Z_1+Y_1)+Y_1*Z_1)+Y_1 & (1') \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} ((Y_1*Z_1+Y_1*Z_1)+Y_1)+Y_1 \approx Y_1*Z_1+(Y_1+(Y_1*Z_1+Y_1)) & (2') \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} (Y_1*Z_1+Y_1*Z_1)+(Y_1+Y_1) \approx (Y_1*Z_1+Y_1)+(Y_1*Z_1+Y_1) & (3') \\ \hline \emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa_2\}} (Y_1*Z_1+(Y_1*Z_1+Y_1))+Y_1 \approx (Y_1*Z_1+Y_1)+(Y_1*Z_1+Y_1) & (4') \\ \hline \end{array}$$

Seule la réécriture vers $(1')$ produit un sous-but plus petit par rapport à $<$. Choisissons donc la réécriture qui conduit vers $(1')$. On peut alors de nouveau appliquer $\mathbf{Rewrite}_2$, cette fois-ci avec l'égalité $(x+y)+z \approx (x+z)+y$. Cependant, celle-ci peut être utilisée de quatre façons différentes :

$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_1,\kappa_2,\kappa_3\}} ((Y_1 * Z_1 + Y_1) + Y_1 * Z_1) + Y_1 \approx ((Y_1 * Z_1 + Y_1) + Y_1 * Z_1) + Y_1$	$(1'')$
$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_1,\kappa_2,\kappa_3\}} ((Y_1 * Z_1 + Y_1 * Z_1) + Y_1) + Y_1 \approx ((Y_1 * Z_1 + Y_1 * Z_1) + Y_1) + Y_1$	$(2'')$
$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_1,\kappa_2,\kappa_3\}} ((Y_1 * Z_1 + Y_1 * Z_1) + Y_1) + Y_1 \approx ((Y_1 * Z_1 + Y_1) + Y_1 * Z_1) + Y_1$	$(3'')$
$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_1,\kappa_2,\kappa_3\}} ((Y_1 * Z_1 + Y_1 * Z_1) + Y_1) + Y_1 \approx ((Y_1 * Z_1 + Y_1) + Y_1) + Y_1 * Z_1$	$(4'')$

Seule la réécriture vers $(2'')$ conduit à un sous-but plus petit pour l'ordre $<$. On choisit donc la réécriture vers $(2'')$ et on peut alors appliquer la règle **Trivial**.

2.2 Règles d'inférence du système de recherche de preuve OptIndNarrow

Induce $\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q[t]_{i.\omega_i}[\![I]\!] \rightarrowtail \bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma = mgu(t,l)}} \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}'_2} (Q[r]_{i.\omega_i})\sigma[\![I]\!]$

si $i.\omega_i \in \mathcal{DM}(Q)$ et $\mathcal{RE}'_2 = L_i(\mathcal{RE}_2\sigma) \cup \{\mathcal{RE}_{ind}(Q,<)\sigma : i\}$

Push$_1$ $\Gamma_1, l \approx r \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \approx r\} \vert \mathcal{RE}_2} Q[\![I]\!]$

Orient$_1$ **a** $\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\} \vert \mathcal{RE}_2} Q[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \to r\} \vert \mathcal{RE}_2} Q[\![I]\!]$

$(\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $l > r$

Orient$_1$ **b** $\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\} \vert \mathcal{RE}_2} Q[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \overset{\gtrless}{\to} r\} \vert \mathcal{RE}_2} Q[\![I]\!]$

$(\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $l \gtrless r$

Push$_2$ $\Gamma_1 \vert \Gamma_2, l \approx r \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{l \approx r\}} Q[\![I]\!]$

Orient$_2$ **a** $\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{\kappa\}} Q[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{\kappa'\}} Q[\![I]\!]$

$(((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $\kappa' = l \to r)$ ou

$((\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$ ou $\kappa = \mathcal{RE}_{ind}(r \approx l)\mu)$ et $\kappa' = \mathcal{RE}_{ind}(l \to r)\mu))$ et $l > r$

Orient$_2$ **b** $\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{\kappa\}} Q[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{\kappa',\kappa''\}} Q[\![I]\!]$

$(((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $\kappa' = l \overset{\gtrless}{\to} r$ et $\kappa'' = r \overset{\gtrless}{\to} l)$

ou $((\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$ ou $\kappa = \mathcal{RE}_{ind}(r \approx l)\mu)$

et $\kappa' = \mathcal{RE}_{ind}(l \overset{\gtrless}{\to} r)\mu$ et $\kappa'' = \mathcal{RE}_{ind}(r \overset{\gtrless}{\to} l)\mu))$ et $l \gtrless r$

'$\bullet$' est un opérateur AC dont '$\diamond$' est l'élément neutre.

FIG. 2.1 – OptIndNarrow(première partie)

Dans l'intention de tirer parti de la décroissance du but tout au long du processus, le système de recherche de preuve OptIndNarrow doit diviser et raffiner les règles d'inférence **Orient** et **Rewrite** du système IndNarrow , et leur ajouter plusieurs ingrédients additionnels : des contraintes d'ordre sur les buts, des indices dans $\mathcal{RE}_2$, pour garder la trace de certaines

Rewrite$_1$ a	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\} \vert \mathcal{RE}_2} Q[l\sigma]_{i.\omega_i}[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\} \vert \mathcal{RE}'_2} Q[r\sigma]_{i.\omega_i}[\![I]\!]$
	$\kappa = l \rightarrow r$ ou $((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $Q_{\vert i}[l\sigma]_{\omega_i} > Q_{\vert i}[r\sigma]_{\omega_i})$ et
	$\mathcal{RE}'_2 = L_i(\mathcal{RE}_2)$
Rewrite$_1$ b	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\} \vert \mathcal{RE}_2} Q[l\sigma]_{i.\omega_i}[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\} \vert \mathcal{RE}_2} Q[r\sigma]_{i.\omega_i}[\![I]\!]$
	$\kappa = l \xrightarrow{\gtrless} r$ ou $((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $Q_{\vert i}[l\sigma]_{\omega_i} \gtrless Q_{\vert i}[r\sigma]_{\omega_i})$
Rewrite$_2$ a	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{\kappa\}} Q[l\sigma]_{i.\omega_i}[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}'_2} Q[r\sigma]_{i.\omega_i}[\![I']\!]$
	$(\kappa = \kappa_0$ ou $\kappa = \mathcal{RE}_{ind}(\kappa_0, \leqslant)\mu$ ou $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu : j)$ et
	$(\kappa_0 = l \rightarrow r$ ou $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l)$ et
	$I' = I \ \cup (\bigcup_{x \in Var(\kappa_0)} \{x\sigma \leqslant x\mu\})$ et
	$\mathcal{RE}'_2 = L_i(\mathcal{RE}_2 \cup \{\kappa\})$ et
	(si $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l$ alors $Q_{\vert i}[l\sigma]_{\omega_i} > Q_{\vert i}[r\sigma]_{\omega_i})$ et
	(si $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu : j,\ \omega_i = \varepsilon$ et $j \neq i$ alors $\kappa_0\sigma <_e \kappa_0\mu)$
Rewrite$_2$ b	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{\kappa\}} Q[l\sigma]_{i.\omega_i}[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}'_2} Q[r\sigma]_{i.\omega_i}[\![I']\!]$
	$(\kappa = \kappa_0$ ou $\kappa = \mathcal{RE}_{ind}(\kappa_0, \leqslant)\mu$ ou $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu : j)$ et
	$(\kappa_0 = l \xrightarrow{\gtrless} r$ ou $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l)$ et
	$I' = I \ \cup (\bigcup_{x \in Var(\kappa_0)} \{x\sigma \leqslant x\mu\})$ et
	(si $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l$ alors $Q_{\vert i}[l\sigma]_{\omega_i} \gtrless Q_{\vert i}[r\sigma]_{\omega_i})$ et
	(si $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu : j,\ \omega = \varepsilon$ et $j \neq i$ alors $\kappa_0\sigma <_e \kappa_0\mu)$
Trivial	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} t \approx t[\![I]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} \diamond[\![I]\!]$
Elim	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q[\![I \cup \{t \leqslant t'\}]\!] \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q[\![I]\!]$ si $t \leqslant t'$
Refutation	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q[\![\emptyset]\!] \rightarrowtail$ Refutation
	si aucune autre règle ne s'applique

FIG. 2.2 – OptIndNarrow(seconde partie)

réductions, et des opérateurs pour transformer des conditions d'ordre strict en conditions d'ordre large. Toute ces notions seront définies formellement à la section 2.4, mais nous allons d'abord les introduire d'une façon plus intuitive, en examinant quelques situations caractéristiques.
Souvenons-nous dans un premier temps que, dans le système de recherche de preuve OptIndNarrow, lorsqu'une réduction s'effectue à l'intérieur d'un membre q_i d'un but $q_1 \approx q_2$, celui-ci décroît ou reste équivalent à lui-même par rapport à $\leqslant$. L'effet d'une telle réduction ne sera bien-sûr pas le même suivant que l'on se situe dans un cas ou dans l'autre. Cela suggère de diviser les règles d'inférence **Rewrite$_1$** et **Rewrite$_2$** en deux familles :

1. **Rewrite$_1$ a** et **Rewrite$_2$ a** si le membre décroît effectivement.
2. **Rewrite$_1$ b** et **Rewrite$_2$ b** si le membre reste équivalent.

Maintenant, considérons une égalité $l \approx r$. Si $l > r$, et puisque $>$ est un ordre de réduction, réécrire un membre d'un but Q avec une égalité $l \approx r$ le fait décroître (il reste équivalent si $l \gtrless r$).

Cela incite à donner la possibilité d'orienter les règles de $\mathcal{RE}_2$, ainsi qu'à faire une distinction entre les règles $l \to r$ telles que $l > r$ ou $l \gtrless r$. C'est donc ce qui a motivé l'introduction des règles d'inférence **Orient**$_1$ **a**, **Orient**$_1$ **b**, **Orient**$_2$ **a** et **Orient**$_2$ **b**. On peut déjà faire deux remarques :

1. **Orient**$_2$ **a** et **Orient**$_2$ **b** peuvent être appliquées à une hypothèse de récurrence ;
2. **Orient**$_1$ **b** et **Orient**$_2$ **b** produisent des règles que l'on note avec le symbole $\overset{\gtrless}{\to}$, pour souligner que leurs deux membres sont équivalents par rapport à $\leqslant$.

Considérons maintenant la règle d'inférence **Induce**. Le système de recherche de preuve instancie l'hypothèse de récurrence par la substitution surréductrice σ et l'annote avec un label (un entier i) qui indique où la réduction s'est opérée (i.e dans quel membre du but équationnel). On verra que ceci permettra à un opérateur L_i de transformer les conditions d'ordre strict dans les buts précédents en des conditions d'ordre large plus faibles. Ces conditions sont alors enregistrées dans un ensemble de contraintes I. Pour avoir une meilleure intuition de la façon dont **Induce** fonctionne avec ces ingrédients additionnels, examinons la situation suivante.

Exemple 2.1 Considérons deux applications consécutives de **Induce** :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![I]\!] \underset{\textbf{Induce}}{\rightarrowtail} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} Q'[\![I]\!] \underset{\textbf{Induce}}{\rightarrowtail} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}''_2} Q''[\![I]\!]$$

Par définition de **Induce**, il existe $i, i' \in \{1,2\}$, $\omega_i \in \mathcal{D}om(Q_{|i})$, $\omega_{i'} \in \mathcal{D}om(Q'_{|i'})$, $\sigma, \sigma' \in Subst^{\Sigma}$, et $t, t' \in \mathcal{T}(\Sigma, \mathcal{X})$, tels que :

1. $Q = Q[t]_{i.\omega_i}$, $l \to r \in \mathcal{RE}_1$, $\sigma = mgu(t,l)$, $Q' = Q[r\sigma]_{i.\omega_i}$,
   $\mathcal{RE}'_2 = L_i(\mathcal{RE}_2\sigma) \cup \{\mathcal{RE}_{ind}(Q,<)\sigma : i\}$
2. $Q' = Q'[t']_{i'.\omega_{i'}}$, $l' \to r' \in \mathcal{RE}_1$, $\sigma' = mgu(t',l')$, $Q'' = Q'[r'\sigma']_{i'.\omega_{i'}}$,
   $\mathcal{RE}''_2 = L_{i'}(\mathcal{RE}'_2\sigma') \cup \{\mathcal{RE}_{ind}(Q',<)\sigma' : i'\}$

Observons l'hypothèse de récurrence $\mathcal{RE}_{ind}(Q,<)\sigma : i$ produite par la première application de **Induce**. Remarquons que $\mathcal{RE}'_2$ est remplacé par $L_{i'}(\mathcal{RE}'_2\sigma)$ après la seconde application de **Induce**. Et puisque la règle $\mathcal{RE}_{ind}(Q,<)\sigma : i$ est un élément de $\mathcal{RE}'_2$, quel effet cela a-t-il sur celle-ci ? La notation suggère deux étapes :

- Premièrement : $\mathcal{RE}_{ind}(Q,<)\sigma : i$ est remplacée par $\mathcal{RE}_{ind}(Q,<)\sigma\sigma' : i$.
- Deuxièmement : l'opérateur $L_{i'}$ relâche les conditions d'ordre sur les buts indexés par i'.
  - Cas 1 : $i' \neq i$. L'opérateur $L_{i'}$ ôte le label i, et remplace le symbole d'ordre strict $<$ par le symbole d'ordre large correspondant $\leqslant$, donc $\mathcal{RE}_{ind}(Q,<)\sigma\sigma' : i$ est remplacé par $\mathcal{RE}_{ind}(Q,\leqslant)\sigma\sigma'$.
  - Cas 2 : $i' = i$. L'opérateur $L_{i'}$ n'a pas d'effet sur $\mathcal{RE}_{ind}(Q,<)\sigma\sigma' : i$, donc $\mathcal{RE}_{ind}(Q,<)\sigma\sigma : i$ n'est pas remplacée.

Il faut remarquer que chaque règle d'inférence **Rewrite**$_1$ **a** et **Rewrite**$_2$ **a** travaille avec l'opérateur L_i, alors qu'il n'en n'est pas de même pour **Rewrite**$_1$ **b** et **Rewrite**$_2$ **b**. On peut alors s'apercevoir que cet opérateur n'intervient qu'à partir du moment où un membre du but décroît après une étape de réécriture. Mais si l'on veut réécrire un but à l'aide d'un but précédent, il y a encore des précautions à prendre, comme dans la situation suivante par exemple.

Exemple 2.2 Revenons sur l'exemple 2.1, et supposons que Q'' contienne en outre une instance d'un des membres de l'égalité Q. Cela se formalise en écrivant qu'il existe $\omega'' \in \mathcal{D}om(Q'')$, $\nu \in Subst^{\Sigma,\mathcal{X}}$, $q_1, q_2 \in \mathcal{T}(\Sigma, \mathcal{X})$, $i'', j \in \{1,2\}$, tels que

1. $Q = q_1 \approx q_2$;

2. $Q'' = Q''[q_j\nu]_{i''.\omega''}$ (avec $j \in \{1, 2\}$).

On est tenté de réécrire le but à l'aide de l'égalité Q, mais cela n'est possible que sous certaines conditions. Observons comment les règles d'inférence **Rewrite**$_2$ **a** ou **Rewrite**$_2$ **b** gèrent les conditions. En revenant aux deux cas de l'exemple 2.1, on a :

- Cas 1 : $i' \neq i$.
  On a vu dans l'exemple 2.1 que la règle de réécriture κ à utiliser est alors $\kappa = \mathcal{RE}_{ind}(Q, \leqslant)\sigma\sigma'$. Si $\overrightarrow{x} \in \mathcal{X}^n$ désigne le vecteur des variables de Q, réécrire le but avec κ est permis seulement si $\overrightarrow{x}\nu \leqslant_n \overrightarrow{x}\sigma\sigma'$. On peut remarquer cependant que vérifier la condition $\overrightarrow{x}\nu \leqslant_n \overrightarrow{x}\sigma\sigma'$ revient à prouver l'inégalité $x_k\nu \leqslant x_k\sigma\sigma'$ pour chaque composante x_k de $\overrightarrow{x}$. Ainsi, la règle d'inférence **Rewrite**$_2$ **a** ou **Rewrite**$_2$ **b** réécrit le but avec κ, et ajoute ces inégalités portant sur les composantes dans l'ensemble I des contraintes.
- Cas 2 : $i' = i$.
  La règle de réécriture est $\mathcal{RE}_{ind}(Q, <)\sigma\sigma' : i$. On peut ici observer que les règles d'inférence **Rewrite**$_2$ **a** et **Rewrite**$_2$ **b** distinguent un sous-cas spécial :
  - Sous-cas 1 : $i'' \neq i$ et $\omega'' = \varepsilon$.
    Réécrire le but avec κ dans ce cas nécessite de vérifier d'abord l'inégalité additionnelle $Q\nu <_e Q\sigma\sigma'$ (il faut garder présent à l'esprit que le symbole $<_e$ désigne l'ordre sur l'ensemble des égalités défini à la section 1.3 de la partie II). Après cela, la règle d'inférence **Rewrite**$_2$ **a** ou **Rewrite**$_2$ **b** agit comme au cas 1.
  - Sous-cas 2 : Sinon.
    Rewrite$_2$ **a** ou **Rewrite**$_2$ **b** agissent comme au cas 1.

Revenons maintenant à l'exemple 2.1. Supposons que le système de recherche de preuve OptInd-Narrow construit ultérieurement une suite de dérivations du type :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2''} Q''[\![I]\!] \overset{*}{\rightarrowtail} {\Gamma_1}^n|{\Gamma_2}^n \vdash_{\mathcal{RE}_1{}^n|\mathcal{RE}_2{}^n} \diamond[\![I^n]\!]$$

Alors, la règle d'inférence **Elim** peut finir le travail en testant toutes les inégalités stockées dans l'ensemble I^n.

2.3 L'exemple simple revisité

Pour avoir une meilleure compréhension de la façon dont **OptIndNarrow** travaille, observons la preuve de la commutativité de l'addition dans l'arithmétique de Peano, par référence à l'exemple simple développé à la section 1.2 du chapitre 1 de cette partie. On affecte ici au symbole + un statut multi-ensemble.

1. Considérons la première application de la règle d'inférence **Induce**, et associons maintenant le label 1 à l'hypothèse de récurrence, dans l'intention d'indiquer que la réduction s'effectue dans le premier membre. Ainsi, les deux sous-buts sont maintenant notés (remarquer le :1) :

$\emptyset\|\Gamma_2$	$\vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(X+Y\approx Y+X)\{X\mapsto X_1;Y\mapsto 0\}:1}$	$X_1 \approx 0 + X_1$
$\emptyset\|\Gamma_2$	$\vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(X+Y\approx Y+X)\{X\mapsto X_1;Y\mapsto s(Y_1)\}:1}$	$s(X_1 + Y_1) \approx s(Y_1) + X_1$

2. Considérons la seconde application de la règle d'inférence **Induce** au premier sous-but, où la réduction a lieu maintenant dans le second membre. Avec IndNarrow , on obtenait les sous-buts suivants :

$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \ \mathcal{RE}_{ind}(X+Y \approx Y+X,<)\{X \mapsto 0; Y \mapsto 0\} \ \mathcal{RE}_{ind}(X_1 \approx 0+X_1,<)\{X_1 \mapsto 0\}}$	$0 \approx 0$
$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \ \mathcal{RE}_{ind}(X+Y \approx Y+X,<)\{X \mapsto s(X_2); Y \mapsto 0\} \ \mathcal{RE}_{ind}(X_1 \approx 0+X_1,<)\{X_1 \mapsto s(X_2)\}}$	$s(X_2) \approx s(0+X_2)$

Le système OptIndNarrow va alors introduire les modifications suivantes :

(a) il associe d'abord le label 2 à la nouvelle hypothèse de récurrence dans chaque sous-but, donc à $\mathcal{RE}_{ind}(X_1 \approx 0+X_1,<)\{X_1 \mapsto 0\}$ et $\mathcal{RE}_{ind}(X_1 \approx 0+X_1,<)\{X_1 \mapsto s(X_2)\}$;

(b) il transforme l'hypothèse de récurrence $\mathcal{RE}_{ind}(X+Y \approx Y+X)\{X \mapsto X_1; Y \mapsto 0\} : 1$ introduite par la première application de **Induce** de la façon suivante :

i. elle est d'abord instanciée par $\{X_1 \mapsto 0\}$ dans le premier sous-but, et par $\{X_1 \mapsto s(X_2)\}$ dans le deuxième. Ces deux instances s'écriront donc :

$\mathcal{RE}_{ind}(X+Y \approx Y+X,<)\{X \mapsto 0; Y \mapsto 0\} : 1$	(κ_1)
$\mathcal{RE}_{ind}(X+Y \approx Y+X,<)\{X \mapsto s(X_2); Y \mapsto 0\} : 1$	(κ_2)

Elles notent en fait les règles suivantes :

$(\underline{x},\underline{y}) <_2 (0,0) \Rightarrow \underline{x}+\underline{y} \approx \underline{y}+\underline{x} : 1$	(κ_1)
$(\underline{x},\underline{y}) <_2 (s(X_2),0) \Rightarrow \underline{x}+\underline{y} \approx \underline{y}+\underline{x} : 1$	(κ_2)

ii. il ôte ensuite le label 1 de ces deux règles, et remplace dans chacune le symbole d'ordre strict $<$ par le symbole de quasi-ordre correspondant $\leqslant$. En effet, considérons la règle (κ_2), par exemple. Puisqu'une première réduction s'est opérée dans le membre gauche du but et une seconde dans le membre droit, on verra à la section 2.4.3 que, dès qu'un futur sous-but Q contiendra un sous-terme du type $\underline{x}\sigma + \underline{y}\sigma$ ou $\underline{y}\sigma + \underline{x}\sigma$, les égalités syntaxiques $\underline{x}\sigma = s(X_2)$ et $\underline{y}\sigma = 0$ ne pourront pas avoir lieu simultanément. Donc, il suffira de vérifier les inégalités larges $\underline{x}\sigma \leqslant s(X_2)$ et $\underline{y}\sigma \leqslant 0$ pour être sûr que la condition $(\underline{x}\sigma, \underline{y}\sigma) <_2 (s(X_2), 0)$ soit satisfaite.

Ces modifications étant introduites, les deux nouveaux sous-buts s'écriront :

$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \ \mathcal{RE}_{ind}(X+Y \approx Y+X,\leqslant)\{X \mapsto 0; Y \mapsto 0\} \ \mathcal{RE}_{ind}(X_1 \approx 0+X_1,<)\{X_1 \mapsto 0\} : 2}$	$0 \approx 0$
$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \ \mathcal{RE}_{ind}(X+Y \approx Y+X,\leqslant)\{X \mapsto s(X_2); Y \mapsto 0\} \ \mathcal{RE}_{ind}(X_1 \approx 0+X_1,<)\{X_1 \mapsto s(X_2)\} : 2}$	$s(X_2) \approx s(0+X_2)$

3. **Trivial** élimine le premier.

4. Puisque $X+Y \gtreqless Y+X$, et du fait que $\leqslant$ est un quasi-ordre de simplification, réécrire un terme à l'aide de cette égalité en crée un nouveau qui est équivalent par rapport à ce quasi-ordre. Pour garder cela en mémoire, la nouvelle règle d'inférence **Orient b** remplace l'égalité $X+Y \approx Y+X$ dans les règles de réécriture de $\mathcal{RE}_2$ par $X+Y \overset{\geqslant}{\to} Y+X$ et $Y+X \overset{\geqslant}{\to} X+Y$. Le sous-but est maintenant noté :

$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \ \mathcal{RE}_{ind}(X+Y \overset{\geqslant}{\to} Y+X,\leqslant)\{X \mapsto s(X_2); Y \mapsto 0\} \ \mathcal{RE}_{ind}(Y+X \overset{\geqslant}{\to} X+Y,\leqslant)\{X \mapsto s(X_2); Y \mapsto 0\} \ \mathcal{RE}_{ind}(X_1 \approx 0+X_1,<)\{X_1 \mapsto s(X_2)\} : 2}$	$s(X_2) \approx s(0+X_2)$

5. Dans l'intention d'appliquer la règle conditionnelle :

$$\mathcal{RE}_{ind}(Y+X \overset{\geqslant}{\to} X+Y,\leqslant)\{X \mapsto s(X_2); Y \mapsto 0\}$$

sur $0+X_2$, comme nous venons juste de le remarquer ci-dessus, on doit vérifier les inégalités $0 \leqslant 0$ et $X_2 \leqslant s(X_2)$. Cela est réalisé par la règle d'inférence **Rewrite**$_2$ **b** qui réécrit le but avec cette hypothèse de récurrence, et stocke les conditions dans un ensemble de contraintes. Le sous-but est ainsi noté :

$$\emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\ \mathcal{RE}_{ind}(X+Y \overset{\geqslant}{\to} Y+X, \leqslant)\{X \mapsto s(X_2); Y \mapsto 0\}\ \mathcal{RE}_{ind}(Y+X \overset{\geqslant}{\to} X+Y, \leqslant)\{X \mapsto s(X_2); Y \mapsto 0\}\ \mathcal{RE}_{ind}(X_1 \approx 0+X_1, <)\{X_1 \mapsto s(X_2)\}:2} s(X_2) \approx s(X_2+0)[\![0 \leqslant 0; X_2 \leqslant s(X_2)]\!]$$

6. **Rewrite**$_1$ **a** réécrit le but avec la règle $X+0 \to X$. Observons que la réduction s'effectue dans le second membre du but, le label 2 ne sera donc pas ôté. Le nouveau sous-but s'écrit ainsi :

$$\emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\ \mathcal{RE}_{ind}(X+Y \overset{\geqslant}{\to} Y+X, \leqslant_2, T^2_\Sigma)\{X \mapsto s(X_2); Y \mapsto 0\}\ \mathcal{RE}_{ind}(Y+X \overset{\geqslant}{\to} X+Y, \leqslant_2, T^2_\Sigma)\{X \mapsto s(X_2); Y \mapsto 0\}\ \mathcal{RE}_{ind}(X_1 \approx 0+X_1, \leqslant, T_\Sigma)\{X_1 \mapsto s(X_2)\}} s(X_2) \approx s(X_2)[\![0 \leqslant 0; X_2 \leqslant s(X_2)]\!]$$

7. **Trivial** élimine l'égalité syntaxique, mais les contraintes demeurent :

$$\emptyset|\Gamma_2 \vdash_{\mathcal{RE}_1|\ \mathcal{RE}_{ind}(X+Y \overset{\geqslant}{\to} Y+X, \leqslant_2, T^2_\Sigma)\{X \mapsto s(X_2); Y \mapsto 0\}\ \mathcal{RE}_{ind}(Y+X \overset{\geqslant}{\to} X+Y, \leqslant_2, T^2_\Sigma)\{X \mapsto s(X_2); Y \mapsto 0\}\ \mathcal{RE}_{ind}(X_1 \approx 0+X_1, <, T_\Sigma)\{X_1 \mapsto s(X_2)\}:2} \emptyset[\![0 \leqslant 0; X_2 \leqslant s(X_2)]\!]$$

8. **Elim** vérifie maintenant les inégalités (ici triviales) stockées dans l'ensemble des contraintes et élimine définitivement le sous-but.
9. On est quitte avec le sous-but $s(X_1+Y_1) \approx s(Y_1)+X_1$ et on va voir plus tard comment la recherche de preuve se termine.

2.4 Justification théorique

2.4.1 Le système d'inférence DecIndNarrow

Pour parvenir à démontrer la correction de OptIndNarrow, il est nécessaire d'introduire le système de recherche de preuve intermédiaire DecIndNarrow, obtenu à partir de IndNarrow par l'adjonction des nouvelles règles d'inférence données Figure 2.3. En effet, nous allons voir ci-dessous que la correction de DecIndNarrow est une conséquence directe de celle de IndNarrow, et à la section 2.4.3, que la correction de OptIndNarrow découle de celle de DecIndNarrow. Si on note $\Pi(\mathcal{RE})$ le système de réécriture obtenu en remplaçant chaque règle de réécriture de $\mathcal{RE}$ par l'égalité correspondante, on vérifie aisément le lemme suivant :

Lemme 2.1 Pour tous contextes Γ_1, Γ_2, Γ'_1, Γ'_2, pour tous systèmes de réécriture $\mathcal{RE}_1$, $\mathcal{RE}_2$, $\mathcal{RE}'_1$, $\mathcal{RE}'_2$, si on a :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathsf{DecIndNarrow}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'$$

alors on a :

$$\Gamma_1|\Gamma_2 \vdash_{\Pi(\mathcal{RE}_1)|\Pi(\mathcal{RE}_2)} Q \overset{*}{\rightarrowtail}_{\mathsf{IndNarrow}} \Gamma'_1|\Gamma'_2 \vdash_{\Pi(\mathcal{RE}'_1)|\Pi(\mathcal{RE}'_2)} Q'$$

Corollaire 2.1 Le système de recherche de preuve DecIndNarrow est correct et réfutationnellement complet.

Preuve. D'après la Proposition 2.1 □

Orient$_1$ a $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{\kappa\}|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{l\to r\}|\mathcal{RE}_2} Q$

$(\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $l > r$

Orient$_1$ b $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{\kappa\}|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{l\overset{\gtrless}{\to} r, r\overset{\gtrless}{\to} l\}|\mathcal{RE}_2} Q$

$(\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $l \gtrless r$

Orient$_2$ a $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa\}} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa'\}} Q$

$(((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $\kappa' = l \to r)$ ou

$((\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$ ou $\kappa = \mathcal{RE}_{ind}(r \approx l)\mu)$ et $\kappa' = \mathcal{RE}_{ind}(l \to r)\mu))$ et $l > r$

Orient$_2$ b $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa\}} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa',\kappa''\}} Q$

$(((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $\kappa' = l \overset{\gtrless}{\to} r$ et $\kappa'' = r \overset{\gtrless}{\to} l)$

ou $((\kappa = \mathcal{RE}_{ind}(l \approx r)\mu$ ou $\kappa = \mathcal{RE}_{ind}(r \approx l)\mu)$

et $\kappa' = \mathcal{RE}_{ind}(l \overset{\gtrless}{\to} r)\mu$ et $\kappa'' = \mathcal{RE}_{ind}(r \overset{\gtrless}{\to} l)\mu))$ et $l \gtrless r$

Rewrite$_1$ a $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{\kappa\}|\mathcal{RE}_2} Q[l\sigma]_{i.\omega_i} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{\kappa\}|\mathcal{RE}_2} Q[r\sigma]_{i.\omega_i}$

$\kappa = l \to r$ ou $((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $Q_{|i}[l\sigma]_{\omega_i} > Q_{|i}[r\sigma]_{\omega_i})$

Rewrite$_1$ b $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{\kappa\}|\mathcal{RE}_2} Q[l\sigma]_{i.\omega_i} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1\cup\{\kappa\}|\mathcal{RE}_2} Q[r\sigma]_{i.\omega_i}$

$\kappa = l \overset{\gtrless}{\to} r$ ou $((\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $Q_{|i}[l\sigma]_{\omega_i} \gtrless Q_{|i}[r\sigma]_{\omega_i})$

Rewrite$_2$ a $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa\}} Q[l\sigma]_{i.\omega_i} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa\}} Q[r\sigma]_{i.\omega_i}$

$(\kappa = \kappa_0$ ou $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu)$ et

$(\kappa_0 = l \to r$ ou $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l)$ et

(si $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l$ alors $Q_{|i}[l\sigma]_{\omega_i} > Q_{|i}[r\sigma]_{\omega_i}$) et

(si $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu$ alors $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$ avec $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(l \approx r)}$)

Rewrite$_2$ b $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\kappa\}} Q[l\sigma]_{i.\omega_i} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} Q[r\sigma]_{i.\omega_i}$

$(\kappa = \kappa_0$ ou $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu)$ et

$(\kappa_0 = l \overset{\gtrless}{\to} r$ ou $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l)$ et

(si $\kappa_0 = l \approx r$ ou $\kappa_0 = r \approx l$ alors $Q_{|i}[l\sigma]_{\omega_i} \gtrless Q_{|i}[r\sigma]_{\omega_i}$) et

(si $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu$ alors $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$ avec $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(l \approx r)}$)

FIG. 2.3 – Règles spécifiques de DecIndNarrow

2.4.2 Labels

La question que l'on a envie de poser maintenant est la suivante : dans quelles circonstances est-on autorisé à utiliser une hypothèse de récurrence provenant de l'application de la règle **Induce** à un but Q ? Ici, on va utiliser le théorème de compatibilité principale (théorème 1.7 de la partie II), et pour cela, il est nécessaire de garder une trace de la succession des réductions depuis l'application de la règle **Induce** au but Q. Cela justifie les modifications suivantes :

1. Supposons que la règle **Induce** soit appliquée au but équationnel Q, à la position $\omega = j.\omega_j$, et à l'aide de l'unificateur le plus général σ, alors, au lieu de noter $\mathcal{RE}_{ind}(Q, <)\sigma$ la nouvelle

règle produite par cette application, on va la noter $\mathcal{RE}_{ind}(Q,<)\sigma : j$, pour indiquer dans quel membre du but la réduction s'est effectuée.

2. Pour toute égalité Q, pour toute substitution σ, pour tous $i, j \in \{1,2\}$, on définit l'opérateur L_i sur l'ensemble des règles de réécritures munies d'un label de la façon suivante :
   (a) $L_i(\mathcal{RE}_{ind}(Q,<)\sigma : j) = \mathcal{RE}_{ind}(Q,<)\sigma : j$ si $i = j$;
   (b) $L_i(\mathcal{RE}_{ind}(Q,<)\sigma : j) = \mathcal{RE}_{ind}(Q,\leqslant)\sigma$ si $i \neq j$.

Notation: Pour tout système de réécriture $\mathcal{RE}$, on note $L_i(\mathcal{RE})$ le système de réécriture obtenu par le remplacement de chaque règle de $\mathcal{RE}$ de la forme $\mathcal{RE}_{ind}(Q,<)\sigma : j$ par la règle correspondante $L_i(\mathcal{RE}_{ind}(Q,<)\sigma : j)$.

2.4.3 Correction et complétude réfutationnelle de OptIndNarrow

L'objectif de ce paragraphe est de démontrer la correction et la complétude réfutationnelle de OptIndNarrowà partir de celles de DecIndNarrow.

Pour tout système de réécriture $\mathcal{RE}$, posons $\Pi(\mathcal{RE})$ le système de réécriture obtenu par le remplacement de chaque règle de réécriture de la forme $\mathcal{RE}_{ind}(\kappa_0,<)\mu : j$ ou $\mathcal{RE}_{ind}(\kappa_0,\leqslant)\mu$ par la règle correspondante $\mathcal{RE}_{ind}(\kappa_0,<)\mu$.

Lemme 2.2 Si il existe une dérivation du type suivant :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![I]\!] \overset{*}{\rightarrowtail}_{\mathsf{OptIndNarrow}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'[\![I']\!] \rightarrowtail_{\mathsf{OptIndNarrow}} \Gamma''_1|\Gamma''_2 \vdash_{\mathcal{RE}''_1|\mathcal{RE}''_2} Q''[\![I'']\!]$$

telle que :

1. $\mathcal{RE}_2$ ne contienne pas de règles de la forme : $\mathcal{RE}_{ind}(Q,<)\mu : j$ ou $\mathcal{RE}_{ind}(Q,\leqslant)\mu$;
2. toutes les inégalités dans I'' soient satisfaites ;

alors, il existe une étape de dérivation :

$$\Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\Pi(\mathcal{RE}'_2)} Q' \rightarrowtail_{\mathsf{DecIndNarrow}} \Gamma''_1|\Gamma''_2 \vdash_{\mathcal{RE}''_1|\Pi(\mathcal{RE}''_2)} Q''$$

Preuve. Considérons l'unique cas non trivial :

$$\Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'[\![I']\!] \rightarrowtail_{\mathbf{Rewrite_2}} \Gamma''_1|\Gamma''_2 \vdash_{\mathcal{RE}''_1|\mathcal{RE}''_2} Q''[\![I'']\!]$$

Posons $\kappa \in \mathcal{RE}'_2$ la règle de réécriture requise pour cette étape d'inférence.

Si κ n'est pas de la forme $\mathcal{RE}_{ind}(\kappa_0,<)\mu : j$ ou $\mathcal{RE}_{ind}(\kappa_0,\leqslant)\mu$, la conclusion est immédiate.
Supposons donc qu'Il existe κ_0 et une substitution μ telles que $\kappa = \mathcal{RE}_{ind}(\kappa_0,<)\mu : j$ ou $\kappa = \mathcal{RE}_{ind}(\kappa_0,\leqslant)\mu$ (avec κ_0 de la forme $l \to r$, $l \gtrless$, ou $l \approx r$). Alors, par définition des règles **Rewrite 2** dans le système de recherche de preuve OptIndNarrow :

1. il existe $i \in \{1,2\}$, une position $\omega_i \in \mathcal{D}om(Q'_{|i})$, et une substitution σ, telles que :

$$Q' = Q'[l\sigma]_{i.\omega_i} \quad \text{et} \quad Q'' = Q'[r\sigma]_{i.\omega_i}$$

2. $$I'' = I' \cup (\bigcup_{x \in \mathcal{V}ar(l \approx r)} \{x\sigma \leqslant x\mu\})$$

Pour montrer que l'on a l'étape de DecIndNarrow -dérivation :

$$\Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\Pi(\mathcal{RE}'_2)} Q' \quad \rightarrowtail \mathbf{Rewrite}_2 \quad \Gamma''_1|\Gamma''_2 \vdash_{\mathcal{RE}''_1|\Pi(\mathcal{RE}''_2)} Q''$$

utilisant la règle de réécriture $\Pi(\kappa) = \mathcal{RE}_{ind}(\kappa_0, <)\mu$, il suffit donc de prouver l'inégalité $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$, avec $\overrightarrow{x} \in \mathcal{X}^n$ défini par $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(l \approx r)}$.
Comme $(\bigcup_{x \in \mathcal{V}ar(l \approx r)} \{x\sigma \leqslant x\mu\}) \subset I''$ d'après 2 ci-dessus, et comme toutes les inégalités de I'' sont satisfaites (hypothèse 2), on a $\overrightarrow{x}\sigma \leqslant_n \overrightarrow{x}\mu$.
Donc, d'après le lemme 1.2 du chapitre 1 de la partie II, pour montrer la condition $\overrightarrow{x}\sigma <_n \overrightarrow{x}\mu$, il suffira de vérifier l'inégalité :

$$(l \approx r)\sigma <_e (l \approx r)\mu \tag{2.1}$$

Remarquons maintenant que, d'après l'hypothèse 1, la règle de réécriture κ a été produite par une application de **Induce**.
Par l'hypothèse 2, on a la dérivation ci-dessous :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![I]\!] \quad \overset{*}{\rightarrowtail}_{\mathsf{OptIndNarrow}} \quad \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'[\![I']\!]$$

Notons q sa longueur, et, pour tout entier k tel que $0 \leq k \leq q-1$, posons :

$${\Gamma_1}^k|{\Gamma_2}^k \vdash_{{\mathcal{RE}_1}^k|{\mathcal{RE}_2}^k} Q^k[\![I^k]\!] \quad \rightarrowtail \quad {\Gamma_1}^{k+1}|{\Gamma_2}^{k+1} \vdash_{{\mathcal{RE}_1}^{k+1}|{\mathcal{RE}_2}^{k+1}} Q^{k+1}[\![I^{k+1}]\!]$$

l'étape de rang k.
Supposons que :

1. $i_1, i_2, \ldots, i_m$ sont les rangs des étapes dans cette dérivation D_n correspondant à une application de la règle d'inférence **Induce**
2. κ est produite par l'application de la règle d'inférence **Induce** à l'étape de rang i_k, où elle est appliquée à l'occurence $j.\omega_j$, et en utilisant la règle de réécriture $g \to d \in {\mathcal{RE}_1}^{i_k}$.

Considérons maintenant les points suivants :
- la règle $\mathcal{RE}_{ind}(Q_{i_k}, <)\sigma_{i_k} : j$ est introduite à l'étape de rang i_k par l'application de **Induce** ;
- à la prochaine application de **Induce**, donc à l'étape de rang i_{k+1}, $\mathcal{RE}_{ind}(Q_{i_k}, <)\sigma_{i_k} : j$ est remplacée par $\mathcal{RE}_{ind}(Q_{i_k}, \leqslant)\sigma_{i_k}\sigma_{i_{k+1}}$ ou $\mathcal{RE}_{ind}(Q_{i_k}, <)\sigma_{i_k}\sigma_{i_{k+1}} : j$;
- à l'étape de rang i_m, celle-ci devient $\mathcal{RE}_{ind}(Q_{i_k}, \leqslant)\sigma_{i_k} \ldots \sigma_{i_m}$ ou $\mathcal{RE}_{ind}(Q_{i_k}, <)\sigma_{i_k} \ldots \sigma_{i_m} : j$;
- à l'étape de rang q, elle est encore de l'une ou de l'autre de ces formes, et c'est notre règle κ.

On en déduit :

1. $Q_{i_k} = l \approx r$ ou $Q_{i_k} = r \approx l$;
2. $\mu = \sigma_{i_k} \ldots \sigma_{i_m}$.

Maintenant, posons R_1, R_2, R_3, R_4 les égalités suivantes :

$$R_1 = Q_{i_k}\mu \qquad R_2 = (Q_{i_k}[d]_{j.\omega_j})\mu \qquad R_3 = Q' \qquad R_4 = Q''$$

Nous allons démontrer les quatre points suivants :

$$1.\ R_1 \rightarrow_{[g \to d, j.\omega_j, \mu]} R_2 \qquad 2.\ R_3 \leq_e R_2 \qquad 3.\ R_3 \rightarrow_{[l \approx r, i.\omega_i, \sigma]} R_4 \qquad 4.\ R_4 \leqslant_e R_3 \tag{2.2}$$

Preuve. 1. par hypothèse

2. Considérons les points suivants :
   - $Q_{i_k}[d]_{j.\omega_j}\sigma_{i_k} \geq_e Q_{i_{k+1}}$, car chaque application d'une règle (autre que **Induce**) fait décroître (au sens large) le membre auquel elle est appliquée ;
   - $Q_{i_{k+1}}\sigma_{i_{k+1}} \geq_e Q_{i_{k+2}}$, car une surréduction a été appliquée à l'égalité $Q_{i_{k+1}}$ à l'aide de l'unificateur le plus général $\sigma_{i_{k+1}}$ (on devrait mettre en fait un symbole d'inégalité stricte, mais ce n'est pas nécessaire ici) ;
   - donc $Q_{i_k}[d]_{j.\omega_j}\sigma_{i_k}\sigma_{i_{k+1}} \geq_e Q_{i_{k+2}}$, d'après les deux inégalités précédentes, et du fait que la relation $\leq_e$ est stable par substitution ;
   - $Q_{i_{k+2}}\sigma_{i_{k+2}} \geq_e Q_{i_{k+3}}$;
   - $Q_{i_k}[d]_{j.\omega_j}\sigma_{i_k}\sigma_{i_{k+1}}\sigma_{i_{k+2}} \geq_e Q_{i_{k+3}}$;

   $\vdots$
   - $Q_{i_k}[d]_{j.\omega_j}\sigma_{i_k}\sigma_{i_{k+1}}\sigma_{i_{k+2}}\ldots\sigma_{i_{s-1}} \geq_e Q_{i_m}$;
   - $Q_{i_m}\sigma_{i_m} \geq_e Q'$;
   - $Q_{i_k}[d]_{j.\omega_j}\sigma_{i_k}\sigma_{i_{k+1}}\sigma_{i_{k+2}}\ldots\sigma_{i_{s-1}}\sigma_{i_s} \geq_e Q'$.
3. Par hypothèse.
4. Car on $Q''_{|i} \leq_e Q'_{|i}$ et d'après le point 1 du lemme 1.2 du chapitre 1 de la partie II.

□

Nous retrouvons ici exactement les hypothèses du théorème de compatibilité principale (théorème 1.7 de la partie II). On peut donc distinguer deux cas :

Cas 1 : $\omega_i \neq \varepsilon$ ou $i = j$, alors, d'après le théorème , $(l \approx r)\sigma <_e R_1$, et, du fait que $R_1 = (l \approx r)\mu$ ou $R_1 = (r \approx l)\mu$, on obtient $(l \approx r)\sigma <_e (l \approx r)\mu$. L'inégalité 2.1 est ainsi démontrée, et on a vu que cela suffit pour justifier l'étape d'inférence :

$$\Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\Pi(\mathcal{RE}'_2)} Q' \quad \rightarrowtail_{\mathsf{DecIndNarrow}} \quad \Gamma''_1|\Gamma''_2 \vdash_{\mathcal{RE}''_1|\Pi(\mathcal{RE}''_2)} Q''$$

Cas 2 : $\omega_i = \varepsilon$ et $i \neq j$

Sous-cas 1 : $\kappa = \mathcal{RE}_{ind}(\kappa_0, <)\mu : j$

Par définition des règles **Rewrite**$_2$ dans le système de recherche de preuve OptIndNarrow, on a $\kappa_0\sigma <_e \kappa_0\mu$, donc, $(l \approx r)\sigma <_e (l \approx r)\mu$, et c'est terminé.

Sous-cas 2 : $\kappa = \mathcal{RE}_{ind}(\kappa_0, \leqslant)\mu$

Soit $\underline{j}$ tel que :

$$\underline{j} \in \{1, 2\} \text{ et } \underline{j} \neq j \tag{2.3}$$

En considérant le système de recherche de preuve OptIndNarrow on conclut qu'une règle d'inférence **Rewrite** a ou **Induce** a été utilisée pour réduire un but Q_p, avec p tel que $i_k < p < q$, et que la réduction s'est effectuée dans le membre $\underline{j}$ de ce but.

Sous-sous-cas 1 : **Induce** est appliquée au but Q_p

Alors il existe un entier s tel que $k < s \leq m$ et $p = i_s$.

Considérons les points suivants :
- $Q_{i_k}\sigma_{i_k} \geq_e Q_{i_{k+1}}$
- $Q_{i_{k+1}}\sigma_{i_{k+1}} \geq_e Q_{i_{k+2}}$
- $Q_{i_k}\sigma_{i_k}\sigma_{i_{k+1}} \geq_e Q_{i_{k+2}}$ d'après les deux premiers points, et puisque $\leq_e$ est stable par substitution.

  $\vdots$
-
$$Q_{i_k}\sigma_{i_k}\ldots\sigma_{i_{s-1}} \geq_e Q_{i_s} \tag{2.4}$$

Considérons maintenant les égalités suivantes :

$$R'_1 = Q_{i_s}\sigma_{i_s}\ldots\sigma_{i_m} \qquad R'_2 = Q_{i_s+1}\sigma_{i_{s+1}}\ldots\sigma_{i_m} \qquad R'_3 = Q' \qquad R'_4 = Q''$$

Par un raisonement analogue à celui effectué pour montrer (2.2), on vérifie qu'il existe une règle de réécriture $l' \to r' \in \mathcal{RE}_1$ et une substitution σ', telles que les affirmations suivantes soient correctes :

1. $R'_1 \to_{[l'\to r',\ \underline{j}.\omega_{\underline{j}},\ \sigma']} R'_2$ 2. $R'_2 \geq_e R'_3$ 3. $R'_3 \to_{[\kappa_0,\ i.\omega_i,\ \sigma]} R'_4$ 4. $R'_4 \leqslant_e R'_3$

Puisque $i,\ j,\ \underline{j} \in \{1,2\}$, $j \neq \underline{j}$, et $i \neq j$, on a :

$$i = \underline{j} \tag{2.5}$$

Donc, d'après le théorème 1.7, on obtient $\kappa_0\sigma <_e R'_1$, soit $(l \approx r)\sigma <_e R'_1$. Ce qui s'écrit encore, avec nos notations :

$$(l \approx r)\sigma <_e Q_{i_s}\sigma_{i_s}\ldots\sigma_{i_m} \tag{2.6}$$

et, du fait que $Q_{i_s} \leq_e Q_{i_k}\sigma_{i_k}\ldots\sigma_{i_{s-1}}$ (2.4), combiné avec (2.6), sachant que $\leq_e$ est stable par substitution et $\leq_e \subset \leqslant_e$, on a :

$$(l \approx r)\sigma <_e Q_{i_k}\sigma_{i_k}\ldots\sigma_{i_m}$$

et enfin, puisque $\mu = \sigma_{i_k}\ldots\sigma_{i_m}$ et $Q_{i_k} = l \approx r$, cela nous conduit à l'inégalité suivante :

$$(l \approx r)\sigma <_e (l \approx r)\mu$$

Sous-sous-cas 2 : **Rewrite**$_1$ **a** ou **Rewrite**$_2$ **a** est appliquée au but Q_p

Alors il existe un entier s tel que $k \leq s \leq m$ et $i_s < p < i_{s+1}$ (en posant $i_{m+1} = q$).

Premièrement, on a :

$$Q_{i_s} \leq_e Q_{i_k}\sigma_{i_k}\ldots\sigma_{i_{s-1}} \tag{2.7}$$

Introduisons les notations suivantes :

$$R''_1 = Q_p\sigma_{i_{s+1}}\ldots\sigma_{i_m} \qquad R''_2 = Q_{p+1}\sigma_{i_{s+1}}\ldots\sigma_{i_m} \qquad R''_3 = Q' \qquad R''_4 = Q''$$

Il existe une règle de réécriture $l'' \to r''$ ou une égalité $l'' \approx r''$, et une substitution σ'' telles que les conditions suivantes soient satisfaites :

1. $R''_1 \to_{[l''\to r'',\ \underline{j}.\omega_{\underline{j}},\ \sigma'']} R''_2$ 2. $R''_2 \geq_e R''_3$ 3. $R''_3 \to_{[\kappa_0, i.\omega_i, \sigma]} R''_4$ 4. $R''_4 \leqslant_e R''_3$

Puisque $i,\ j,\ \underline{j} \in \{1,2\}$, $j \neq \underline{j}$, et comme $i \neq j$, on a :

$$i = \underline{j} \tag{2.8}$$

Donc, d'après le lemme 1.7, on obtient $\kappa_0\sigma <_e R''_1$.
Ainsi, $(l \approx r)\sigma <_e R''_1$, donc, par définition :

$$(l \approx r)\sigma <_e Q_p\sigma_{i_{s+1}} \dots \sigma_{i_m} \tag{2.9}$$

Mais comme $i_s < p < i_{s+1}$, on a $Q_p \leq_e Q_{i_s}\sigma_{i_s}$, et comme $\leq_e$ est stable par substitution, cela entraîne $Q_p\sigma_{i_{s+1}} \dots \sigma_{i_m} \leq_e Q_{i_s}\sigma_{i_s}\sigma_{i_{s+1}} \dots \sigma_{i_m}$. Mais comme $Q_{i_s} \leq_e Q_{i_k}\sigma_{i_k} \dots \sigma_{i_{s-1}}$ (d'après (2.7)), on en déduit l'inégalité $Q_p\sigma_{i_{s+1}} \dots \sigma_{i_m} \leq_e Q_{i_k}\sigma_{i_k} \dots \sigma_{i_m}$ (stabilité par substitution et transitivité de $\leq_e$). En combinant avec l'inégalité (2.9), et du fait de l'inclusion $\leq_e \subset \leqslant_e$, on obtient finalement $(l \approx r)\sigma <_e Q_{i_k}\sigma_{i_k} \dots \sigma_{i_m}$, ce qui est exactement $(l \approx r)\sigma < (l \approx r)\mu$. □

Corollaire 2.2 Si il existe une dérivation du type suivant :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![I]\!] \quad \overset{*}{\rightarrowtail}_{\mathsf{OptIndNarrow}} \quad \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'[\![I']\!]$$

telle que :

1. $\mathcal{RE}_2$ ne contienne pas de règles de la forme : $\mathcal{RE}_{ind}(Q,<)\mu : j$ ou $\mathcal{RE}_{ind}(Q,\leqslant)\mu$;
2. toutes les inégalités dans I' soient satisfaites ;

alors, il existe une dérivation :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \quad \overset{*}{\rightarrowtail}_{\mathsf{DecIndNarrow}} \quad \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\Pi(\mathcal{RE}'_2)} Q'$$

Preuve. Par récurrence sur la longueur de la dérivation, et grâce au lemme 2.2 □

Corollaire 2.3 S'il existe une dérivation du type :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![\emptyset]\!] \quad \overset{*}{\rightarrowtail}_{\mathsf{OptIndNarrow}} \quad \diamond$$

et telle que $\mathcal{RE}_2$ ne contienne pas de règles de la forme : $\mathcal{RE}_{ind}(Q,<)\mu : j$ ou $\mathcal{RE}_{ind}(Q,\leqslant)\mu$, alors, si $\overrightarrow{x} \in \mathcal{X}^n$ désigne le vecteur des variables libres de $\mathcal{RE}_2 \cup \{Q\}$, le séquent $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ admet une preuve dans le calcul des séquents modulo.

Preuve. La dérivation peut se décomposer en :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![\emptyset]\!] \quad \overset{*}{\rightarrowtail}_{\mathsf{OptIndNarrow}} \quad \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} t' \approx t'[\![I']\!] \rightarrowtail_{\mathbf{Trivial}} \diamond[\![I']\!] \quad \overset{*}{\rightarrowtail}_{\mathbf{Elim}} \quad \diamond$$

Comme la règle **Elim** est appliquée avec succès, toutes les inégalités de I' sont satisfaites. D'après le corollaire 2.2, on en déduit qu'il existe une dérivation :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \quad \overset{*}{\rightarrowtail}_{\mathsf{DecIndNarrow}} \quad \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\Pi(\mathcal{RE}'_2)} t' \approx t' \rightarrowtail_{\mathsf{DecIndNarrow}} \diamond$$

et le résultat est démontré, d'après le corollaire 2.1. □

Corollaire 2.4 On reprend les hypothèses du paragraphe 1.5. S'il existe une dérivation du type :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![\emptyset]\!] \overset{*}{\rightarrowtail}_{\textbf{Refutation}} \text{Refutation}$$

et telle que $\mathcal{RE}_2$ ne contienne pas de règles de la forme : $\mathcal{RE}_{ind}(Q,<)\mu : j$ ou $\mathcal{RE}_{ind}(Q,\leqslant)\mu$, alors, si $\overrightarrow{x} \in \mathcal{X}^n$ désigne le vecteur des variables libres de $\mathcal{RE}_2 \cup \{Q\}$, le séquent $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ n'admet pas de preuve dans le calcul des séquents modulo.

Preuve. La dérivation peut se décomposer en :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q[\![\emptyset]\!] \overset{*}{\rightarrowtail}_{\mathsf{OptIndNarrow}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'[\![\emptyset]\!] \rightarrowtail_{\textbf{Refutation}} \text{Refutation}$$

D'après le corollaire 2.2, on en déduit qu'il existe une dérivation :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \overset{*}{\rightarrowtail}_{\mathsf{DecIndNarrow}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\Pi(\mathcal{RE}'_2)} Q' \rightarrowtail_{\textbf{Refutation}} \text{Refutation}$$

et le résultat est démontré, d'après le corollaire 2.1. □

2.5 Exemple

Nous effectuons ici une preuve complète de la commutativité de l'addition avec notre système.

	$x+0 \approx x, x+s(y) \approx s(x+y) \vert \Gamma_2 \vdash X+Y \approx Y+X$	
$\rightarrowtail$	$x+0 \approx x \vert \Gamma_2 \vdash_{x+s(y)\approx s(x+y)} X+Y \approx Y+X$	**Push**$_1$
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{x+0\approx x, x+s(y)\approx s(x+y)} X+Y \approx Y+X$	**Push**$_1$
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{x+0\approx x, x+s(y)\to s(x+y)} X+Y \approx Y+X$	**Orient**$_1$ **a**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1} X+Y \approx Y+X$	**Orient**$_1$ **a**

Avec : $\mathcal{RE}_1 = \{x+0 \to x, x+s(y) \to s(x+y)\}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_1:1\}} X_1 \approx 0+X_1$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2:1\}} s(X_1+Y_1) \approx s(Y_1)+X_1$	**Induce**

Avec : $\begin{cases} \kappa_1 = & (\underline{x},\underline{y}) <_2 (X_1,0) \Rightarrow \underline{x}+\underline{y} \approx \underline{y}+\underline{x} \\ \kappa_2 = & (\underline{x},\underline{y}) <_2 (X_1,s(Y_1)) \Rightarrow \underline{x}+\underline{y} \approx \underline{y}+\underline{x} \end{cases}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_3,\kappa_4:2\}} 0 \approx 0$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_5,\kappa_6:2\}} s(X_2) \approx s(0+X_2)$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2:1\}} s(X_1+Y_1) \approx s(Y_1)+X_1$	**Induce**

Avec : $\begin{cases} \kappa_3 = (\underline{x},\underline{y}) \leqslant_2 (0,0) \Rightarrow \underline{x}+\underline{y} \approx \underline{y}+\underline{x} \\ \kappa_4 = \underline{x} < 0 \Rightarrow \underline{x} \approx 0+\underline{x} \\ \kappa_5 = (\underline{x},\underline{y}) \leqslant_2 (s(X_2),0) \Rightarrow \underline{x}+\underline{y} \approx \underline{y}+\underline{x} \\ \kappa_6 = \underline{x} < s(X_2) \Rightarrow \underline{x} \approx 0+\underline{x} \end{cases}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_5,\kappa_6:2\}} s(X_2) \approx s(0+X_2)$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2:1\}} s(X_1+Y_1) \approx s(Y_1)+X_1$	**Trivial**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_7,\kappa_8,\kappa_6:2\}} s(X_2) \approx s(0+X_2)$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2:1\}} s(X_1+Y_1) \approx s(Y_1)+X_1$	**Orient**$_2$ **b**

Avec : $\begin{cases} \kappa_7 = (\underline{x}, \underline{y}) \leqslant_2 (s(X_2), 0) \Rightarrow \underline{x} + \underline{y} \xrightarrow{\gtrless} \underline{y} + \underline{x} \\ \kappa_8 = (\underline{x}, \underline{y}) \leqslant_2 (s(X_2), 0) \Rightarrow \underline{y} + \underline{x} \xrightarrow{\gtrless} \underline{x} + \underline{y} \end{cases}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_7, \kappa_8, \kappa_6 : 2\}} s(X_2) \approx s(X_2 + 0) [\![I^{(1)}]\!]$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2 : 1\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$	**Rewrite$_2$ b**

Avec : $I^{(1)} = \{0 \leq 0, X_2 \leq s(X_2)\}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_7, \kappa_8, \kappa_6 : 2\}} s(X_2) \approx s(X_2) [\![I^{(1)}]\!]$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2 : 1\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$	**Rewrite$_1$ a**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_7, \kappa_8, \kappa_6 : 2\}} \diamond [\![I^{(1)}]\!]$ $\vert \emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2 : 1\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$	**Trivial**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_2 : 1\}} s(X_1 + Y_1) \approx s(Y_1) + X_1$	**Elim**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_9, \kappa_{10} : 2\}} s(0 + Y_3) \approx s(Y_3)$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{11}, \kappa_{12} : 2\}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$	**Induce**

Avec : $\begin{cases} \kappa_9 = (\underline{x}, \underline{y}) \leqslant_2 (0, s(Y_3)) \Rightarrow \underline{x} + \underline{y} \approx \underline{y} + \underline{x} \\ \kappa_{10} = (\underline{x}, \underline{y}) <_2 (0, Y_3) \Rightarrow s(\underline{x} + \underline{y}) \approx s(\underline{y}) + \underline{x} \\ \kappa_{11} = (\underline{x}, \underline{y}) \leqslant_2 (s(X_3), s(Y_3)) \Rightarrow \underline{x} + \underline{y} \approx \underline{y} + \underline{x} \\ \kappa_{12} = (\underline{x}, \underline{y}) <_2 (s(X_3), Y_3) \Rightarrow s(\underline{x} + \underline{y}) \approx s(\underline{y}) + \underline{x} \end{cases}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{13}, \kappa_{14}, \kappa_{10} : 2\}} s(0 + Y_3) \approx s(Y_3)$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{11}, \kappa_{12} : 2\}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$	**Orient$_2$b**

Avec : $\begin{cases} \kappa_{13} = (\underline{x}, \underline{y}) \leqslant_2 (0, s(Y_3)) \Rightarrow \underline{x} + \underline{y} \xrightarrow{\gtrless} \underline{y} + \underline{x} \\ \kappa_{14} = (\underline{x}, \underline{y}) \leqslant_2 (0, s(Y_3)) \Rightarrow \underline{y} + \underline{x} \xrightarrow{\gtrless} \underline{x} + \underline{y} \end{cases}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{13}, \kappa_{14}, \kappa_{10} : 2\}} s(Y_3 + 0) \approx s(Y_3) [\![I^{(2)}]\!]$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{11}, \kappa_{12} : 2\}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$	**Rewrite$_2$ b**

Avec : $I^{(2)} = \{0 \leq 0, Y_3 \leq s(Y_3)\}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{13}, \kappa_{14}, \kappa_{15}\}} s(Y_3) \approx s(Y_3) [\![I^{(2)}]\!]$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{11}, \kappa_{12} : 2\}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$	**Rewrite$_1$a**

Avec : $\kappa_{15} = (\underline{x}, \underline{y}) \leqslant_2 (0, Y_3) \Rightarrow s(\underline{x} + \underline{y}) \approx s(\underline{y}) + \underline{x}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{13}, \kappa_{14}, \kappa_{15}\}} \diamond [\![I^{(2)}]\!]$ $\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{11}, \kappa_{12} : 2\}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$	**Trivial**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{11}, \kappa_{12} : 2\}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$	**Elim**

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{16}, \kappa_{17}, \kappa_{12} : 2\}} s(s(X_3) + Y_3) \approx s(s(Y_3) + X_3)$	**Orient$_2$ b**

Avec $\begin{cases} \kappa_{16} = (\underline{x}, \underline{y}) \leqslant_2 (s(X_3), s(Y_3)) \Rightarrow \underline{x} + \underline{y} \xrightarrow{\gtrless} \underline{y} + \underline{x} \\ \kappa_{17} = (\underline{x}, \underline{y}) \leqslant_2 (s(X_3), s(Y_3)) \Rightarrow \underline{y} + \underline{x} \xrightarrow{\gtrless} \underline{x} + \underline{y} \end{cases}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{16}, \kappa_{17}, \kappa_{12} : 2\}} s(Y_3 + s(X_3)) \approx s(s(Y_3) + X_3) [\![I^{(3)}]\!]$	**Rewrite$_2$ b**

Avec $I^{(3)} = \{s(X_3) \leq s(X_3); Y_3 \leq s(Y_3)\}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{16},\kappa_{17},\kappa_{18}\}} s(s(Y_3 + X_3)) \approx s(s(Y_3) + X_3)[\![I^{(3)}]\!]$	**Rewrite**$_1$ **a**

Avec : $\kappa_{18} = (\underline{x}, \underline{y}) \leqslant_2 (s(X_3), Y_3) \Rightarrow s(\underline{x} + \underline{y}) \approx s(\underline{y}) + \underline{x}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{16},\kappa_{17},\kappa_{18}\}} s(s(Y_3 + X_3)) \approx s(X_3 + s(Y_3))[\![I^{(4)}]\!]$	**Rewrite**$_2$ **b**

Avec : $I^{(4)} = \{s(X_3) \leq s(X_3); Y_3 \leq s(Y_3);\ s(Y_3) \leq s(Y_3); X_3 \leq s(X_3)\}$

$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{16},\kappa_{17},\kappa_{18}\}} s(s(Y_3 + X_3)) \approx s(s(X_3 + Y_3))[\![I^{(4)}]\!]$	**Rewrite**$_1$ **a**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{16},\kappa_{17},\kappa_{18}\}} s(s(X_3 + Y_3)) \approx s(s(X_3 + Y_3))[\![I^{(4)}]\!]$	**Rewrite**$_2$ **b**
$\rightarrowtail$	$\emptyset \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\kappa_{16},\kappa_{17},\kappa_{18}\}} \diamond[\![I^{(4)}]\!]$	**Trivial**
$\rightarrowtail$	$\diamond$	**Elim**

Conclusion

Nous venons d'observer comment la surréduction est capable de fournir le mécanisme d'inférence qui permet de réaliser une recherche de preuve par récurrence. Au lieu de définir au préalable des schémas et des variables de récurrence, cette méthode a l'avantage de cibler exactement les variables à instancier, et comment. En outre, du fait que la méthode est issue directement du cadre de la déduction modulo, on a l'avantage de pouvoir directement traduire une recherche de preuve réalisée avec succès par notre système en une dérivation du calcul des séquents modulo. En ce qui concerne la preuve, le cadre général de la déduction modulo est conçu pour permettre de déléguer toutes les étapes de calcul sur les propositions ou les termes à des démonstrateurs spécialisés utilisant des techniques équationnelles et de réécriture, et de les séparer ainsi des étapes de déduction comme le modus ponens par exemple. Ainsi, certaines parties de la preuve peuvent être réalisées par des calculs parallèles, pendant que le véritable squelette de la preuve est construit au fur et à mesure. En ce qui concerne la vérification, les expériences décrites dans [DEP 03] de traduction de preuves équationnelles et par récurrence en termes de preuve pour Coq devront s'avérer pertinentes. Si l'approche est fructueuse sur le plan théorique et souligne la relation entre les méthodes de recherche de preuve par récurrence fondées sur la réécriture et la récurrence noethérienne, nous sommes clairement à la recherche d'une implémentation des résultats présentés ici. Notre but sera de réaliser cela en permettant de mécaniser la recherche de preuve dans un assistant qui utiliserait à la fois la théorie des types et le calcul de réécriture [BAR 03, WAC 05]. En outre, notre approche donne la possibilité d'utiliser un principe de récurrence utilisant l'ordre induit par des systèmes de réécriture noethériens, donc beaucoup plus performant que le principe de récurrence structurelle qui est, en pratique, utilisé dans la plupart des assistants de preuve habituels. Cette approche fondée sur la surréduction est la porte ouverte à de nouvelles questions fondamentales. Citons-en trois. La première concerne sa relation avec la stratégie très utile que l'on appelle rippling [BUN 05]. En effet, d'une manière qui s'apparente à cette stratégie, la surréduction rend explicite et relie à un système de réécriture noethérien ce dont on doit se servir pour démontrer un but par récurrence. Cette analogie devrait être étudiée de manière approfondie et peut-être être exploitée. La seconde, que nous développons dans la dernière partie de cette thèse, concerne la possibilité d'étendre la recherche de preuve par récurrence fondée sur la réécriture développée jusqu'à présent à la réécriture de classe. Cela a déjà été exploré en particulier dans [BER 95] pour les théories associatives commutatives. Le caractère générique de la surréduction modulo peut faciliter l'usage de tels systèmes de réécriture de classe pour la recherche de preuve par récurrence. La troisième concerne les preuves par consistance, qui sont en fait à la source de l'usage des techniques de réécriture pour la récurrence [MUS 80b, GAN 92, COM 00, STE]. La relation entre la déduction modulo et la démarche utilisée dans les preuves par consistance mérite d'être mieux comprise.

Quatrième partie

Extension au cas de la réécriture modulo

1

Un système de recherche de preuve pour la récurrence modulo

Sommaire

1.1 Le système de recherche de preuve IndNarrowMod 127
1.2 Correction 128
1.3 Correction réfutationnelle 131
1.4 Complétude réfutationnelle 133
1.5 Exemple 133

Nous étendons à présent notre système de recherche de preuve dans l'intention de pouvoir réaliser des preuves par récurrence dans des théories contenant des axiomes non orientables, par exemple des axiomes commutatifs. Le système de recherche de preuve IndNarrowMod pour les preuves par récurrence introduit dans cette section est fondé sur la surréduction modulo. Avec ce système, l'étape de récurrence est de nouveau réalisée par surréduction au niveau d'une position *définie-maximale*. En outre, si on suppose que la théorie est axiomatisée par un système de réécriture équationnel $(\mathcal{R}, E)$ terminant sur les termes clos, suffisamment complet, et dont le système d'égalités E préserve les termes constructeurs, les substitutions surréductrices appartiennent à un ensemble complet d'unificateurs constructeurs. Nous avons déjà observé au chapitre 2 que ces ensembles peuvent être très faciles à déterminer. La correction et la complétude réfutationelle de notre système sont validées par le calcul des séquents modulo.

1.1 Le système de recherche de preuve IndNarrowMod

On fait les mêmes hypothèses qu'à la section 1 de la partie III, et on suppose également que :

1. E est une théorie équationnelle présente dans Γ_1 ;
2. $\mathcal{RE}_1$ est terminant sur les termes clos et suffisamment complet modulo E par rapport à un ensemble de constructeurs libres ;
3. $(\mathcal{R}, E)$ est compatible avec $\leqslant$ (voir définition 1.4 de la partie II).

La règle principale est bien-sûr de nouveau **Induce**. Elle utilise la surréduction modulo E pour choisir à la fois les variables de récurrence et le schéma d'instanciation. La surréduction est appliquée de nouveau uniquement au niveau de positions *définie-maximales* du but courant Q. Les autres règles effectuent les tâches suivantes : **Trivial** élimine une égalité entre deux termes

Induce	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q \rightarrowtail$
	$\bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma' \in CSUC_E(Q'_{\vert \omega'}, l)}} \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} (Q'[r]_{\omega'})\sigma'$
	si $Q' =_E Q$ et $\omega' \in \mathcal{DM}(Q')$
Orient	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\} \vert \mathcal{RE}_2} Q \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \to r\} \vert \mathcal{RE}_2} Q$
	si $(\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $l > r$
Push$_1$	$\Gamma_1, l \approx r \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \approx r\} \vert \mathcal{RE}_2} Q$
Push$_2$	$\Gamma_1 \vert \Gamma_2, l \approx r \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2 \cup \{l \approx r\}} Q$
Rewrite$_1$	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q'$
	si $Q \to_{\mathcal{RE}_1/E} Q'$
Rewrite$_2$	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q \rightarrowtail \Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q'$
	si $Q \to_{\mathcal{RE}_2/E} Q'$
Trivial	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} t \approx t' \rightarrowtail \diamond$
	si $t =_E t'$
Refutation	$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q \rightarrowtail$ Refutation
	si aucune autre règle ne peut s'appliquer

FIG. 1.1 – Le système de recherche de preuve IndNarrowMod

E-équivalents, **Push** pousse une hypothèse équationnelle de la partie déduction vers la partie calcul, **Orient** oriente une égalité de la partie calcul pour la transformer en règle de réécriture, suivant le quasi-ordre $\leqslant$ qui aura été choisi, **Rewrite** (1 ou 2) réécrit modulo E-équivalence à l'aide d'une règle, d'une égalité, ou d'une instance plus petite d'un but précédent. **Push** et **Rewrite** sont dupliquées du fait de la distinction $\Gamma_1/\mathcal{RE}_1$ et $\Gamma_2/\mathcal{RE}_2$.

1.2 Correction

Montrer la correction revient à prouver que, pour chaque règle de réécriture du système de recherche de preuve IndNarrowMod de la forme :

$$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q \rightarrowtail \bullet_{i \in I} \Gamma_1^i \vert \Gamma_2^i \vdash_{\mathcal{RE}_1^i \vert \mathcal{RE}_2^i} Q^i$$

alors $\Gamma_1 \cup \Gamma_2, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ est dérivable dans le calcul des séquents modulo, à partir du moment où tous les $\Gamma_1^i \cup \Gamma_2^i, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2^i \cup \{Q^i\})} \in \mathcal{T}(\Sigma)^{n^i} \vdash_{\mathcal{RE}_1^i \cup \mathcal{RE}_2^i} Q^i$ le sont.

Théorème 1.1 *Si :*

1. **Induce** *est appliquée sur*
   $\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_2} Q$

pour obtenir :

- $\underset{\sigma' \in CSUC_E(Q'_{|\omega'}, l)}{l \to r \in \mathcal{RE}_1}$ $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} (Q'[r]_{\omega'})\sigma'$ *(rappel : $\omega' \in \mathcal{DM}(Q')$, $Q' =_E Q$) ;*

2. *$<$ est noethérien sur $\mathcal{T}(\Sigma)$;*

3. *chaque séquent* $\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_{\sigma'} \in \mathcal{T}(\Sigma)^{n_{\sigma'}} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} (Q'[r]_{\omega'})\sigma'$ *est dérivable dans le calcul des séquents modulo, avec $\overrightarrow{x}_{\sigma'} \in \mathcal{X}^{n_{\sigma'}}$ le vecteur des variables libres de $\mathcal{RE}_2\sigma' \cup \{Q\sigma'\}$;*

alors, si $\overrightarrow{x} \in \mathcal{X}^n$ désigne le vecteur des variables libres de $\mathcal{RE}_2 \cup \{Q\}$, le séquent $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ est dérivable dans le calcul des séquents modulo.

Preuve. Tout d'abord, comme E préserve les termes constructeurs, E est régulière, donc :

$$\mathcal{V}ar(Q) = \mathcal{V}ar(Q') \tag{1.1}$$

Soit α une substitution close telle que $\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\}) \subseteq \mathcal{D}om(\alpha)$, $\alpha \downarrow$ l'une de ses $\mathcal{RE}_1, E$-formes normales, et $V \subset \mathcal{X}$ tel que $\mathcal{D}om(\alpha) \subseteq V$.
$\omega' \in \mathcal{DM}(Q')$, donc Q' est $\mathcal{R}, E$-inductivement réductible à la position ω' (lemme 2.1 de la partie I). D'après la proposition 2.3 de la partie II, on en déduit qu'il existe une règle de réécriture $l \to r$, et deux substitutions σ'', μ'', telles que $\sigma'' \in CSU_E(Q'_{|\omega'}, l)$, $Im(\sigma'') \subset \mathcal{T}(\mathcal{C}, \mathcal{X})$, et :

$$\sigma''\mu'' =_E \alpha \downarrow [V] \tag{1.2}$$

Maintenant, comme σ'' est un unificateur constructeur de $Q'_{|\omega'}$ et de l, il existe deux substitutions σ' et θ', telles que $\sigma' \in CSUC_E(Q'_{|\omega'}, l)$ et

$$\sigma'' =_E \sigma'\theta'[\mathcal{V}ar(Q'_{|\omega'}) \cup \mathcal{V}ar(l)] \tag{1.3}$$

Observons cependant que l'on peut supposer $\mathcal{R}an(\sigma') \cap V = \emptyset$, donc, en posant $\mu = \theta'_{|\mathcal{R}an(\sigma')} + \sigma''_{|V}$, on déduit aisément de (1.3) que l'on a $\sigma'' =_E \sigma'\mu[V]$. En remplaçant dans l'égalité (1.2), et en posant $\mu' = \mu\mu''$, on obtient :

$$\sigma'\mu' =_E \alpha \downarrow [V] \tag{1.4}$$

Considérons maintenant les dérivations suivantes.

Π_1

$$\dfrac{\dfrac{\dfrac{}{x\alpha \downarrow\approx x\sigma'\mu' \vdash_{\mathcal{RE}_1 \cup E} x\alpha \approx x\sigma'\mu'}\ Ax}{E, x\alpha \downarrow\approx x\sigma'\mu' \vdash_{\mathcal{RE}_1} x\alpha \approx x\sigma'\mu'}\ pop}{\Gamma_1 \cup \Gamma_2, x \in V, x\alpha \downarrow\approx x\sigma'\mu' \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} x\alpha \approx x\sigma'\mu'}\ w + push(\text{ car } E \subset \Gamma_1)$$

Π_2

$$\dfrac{\dfrac{}{\Gamma_1 \cup \Gamma_2, x \in V \vdash x\alpha \downarrow\approx x\sigma'\mu'}\ r(E) \text{ (d'après (1.4))}}{\Gamma_1 \cup \Gamma_2, x \in V \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} x\alpha \downarrow\approx x\sigma'\mu', x\alpha \approx x\sigma'\mu'}\ w + push$$

Π_3

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma_1 \cup \Gamma_2, x \in V \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} x\alpha \approx x\sigma'\mu'}\ cut$$

$\Pi_{1,\sigma'}$:

$$\dfrac{\dfrac{}{(Q'[r]_{\omega'})\sigma' \vdash_{\mathcal{RE}_1} (Q'[l]_{\omega'})\sigma'}\ Ax}{(Q'[r]_{|\omega'})\sigma' \vdash_{\mathcal{RE}_1 \cup E} (Q'[l]_{|\omega'})\sigma',\ Q'\sigma'}\ w + push$$

$\Pi_{2,\sigma'}$:

$$\dfrac{\dfrac{}{(Q'[l]_{|\omega'})\sigma' \vdash_E Q'\sigma'}\ Ax\ (\text{car } \sigma' \in CSUC_E(Q'_{|\omega'}, l))}{(Q'[r]_{\omega'})\sigma',\ (Q'[l]_{\omega'})\sigma' \vdash_{\mathcal{RE}_1 \cup E} Q'\sigma'}\ w + push$$

$\Pi_{3,\sigma'}$:

$$\dfrac{\dfrac{\dfrac{\Pi_{1,\sigma'} \quad \Pi_{2,\sigma'}}{(Q'[r]_{\omega'})\sigma' \vdash_{\mathcal{RE}_1 \cup E} Q'\sigma'}\ cut}{E,\ (Q'[r]_{\omega'})\sigma' \vdash_{\mathcal{RE}_1} Q'\sigma'}\ pop}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x_{\sigma'}} \in \mathcal{T}(\Sigma)^{n'_\sigma}, (Q'[r]_{\omega'})\sigma' \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} Q'\sigma'}\ w + push$$

$\Pi_{4,\sigma'}$:

$$\dfrac{\Gamma_1 \cup \Gamma_2, \overrightarrow{x_{\sigma'}} \in \mathcal{T}(\Sigma)^{n'_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} (Q'[r]_{\omega'})\sigma'\ (\text{supposé})}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x_{\sigma'}} \in \mathcal{T}(\Sigma)^{n'_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} Q'\sigma', (Q'[r]_{\omega'})\sigma'}\ w$$

$\Pi_{5,\sigma'}$:

$$\dfrac{\Pi_{3,\sigma'} \quad \Pi_{4,\sigma'}}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x_{\sigma'}} \in \mathcal{T}(\Sigma)^{n'_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} Q'\sigma'}\ cut$$

Posons $\mathcal{PE}_{ind}(Q,<)$ la proposition canonique associée à $\mathcal{RE}_{ind}(Q,<)$.

$\Pi_{6,\sigma'}$

$$\dfrac{\dfrac{\Pi_{5,\sigma'}}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x_{\sigma'}} \in \mathcal{T}(\Sigma)^{n'_\sigma}, \mathcal{PE}_{ind}(Q,<)\sigma' \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma'} Q'\sigma'}\ pop}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x_{\sigma'}} \in \mathcal{T}(\Sigma)^{n'_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma'} \mathcal{PE}_{ind}(Q,<)\sigma' \Rightarrow Q'\sigma'}\ \Rightarrow -r$$

$\Pi_{\sigma',\mu'}$:

$$\dfrac{\Pi_{6,\sigma'}}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\sigma'\mu'} (\mathcal{PE}_{ind}(Q,<) \Rightarrow Q')\sigma'\mu'}\ r_{\mu'}$$

Π_α

$$\dfrac{\Pi_{\sigma',\mu'} \quad \Pi_3}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2\alpha} (\mathcal{PE}_{ind}(Q,<) \Rightarrow Q')\alpha}\ r_e$$

Et puisque α a été choisie de façon quelconque, on a :
Π_4 :

$$\dfrac{\dfrac{\dfrac{\bigwedge_{\alpha \in Subst^\Sigma \wedge \mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\}) \subseteq \mathcal{D}om(\alpha)} \Pi_\alpha}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} \mathcal{PE}_{ind}(Q,<) \Rightarrow Q'}\ r_{\overrightarrow{x}}}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, \mathcal{PE}_{ind}(Q,<) \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q'}\ imp}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, \mathcal{PE}_{ind}(Q',<), \mathcal{PE}_{ind}(Q,<) \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q'}\ w$$

Π_5 :

$$\dfrac{\dfrac{\overline{\mathcal{PE}_{ind}(Q',<) \vdash_E \mathcal{PE}_{ind}(Q,<)}\ Ax}{\mathcal{PE}_{ind}(Q',<), E \vdash \mathcal{PE}_{ind}(Q,<)}\ pop}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, \mathcal{PE}_{ind}(Q',<) \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} \mathcal{PE}_{ind}(Q,<), Q'}\ w + push \text{ car } (E \subset \Gamma_1)$$

Π_6 :

$$\dfrac{\dfrac{\Pi_4 \quad \Pi_5}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, \mathcal{PE}_{ind}(Q',<) \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q'}\ cut}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n, \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \cup \{\mathcal{RE}_{ind}(Q',<)\}} Q'}\ push$$

Π

$$\dfrac{\Pi_6 \qquad \Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Noeth(<, \mathcal{T}(\Sigma))}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q'}\ r_I$$

et la preuve est terminée. □

Théorème 1.2 *IndNarrowMod est correct.*

Preuve. Le théorème 1.1 prouve la correction de **Induce**, et les preuves de correction des autres règles sont analogues à celles qui ont été effectuées pour le système IndNarrow □

1.3 Correction réfutationnelle

Démontrer la correction réfutationnelle revient à prouver que, pour chaque règle du système IndNarrowMod de la forme

$$\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q \rightarrowtail \bullet_{i \in I}\ \Gamma_1^i | \Gamma_2^i \vdash_{\mathcal{RE}_1^i | \mathcal{RE}_2^i} Q^i$$

tous les $\Gamma_1^i \cup \Gamma_2^i, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2^i \cup \{Q^i\})} \in \mathcal{T}(\Sigma)^{n^i} \vdash_{\mathcal{RE}_1^i \cup \mathcal{RE}_2^i} Q^i$ sont dérivables si $\Gamma_1 \cup \Gamma_2, \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ l'est. Nous détaillons dans la preuve du théorème suivant le point le plus délicat qui est de nouveau le cas de la règle **Induce**.

Théorème 1.3 *Si :*

1. $\Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2} Q \rightarrowtail_{\mathbf{Induce}} \bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma' \in CSUC_E(Q'_{|\omega'}, l)}} \Gamma_1 | \Gamma_2 \vdash_{\mathcal{RE}_1 | \mathcal{RE}_2 \sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} Q'[r]_{\omega'} \sigma'$ *(rappel :* $\omega' \in \mathcal{DM}(Q')$, $Q' =_E Q$*) ;*
2. *le séquent* $\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q$ *admet une preuve dans le calcul des séquents modulo, avec* $\overrightarrow{x} \in \mathcal{X}^n$ *défini par* $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE}_2 \cup \{Q\})}$ *;*

alors, chaque séquent :

$$\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_{\sigma'} \in \mathcal{T}(\Sigma)^{n_{\sigma'}} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma' \cup \{\mathcal{RE}_{ind}(Q,<)\sigma'\}} Q'[r]_{\omega'} \sigma'$$

est également démontrable dans le calcul des séquents modulo, en posant $\overrightarrow{x}_{\sigma'} \in \mathcal{X}^{n_{\sigma'}}$ *le vecteur des variables libres de* $\mathcal{RE}_2 \sigma' \cup \{Q\sigma'\}$.

Preuve. Soit $\sigma \in CSUC_E(t,l)$. Pour toute substitution close μ telle que $\mathcal{V}ar(\mathcal{RE}_2\sigma \cup \{Q\sigma\}) \subseteq \mathcal{D}om(\mu)$, on a :

$\Pi_{\sigma,\mu}$

$$\frac{\Gamma_1 \cup \Gamma_2, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2} Q}{\Gamma_1 \cup \Gamma_2 \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma\mu} Q\sigma\mu}\ r_{\sigma\mu}$$

Considérons maintenant les dérivations suivantes :

$\Pi_{1,\sigma}$:

$$\frac{\bigwedge_{\mu \in Subst^{\Sigma} \wedge \mathcal{V}ar(\mathcal{RE}_2\sigma \cup \{Q\sigma\}) \subseteq \mathcal{D}om(\mu)} \Pi_{\sigma,\mu}}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma} Q\sigma}\ r_{\overrightarrow{x}}$$

$\Pi_{2,\sigma}$:

$$\frac{\Pi_{1,\sigma}}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma} Q\sigma, (Q[r]_\omega)\sigma}\ w$$

$\Pi_{3,\sigma}$:

$$\frac{\dfrac{}{(Q[l]_{|\omega})\sigma \vdash_{\{l \to r\}} (Q[r]_{|\omega})\sigma}\ Ax}{\Gamma_1 \cup \Gamma_2,\ Q\sigma,\ (Q[l]_\omega)\sigma \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma} (Q[r]_\omega)\sigma}\ w + push\ (\text{car } l \to r \in \mathcal{RE}_1)$$

$\Pi_{4,\sigma}$:

$$\frac{\dfrac{\dfrac{}{Q\sigma \vdash_E (Q[l]_\omega)\sigma}\ Ax\ (\text{car } \sigma \in CSUC_E(Q_{|\omega}, l))}{E,\ Q\sigma \vdash (Q[l]_\omega)\sigma}\ pop}{\Gamma_1 \cup \Gamma_2,\ Q\sigma \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma} (Q[l]_\omega)\sigma,\ (Q[r]_\omega)\sigma}\ w + push$$

$\Pi_{5,\sigma}$:

$$\frac{\dfrac{\Pi_{3,\sigma} \quad \Pi_{4,\sigma}}{\Gamma_1 \cup \Gamma_2,\ Q\sigma \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma} (Q[r]_\omega)\sigma}\ cut}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma}, Q\sigma \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma} (Q[r]_\omega)\sigma}\ w$$

$\Pi_{6,\sigma}$:

$$\frac{\dfrac{\Pi_{2,\sigma} \quad \Pi_{5,\sigma}}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma} (Q[r]_\omega)\sigma}\ cut}{\Gamma_1 \cup \Gamma_2, \overrightarrow{x}_\sigma \in \mathcal{T}(\Sigma)^{n_\sigma} \vdash_{\mathcal{RE}_1 \cup \mathcal{RE}_2 \sigma \cup \mathcal{RE}_{ind}(Q)\sigma} (Q[r]_\omega)\sigma}\ w + push$$

ce qui termine la preuve. □

En guise de corollaire du théorème 1.5, on obtient :

Théorème 1.4 *Le système* IndNarrowMod *est réfutationellement correct.*

Preuve. Le théorème 1.3 prouve la correction de **Induce**, et les preuves de correction des autres règles sont analogues à celles qui ont été effectuées pour le système IndNarrow □

1.4 Complétude réfutationnelle

Théorème 1.5 *Le système* IndNarrowMod *est réfutationellement complet.*

Preuve. L'argumentation est analogue à celle développée au paragraphe 1.5 de la partie III □

1.5 Exemple

Considérons la spécification **liste** présentée à la figure 1.2. Notre but ici est de prouver la

- **Sortes :** nat, list.
- **Constructeurs :**

$$0 : \to nat; \quad s : nat \to nat; \quad Nil : \to list; \quad :: : nat \times list \to list.$$

- **Fonctions définies :**

$$+ : nat \times nat \to nat; \quad <> : list \times list \to list; \quad rev : list \to list$$

- **Axiomes** :

$$\begin{array}{lll} 0 + x \approx x & Nil <> l \approx l & rev(Nil) \approx Nil \\ s(x) + y \approx s(x+y) & (x :: l) <> m \approx x :: (l <> m) & rev(x :: l) \approx rev(l) <> (x :: Nil) \end{array}$$

FIG. 1.2 – **liste**

conjecture :

$$L <> M \approx rev(L) <> M$$

Commençons par introduire les notations suivantes :

1. $E(<>) = \{(L <> M) <> N \approx (M <> L) <> N\}$ (nous avons déjà pu constater dans l'exemple 2.1 de la partie II que $E(<>)$ est une théorie infinitaire) ;
2. Γ'_1 = ensemble des axiomes de la spécification **liste** ;
3. $\mathcal{RE}_1$ le système de réécriture obtenu par remplacement dans Γ'_1 du symbole $\approx$ par $\to$;
4. $\Gamma_1 = \Gamma'_1 \cup E(<>) \quad \Gamma_2 = \{L(\approx), NI,\ Noeth(<_2, \mathcal{T}(\Sigma)^2)\}$;
5. $Q = L <> M \approx rev(L) <> M$.

Posons donc le but :

$$\boxed{\Gamma_1, E(<>) | \Gamma_2 \vdash_{\mathcal{RE}_1 | \emptyset} Q}$$

En appliquant **Push**$_1$ et **Orient** de façon répétée, on obtient :

$\emptyset,\ E(<>) | \Gamma_2 \vdash_{\mathcal{RE}_1 | \emptyset} Q$

On peut maintenant appliquer **Induce**, car $\mathcal{RE}_1$ est terminant et suffisamment complet modulo $E(<>)$. Les positions *définie-maximales* dans $\mathcal{D}om(Q)$ sont 1 et 2.1. On choisit arbitrairement la position 2.1, et on doit donc calculer les ensembles $CSUC_{E(<>)}(rev(L), l)$ pour chaque règle $l \to r$ de $\mathcal{RE}_1$. On peut se restreindre aux règles telles que $l(\varepsilon) = rev$, et on obtient :

l	$CSUC_{E(<>)}(rev(L), l)$
$rev(Nil)$	$\sigma_1 = \{L \to Nil\}$
$rev(x :: l)$	$\sigma_2 = \{L \to X_1 :: L_1; x \to X_1; l \to L_1\}$

On est amené à prouver les deux sous-buts :

$$\boxed{\emptyset, E(<>)|\Gamma_2 \vdash_{\mathcal{RE}_1|\{\kappa_1\}} Nil <> M \approx Nil <> M}$$
$$\boxed{\emptyset, E(<>)|\Gamma_2 \vdash_{\mathcal{RE}_1|\{\kappa_2\}} ((X_1 :: L_1) <> M) \approx (rev(L_1) <> (X_1 :: Nil)) <> M}$$

avec :

$$\begin{cases} \kappa_1 = (\underline{l}, \underline{m}) <_2 (Nil, M) \Rightarrow \underline{l} <> \underline{m} \approx rev(\underline{l}) <> \underline{m} \\ \kappa_2 = (\underline{l}, \underline{m}) <_2 (X_1 :: L_1, M) \Rightarrow \underline{l} <> \underline{m} \approx rev(\underline{l}) <> \underline{m} \end{cases}$$

Trivial élimine le premier sous-but, et sachant que l'on travaille modulo $E(<>)$ équivalence, ce deuxième sous-but peut encore s'écrire :

$$\boxed{\emptyset, E(<>)|\Gamma_2 \vdash_{\mathcal{RE}_1|\{\kappa_2\}} ((X_1 :: L_1) <> M) \approx ((X_1 :: Nil) <> rev(L_1)) <> M}$$

ce qui donne, après normalisation :

$$\boxed{\emptyset, E(<>)|\Gamma_2 \vdash_{\mathcal{RE}_1|\{\kappa_2\}} X_1 :: (L_1 <> M) \approx X_1 :: (rev(L_1) <> M)}$$

Maintenant, comme $(L_1, M) <_2 (X_1 :: L_1, M)$, on peut appliquer **Rewrite**$_2$ à ce dernier sous-but, on obtient ainsi :

$$\boxed{\emptyset, E(<>)|\Gamma_2 \vdash_{\mathcal{RE}_1|\{\kappa_2\}} X_1 :: (L_1 <> M) \approx X_1 :: (L_1 <> M)}$$

et **Trivial** élimine ce dernier sous-but.

2

Spécialisation au cas des théories associatives-commutatives ou associatives

Sommaire

2.1 Formes plates : étude plus approfondie 136
2.2 Les systèmes de recherche de preuve IndNarrowModAC et IndNarrowModA 140
2.3 Deux exemples simples 141
2.4 Correction et complétude réfutationnelle de IndNarrowModAC et IndNarrowModA 144
2.4.1 IndNarrowModAC et IndNarrowModA : deux instances de IndNarrowMod 145
2.5 Autres exemples 148
2.5.1 Exemples AC 148
2.5.2 Exemple A 153

Nous présentons dans ce chapitre deux systèmes de recherche de preuve : IndNarrowModAC et IndNarrowModA qui sont des succédanés du système de recherche de preuve IndNarrowMod présenté au chapitre précédent. Comme leurs noms l'indiquent, le premier effectue des preuves dans les théories associatives commutatives, et le second dans les théories associatives. Ces systèmes travaillent dans l'algèbre des termes variadiques, définie par C.Marché [MAR 93]. Nous avons déjà fait observer que celle-ci est une généralisation de la notion d'algèbre de termes classiques, où certains symboles de fonction ont une arité non fixée. L'intérêt de cette généralisation est de pouvoir coder les axiomes de commutativité et d'associativité dans la structure des termes, en leur faisant subir une transformation appelée *aplatissement*. Cette transformation a pour conséquence immédiate d'alléger l'écriture, et on verra dans les exemples que ceci n'est pas uniquement une commodité pratique. En effet, on s'apercevra qu'il est plus facile de repérer les symétries dans un but équationnel lorsque les deux membres sont aplatis, et que cela est très utile pour savoir à quelle position il est préférable d'appliquer la règle **Induce** : par symétrie, certaines positions sont clairement équivalentes, et on privilégiera celles qui engendrent des instances du but auxquelles on peut appliquer davantage de réductions ultérieures. L'aplatissement permet également de contourner la difficulté qui est le choix du représentant de la classe d'équivalence du but lorsque l'on veut réaliser l'étape de récurrence. En effet, deux égalités d'une même classe

d'équivalence modulo A ont exactement la même forme plate, et deux égalités d'une même classe d'équivalence modulo AC ont la même forme plate à permutation près des arguments des symboles associatifs et commutatifs. On verra alors que la question du choix du meilleur représentant ne se pose plus dans les deux cas.

2.1 Formes plates : étude plus approfondie

Une des difficultés posée par les termes aplatis à C. Marché [MAR 93] a été la représentation des positions. Par exemple, si $+$ est un symbole AC, quelle est la position de $b+c$ dans $a+b+c+d$? C.Marché a proposé une définition qui donne à $b+c$ la position $\{2,3\}$. De cette définition des positions on en déduit une autre des sous-termes. Ces définitions ont ensuite permis à C.Marché d'introduire une transformation, appelée *aplatissement des positions*, et qui fait correspondre, à toute position d'un terme, une position à l'intérieur de sa forme plate. Un des intérêts majeurs de cette transformation est qu'elle permet de repérer, à l'intérieur de la forme plate d'un terme s, les sous-termes qui sont eux-mêmes les formes plates de sous-termes de s. Ceci est le point clé qui crée un pont entre les systèmes IndNarrowModAC et IndNarrowModA d'une part et le système générique IndNarrowMod d'autre part. Dans ce qui suit, $+$ désigne un symbole variadique quelconque, et on adoptera les conventions suivantes :

Notation: Soit $\overline{s}$ un terme aplati quelconque :

- si $\overline{s}(\varepsilon) \notin \mathcal{V}$, on écrira $\overline{s} = f\overline{s_1}\ldots\overline{s_n}$, en précisant $f \notin \mathcal{V}$;
- si $\overline{s}(\varepsilon) \in \mathcal{V}$, on écrira $\overline{s} = \overline{s_1}+\ldots+\overline{s_n}$, chaque $\overline{s_i}$ désignant un sous-terme aplati de $\overline{s}$ muni d'un symbole de tête différent de $+$.

Commençons par introduire les définitions de C.Marché [MAR 93].

On se place de nouveau dans une algèbre de termes variadiques définie dans la partie II (définition 1.10), et on reprend les notations qui ont été introduites alors. On définit $A(\mathcal{V})$ et $AC(\mathcal{V})$ par :

- $A(\mathcal{V}) = \{f(fxy)z = fx(fyz) \mid f \in \mathcal{V}\}$
- $AC(\mathcal{V}) = A(\mathcal{V}) \cup \{fxy = fyx \mid f \in \mathcal{V}\}$

ou encore A et AC, s'il ne peut y avoir d'ambigüités.

Définition 2.1 L'ensembles des positions d'un terme aplati $\overline{s}$, noté $\mathcal{D}om(\overline{s})$, est défini de la façon suivante :

- $\mathcal{D}om(f\overline{s_1}\ldots\overline{s_n}) = \{\varepsilon\} \cup 1.\mathcal{D}om(\overline{s_1}) \cup \ldots \cup n.\mathcal{D}om(\overline{s_n})$, si $f \notin \mathcal{V}$;
- $\mathcal{D}om(\overline{s_1}+\ldots+\overline{s_n}) = \{\varepsilon\} \cup 1.\mathcal{D}om(\overline{s_1}) \cup \ldots \cup n.\mathcal{D}om(\overline{s_n}) \cup \mathcal{P}_n$ avec $\mathcal{P}_n$ l'ensemble des parties de $\{1,\ldots,n\}$ de cardinal ≥ 2

Remarque: La position $\{1,\ldots,n\}$ sera identifiée avec ε.

Exemple 2.1 [BER 97] Supposons $f \notin \mathcal{V}$, et posons $\overline{s} = a+b+f(c+d)$. Alors :

$$\mathcal{D}om(\overline{s}) = \{\varepsilon,\ 1,\ 2,\ 3,\ \{1,2\},\ \{1,3\},\ \{2,3\},\ 3.1,\ 3.1.1,\ 3.1.2\}$$

Les positions dans $\mathcal{P}_n$ sont appelées *positions de niveau 0*. Plus généralement, le *niveau* d'une position est la longueur de la chaîne obtenue en effaçant l'ensemble se trouvant éventuellement à la fin de cette position.

Définition 2.2 Un terme aplati $\overline{s} = \overline{s_1}+\ldots+\overline{s_n}$ étant donné, on définit récursivement le niveau d'une position $\omega \in \mathcal{D}om(\overline{s})$, noté $Level(\omega)$, par :

- $Level(\varepsilon) = \varepsilon$;
- $Level(i.\omega) = i.Level(\omega)$;

– $Level(\{i_1, \ldots, i_k\}) = \varepsilon$, avec $k \geq 2$.

Remarque: La définition reste cohérente si $\{1, \ldots, n\}$ est identifiée avec ε.

Exemple 2.2 Dans l'exemple 2.1, $Level(1) = 1$, $Level(\{1, 2\}) = \varepsilon$, $Level(3.1) = 3.1$.

Après avoir défini les positions, on peut naturellement définir les sous-termes :

Définition 2.3 Un terme aplati $\overline{s}$ et une position $\omega \in \mathcal{D}om(\overline{s})$ étant donnés, le sous-terme de $\overline{s}$ à la position ω, noté $\overline{s}_{|\omega}$, est défini par :
- $\overline{s}_{|\varepsilon} = \overline{s}$;
- $\overline{s}_{|i.\omega_i} = \overline{s_i}_{|\omega_i}$;
- $(\overline{s_1} + \ldots + \overline{s_n})_{|\{i_1,\ldots,i_p\}} = \overline{s_{i_1}} + \ldots + \overline{s_{i_p}}$.

Exemple 2.3 Dans l'exemple 2.1 , $\overline{s}_{|\{1,3\}} = a + f(c + d)$ et $\overline{s}_{|3.1} = c + d$

Quand on aplatit un terme s, l'ensemble de ses positions est transformé, cela est défini formellement comme suit :

Définition 2.4 [MAR 93] Un terme $s = fs_1 \ldots s_n \in \mathcal{T}(\Sigma, \mathcal{X})$ étant donné, il existe une application de $\mathcal{D}om(s)$ vers $\mathcal{D}om(\overline{s})$, notée $flat_s$, définie récursivement de la façon suivante :
- si il existe $f \in \Sigma \setminus \mathcal{V}$, $s_1, \ldots s_n \in \mathcal{T}(\Sigma, \mathcal{X})$, tels que $s = fs_1 \ldots s_n$, alors :
  - $flat_s(\varepsilon) = \varepsilon$
  - $flat_s(i.\omega) = i.flat_{s_i}(\omega)$
- si il existe $s_1, \ldots s_n,\ t_1 \ldots t_n \in \mathcal{T}(\Sigma, \mathcal{X})$, tels que $\overline{s} = \overline{s_1} + \ldots + \overline{s_n}$ et $\overline{t} = \overline{t_1} + \ldots \overline{t_m}$ ($n \geq 1$, $m \geq 1$) , alors :
  - $flat_{s+t}(\varepsilon) = \varepsilon$
  - si $n = 1$, $flat_{s+t}(1.\omega) = 1.flat_s(\omega)$
  - si $m = 1$, $flat_{s+t}(2.\omega) = (n + 1).flat_t(\omega)$
  - si $n \geq 2$, $flat_{s+t}(1) = \{1, \ldots,\ n\}$
  - si $m \geq 2$, $flat_{s+t}(2) = \{n + 1, \ldots,\ n + m\}$
  - si $n \geq 2$, $flat_{s+t}(1.i.\omega) = flat_s(i.\omega)$
  - si $m \geq 2$ et $flat_t(i.\omega) = j.\omega'$, alors $flat_{s+t}(2.i.\omega) = (n + j).\omega'$
  - si $m \geq 2$ et $flat_t(i.\omega) = \{i_1, \ldots,\ i_k\}$, alors $flat_{s+t}(2.i.\omega) = \{n + i_1, \ldots,\ n + i_k\}$

Exemple 2.4 Pour $s = (a + b) + f(c, d)$, on a :

$$\begin{array}{cc} \multicolumn{2}{c}{flat_s(\varepsilon) = \varepsilon} \\ \multicolumn{2}{c}{flat_s(1) = \{1, 2\} \quad flat_s(2) = 3} \\ flat_s(1.1) = 1 & flat_s(2.1) = 3.1 \\ flat_s(1.2) = 2 & flat_s(2.2) = 3.2 \end{array}$$

On a écrit, au début, que l'intérêt de la définition précédente est qu'elle permet de repérer, à l'intérieur de la forme plate d'un terme s, des sous-termes qui sont eux-mêmes des formes plates de sous-termes de s. Cela est formalisé comme suit :

Lemme 2.1 [MAR 93] $\overline{s_{|\omega}} = \overline{s}_{|flat_s(\omega)}$.

Il est alors naturel de se demander sous quelles conditions une position ω, choisie à l'intérieur d'un terme aplati, est l'image d'une position d'un terme s par l'application $flat_s$. C'est le cas si le dernier élément de la suite définissant la position ω est un entier, ou un ensemble d'entiers *consécutifs*. Plus précisément, cela s'énonce comme suit :

Lemme 2.2 Les propositions 1 et 2 ci-dessous sont équivalentes :

1. $(\omega \in \mathcal{D}om(\overline{s}))$, il existe $i, k \in \mathbb{N}$, et une séquence d'entiers ω_0, tels que $\omega = \omega_0.i$ ou $\omega = \omega_0.\{i, i+1, \ldots, i+k\}$.
2. Il existe $s' \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega' \in \mathcal{D}om(s')$, tels que $\overline{s'} = \overline{s}$ et $\omega = flat_{s'}(\omega')$

Preuve. c'est le lemme C.34 démontré en annexe □

La proposition précédente justifie la définition suivante :

Définition 2.5 Un terme aplati $\overline{s}$ étant donné, une position $\omega \in \mathcal{D}om(\overline{s})$ est dite *position aplatie*, si il existe $i, k \in \mathbb{N}$ et une séquence d'entiers ω_0 vérifiant $\omega = \omega_0.i$ ou $\omega = \omega_0.\{i, i+1, \ldots, i+k\}$.

Une autre difficulté posée par les termes variadiques est la définition du remplacement d'un sous-terme à une occurence donnée. Par exemple, a, b, c, d, e, f et g étant des constantes données, comment peut-on définir le remplacement du sous-terme de $a+b+c+d$ à la position $\{1,3\}$ par $e+f+g$? C. Marché [MAR 93] a proposé une définition selon laquelle $(a+b+c+d)[e+f+g]_{\{1,3\}}$ est défini par l'addition (suivi d'un réaplatissement) du terme $a+b+c+d$ privé de a et c, avec $e+f+g$. Cela s'écrit : $a+b+c+d[e+f+g]_{\{1,3\}} = b+d+e+f+g$. De la même manière, on peut écrire : $(a+b+c+d)[e+f+g]_{\{2,3\}} = a+d+e+f+g$. Observons maintenant que le terme $a+b+c+d$ peut être obtenu en aplatissant $s = (a+(b+c))+d$, et il est facile de vérifier que si $\omega \in \mathcal{D}om(s)$ est défini par $\omega = 1.2$, $flat_s(\omega) = \{2,3\}$. Si on pose maintenant $t = e+(f+g)$, on a $\overline{t} = e+f+g$, et le terme $(a+b+c+d)[e+f+g]_{\{2,3\}}$ peut s'écrire avec ces notations $\overline{s}[\overline{t}]_{flat_s(\omega)}$. Cependant, on a $s[t]_\omega = (a+(e+(f+g)))+d$, donc par conséquent $\overline{s[t]_\omega} = a+e+f+g+d$. On voit alors que l'on a $\overline{s[t]_\omega} \equiv_p \overline{s}[\overline{t}]_{flat_s(\omega)}$, mais que l'on n'a pas $\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{flat_s(\omega)}$. Il nous a pourtant semblé qu'il serait très pratique de pouvoir affirmer que cette dernière égalité est vraie pour tout choix de s, t et ω. C'est pourquoi nous avons proposé une autre définition du remplacement d'un sous-terme à une occurence donnée que celle de C.Marché. Observons tout d'abord que l'égalité souhaitée $\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{flat_s(\omega)}$ incite à ne modifier cette définition qu'au niveau d'une position aplatie. Sachant que celle-ci se termine toujours par un entier isolé ou un ensemble d'entiers consécutifs, on peut proposer une définition très intuitive. Dans la situation précédente, par exemple, une position aplatie à l'intérieur du terme $a+b+c+d$ peut être $\omega = \{2,3\}$. On peut alors simplement poser $a+b+c+d[e+f+g]_{\{2,3\}} = a+e+f+g+d$. Formellement, cela s'écrit :

Définition 2.6 Etant donné deux termes aplatis $\overline{s} = f\overline{s_1}\ldots\overline{s_n}$, $\overline{t}$, et une position $\omega \in \overline{s}$, on définit le remplacement par $\overline{t}$ dans $\overline{s}$ à la position ω récursivement de la façon suivante :

- $\overline{s}[\overline{t}]_\varepsilon = \overline{t}$;
- si $\omega \in \{1, \ldots, n\}$:
  - cas 1 : il existe $i,\ k \in \mathbb{N}$, tels que $\omega = \{i, i+1, \ldots, i+k\}$.
    $\overline{s}[\overline{t}]_\omega = \overline{f\overline{s_1}\ldots\overline{s_{i-1}}\overline{t}\overline{s_{i+k+1}}\ldots\overline{s_n}}$
  - cas 2 : sinon, soit $\{i_1, \ldots, i_k\} = \{1, \ldots, n\} - \omega$.
    $\overline{s}[\overline{t}]_\omega = \overline{f\overline{s_{i_1}}\ldots\overline{s_{i_k}}\overline{t}}$.
- $\overline{s}[\overline{t}]_{i.\omega_i} = \overline{f\overline{s_1}\ldots\overline{s_i}[\omega_i \leftarrow \overline{t}]\ldots\overline{s_n}}$.

Il faut remarquer en ce qui concerne le deuxième point que le réaplatissement est nécessaire si $\overline{t}$ admet + pour symbole de tête.
Cette définition permet alors d'énoncer la proposition suivante :

Proposition 2.1 Etant donné un terme s, une règle $l \to r$, une position $\omega \in \mathcal{D}om(s)$, et une substitution σ, on a l'égalité suivante :

$$\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{flat_s(\omega)}$$

Preuve. Voir proposition C.5 démontrée en annexe. □

A partir de cette proposition, une relation de réécriture sur les termes se décline sur les termes aplatis de la façon suivante :

Définition 2.7 Un système de réécriture (non conditionnel) $\mathcal{R}$ étant donné, on définit la relation $\to_{\overline{\mathcal{R}}}$ sur l'ensemble des termes aplatis par $\overline{s} \to_{\overline{\mathcal{R}}} \overline{t}$ s'il existe une règle $l \to r$ dans $\mathcal{R}$, une position aplatie $\omega \in \mathcal{D}om(\overline{s})$ et une substitution σ telles que : $\overline{s}_{|\omega} = \overline{l\sigma}$, $\overline{t} = \overline{s}[\overline{r\sigma}]_\omega$.

Exemple 2.5 Considérons la spécification **liste** définie à la section 1.5 du chapitre 1 (figure 1.2). Posons $\mathcal{R}$ le système de réécriture obtenu en orientant les axiomes de gauche à droite, et supposons que $+$ soit variadique. Si $\overline{s} = x + rev((s(0) + y + 0 + z) :: Nil)$, alors $\overline{s} \to_{\overline{\mathcal{R}}} x + rev((s(0) + y + z) :: Nil)$

On peut ensuite étendre cette relation aux classes d'équivalence des termes aplatis modulo permutation des arguments des symboles variadiques :

Définition 2.8 Un système de réécriture $\mathcal{R}$ étant donné, on définit la relation $\to_{\overline{\mathcal{R}}} / \equiv_p$ sur l'ensemble des termes aplatis par $\overline{s} \to_{\overline{\mathcal{R}}/\equiv_p} \overline{t}$ si il existe deux termes s' et t' tels que $\overline{s} \equiv_p \overline{s'}$, $\overline{s'} \to_{\overline{\mathcal{R}}} \overline{t'}$, et $\overline{t'} \equiv_p \overline{t}$.

La proposition ci-dessous montre le lien qui existe entre la réécriture modulo sur les termes et celle sur les termes aplatis définie plus haut. Cette proposition sera utile pour montrer que les systèmes IndNarrowModAC et IndNarrowModA sont des instances du système générique IndNarrowMod .

Proposition 2.2 Deux termes $s,\ t \in \mathcal{T}(\Sigma, \mathcal{X})$ et un système de réécriture $\mathcal{R}$ étant donnés,

1. $\overline{s} \to_{\overline{\mathcal{R}}} \overline{t}$ *ssi* $s \to_{\mathcal{R}/A} t$
2. $\overline{s} \to_{\overline{\mathcal{R}}/\equiv_p} \overline{t}$ *ssi* $s \to_{\mathcal{R}/AC} t$

Preuve. 1. – $\Rightarrow$: supposons que $\overline{s} \to_{\overline{\mathcal{R}}} \overline{t}$. Par définition, il existe une règle $l \to r$ dans $\mathcal{R}$, une position aplatie $\omega \in \mathcal{D}om(\overline{s})$ et une substitution σ telles que : $\overline{s}_{|\omega} = \overline{l\sigma}$. Or, d'après le lemme 2.2, il existe $s' \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega' \in \mathcal{D}om(s')$, tels que :

$$\overline{s'} = \overline{s} \quad \text{et} \quad \omega = flat_{s'}(\omega') \tag{2.1}$$

On peut alors écrire $\overline{s}_{|\omega} = \overline{s'}_{|flat_{s'}(\omega')}$. De plus, d'après le lemme 2.1, on a : $\overline{s'_{|\omega'}} = \overline{s'}_{|flat_{s'}(\omega')}$. On déduit alors des égalités précédentes l'égalité $\overline{s'_{|\omega'}} = \overline{l\sigma}$, et cela peut encore s'écrire :

$$s'_{|\omega'} =_A l\sigma \tag{2.2}$$

Mais comme $\overline{s'} = \overline{s}$ (d'après 2.1), on a :

$$s =_A s' \tag{2.3}$$

De (2.3) et (2.2), on tire :

$$s =_A s'[l\sigma]_{|\omega'} \tag{2.4}$$

Observons maintenant que, par définition, $\overline{t} = \overline{s}[\overline{r\sigma}]_\omega$, et, en considérant (2.1), cela peut encore s'écrire $\overline{t} = \overline{s'}[\overline{r\sigma}]_{flat_{s'}(\omega')}$. Ceci, combiné avec la proposition 2.1, permet d'écrire : $\overline{t} = \overline{s'[r\sigma]_{\omega'}}$, ou encore :

$$t =_A s'[r\sigma]_{\omega'} \tag{2.5}$$

(2.4) et (2.5) entrainent finalement $s \rightarrow_{\mathcal{R}/A} t$.

– $\Leftarrow$: Soient s, t deux termes tels que $s \rightarrow_{\mathcal{R}/A} t$. On peut alors écrire : $s =_A s'[l\sigma]_{\omega'}$, $t =_A s'[r\sigma]_{\omega'}$, avec $l \rightarrow r \in \mathcal{R}$. Ceci, combiné avec la proposition 2.1, permet d'écrire : $\overline{s} = \overline{s'}[\overline{l\sigma}]_{flat_{s'}(\omega')}$, et $\overline{t} = \overline{s'}[\overline{r\sigma}]_{flat_{s'}(\omega')}$, ce qui implique $\overline{s} \rightarrow_{\overline{\mathcal{R}}} \overline{t}$.

2. : conséquence immédiate de 1, et de l'équivalence :$s =_{AC} t \Leftrightarrow \overline{s} \equiv_p \overline{t}$ (lemme 1.3 de la partie II).

□

Lorsqu'on réalise l'étape de récurrence avec les systèmes IndNarrowModAC et IndNarrowModA , on commence par repérer des positions *définie-maximales* à l'intérieur des conjectures à démontrer. Bien qu'intuitif, ce concept mérite d'être clairement défini dans le cas des termes aplatis :

Définition 2.9 Pour tout $s \in \mathcal{T}(\Sigma, \mathcal{X})$, et pour tout $\omega \in \mathcal{D}om(\overline{s})$, la position ω est appelée *définie-maximale*, si il existe $f \in \mathcal{D}$ et des termes $s_1, \ldots, s_n \in \mathcal{T}(\mathcal{C}, \mathcal{X})$, tels que $\overline{s}_{|\omega} = f s_1 \ldots s_n$, avec $n = 2$ si f est un symbole AC.

Exemple 2.6 Considérons la spécification **liste** définie à la section 1.5 du chapitre 1 (figure 1.2), supposons que + soit variadique, et posons $\overline{s} = x + rev((s(0) + y + 0 + z) :: Nil)$. Les positions $2.1.\{1,2\}$, $2.1.\{1,3\}$, $2.1.\{1,4\}$, $2.1.\{2,3\}$, $2.1.\{2,4\}$, et $2.1.\{3,4\}$ sont *définie-maximales* dans $\overline{s}$.

Remarque: Un terme $s \in \mathcal{T}(\Sigma, \mathcal{X})$ et une position $\omega \in \mathcal{DM}(\overline{s})$ étant donnés, $\overline{s}_{|\omega}$ peut être considéré comme un terme qui coïncide avec sa propre forme plate.

La définition 2.9 est justifiée par la proposition suivante :

Proposition 2.3 *$\omega \in \mathcal{DM}(s)$ si, et seulement si $flat_s(\omega) \in \mathcal{DM}(\overline{s})$*

Preuve. c'est une conséquence directe du lemme 2.1 □

La proposition ci-dessous peut être utile dans le cas où on cherche à savoir si une position donnée à l'intérieur d'un terme aplati est *définie-maximale* :

Proposition 2.4 Si $\overline{s}_{|\omega} \equiv_p \overline{s'}_{|\omega'}$, et si $\omega \in \mathcal{DM}(\overline{s})$, alors $\omega' \in \mathcal{DM}(\overline{s'})$

Preuve. c'est une conséquence directe des définitions 1.12 et 2.9. □

2.2 Les systèmes de recherche de preuve IndNarrowModAC et IndNarrowModA

La règle **Induce** du système de recherche de preuve IndNarrowMod est très difficile à utiliser et à automatiser, car elle impose de parcourir la classe de congruence du but à prouver. L'idée de base que l'on va essayer de mettre en oeuvre dans la suite est d'utiliser les formes plates pour contourner cette difficulté dans le cas des théories associatives commutatives, ou associatives uniquement. Plus précisément, on cherche à construire un ensemble de règles d'inférence, tel

InduceAC $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail$

$\bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma \in CSUC_{AC}(\overline{Q}_{|\omega}, l)}} \quad \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} (\overline{Q}[\overline{r}]_\omega)\sigma$

si $\omega \in \mathcal{DM}(\overline{Q})$

OrientAC $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \to r\}|\mathcal{RE}_2} \overline{Q}$

si $(\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $l > r$

Push$_1$AC $\Gamma_1, l \approx r|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \approx r\}|\mathcal{RE}_2} \overline{Q}$

Push$_2$AC $\Gamma_1|\Gamma_2, l \approx r \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{l \approx r\}} \overline{Q}$

Rewrite$_1$AC $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q'}$

si $\overline{Q} \to_{\overline{\mathcal{RE}_1}/\equiv_p} \overline{Q'}$

Rewrite$_2$AC $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q'}$

si $\overline{Q} \to_{\overline{\mathcal{RE}_2}/\equiv_p} \overline{Q'}$

TrivialAC $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{t} \approx \overline{t'} \rightarrowtail \diamond$

si $\overline{t} \equiv_p \overline{t'}$

RefutationAC $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail$ Refutation

si aucune autre règle ne s'applique

FIG. 2.1 – Le système de recherche de preuve IndNarrowModAC

qu'à chaque étape de dérivation dans le système IndNarrowMod , on puisse associer une étape de dérivation sur la forme plate du but correspondant. On obtiendra ainsi deux systèmes de recherche de preuve IndNarrowModAC et IndNarrowModA capables d'effectuer les mêmes preuves que le système IndNarrowMod , et qui réalisent l'étape de récurrence directement sur la forme plate du but à prouver.
A l'aide de la mise au point concernant les termes aplatis faite à la section précédente, on peut introduire les sytèmes de recherche de preuve spécifiques IndNarrowModAC (Figure 2.1) et IndNarrowModA (Figure 2.2).

2.3 Deux exemples simples

Pour avoir une meilleure intuition de la façon dont cet ensemble de règles fonctionne, nous allons considérer deux exemples. Dans ce qui suit, nous ferons toujours référence à la spécification donnée dans la figure 1 de l'introduction. Dans le premier exemple, on utilise les propriétés AC des symboles $+$ et $*$. Dans le second exemple, on effectue une preuve dans la même spécification que dans le premier, mais on suppose que les symboles $+$ et $*$ sont uniquement associatifs.

Un example AC **:** Supposons que $\mathcal{RE}_1$ contienne les règles de la spécification **Arithmétique simple** donnée dans la figure 1 de l'introduction. $\mathcal{RE}_1$ est terminant et suffisamment complet modulo l'associativité et la commutativité des opérateurs $*$ et $+$ (notée $AC(+,*)$).

InduceA $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail$

$\bullet_{\substack{l \to r \in \mathcal{RE}_1 \\ \sigma \in CSUC_A(\overline{Q}_{|\omega}, l)}} \quad \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\sigma \cup \{\mathcal{RE}_{ind}(Q,<)\sigma\}} (\overline{Q}[\overline{r}]_\omega)\sigma$

si $\omega \in \mathcal{DM}(\overline{Q})$ et ω aplatie

OrientA $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{\kappa\}|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \to r\}|\mathcal{RE}_2} \overline{Q}$

si $(\kappa = l \approx r$ ou $\kappa = r \approx l)$ et $l > r$

Push$_1$A $\Gamma_1, l \approx r|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1 \cup \{l \approx r\}|\mathcal{RE}_2} Q$

Push$_2$A $\Gamma_1|\Gamma_2, l \approx r \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{l \approx r\}} Q$

Rewrite$_1$A $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q'}$

si $\overline{Q} \to_{\overline{\mathcal{RE}_1}} \overline{Q'}$

Rewrite$_2$A $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q'}$

si $\overline{Q} \to_{\overline{\mathcal{RE}_2}} \overline{Q'}$

TrivialA $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{t} \approx \overline{t'} \rightarrowtail \diamond$

si $\overline{t} = \overline{t'}$

RefutationA $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail$ Refutation

si aucune autre règle ne s'applique

FIG. 2.2 – Le système de recherche de preuve IndNarrowModA

Posons $\Gamma_1 = AC(+,*)$, $\Gamma_2 = L(\approx) \cup \{NI,\ Noeth(<_4, \mathcal{T}(\Sigma)^4)\}$, et $Q = (X_1 + X_2 + X_3) * X_4 \approx X_1 * X_4 + X_2 * X_4 + X_3 * X_4$. Considérons le but suivant :
$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\emptyset} Q$
Appliquons d'abord la règle **InduceAC**. Les positions *définie-maximales* dans Q sont $1.1.\{1,2\}$, $1.1.\{1,3\}$, $1.1.\{2,3\}$, 2.1, 2.2 et 2.3. Observons maintenant que le but reste équivalent par permutation des variables X_1, X_2 et X_3. Il reste donc deux possibilités : surréduire à une position où le symbole $+$ apparaît, ou à une position où le symbole $*$ apparaît. Comme la surréduction à une position où le symbole $*$ apparaît va créer plus de réductions ultérieures, on choisit de surréduire à la position 2.1 du but. On doit donc déterminer l'ensemble $CSUC_{AC(+,*)}(X_1 * X_4, l)$ pour chaque règle $l \to r$ de $\mathcal{RE}_1$. On peut se restreindre aux règles telles que $l(\varepsilon) = *$, et on obtient, par application du corollaire 2.2 de la partie II :

l	$CSUC_{AC(+,*)}(X_1 * X_4, l)$
$x * 0$	$\sigma_1 = \{X_1 \to Y_1; x \to Y_1; X_4 \to 0\}$ $\sigma_2 = \{X_1 \to 0; x \to Y_4; X_4 \to Y_4\}$
$x * s(y)$	$\sigma_3 = \{X_1 \to Y_1; x \to Y_1; y \to Y_4; X_4 \to s(Y_4)\}$ $\sigma_4 = \{X_1 \to s(Y_1); x \to Y_4; y \to Y_1; X_4 \to Y_4\}$

Après normalisation, cela nous conduit à prouver les quatre séquents :

$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_1}$	$0 \approx 0$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_2}$	$(X_2 + X_3) * Y_4 \approx X_2 * Y_4 + X_3 * Y_4$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_3}$	$(Y_1 + X_2 + X_3) * Y_4 + Y_1 + X_2 + X_3$ $\approx Y_1 * Y_4 + Y_1 + X_2 * Y_4 + X_2 + X_3 * Y_4 + X_3$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_4}$	$(Y_1 + X_2 + X_3) * Y_4 + Y_4 \approx Y_1 * Y_4 + Y_4 + X_2 * Y_4 + X_3 * Y_4$

Trivial élimine le premier. Puisque $(Y_1, X_2, X_3, Y_4) <_4 (Y_1, X_2, X_3, s(Y_4))$, **Rewrite**$_2$ peut s'appliquer sur le troisième, et comme $(Y_1, X_2, X_3, Y_4) <_4 (s(Y_1), X_2, X_3, Y_4)$, **Rewrite**$_2$ peut également s'appliquer sur le quatrième. On obtient ainsi

$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_2}$	$(X_2 + X_3) * Y_4 \approx X_2 * Y_4 + X_3 * Y_4$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_3}$	$Y_1 * Y_4 + X_2 * Y_4 + X_3 * Y_4 + Y_1 + X_2 + X_3$ $\approx Y_1 * Y_4 + Y_1 + X_2 * Y_4 + X_2 + X_3 * Y_4 + X_3$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_4}$	$Y_1 * Y_4 + X_2 * Y_4 + X_3 * Y_4 + Y_4$ $\approx Y_1 * Y_4 + Y_4 + X_2 * Y_4 + X_3 * Y_4$

Trivial élimine les deux derniers sous-buts. Il faut appliquer **Induce** au premier. Posons alors $Q' = (X_2 + X_3) * Y_4 \approx X_2 * Y_4 + X_3 * Y_4$. Les positions *définie-maximales* dans Q' sont1.1, 2.1, et 2.2. Observons cependant que le but est équivalent à permutation près des variables X_2 et X_3. Il reste donc deux possibilités : surréduire à une position où le symbole $+$ apparaît, ou à une position où c'est le symbole $*$ qui apparaît. On privilégie le symbole $*$, car cela permettra d'obtenir davantage de réductions ultérieures, et on choisit ainsi l'occurence 2.1. Il faut donc déterminer $CSUC_{AC}(X_2 * Y_4, l)$ pour chaque règle $l \to r$ de $\mathcal{RE}_1$. Par application du corollaire 2.2 de la partie II, on a :

l	$CSUC_{AC(+,*)}(X_2 * Y_4, l)$
$x * 0$	$\sigma_5 = \{X_2 \to Y_2; x \to Y_2; Y_4 \to 0\}$ $\sigma_6 = \{X_2 \to 0; x \to Z_4; Y_4 \to Z_4\}$
$x * s(y)$	$\sigma_7 = \{X_2 \to Y_2; x \to Y_2; y \to Z_4; Y_4 \to s(Z_4)\}$ $\sigma_8 = \{X_2 \to s(Y_2); x \to Z_4; y \to Y_2; Y_4 \to Z_4\}$

Après normalisation, cela nous conduit à prouver les quatre séquents :

$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\mathcal{RE}_{ind}(Q')\sigma_2, \mathcal{RE}_{ind}(Q')\sigma_5\}}$	$0 \approx 0$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\mathcal{RE}_{ind}(Q')\sigma_2, \mathcal{RE}_{ind}(Q')\sigma_6\}}$	$X_3 * Z_4 \approx X_3 * Z_4$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\mathcal{RE}_{ind}(Q')\sigma_2, \mathcal{RE}_{ind}(Q')\sigma_7\}}$	$(Y_2 + X_3) * Z_4 + Y_2 + X_3 \approx Y_2 * Z_4 + Y_2 + X_3 * Z_4 + X_3$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\mathcal{RE}_{ind}(Q')\sigma_2, \mathcal{RE}_{ind}(Q')\sigma_8\}}$	$(Y_2 + X_3) * Z_4 + Z_4 \approx Y_2 * Z_4 + Z_4 + X_3 * Z_4$

Trivial élimine les deux premiers sous-buts. Observons maintenant que l'on a :

$$\begin{cases} \mathcal{RE}_{ind}(Q')\sigma_7 = (\underline{x_2}, \underline{x_3}, \underline{y_4}) <_3 (Y_2, X_3, s(Z_4)) \Rightarrow (\underline{x_2} + \underline{x_3}) * \underline{y_4} \approx \underline{x_2} * \underline{y_4} + \underline{x_3} * \underline{y_4}; \\ \mathcal{RE}_{ind}(Q')\sigma_8 = (\underline{x_2}, \underline{x_3}, \underline{y_4}) <_3 (s(Y_2), X_3, Z_4) \Rightarrow (\underline{x_2} + \underline{x_3}) * \underline{y_4} \approx \underline{x_2} * \underline{y_4} + \underline{x_3} * \underline{y_4}; \end{cases}$$

et comme $(Y_2, X_3, Z_4) <_3 (Y_2, X_3, s(Z_4))$ et $(Y_2, X_3, Z_4) <_3 (s(Y_2), X_3, Z_4)$, **Rewrite**$_2$ peut être appliquée sur le troisième et le quatrième à l'aide des règles $\mathcal{RE}_{ind}(Q')\sigma_7$ et $\mathcal{RE}_{ind}(Q')\sigma_8$

respectivement. On obtient ainsi :

$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\mathcal{RE}_{ind}(Q)\sigma_2, \mathcal{RE}_{ind}(Q')\sigma_7\}}$	$Y_2 * Z_4 + X_3 * Z_4 + Y_2 + X_3$ $\approx Y_2 * Z_4 + Y_2 + X_3 * Z_4 + X_3$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \{\mathcal{RE}_{ind}(Q)\sigma_2, \mathcal{RE}_{ind}(Q')\sigma_8\}}$	$Y_2 * Z_4 + X_3 * Z_4 + Z_4$ $\approx Y_2 * Z_4 + Z_4 + X_3 * Z_4$

Et **Trivial** élimine ces deux derniers sous-buts.

Un exemple A **:** Supposons que $\mathcal{RE}_1$ contienne les règles de la spécification **Arithmétique simple** donnée dans la figure 1 de l'introduction. $\mathcal{RE}_1$ est terminant et suffisamment complet modulo l'associativité des opérateurs $*$ et $+$ (notée $A(+,*)$). Montrons que la distributivité de $*$ sur $+$ est un théorème inductif. Posons $\Gamma_1 = A(+,*)$, $\Gamma_2 = L(\approx) \cup \{NI,\ Noeth(<_3, \mathcal{T}(\Sigma)^3)\}$, et $Q = X_1 * (X_2 + X_3) \approx X_1 * X_2 + X_1 * X_3$. On part du but suivant :
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \emptyset} Q$
On peut appliquer **InduceA** au niveau des positions *définie-maximales* 1.2, 2.1 et 2.2 dans Q. Comme la surréduction à la position 2.1 va créer moins de réductions ultérieures qu'au niveau des deux autres, on préfère surréduire à la position 2.2 du but. Il est donc nécessaire de calculer $CSUC_{A(+,*)}(X_1 * X_3, l)$ pour chaque règle $l \to r$ de $\mathcal{RE}_1$. On peut bien-sûr se restreindre aux règles telles que $l(\varepsilon) = *$, et on obtient, par application du théorème 2.1 de la partie II :

l	$CSUC_{A(+,*)}(X_1 * X_3, l)$
$x * 0$	$\sigma_1 = \{X_1 \to Y_1; x \to Y_1; X_3 \to 0\}$
$x * s(y)$	$\sigma_2 = \{X_1 \to Y_1; x \to Y_1; y \to Y_3; X_3 \to s(Y_3)\}$

Après normalisation, on obtient les sous-buts :

$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_1} Y_1 * X_2 \approx Y_1 * X_2$
$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_2} Y_1 * (X_2 + Y_3) + Y_1 \approx Y_1 * X_2 + Y_1 * Y_3 + Y_1$

Trivial élimine le premier. Comme $(Y_1, X_2, Y_3) <_3 (Y_1, X_2, s(Y_3))$, **Rewrite**$_2$**A** s'applique sur le second. On obtient alors :

$\Gamma_1 \vert \Gamma_2 \vdash_{\mathcal{RE}_1 \vert \mathcal{RE}_{ind}(Q)\sigma_2} Y_1 * X_2 + Y_1 * Y_3 + Y_1 \approx Y_1 * X_2 + Y_1 * Y_3 + Y_1$

et **Trivial** élimine ce dernier sous-but.

2.4 Correction et complétude réfutationnelle de IndNarrowModAC et IndNarrowModA .

Les systèmes IndNarrowModAC et IndNarrowModA sont beaucoup plus commodes d'utilisation que IndNarrowMod , car ils n'imposent pas de parcourir la classe d'équivalence du but à prouver avant de réaliser l'étape de récurrence. Nous allons démontrer maintenant qu'il existe une correspondance entre une déduction sur un but Q effectuée avec le système IndNarrowMod et une déduction sur le but aplati correspondant effectuée avec le système IndNarrowModAC ou IndNarrowModA. On en déduira alors automatiquement la correction et la complétude réfutationnelle de IndNarrowModAC et IndNarrowModA .

2.4.1 IndNarrowModAC et IndNarrowModA : deux instances de IndNarrowMod

Nous commençons par introduire quelques lemmes utiles (certaines preuves sont effectuées en annexe).

Lemme 2.3 Soient s, s' deux termes, et $\omega \in \mathcal{D}om(\overline{s})$. Si $\overline{s} \equiv_p \overline{s'}$, il existe $\omega' \in \mathcal{D}om(\overline{s'})$, tel que :
- $\overline{s}_{|\omega} \equiv_p \overline{s'}_{|\omega'}$
- Pour tout terme t, $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{|\omega'}$

Preuve. Voir lemme C.33 démontré en annexe. □

Lemme 2.4 Pour tous termes s, $t \in \mathcal{T}(\Sigma, \mathcal{X})$ et pour tout $\omega \in \mathcal{D}om(\overline{s})$, il existe $s' \in \mathcal{T}(\Sigma, \mathcal{X})$ et $\omega' \in \mathcal{D}om(s')$, tels que $\overline{s} \equiv_p \overline{s'}$, $\overline{s}_{|\omega} = \overline{s'}_{|flat_{s'}(\omega')}$, et $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{flat_{s'}(\omega')}$

Preuve. Voir lemme C.35 démontré en annexe. □

Lemme 2.5 Soient $s \in \mathcal{T}(\Sigma, \mathcal{X})$, et $\omega \in \mathcal{DM}(s)$. Alors, $CSUC_{AC}(s_{|\omega}, l) = CSUC_{AC}(\overline{s}_{|flat_s(\omega)}, l)$.

Preuve. Soit σ un AC-unificateur constructeur de $s_{|\omega}$ et de l. On a $s_{|\omega}\sigma =_{AC} l\sigma$, ce qui entraîne (d'après le lemme 1.3 de la partie II) $\overline{s_{|\omega}\sigma} \equiv_p \overline{l\sigma}$, ou encore $\overline{s_{|\omega}}\sigma \equiv_p \overline{l\sigma}$ (1), car σ est une substitution constructeur et on a supposé que les symboles constructeurs ne sont pas variadiques. Maintenant, si $flat_s(\omega)$ est l'aplatissement de la position ω, on a $\overline{s_{|\omega}} = \overline{s}_{|flat_s(\omega)}$ (2) (lemme 2.1). Sachant que $\omega \in \mathcal{DM}(s)$, et d'après la proposition 2.3, on a $flat_s(\omega) \in \mathcal{DM}(\overline{s})$. Et puisque σ est une substitution constructeur, on peut alors poser $\overline{s}_{|flat_s(\omega)}\sigma = \overline{\overline{s}_{|flat_s(\omega)}\sigma}$ (3). On déduit alors de (1), (2) et (3) la congruence $\overline{\overline{s}_{|flat_s(\omega)}\sigma} \equiv_p \overline{l\sigma}$, ce qui s'écrit encore $\overline{s}_{|flat_s(\omega)}\sigma =_{AC} l\sigma$, donc σ est un AC-unificateur de $\overline{s}_{|flat_s(\omega)}$ et de l. L'inclusion réciproque se démontre de façon symétrique. □

Lemme 2.6 Soient $s \in \mathcal{T}(\Sigma, \mathcal{X})$, et $\omega \in \mathcal{DM}(s)$. Alors, $CSUC_A(s_{|\omega}, l) = CSUC_A(\overline{s}_{|flat_s(\omega)}, l)$.

Preuve. la preuve est analogue à celle du lemme 2.5 : il suffit de remplacer AC par A, et $\equiv_p$ par $=$. □

Remarque: On étend les lemmes précédents aux égalités, i.e on peut remplacer s par une égalité Q quelconque.

Théorème 2.1 *Soit $E = AC(\mathcal{V})$.*

1. *Si $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathsf{IndNarrowMod}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} R$, alors $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathsf{IndNarrowModAC}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} \overline{R}$.*

2. *Si $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathsf{IndNarrowModAC}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} \overline{R}$, alors il existe R' tel que $R' =_{AC} R$, et $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathsf{IndNarrowMod}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} R'$*

Preuve. – **Induce**.
- $1 \Rightarrow 2$:
  Supposons l'étape d'inférence : $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathbf{Induce}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\mathcal{RE}_{ind}(Q')\sigma'\}} (Q'[r]_{\omega'})\sigma'$. Par définition, il existe $l \rightarrow r \in \mathcal{RE}_1$ tel que $\sigma' \in CSUC_{AC}(Q'_{|\omega'}, l)$. Sachant que $\omega' \in \mathcal{DM}(Q')$, le lemme 2.5 permet

alors d'en déduire $\sigma' \in CSUC_{AC}(\overline{Q'}_{|flat_{Q'}(\omega')}, l)$ (1). Maintenant, par définition de **Induce**, $Q' =_{AC} Q$, donc, d'après le lemme 2.3, on en déduit qu'il existe $\omega \in \mathcal{D}om(\overline{Q})$ tel que $\overline{Q'}_{|flat_{Q'}(\omega')} \equiv_p \overline{Q}_{|\omega}$ et $\overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')} \equiv_p \overline{Q}[\overline{r}]_\omega$. Observons que l'on a $\omega' \in \mathcal{DM}(Q')$, donc, d'après la proposition 2.3, $flat_{Q'}(\omega') \in \mathcal{DM}(\overline{Q'})$. La congruence $\overline{Q'}_{|flat_{Q'}(\omega')} \equiv_p \overline{Q}_{|\omega}$ permet alors d'en déduire que l'on a également $\omega \in \mathcal{DM}(\overline{Q})$ (2). Les termes $\overline{Q'}_{|flat_{Q'}(\omega')}$ et $\overline{Q}_{|\omega}$ sont ainsi égaux à leurs propres formes plates (voir remarque précédente La congruence $\overline{Q'}_{|flat_{Q'}(\omega')} \equiv_p \overline{Q}_{|\omega}$ peut donc aussi s'écrire $\overline{\overline{Q'}_{|flat_{Q'}(\omega')}} \equiv_p \overline{\overline{Q}_{|\omega}}$, ce qui entraîne $\overline{Q'}_{|flat_{Q'}(\omega')} =_{AC} \overline{Q}_{|\omega}$, et, en combinant avec (1), on obtient $\sigma' \in CSUC_{AC}(\overline{Q}_{|\omega}, l)$ (3). De (2) et (3), on déduit l'étape d'inférence :
$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{InduceAC}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \mathcal{RE}_{ind}(Q,<)\sigma'} (\overline{Q}[\overline{r}]_\omega)\sigma'$ (4).
Souvenons-nous maintenant que l'on a $\overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')} \equiv_p \overline{Q}[\overline{r}]_\omega$ (5), et, d'après la proposition 2.1, on a également l'égalité $\overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')} = \overline{Q'[r]_{\omega'}}$ (6). De (4), (5), (6), et du fait que σ' est une substitution constructeur (donc telle que l'image est constituée de termes non variadiques), on obtient ainsi :
$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{InduceAC}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \mathcal{RE}_{ind}(Q,<)\sigma} \overline{Q'[r]_{\omega'}\sigma'}$ et le résultat est démontré.

- $2 \Rightarrow 1$:
  Supposons : $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{InduceAC}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{RE_{ind}(Q)\sigma\}} (\overline{Q}[\overline{r}]_\omega)\sigma$.
  Par définition de la règle **InduceAC**, on a $\omega \in \mathcal{DM}(\overline{Q})$ et $\sigma \in CSUC_{AC}(\overline{Q}_{|\omega}, l)$ (1).
  D'après le lemme 2.4, il existe une égalité Q' et une position $\omega' \in \mathcal{D}om(Q')$, telles que $\overline{Q'} \equiv_p \overline{Q}$ (2), $\overline{Q}_{|\omega} = \overline{Q'}_{|flat_{Q'}(\omega')}$ (3), et $\overline{Q}[\overline{r}]_\omega \equiv_p \overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')}$ (4). Maintenant, comme $\omega \in \mathcal{DM}(\overline{Q})$, et d'après l'égalité (3), on tire $flat_{Q'}(\omega') \in \mathcal{DM}(\overline{Q'})$. D'après la proposition 2.3, cela entraîne alors $\omega' \in \mathcal{DM}(Q')$ (5). (1), (3), (5), et le lemme 2.5 permettent alors d'en déduire la relation $\sigma \in CSUC_{AC}(Q'_{|\omega'}, l)$ (6). De (2), (5) et (6), on en déduit ainsi l'étape d'inférence :
  $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathbf{Induce}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\mathcal{RE}_{ind}(Q)\sigma\}} (Q'[r]_{\omega'})\sigma$ (7). Maintenant, du fait que σ est une substitution constructeur, on peut écrire $\overline{(Q'[r]_{\omega'})\sigma} = \overline{(Q'[r]_{\omega'})}\sigma$. Et, d'après la proposition 2.1, on en déduit $\overline{(Q'[r]_{\omega'})\sigma} = (\overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')})\sigma$. D'où, par (4), $\overline{(Q'[r]_{\omega'})\sigma} \equiv_p (\overline{Q}[\overline{r}]_\omega)\sigma$, et, en combinant avec (7) et l'hypothèse, le résultat est démontré.

- **Orient** : clair.
- **Push** : clair.
- **Rewrite** :
  - $1 \Rightarrow 2$:
    Supposons que :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathbf{Rewrite}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'$$

    alors $Q \rightarrow_{\mathcal{RE}/AC} Q'$, donc, d'après la proposition 2.2, $\overline{Q} \rightarrow_{\overline{\mathcal{RE}}/\equiv_p} \overline{Q'}$, et on obtient l'étape de dérivation suivante :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{RewriteAC}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} \overline{Q'}$$

  - $2 \Rightarrow 1$: symétrique.
- **Trivial**

– $1 \Rightarrow 2$:
Supposons que l'on ait l'étape de dérivation suivante :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} q_1 \approx q_2 \rightarrowtail_{\mathbf{Trivial}} \diamond$$

alors $q_1 =_{AC} q_2$, donc, $\overline{q_1} \equiv_p \overline{q_2}$, et cela implique immédiatement l'étape de dérivation :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{q_1} \approx \overline{q_2} \rightarrowtail_{\mathbf{TrivialAC}} \diamond$$

– $2 \Rightarrow 1$: symétrique.

□

Théorème 2.2 *Soit $E = A(\mathcal{V})$.*

1. *Si $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathsf{IndNarrowMod}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} R$, alors $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathsf{IndNarrowModA}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} \overline{R}$.*

2. *Si $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathsf{IndNarrowModA}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} \overline{R}$, alors il existe R' tel que $R' =_A R$, et $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathsf{IndNarrowMod}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}'_2} R'$*

Preuve. – **Induce**.

– $1 \Rightarrow 2$:
Supposons l'étape d'inférence : $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathbf{Induce}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\mathcal{RE}_{ind}(Q')\sigma'\}} (Q'[r]_{\omega'})\sigma'$. Par définition, $\omega' \in \mathcal{DM}(Q')$, et $\sigma' \in CSUC_A(Q'_{|\omega'}, l)$. Sachant que $\omega' \in \mathcal{DM}(Q')$, le lemme 2.6 permet alors d'en déduire $\sigma' \in CSUC_A(\overline{Q'}_{|flat_{Q'}(\omega')}, l)$ (1). Posons $\omega = flat_{Q'}(\omega')$. Par définition de **Induce**, $Q' =_A Q$, donc $\overline{Q'} = \overline{Q}$ (2) . Observons que l'on a $\omega' \in \mathcal{DM}(Q')$, donc, d'après la proposition 2.3, et par définition de ω, $\omega \in \mathcal{DM}(\overline{Q'})$, et d'après l'égalité (2), on a également $\omega \in \mathcal{DM}(\overline{Q})$ (3). Remarquons en outre que (1) et (2) entraînent $\sigma' \in CSUC_A(\overline{Q}_{|\omega}, l)$ (4). De (3), (4), et en observant que ω est une position aplatie, on déduit l'étape d'inférence :
$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{InduceA}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \mathcal{RE}_{ind}(Q,<)\sigma'} (\overline{Q}[\overline{r}]_\omega)\sigma'$ (5). Maintenant, par définition de ω, et d'après (2), on peut écrire $\overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')} = \overline{Q}[\overline{r}]_\omega$ (6), et, d'après la proposition 2.1, on a également l'égalité $\overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')} = \overline{Q'[r]_{\omega'}}$ (7). De (5), (6), (7), et du fait que σ' est une substitution constructeur (donc telle que l'image est constituée de termes non variadiques), on obtient ainsi :
$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{InduceA}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \mathcal{RE}_{ind}(Q,<)\sigma} \overline{Q'[r]_{\omega'}\sigma'}$ et le résultat est démontré.

– $2 \Rightarrow 1$:
Supposons : $\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{InduceA}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2 \cup \{\mathcal{RE}_{ind}(Q)\sigma\}} (\overline{Q}[\overline{r}]_\omega)\sigma$. Par définition de la règle **InduceA**, il existe une position aplatie $\omega \in \mathcal{DM}(\overline{Q})$ et $l \rightarrow r \in \mathcal{RE}_1$, telles que $\sigma \in CSUC_A(\overline{Q}_{|\omega}, l)$ (1). D'après la définition 2.5 et le lemme 2.2, il existe une égalité Q' et une position $\omega' \in \mathcal{D}om(Q')$, telles que $\overline{Q'} = \overline{Q}$ (qui peut également s'écrire $Q' =_A Q$) (2) et $\omega = flat_{Q'}(\omega')$ (3). Maintenant, comme $\omega \in \mathcal{DM}(\overline{Q})$, et d'après l'égalité (3), on tire $flat_{Q'}(\omega') \in \mathcal{DM}(\overline{Q'})$. D'après la proposition 2.3, cela entraîne alors $\omega' \in \mathcal{DM}(Q')$ (4). (1), (2), (3), (4), et le lemme 2.6 permettent alors d'en déduire la relation $\sigma \in CSUC_A(Q'_{|\omega'}, l)$ (5). De (2), (4) et (5), on en déduit ainsi l'étape d'inférence :

$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathbf{Induce}} \Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma\}} (Q'[r]_{\omega'})\sigma$ (6). Maintenant, du fait que σ est une substitution constructeur, on peut écrire $\overline{(Q'[r]_{\omega'})\sigma} = \overline{(Q'[r]_{\omega'})}\sigma$. Et, d'après la proposition 2.1, on en déduit $\overline{(Q'[r]_{\omega'})\sigma} = (\overline{Q'}[\overline{r}]_{flat_{Q'}(\omega')})\sigma$. D'où, par (2) et (3), $\overline{(Q'[r]_{\omega'})\sigma} = (\overline{Q}[\overline{r}]_{\omega})\sigma$, et, en combinant avec (6), le résultat est démontré.

- **Orient** : clair.
- **Push** : clair.
- **Rewrite** :
  - $1 \Rightarrow 2$:
    Supposons que :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q \rightarrowtail_{\mathbf{Rewrite}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} Q'$$

    alors $Q \rightarrow_{\mathcal{RE}/A} Q'$, donc, d'après la proposition 2.2, $\overline{Q} \rightarrow_{\overline{\mathcal{RE}}} \overline{Q'}$, et on obtient l'étape de dérivation suivante :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{Q} \rightarrowtail_{\mathbf{RewriteA}} \Gamma'_1|\Gamma'_2 \vdash_{\mathcal{RE}'_1|\mathcal{RE}'_2} \overline{Q'}$$

  - $2 \Rightarrow 1$: symétrique.
- **Trivial**
  - $1 \Rightarrow 2$:
    Supposons que l'on ait l'étape de dérivation suivante :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} q_1 \approx q_2 \rightarrowtail_{\mathbf{Trivial}} \diamond$$

    alors $q_1 =_A q_2$, donc, $\overline{q_1} = \overline{q_2}$, et cela implique immédiatement l'étape de dérivation :

$$\Gamma_1|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} \overline{q_1} \approx \overline{q_2} \rightarrowtail_{\mathbf{TrivialA}} \diamond$$

  - $2 \Rightarrow 1$: symétrique.

□

2.5 Autres exemples

Considérons la spécification *Binôme* présentée à la figure 2.3.

2.5.1 Exemples AC

Nous traitons dans ce paragraphe, et avec notre méthode, des exemples qui ont déjà été présentés dans [BER 97]. On suppose que $+$ et $*$ sont AC.

1. Nous allons démontrer la conjecture :

$$exp(X_1 * X_2, X_3) \approx exp(X_1, X_3) * exp(X_2, X_3)$$

   (a) Γ'_1 = ensemble des axiomes de la spécification dont le symbole de tête du membre gauche est $+$, $*$ ou exp ;

   (b) $\mathcal{RE}_1$ le système de réécriture obtenu par remplacement dans Γ'_1 du symbole $\approx$ par $\rightarrow$;

   (c) $\Gamma_1 = \Gamma'_1 \cup AC(+,*) \quad \Gamma_2 = \{L(\approx), NI,\ Noeth(<_4, \mathcal{T}(\Sigma)^4)\}$

   (d) $Q = exp(X_1 * X_2, X_3) \approx exp(X_1, X_3) * exp(X_2, X_3)$

- **Spécification :** binôme
- **Sortes :** nat, list ;
- **Constructeurs :**

$$0 : \to Nat \quad s : \ Nat \to Nat \quad Nil : \to list \quad :: : \ Nat \times list \to list$$

- **Fonctions définies :**

$$+ : Nat \times Nat \to Nat \quad * : Nat \times Nat \to Nat \quad exp : Nat \times Nat \to Nat \quad bin : Nat \to list$$
$$ps : list \times list \to list \quad seq : Nat \times list \to Nat \quad <> : list \times list \to list$$

- **Axiomes** :

$$x + 0 \approx x \qquad x * 0 \approx 0 \qquad exp(x, 0) \approx s(0)$$
$$x + s(y) \approx s(x + y) \qquad x * s(y) \approx x * y + x \qquad exp(x, s(y)) \approx x * exp(x, y)$$
$$x * (y + z) \approx x * y + x * z$$

$$seq(x, Nil) \approx 0 \qquad x <> Nil \approx x \qquad rev(Nil) \approx Nil$$
$$seq(x, y :: z) \approx y + (x * seq(x, z)) \qquad x <> y :: z \approx y :: (z <> x) \qquad rev(x :: l) \approx rev(l) <> (x :: Nil)$$

$$ps(Nil, l) \approx l; \quad ps(l, Nil) \approx l \qquad bin(0) \approx s(0) :: Nil$$
$$ps(x_1 :: l_1, x_2 :: l_2) \approx \qquad bin(s(x)) \approx$$
$$(x_1 + x_2) :: ps(l_1, l_2) \qquad ps(0 :: bin(x), bin(x))$$

FIG. 2.3 – **Binôme**

Le but est donc :

$\boxed{\Gamma_1 | \Gamma_2 \vdash_{\emptyset|\emptyset} Q}$

L'application répétée de **Push**$_1$**AC** et **OrientAC** permet d'obtenir :

$\boxed{\emptyset, \ AC(+, *) | \Gamma_2 \vdash_{\mathcal{RE}_1|\emptyset} Q}$

On peut maintenant appliquer **InduceAC**, car $\mathcal{RE}_1$ est terminant et suffisamment complet modulo $AC(+, *)$. Les positions *définie-maximales* dans Q sont 1.1, 2.1 et 2.2. Observons maintenant que le but reste équivalent par permutation des variables X_1 et X_2. Il reste donc deux possibilités : surréduire à une position où le symbole $*$ apparaît, ou à une position où c'est le symbole *exp* qui apparaît. On préfère la seconde possibilité, car cela va entraîne davantage de réductions ultérieures : on choisit par exemple la position 2.1. Il faut donc déterminer l'ensemble $CSUC_{AC(+,*)}(exp(X_1, X_3), l)$ pour toute règle de réécriture $l \to r$ de $\mathcal{RE}_1$. D'après le lemme 2.4 et le corollaire 2.2 de la partie II, on peut se restreindre aux

– Simplify :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E \cup \{s' \approx t\}, H, G >} \; s \rightarrow_{(\mathcal{R} \cup H)/G} s'$$

– Simplify2 :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E \cup \{s' \approx t\}, H, G >} \; s \overset{*}{\leftrightarrow}_{\mathcal{R} \cup \mathcal{E}} s', s \geqslant s'$$

– Delete :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E, H, G >} \; s \overset{*}{\leftrightarrow}_{G} t$$

– Delete2 :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E, H, G >} \; s \overset{*}{\leftrightarrow}_{\mathcal{R} \cup \mathcal{E}} t$$

– Expand :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E \cup Expd_u(s,t), H \cup \{s \rightarrow t\}, G >} \; u \in B(s), \; s > t$$

– Expand2 :

$$\frac{< E \cup \{s \approx t\}, H, G >}{< E \cup Expd2_{u,v}(s,t), H, G \cup \{s \approx t\} >} \; u \in B(s), \; v \in B(t), \; s \gtrless t$$

FIG. 2.4 – Système d'inférence étendu de Aoto

règles telles que $l(\varepsilon) = exp$ et on obtient :

l	$CSUC_{AC(+,*)}(exp(X_1, X_3), l)$
$exp(x, 0)$	$\sigma_1 = \{X_1 \rightarrow Y_1; x \rightarrow Y_1; X_3 \rightarrow 0\}$
$exp(x, s(y))$	$\sigma_2 = \{X_1 \rightarrow Y_1; x \rightarrow Y_1; y \rightarrow Y_3; X_3 \rightarrow s(Y_3)\}$

Après normalisation, cela nous conduit à prouver les deux séquents :

$\emptyset, \; AC(+,*) \mid \Gamma_2 \vdash_{\mathcal{RE}_1 \mid \mathcal{RE}_{ind}(Q)\sigma_1} s(0) \approx s(0)$
$\emptyset, \; AC(+,*) \mid \Gamma_2 \vdash_{\mathcal{RE}_1 \mid \mathcal{RE}_{ind}(Q)\sigma_2} Y_1 * X_2 * exp(Y_1 * X_2, Y_3) \approx Y_1 * exp(Y_1, Y_3) * X_2 * exp(X_2, Y_3)$

TrivialAC élimine le premier sous-but, et comme $(Y_1, X_2, Y_3) <_3 (Y_1, X_2, s(Y_3))$, on peut appliquer **Rewrite**$_2$ au second, on obtient :

$\emptyset, \; AC(+,*) \mid \Gamma_2 \vdash_{\mathcal{RE}_1 \mid \mathcal{RE}_{ind}(Q)\sigma_2} Y_1 * X_2 * exp(Y_1, Y_3) * exp(X_2, Y_3) \approx Y_1 * exp(Y_1, Y_3) * X_2 * exp(X_2, Y_3)$

que la règle **TrivialAC** élimine.
Supposons maintenant que l'on essaie de démontrer cette conjecture avec la méthode présentée dans [AOT 06]. Pour cela, il est nécessaire d'utiliser une extension du système présenté à la figure 3.2 de la partie I (voir figure 2.4). La correction de ce système étendu repose sur le théorème suivant :

Théorème 2.3 *[AOT 06] Soit $\mathcal{R}$ un système de réécriture suffisamment complet, E, $\mathcal{E}$ des ensembles d'égalités, $\leqslant$ un quasi-ordre de réduction tel que $\mathcal{R} \subseteq \leqslant$. On suppose que les égalités de $\mathcal{E}$ sont des théorèmes inductifs de $\mathcal{R}$. Si il existe des ensembles H, G, tels que $< E, \emptyset, \emptyset > \overset{*}{\rightarrow} < \emptyset, H, G >$, alors les égalités de E sont des théorèmes inductifs de $\mathcal{R}$.*

Pour démontrer la conjecture précédente à l'aide de ce système, il faut poser $\mathcal{E} = AC(+,*)$, $\mathcal{R} = \mathcal{RE}_1$, et mettre le but sous la forme :

$$\boxed{< exp(X_1 * X_2, X_3) \approx exp(X_1, X_3) * exp(X_2, X_3), \emptyset, \emptyset >}$$

La seule règle que l'on puisse appliquer ici est *Expand*. Comme $exp(X_1 * X_2, X_3) > exp(X_1, X_3) * exp(X_2, X_3)$, il est nécessaire d'effectuer la surréduction au niveau du sous-terme $X_1 * X_2$ du but. L'un des sous-buts obtenus sera ainsi :

$$\boxed{< exp(X_1 * X_2 + X_1, X_3) \approx exp(X_1, X_3) * exp(s(X_2), X_3), \emptyset, \emptyset >}$$

Il est de nouveau nécessaire d'appliquer *Expand* à ce dernier, et on sera amené à prouver :

$$\boxed{< exp(X_1 * X_2 + X_1 + X_1, X_3) \approx exp(X_1, X_3) * exp(s(s(X_2)), X_3), \emptyset, \emptyset >}$$

Finalement, on obtient la suite infinie de sous-buts :

$< exp(X_1 * X_2 + X_1, X_3) \approx exp(X_1, X_3) * exp(s(X_2), X_3), \emptyset, \emptyset >$
$< exp(X_1 * X_2 + X_1 + X_1, X_3) \approx exp(X_1, X_3) * exp(s(s(X_2)), X_3), \emptyset, \emptyset >$
$< exp(X_1 * X_2 + X_1 + X_1 + X_1, X_3) \approx exp(X_1, X_3) * exp(s(s(s(X_2))), X_3), \emptyset, \emptyset >$
$\vdots$

et la tentative échoue.

2. Nous allons maintenant démontrer la conjecture :

$$seq(X_1, ps(L_2, L_3)) \approx seq(X_1, L_2) + seq(X_1, L_3)$$

Introduisons d'abord les notations suivantes :

(a) Γ'_1 = ensemble des axiomes de la théorie dont le symbole de tête du membre gauche est $+$, $*$, *seq* ou *ps* ;

(b) $\mathcal{RE}_1$ le système de réécriture obtenu par remplacement dans Γ'_1 du symbole $\approx$ par $\rightarrow$;

(c) $\Gamma_1 = \Gamma'_1 \cup AC(+,*) \quad \Gamma_2 = \{L(\approx), NI,\ Noeth(<_3, \mathcal{T}(\Sigma)^3)\}$;

(d) $Q = seq(X_1, ps(L_2, L_3)) \approx seq(X_1, L_2) + seq(X_1, L_3)$.

Le but est donc :

$\boxed{\Gamma_1 | \Gamma_2 \vdash_{\emptyset|\emptyset} Q}$

L'application répétée de **Push**$_1$**AC** et **OrientAC** permet d'obtenir :

$\boxed{\emptyset,\ AC(+,*) | \Gamma_2 \vdash_{\mathcal{RE}_1|\emptyset} Q}$

On peut maintenant appliquer **InduceAC**, car $\mathcal{RE}_1$ est terminant et suffisamment complet modulo $AC(+,*)$. Les positions *définie-maximales* dans Q sont 1.2, 2.1 et 2.2. Surréduire à la position 1.2 va créer davantage de réductions ultérieures qu'aux positions 2.1 ou 2.2. On choisit donc la position 1.2, et il faut alors déterminer l'ensemble $CSUC_{AC}(ps(L_2, L_3), l)$ pour toute règle de réécriture $l \rightarrow r$ de $\mathcal{RE}_1$. On peut se restreindre aux règles telles que $l(\varepsilon) = ps$, et on obtient :

l	$CSUC_{AC(+,*)}(ps(L_2, L_3), l)$
$ps(Nil, l)$	$\sigma_1 = \{L_2 \rightarrow Nil; L_3 \rightarrow M_3; l \rightarrow M_3\}$
$ps(l, Nil)$	$\sigma_2 = \{L_2 \rightarrow M_2; L_3 \rightarrow Nil; l \rightarrow M_2\}$
$ps(x_1 :: l_1, x_2 :: l_2)$	$\sigma_3 = \{\ L_2 \rightarrow X_2 :: M_2\ ;\ L_3 \rightarrow X_3 :: M_3$ $x_1 \rightarrow X_2\ ;\ l_1 \rightarrow M_2\ ;\ x_2 \rightarrow X_3\ ;\ l_2 \rightarrow M_3\ \}$

On obtient ainsi les sous-buts suivants :

$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(Q)\sigma_1}$	$seq(X_1, M_3) \approx seq(X_1, M_3)$
$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(Q)\sigma_2}$	$seq(X_1, M_2) \approx seq(X_1, M_2)$
$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(Q)\sigma_3}$	$X_2 + X_3 + X_1 * seq(X_1, ps(M_2, M_3))$ $\approx X_2 + X_1 * seq(X_1, M_2) + X_3 + X_1 * seq(X_1, M_3)$

L'application de **TrivialAC** élimine les deux premiers.
Comme $(X_1, M_2, M_3) <_3 (X_1, X_2 :: M_2, X_3 :: M_3)$, $\mathbf{Rewrite}_2$ peut être appliquée sur le troisième. On obtient alors :

$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(Q)\sigma_3}$	$X_2 + X_3 + X_1 * (seq(X_1, M_2) + seq(X_1, M_3))$ $\approx X_2 + X_1 * seq(X_1, M_2) + X_3 + X_1 * seq(X_1, M_3)$

Après une application de $\mathbf{Rewrite}_1\mathbf{AC}$, on obtient :

$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(Q)\sigma_3}$	$X_2 + X_3 + X_1 * seq(X_1, M_2) + X_1 * seq(X_1, M_3)$ $\approx X_2 + X_1 * seq(X_1, M_2) + X_3 + X_1 * seq(X_1, M_3)$

et **TrivialAC** peut éliminer ce dernier sous-but.

3. Nous allons démontrer ici la conjecture :

$$exp(s(X_1), X_2) \approx seq(X_1, bin(X_2))$$

en utilisant l'égalité démontrée au 2. Introduisons d'abord les notations suivantes :

(a) Γ'_1 = ensemble des axiomes de la théorie dont le symbole de tête du membre gauche est $+$, $*$, *exp*, *seq*, *ps*, *bin* ;

(b) $\mathcal{RE}_1$ le système de réécriture obtenu par remplacement dans Γ'_1 du symbole $\approx$ par $\rightarrow$;

(c) $\mathcal{RE}_2 = \{seq(X_1, ps(L_2, L_3)) \approx seq(X_1, L_2) + seq(X_1, L_3)\}$

(d) $\Gamma_1 = \Gamma'_1 \cup AC(+,*) \quad \Gamma_2 = \{L(\approx), NI,\ Noeth(<_2, \mathcal{T}(\Sigma)^2)$
$seq(X_1, ps(L_2, L_3)) \rightarrow seq(X_1, L_2) + seq(X_1, L_3)\}$

(e) $Q = exp(s(X_1), X_2) \approx seq(X_1, bin(X_2))$

Le but est donc :
$\boxed{\Gamma_1|\Gamma_2 \vdash_{\emptyset|\mathcal{RE}_2} Q}$
L'application répétée de $\mathbf{Push}_1\mathbf{AC}$ et **OrientAC** permet d'obtenir :
$\boxed{\emptyset,\ AC|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_2} Q}$
On peut maintenant appliquer **InduceAC**, car $\mathcal{RE}_1$ est terminant et suffisamment complet modulo $AC(+,*)$. Les positions *définie-maximales* dans Q sont 1 and 2.2. Les sous-buts étant ici les mêmes si on surréduit à l'une ou l'autre de ces occurences, on choisit donc arbitrairement l'occurence 2.1. Il faut donc déterminer l'ensemble $CSUC_{AC(+,*)}(bin(X_2), l)$ pour toute règle de réécriture $l \rightarrow r$ de $\mathcal{RE}_1$. On peut se restreindre aux règles telles que $l(\varepsilon) = ps$, et on obtient :

l	$CSUC_{AC(+,*)}(bin(X_2), l)$
$bin(0)$	$\sigma_1 = \{X_2 \rightarrow 0\}$
$bin(s(x))$	$\sigma_2 = \{X_2 \rightarrow s(Y_2), x \rightarrow s(Y_2)\}$

On obtient ainsi les deux sous-buts :

$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma_1\}} s(0) \approx s(0)$
$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma_2\}} s(X_1) * exp(s(X_1), Y_2) \approx seq(X_1, ps(0 :: bin(Y_2), bin(Y_2)))$

TrivialAC élimine le premier, et on termine la preuve de la façon suivante

$\rightarrowtail$**Rewrite**$_2$ $_{AC}$	$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma_2\}}$	$s(X_1) * exp(s(X_1), Y_2) \approx$ $seq(X_1, 0 :: bin(Y_2)) + seq(X_1, bin(Y_2))$
$\rightarrowtail$**Rewrite**$_1$ $_{AC}$	$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma_2\}}$	$s(X_1) * exp(s(X_1), Y_2) \approx$ $X_1 * seq(X_1, bin(Y_2)) + seq(X_1, bin(Y_2))$
$\rightarrowtail$**Rewrite**$_2$ $_{AC}$	$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma_2\}}$	$s(X_1) * exp(s(X_1), Y_2) \approx$ $X_1 * exp(s(X_1), Y_2) + exp(s(X_1), Y_2)$
$\rightarrowtail$**Rewrite**$_1$ $_{AC}$	$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma_2\}}$	$X_1 * exp(s(X_1), Y_2) + exp(s(X_1), Y_2) \approx$ $X_1 * exp(s(X_1), Y_2) + exp(s(X_1), Y_2)$
$\rightarrowtail$**Trivial** $_{AC}$	$\emptyset,\ AC(+,*)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_2\cup\{\mathcal{RE}_{ind}(Q)\sigma_2\}}$	$\diamond$

2.5.2 Exemple A

Considérons la spécification **liste** donnée à la figure 1.2 (chapitre 1), et supposons que $<>$ est associatif. Nous allons démontrer la conjecture :

$$rev(L <> M) \approx rev(M) <> rev(L)$$

On introduit d'abord les notations suivantes :

1. Γ'_1 = Ensemble des axiomes de la théorie ;
2. $\mathcal{RE}_1$ le système de réécriture obtenu par remplacement dans Γ'_1 du symbole $\approx$ par $\rightarrow$;
3. $\Gamma_1 = \Gamma'_1 \cup A(<>) \quad \Gamma_2 = \{L(\approx), NI,\ Noeth(<_2, \mathcal{T}(\Sigma)^2)\}$;
4. $Q = rev(L <> M) \approx rev(M) <> rev(L)$

Le but est donc :

$\boxed{\Gamma_1|\Gamma_2 \vdash_{\emptyset|\emptyset} Q}$

L'application répétée de **Push**$_1$**A** et **OrientA** permet d'obtenir :

$\boxed{\emptyset,\ A(<>)|\Gamma_2 \vdash_{\mathcal{RE}_1|\emptyset} Q}$

On peut maintenant appliquer **InduceA**, car $\mathcal{RE}_1$ est terminant et suffisamment complet modulo $A(<>)$. Les positions *définie-maximales* dans Q sont 1.1, 2.1 et 2.2. Les sous-buts sont les mêmes si on surréduit à la position 1.1 ou 2.2, et la surréduction à la position 2.1 entraîne moins de réductions ultérieures. On choisit donc l'occurence 1.1. Il faut donc déterminer l'ensemble $CSUC_{A(<>)}(L <> M, l)$ pour toute règle de réécriture $l \rightarrow r$ de $\mathcal{RE}_1$. On peut se restreindre aux règles telles que $l(\varepsilon) =<>$, et on obtient :

l	$CSUC_{A(<>)}(L <> M, l)$
$Nil <> l$	$\sigma_1 = \{L \rightarrow Nil; M \rightarrow M_1; l \rightarrow M_1\}$
$x :: l <> m$	$\sigma_2 = \{L \rightarrow X_1 :: L_1; M \rightarrow M_1; x \rightarrow X_1; l \rightarrow L_1; m \rightarrow M_1\}$

Après normalisation, on est donc amené à démontrer les sous-buts suivants :

$\emptyset, A(<>)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(Q)\sigma_1} rev(M_1) \approx rev(M_1)$
$\emptyset, A(<>)\|\Gamma_2 \vdash_{\mathcal{RE}_1\|\mathcal{RE}_{ind}(Q)\sigma_2} rev(L_1 <> M_1) <> (X_1 :: Nil) \approx rev(M_1) <> rev(L_1) <> (X_1 :: Nil))$

TrivialA élimine ce premier sous-but.

Sachant que $(L_1, M_1) <_2 (X_1 :: L_1, M_1)$, **Rewrite$_2$A** peut être appliquée au second. On obtient alors :

$$\boxed{\begin{array}{rl} \emptyset, A(<>)|\Gamma_2 \vdash_{\mathcal{RE}_1|\mathcal{RE}_{ind}(Q)\sigma_2} & rev(M_1) <> rev(L_1) <> (X_1 :: Nil) \\ & \approx rev(M_1) <> rev(L_1) <> (X_1 :: Nil) \end{array}}$$

et **TrivialA** élimine ce dernier sous-but.

Conclusion générale

Dans la partie I, nous avons fait la différence entre conséquence sémantique et conséquence inductive d'une théorie : la première étant satisfaite pour tout modèle de cette théorie, et la seconde pour tout modèle de Herbrand de cette théorie. D'après le théorème de Birkhoff, il est possible de construire un système d'inférence capable de dériver une égalité donnée à partir d'une théorie Γ si, et seulement si cette égalité est une conséquence sémantique de cette théorie. Cela s'écrit habituellement :

$$\Gamma \models E \quad \Leftrightarrow \quad \Gamma \vdash E$$

Ce théorème a été étendu aux cas des égalités conditionnelles par [SEL 72]. Ce théorème fait ainsi la distinction entre deux approches possibles pour démontrer qu'une proposition est une conséquence sémantique d'une théorie Γ : une approche dite *en théorie des modèles* et une autre dite *en théorie de la preuve*.
Comme il n'existe pas de théorème analogue à celui de Birkhoff dans le cas où l'on cherche à démontrer qu'une proposition est une conséquence inductive, il a donc semblé au départ que la seule voie possible pour y parvenir était de démontrer sa valididité dans tout modèle de Herbrand de cette théorie. On a rappelé que si la proposition considérée est une clause positive, il suffit de montrer sa validité dans le modèle initial, et celui-ci est isomorphe au quotient de l'ensemble des termes clos $\mathcal{T}(\Sigma)$ par une congruence. C'est précisément sur ces observations que se sont fondées les premières méthodes de recherche de preuves de théorèmes inductifs. Montrer qu'une égalité $s \approx t$ est une conséquence inductive d'une théorie Γ revient en effet à montrer que s et t sont dans la même classe de congruence, et il est très intéressant d'observer qu'il existe des algorithmes capables dans certains cas de calculer ces classes de congruence. C'est le cas par exemple lorsque la théorie considérée Γ est un ensemble d'égalités (éventuellement conditionnelles). La démarche a été alors souvent de déterminer un système de réécriture confluent et terminant $\mathcal{R}$ permettant de simuler Γ. La congruence de deux termes se ramène alors à l'égalité de leur forme normale. Kaplan [KAP 84a] a par ailleurs montré que si $\mathcal{R}$ est confluent et terminant, il existe un algorithme de calcul de la forme normale si, et seulement si la relation $\rightarrow_{\mathcal{R}}$ est décidable. Dans le cas où Γ est un ensemble d'égalités non conditionnelles, le moyen le plus connu pour déterminer ce système de réécriture $\mathcal{R}$ est la procédure de complétion de Knuth Bendix. Nous avons rappelé que Huet et Hullot ont observé que l'on pouvait appliquer une version légèrement restreinte de cette procédure à l'ensemble constitué des axiomes de la théorie Γ ainsi que des conjectures à démontrer. Fribourg a ensuite démontré que l'on pouvait restreindre cette procédure de façon encore beaucoup plus drastique en ne considérant que les égalités obtenues par la superposition des axiomes au niveau de positions inductivement réductibles des conjectures. En supposant que Γ est déjà mis sous la forme d'un système de réécriture confluent et terminant, il obtient un système de réécriture $\mathcal{R}$ tel que $\rightarrow_{\mathcal{R}} \subseteq \rightarrow_{\Gamma} \circ \overset{*}{\rightarrow}_{\Gamma\cup\mathcal{R}} \circ \overset{*}{\leftarrow}_{\Gamma\cup\mathcal{R}}$. Il en déduit alors par récurrence noethérienne que les conjectures initiales sont des théorèmes inductifs. Aoto [AOT 06] mentionne que Koike et

Toyama [KOI 00] ont révélé que cette démarche caractérise la méthode dite de récurrence par réécriture développée jusqu'alors.

Notre approche est fondamentalement différente de celles rappelées plus haut. En effet, montrer qu'une conjecture est une conséquence inductive d'une théorie donnée avec notre méthode revient à faire une preuve en calcul des séquents modulo. Bien qu'il n'existe pas de théorème semblable à celui de Birkhoff pour les conséquences inductives, nous pensons cependant avoir donné une autre compréhension de la récurrence par réécriture : une compréhension *en théorie de la preuve* . Récapitulons les étapes essentielles de notre parcours. Nous avons d'abord défini un système de recherche de preuve par récurrence démontré correct si les preuves s'effectuent dans une théorie dont les axiomes peuvent être orientés en un système de réécriture terminant sur les termes clos et suffisamment complet par rapport à un ensemble de constructeurs libres. Nous avons également montré que notre système est capable de montrer dans certains cas que des conjectures ne sont pas des théorèmes inductifs. Un avantage de notre méthode est que le schéma de récurrence s'obtient par surréduction au niveau d'une position *définie-maximale* du but, et nous avons montré sur des exemples simples que cette approche permet de réduire significativement le nombre de cas à considérer (donc de limiter les risques de divergence éventuelle). Nous avons ensuite optimisé notre système en rendant les conditions d'application de l'hypothèse de récurrence moins strictes et plus commodes à vérifier. Nous l'avons en outre étendu au cadre des théories où les termes sont considérés modulo une axiomatisation équationnelle E, avec un éclairage particulier dans le cas où E est une théorie associative, ou associative-commutative. Une caractéristique originale de notre approche est que la théorie E est à priori très générale. Un des points clés de la méthode est que l'on considère uniquement les unificateurs qui ont leurs images dans l'ensemble des termes constructeurs, ce qui est un avantage significatif dans le cas AC où l'unification est de complexité doublement exponentielle. Cela permet aussi d'envisager des preuves dans des théories infinitaires.

Pour l'instant, notre système n'est pas en mesure d'effectuer des preuves par récurrence dans des théories conditionnelles. Une difficulté pour cela tient au fait que la correction de notre système est fondée sur les lemmes de relevage (lemme 2.1 et lemme 2.2 de la partie II) qui s'appliquent dans le cas où la surréduction est effectuée par un ensemble de règles non conditionnelles. Il nous semble cependant possible dans un futur proche d'adapter notre système afin de le rendre capable d'effectuer des preuves dans une axiomatisation conditionnelle, à partir du moment où celle-ci contient un sous-ensemble de règles non conditionnelles que l'on peut orienter en un système de réécriture convergent sur les termes clos et suffisamment complet sur des constructeurs libres. On pourrait alors envisager de réaliser l'étape de récurrence au moyen de ces règles uniquement, et d'utiliser les autres pour simplifier les conjectures à prouver. Notons que, dans ce cadre, celles-ci ne seraient plus nécessairement des égalités, mais pourraient également être des clauses.

Observons que l'un des principaux atouts de notre méthode nous semble justement être la puissance du processus de simplification : nous avons ainsi pu observer comment, dans le contexte de la déduction modulo, les hypothèses de récurrence qui apparaissent à partir de buts équationnels peuvent être internalisées dynamiquement dans la relation de réécriture. Cela n'est pas suffisant pour garantir le succès de la preuve. Il y a deux raisons principales à cela :

- d'une part, la nature des hypothèses de récurrence qui apparaissent au fur et à mesure dépend de la relation bien fondée choisie par l'utilisateur , et le nombre de choix possibles pour cette relation est à priori illimité. En effet, dans les exemples que nous avons

présentés, nous avons toujours pu réaliser la récurrence avec un ordre récursif sur les chemins utilisé pour orienter les axiomes de la spécification. De nombreux exemples, issus des mathématiques en particulier, nous prouvent cependant que le choix de la relation noethérienne le mieux adapté à la résolution d'un problème donné est parfois difficile. Considérons par exemple la fonction 91 de John McCarthy, définie sur les entiers naturels par les règles suivantes :

$$x > 100 \Rightarrow f(x) = x - 10$$
$$x \leq 100 \Rightarrow f(x) = f(f(x + 11))$$

et la conjecture C définie par $\forall x\ ((f(x) = 91) \vee (f(x) = x - 10))$. La relation bien fondée la mieux adaptée, que l'on notera $\prec$, pour montrer que la conjecture C est un théorème inductif est définie par :

$$x \prec y \Leftrightarrow y < x \leqslant 101$$

et on imagine bien que ce n'est pas le premier choix qui vient à l'esprit ...

– d'autre part, on a déjà fait remarquer qu'il est fréquemment nécessaire, même dans des cas simples, d'avoir recours à un lemme intermédiaire, ou de devoir généraliser la formule à démontrer. Par exemple, le système IndNarrow échoue dans la situation décrite à l'exemple 3.5 de la partie I. Il serait donc intéressant de rendre notre système capable d'utiliser des heuristiques lui permettant, au cours d'une preuve, de décider de prouver d'abord un lemme intermédiaire ou une généralisation de la conjecture à démontrer. Reprenons la situation décrite dans ce dernier exemple. En rendant notre système apte à analyser le sous-but $rev(rev(t) <> (h :: nil)) \approx h :: t$ avec l'heuristique appelée Rippling introduite dans [BUN 93], il pourrait proposer, dans un premier temps de démontrer une conjecture intermédiaire de la forme $rev(l <> l') \rightarrow f(l, l', rev(l))$, avec f une méta-variable. Dans un second temps, f serait instanciée en $f(l, l', l'') = rev(l') <> l''$, de telle sorte que cette conjecture deviendrait $rev(l <> l') \approx rev(l') <> rev(l)$. Observons cependant que la démarche décrite ci-dessus nécessite d'utiliser des algorithmes d'unification à l'ordre supérieur, et que ceux-ci ne sont pas déterministes. Des techniques ont néanmoins été proposées pour réduire l'espace de recherche : nous renvoyons le lecteur intéressé à [BUN 99] pour les détails.

En dépit de ces obstacles théoriques incontournables, notre système garde l'avantage décisif de pouvoir construire une preuve dans le calcul des séquents modulo, qu'il est ensuite possible de faire valider par un assistant de preuve (*Coq* par exemple). Plus généralement, la démarche suivie par notre système peut être schématisée de la façon suivante :

1. on part de la proposition inductive à prouver ;
2. le mécanisme de recherche de preuve construit une preuve dans le calcul des séquents modulo ;
3. de cette preuve est déduit un terme de preuve ;
4. son type dans un système de type approprié est la proposition initiale, et la boucle est bouclée.

En définitive, la déduction modulo nous a fourni un cadre théorique grâce auquel nous avons pu élaborer un système de recherche de preuve encore limité en soi, mais qui a le pouvoir significatif de collaborer de façon sûre avec d'autres assistants de preuve : les premiers aidant au choix de la relation noethérienne et des conjectures intermédiaires, et les second à la validation de la preuve ainsi réalisée.

A

Lemmes du calcul des séquents modulo

A.1 Lemmes logiques

Lemme A.7 Pour toutes propositions P, Q, pour tous contextes Γ, Δ et pour tout système de réécriture $\mathcal{RE}$, les séquents $\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta$ et $\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta$ sont équivalents.

Preuve. 1. Supposons $\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta$

Posons P et Q deux propositions et considérons les dérivations suivantes :

Π_1 :

$$\dfrac{\dfrac{\dfrac{}{P \vdash_{\mathcal{RE}} P}\,Ax}{\Gamma, P \vdash_{\mathcal{RE}} P, Q, \Delta}\,w \qquad \dfrac{\dfrac{}{Q \vdash_{\mathcal{RE}} Q}\,Ax}{\Gamma, P, Q \vdash_{\mathcal{RE}} Q, \Delta}\,w}{\Gamma, P, P \Rightarrow Q \vdash_{\mathcal{RE}} Q, \Delta}\,\Rightarrow_l$$

Π_2 :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta \text{ (supposé)}}{\Gamma, P \vdash_{\mathcal{RE}} P \Rightarrow Q, Q, \Delta}\,w$$

Π :

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}\,cut$$

2. Supposons $\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta$

$$\dfrac{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta}\,\Rightarrow_r$$

□

Notation: On définit ainsi les deux nouvelles règles :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta}{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}\ \text{lemme } A.7 \qquad \dfrac{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta}\ \text{lemme } A.7$$

Corollaire A.1 Pour toutes propositions P, Q, pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ tel que $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\{P,Q\})}$, si le séquent $\Gamma \vdash_{\mathcal{RE}} \forall \overrightarrow{x}\ P \Rightarrow Q, \Delta$ est dérivable, on peut alors dériver le séquent :

$$\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta$$

Preuve. Π_1

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P \Rightarrow Q \vdash_{\mathcal{RE}} P \Rightarrow Q}\ Ax}{P \Rightarrow Q, P \vdash_{\mathcal{RE}} Q}\ \text{lemme A.7}}{\forall x\ P \Rightarrow Q, P \vdash_{\mathcal{RE}} Q}\ \forall_l}{\Gamma, \forall x\ P \Rightarrow Q, P \vdash_{\mathcal{RE}} Q, \Delta}\ w$$

Π_2 :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} \forall x\ P \Rightarrow Q, \Delta\ \text{(supposé)}}{\Gamma, P \vdash_{\mathcal{RE}} Q, \forall x\ P \Rightarrow Q, \Delta}\ w$$

Π

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}\ cut$$

□

Notation: On définit ainsi la nouvelle règle :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} \forall x\ P \Rightarrow Q, \Delta}{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}\ \text{corollaire } A.1$$

Lemme A.8 Pour tous contextes Γ, Δ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ des variables libres de $\mathcal{RE} \cup \{P\}$, si le séquent $\Gamma \vdash_{\mathcal{RE}} P, \Delta$ a une preuve, il est alors possible de construire une preuve du séquent $\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta$

Preuve. Par récurrence sur le nombre de règles dans $\mathcal{RE}$.

Cas de base : si $\mathcal{RE}$ est vide :

Considérons les dérivations suivantes :

Π_1 :

$$\dfrac{\dfrac{P\{\overrightarrow{t}/\overrightarrow{x}\} \vdash P\{\overrightarrow{t}/\overrightarrow{x}\}}{\forall \overrightarrow{x} P \vdash P\{\overrightarrow{t}/\overrightarrow{x}\}}\ \forall_l}{\Gamma, \forall \overrightarrow{x} P \vdash P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ w$$

Π_2 :

$$\dfrac{\dfrac{\Gamma \vdash P, \Delta \quad \text{(supposé)}}{\Gamma \vdash \forall \overrightarrow{x}\ P, \Delta}\ \forall_r}{\Gamma \vdash \forall \overrightarrow{x}\ P, P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ w$$

Π_3 :

$$\dfrac{\Pi_1 \quad \Pi_2}{\Gamma \vdash P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ cut$$

Cas de récurrence : si $\mathcal{RE}$ contient $n+1$ règles.

Notons κ l'une des règles de $\mathcal{RE}$, et posons P_κ la proposition canonique associée, et $\mathcal{RE}' = \mathcal{RE} \setminus \{\kappa\}$

Considérons la dérivation suivante :
Π_4 :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash_{\mathcal{RE}} P, \Delta \quad \text{(supposé)}}{\Gamma \vdash_{\mathcal{RE}' \cup \{\kappa\}} P, \Delta}\ \text{par définition}}{\Gamma, P_\kappa \vdash_{\mathcal{RE}'} P, \Delta}\ \textit{pop}}{\Gamma \vdash_{\mathcal{RE}'} P_\kappa \Rightarrow P, \Delta}\ \Rightarrow_r}{\Gamma \vdash_{\mathcal{RE}'\{\overrightarrow{t}/\overrightarrow{x}\}} (P_\kappa \Rightarrow P)\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \text{par récurrence}}{\Gamma, P_\kappa\{\overrightarrow{t}/\overrightarrow{x}\} \vdash_{\mathcal{RE}'\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \text{lemme A.7}}{\Gamma \vdash_{\mathcal{RE}'\{\overrightarrow{t}/\overrightarrow{x}\} \cup \{\kappa\{\overrightarrow{t}/\overrightarrow{x}\}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \textit{push}}{\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \text{par définition}$$

□

Notation: On définit ainsi la nouvelle règle :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} P, \Delta}{\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \text{lemme A.8}$$

Remarque: Le lemme A.8 est équivalent à la proposition suivante :

Pour tous contextes Γ, Δ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, pour toute substitution σ, si le séquent $\Gamma \vdash_{\mathcal{RE}} P, \Delta$ admet une preuve, alors on peut également construire une preuve du séquent $\Gamma \vdash_{\mathcal{RE}\sigma} P\sigma, \Delta$

La règle précédente peut donc également être mise sous la forme :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} P, \Delta}{\Gamma \vdash_{\mathcal{RE}\sigma} P\sigma, \Delta}\ \text{lemme A.8}$$

Lemme A.9 Pour tout contexte Γ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ des variables libres de $\{P\} \cup \mathcal{RE}$, et pour tout vecteur $\overrightarrow{t} \in \mathcal{T}(\Sigma, \mathcal{X})^n$, si les deux séquents $\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta$ et $\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta$ admettent une preuve, on peut alors construire une preuve du séquent $\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} Q\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta$

Preuve. Π_1

$$\dfrac{\dfrac{\dfrac{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta \quad \text{(supposé)}}{\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta}\ \Rightarrow_r}{\Gamma \vdash_{\mathcal{RE}} \forall \overrightarrow{x}\ P \Rightarrow Q, \Delta}\ \forall_r}{\Gamma \vdash_{\mathcal{RE}} \forall \overrightarrow{x}\ P \Rightarrow Q, P \Rightarrow Q, \Delta}\ w$$

Π_2 :

$$\dfrac{\dfrac{\dfrac{}{P \Rightarrow Q \vdash_{\mathcal{RE}} P \Rightarrow Q}\ Ax}{\forall \overrightarrow{x}\ P \Rightarrow Q \vdash_{\mathcal{RE}} P \Rightarrow Q}\ \forall_l}{\Gamma, \forall \overrightarrow{x}\ P \Rightarrow Q \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta}\ w$$

Π_3 :

$$\dfrac{\dfrac{\dfrac{\Pi_1 \quad \Pi_2}{\Gamma \vdash_{\mathcal{RE}} P \Rightarrow Q, \Delta}\ cut}{\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} (P \Rightarrow Q)\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \text{lemme A.8}}{\Gamma, P\{\overrightarrow{t}/\overrightarrow{x}\} \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} Q\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \text{lemme A.7}$$

Π_4 :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta \quad \text{(supposé)}}{\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, Q\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ w$$

Π :

$$\dfrac{\Pi_3 \quad \Pi_4}{\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} Q\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ cut$$

□

Notation: On définit ainsi la nouvelle règle :

$$\dfrac{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta \qquad \Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} P\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}{\Gamma \vdash_{\mathcal{RE}\{\overrightarrow{t}/\overrightarrow{x}\}} Q\{\overrightarrow{t}/\overrightarrow{x}\}, \Delta}\ \text{lemme } A.9$$

Remarque: Le lemme A.9 est equivalent à la proposition suivante :

"Pour tous contextes Γ, Δ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ des variables libres de $\{P\} \cup \mathcal{RE}$, et pour toute substitution σ, si les deux séquents $\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta$ et $\Gamma \vdash_{\mathcal{RE}\sigma} P\sigma, \Delta$ ont une preuve, alors il existe une preuve de $\Gamma \vdash_{\mathcal{RE}\sigma} Q\sigma, \Delta$."

La règle précédente peut donc se mettre sous la forme :

$$\dfrac{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta \qquad \Gamma \vdash_{\mathcal{RE}\sigma} P\sigma, \Delta}{\Gamma \vdash_{\mathcal{RE}\sigma} Q\sigma, \Delta}\ \text{lemme } A.9$$

Egalité

Pour toute relation binaire $\approx$ définie sur $\mathcal{T}(\Sigma, \mathcal{X})$, on pose $L(\approx)$ (égalité de Leibniz) la proposition :

$$\forall y\ \forall z\ y \approx z \Rightarrow (\forall P\ \forall x\ P\{y/x\} \Rightarrow P\{z/x\})$$

On admettra les règles suivantes :

– $$\dfrac{}{L(\approx) \vdash x \approx x}\ refl(\approx)$$

– $$\dfrac{}{L(\approx), x \approx y \vdash y \approx x}\ sym(\approx)$$

– $$\dfrac{}{L(\approx), x \approx y, y \approx z \vdash x \approx z}\ trans(\approx)$$

Lemme A.10 Pour toute proposition P, pour tout $x \in \mathcal{X}$, et pour tout $s,\ t \in \mathcal{T}(\Sigma, \mathcal{X})$, le séquent suivant possède une preuve dans le calcul des séquents modulo :

$$L(\approx), s \approx t, P\{s/x\} \vdash P\{t/x\}$$

Preuve. Π :

$$\dfrac{\dfrac{\dfrac{\dfrac{}{s \approx t \Rightarrow (P\{s/x\} \Rightarrow P\{t/y\}) \vdash s \approx t \Rightarrow (P\{s/x\} \Rightarrow P\{t/x\})}\ Ax}{s \approx t \Rightarrow (P\{s/x\} \Rightarrow P\{t/x\}), s \approx t \vdash P\{s/x\} \Rightarrow P\{t/x\}}\ \text{lemme A.7}}{s \approx t \Rightarrow (P\{s/x\} \Rightarrow P\{t/x\}), s \approx t, P\{s/x\} \vdash P\{t/x\}}\ \text{lemme A.7}}{L(\approx), s \approx t, P\{s/x\} \vdash P\{t/x\}}\ \forall_l$$

□

Notation: On obtient ainsi la nouvelle règle :

$$\dfrac{}{L(\approx), s \approx t, P\{s/x\} \vdash P\{t/x\}}\ \text{lemme } A.10$$

Lemme A.11 Pour toutes substitutions α, β, si $\alpha =_E \beta[V]$, le séquent suivant possède une preuve dans le calcul des séquents modulo :

$$L(\approx), E, x \in V \vdash x\alpha \approx x\beta$$

Preuve. Soit $v \in V$.

Π_1 :

$$\dfrac{\dfrac{\dfrac{}{v\alpha \approx v\alpha \vdash_E v\alpha \approx v\beta}\ Ax}{E, v\alpha \approx v\alpha \vdash v\alpha \approx v\beta}\ pop}{L(\approx), E, v\alpha \approx v\alpha \vdash v\alpha \approx v\beta}\ w$$

Π_2 :

$$\dfrac{\dfrac{}{L(\approx) \vdash v\alpha \approx v\alpha}\ refl(\approx)}{L(\approx), E \vdash v\alpha \approx v\alpha, v\alpha \approx v\beta}\ w$$

Π_3 :

$$\dfrac{\dfrac{\Pi_1 \quad \Pi_2}{L(\approx), E \vdash v\alpha \approx v\beta}\ cut}{L(\approx), E, x \approx v \vdash v\alpha \approx v\beta, x\alpha \approx x\beta}\ w$$

Π_4 :

$$\dfrac{\dfrac{}{L(\approx), E, x \approx v, v\alpha \approx v\beta \vdash x\alpha \approx x\beta}\ \text{lemme } A.10}{L(\approx), E, x \approx v, v\alpha \approx v\beta \vdash x\alpha \approx x\beta}\ w$$

Π_v :

$$\dfrac{\Pi_3 \quad \Pi_4}{L(\approx), E, x \approx v \vdash x\alpha \approx x\beta}\ cut$$

Π_5 :

$$\dfrac{\dfrac{\bullet\Pi_v}{L(\approx), E, \bigvee_{v \in V} x \approx v \vdash x\alpha \approx x\beta}\ \vee_l}{L(\approx), E, x \in V, \bigvee_{v \in V} x \approx v \vdash x\alpha \approx x\beta}\ w$$

Π_6 :

$$\dfrac{\dfrac{}{x \in V \vdash \bigvee_{v \in V} x \approx v}\ \text{lemme } A.14}{L(\approx), E, x \in V \vdash \bigvee_{v \in V} x \approx v, x\alpha \approx x\beta}\ w$$

Π

$$\dfrac{\Pi_5 \quad \Pi_6}{L(\approx), E, x \in V \vdash x\alpha \approx x\beta}\ cut$$

□

Notation: On définit ainsi la nouvelle règle :

$$\dfrac{}{L(\approx), E, x \in V \vdash x\alpha \approx x\beta}\ \text{lemme } A.11 \text{ (si } \alpha =_E \beta[V])$$

Lemme A.12 Pour toute proposition P, pour toutes substitutions α, β, et pour tout ensemble V de variables tel que $\mathcal{V}ar(P) \subseteq V$, le séquent suivant possède une preuve dans le calcul des séquents modulo :

$$L(\approx), \bigwedge_{x \in V} x\alpha \approx x\beta, P\alpha \vdash P\beta$$

Preuve. Par récurrence sur le nombre de variables de $\mathcal{D}om(\alpha)$ ou $\mathcal{D}om(\beta)$ contenues dans V.

- Cas de base : aucun élément de $\mathcal{D}om(\alpha)$ ou $\mathcal{D}om(\beta)$ n'est contenu dans V. Dans ce cas, α et β sont sans effet sur P, et la conclusion est immédiate.
- Cas de récurrence.

  Introduisons d'abord les notations suivantes :

  1. $(\mathcal{D}om(\alpha) \cup \mathcal{D}om(\beta)) \cap V = \{x_1, x_2, ..., x_{n+1}\}$
  2. $\mu = \alpha_{|\{x_1,...,x_n\}}$
  3. $\nu = \beta_{|\{x_1,...,x_n\}}$

  On a donc immédiatement :

  1. $(\mathcal{D}om(\mu) \cup \mathcal{D}om(\nu)) \cap V \subset (\mathcal{D}om(\alpha) \cup \mathcal{D}om(\beta)) \cap V$
  2. $$x_{n+1} \in (\mathcal{D}om(\alpha) \cup \mathcal{D}om(\beta)) \cap V \quad \text{et} \quad x_{n+1} \notin (\mathcal{D}om(\mu) \cup \mathcal{D}om(\nu))$$

  On a ainsi $(\mathcal{D}om(\mu) \cup \mathcal{D}om(\nu)) \cap V \subsetneq (\mathcal{D}om(\alpha) \cup \mathcal{D}om(\beta)) \cap V$

  Considérons maintenant les dérivations suivantes :

  Π_1 :

$$\dfrac{\dfrac{\dfrac{L(\approx), \bigwedge_{x \in V} x\mu \approx x\nu, P\mu \vdash P\nu \quad \text{(par récurrence, et d'après l'inclusion } \textit{stricte} \text{ ci-dessus)}}{L(\approx), \bigwedge_{x \in V} x\mu \approx x\nu \vdash P\mu \Rightarrow P\nu}\ \Rightarrow_r}{L(\approx), \bigwedge_{x \in V} x\mu \approx x\nu \vdash P\mu\{x_{n+1}\beta/x_{n+1}\} \Rightarrow P\nu\{x_{n+1}\beta/x_{n+1}\}}\ \textit{lemme } A.8}{L(\approx), \bigwedge_{x \in V} x\mu \approx x\nu, P\mu\{x_{n+1}\beta/x_{n+1}\} \vdash P\nu\{x_{n+1}\beta/x_{n+1}\}}\ \text{lemme A.7}}{L(\approx), \bigwedge_{x \in \{x_1,...,x_n\}} x\alpha \approx x\beta, \bigwedge_{x \in V} x\mu \approx x\nu, x_{n+1}\alpha \approx x_{n+1}\beta, P\mu\{x_{n+1}\beta/x_{n+1}\} \vdash P\nu\{x_{n+1}\beta/x_{n+1}\}}\ w$$

Π_2 :

$$\dfrac{\bigwedge_{x\in\{x_1,\ldots,x_n\}} x\alpha \approx x\beta \vdash \bigwedge_{x\in V} x\mu \approx x\nu \text{ (par définition)}}{L(\approx), \bigwedge_{x\in\{x_1,\ldots,x_n\}} x\alpha \approx x\beta, x_{n+1}\alpha \approx x_{n+1}\beta, P\mu\{x_{n+1}\beta/x_{n+1}\} \vdash P\nu\{x_{n+1}\beta/x_{n+1}\}, \bigwedge_{x\in V} x\mu \approx x\nu}\ w$$

Π_3 :

$$\dfrac{\Pi_1 \quad \Pi_2}{L(\approx), \bigwedge_{x\in\{x_1,\ldots,x_n\}} x\alpha \approx x\beta, x_{n+1}\alpha \approx x_{n+1}\beta, P\mu\{x_{n+1}\beta/x_{n+1}\} \vdash P\nu\{x_{n+1}\beta/x_{n+1}\}}\ cut$$

Π_4 :

$$\dfrac{\dfrac{\dfrac{\Pi_3}{L(\approx), \bigwedge_{x\in\{x_1,\ldots,x_{n+1}\}} x\alpha \approx x\beta, P\mu\{x_{n+1}\beta/x_{n+1}\} \vdash P\nu\{x_{n+1}\beta/x_{n+1}\}}\ \wedge_l}{L(\approx), \bigwedge_{x\in(\mathcal{D}om(\alpha)\cup\mathcal{D}om(\beta))\cap V} x\alpha \approx x\beta, P\mu\{x_{n+1}\beta/x_{n+1}\} \vdash P\beta}\ \text{par définition}}{L(\approx), \bigwedge_{x\in(\mathcal{D}om(\alpha)\cup\mathcal{D}om(\beta))\cap V} x\alpha \approx x\beta, P\mu\{x_{n+1}\beta/x_{n+1}\}, P\alpha \vdash P\beta}\ w$$

Π_5 :

$$\dfrac{\dfrac{\dfrac{\dfrac{}{L(\approx), x_{n+1}\alpha \approx x_{n+1}\beta, P\mu\{x_{n+1}\alpha/x_{n+1}\} \vdash P\mu\{x_{n+1}\beta/x_{n+1}\}}\ \text{lemme } A.10}{L(\approx), x_{n+1}\alpha \approx x_{n+1}\beta, P\alpha \vdash P\mu\{x_{n+1}\beta/x_{n+1}\}}\ \text{par définition}}{L(\approx), \bigwedge_{x\in\{x_1,\ldots x_{n+1}\}} x\alpha \approx x\beta, P\alpha \vdash P\mu\{x_{n+1}\beta/x_{n+1}\}, P\beta}\ w}{L(\approx), \bigwedge_{x\in(\mathcal{D}om(\alpha)\cup\mathcal{D}om(\beta))\cap V} x\alpha \approx x\beta, P\alpha \vdash P\mu\{x_{n+1}\beta/x_{n+1}\}, P\beta}\ \text{par définition}$$

Π :

$$\dfrac{\Pi_4 \qquad \Pi_5}{L(\approx), \bigwedge_{x\in(\mathcal{D}om(\alpha)\cup\mathcal{D}om(\beta))\cap V} x\alpha \approx x\beta, P\alpha \vdash P\beta}\ cut$$

□

Notation: On obtient ainsi la nouvelle règle :

$$\dfrac{}{L(\approx), \bigwedge_{x\in V} x\alpha \approx x\beta, P\alpha \vdash P\beta}\ \text{lemme } A.12 \text{ (si } \mathcal{V}ar(P) \subseteq V)$$

Lemme A.13 Pour tout contexte Γ tel que $L(\approx) \subseteq \Gamma$, pour tout contexte Δ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, pour toutes substitutions α, β, et pour tout ensemble de variables V tel que $\mathcal{V}ar(P) \cup \mathcal{V}ar(\mathcal{RE}) \subseteq V$, si les séquents $\Gamma \vdash_{\mathcal{RE}} \bigwedge_{x\in V} x\alpha \approx x\beta$ et $\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta$ ont une preuve dans le calcul des séquents modulo, alors on peut construire une preuve de $\Gamma \vdash_{\mathcal{RE}\beta} P\beta, \Delta$

Preuve. Par récurrence sur le cardinal de $\mathcal{RE}$

Cas de base : $\mathcal{RE} = \emptyset$

Considérons les dérivations suivantes :

Π_1 :

$$\dfrac{L(\approx), \bigwedge_{x\in V} x\alpha \approx x\beta, P\alpha \vdash P\beta \quad (\text{ lemme } A.12)}{\Gamma, \bigwedge_{x\in V} x\alpha \approx x\beta, P\alpha \vdash P\beta, \Delta}\ w \quad (\text{ car } L(\approx) \in \Gamma)$$

Π_2 :

$$\dfrac{\dfrac{\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta \quad (\text{par hypothèse})}{\Gamma \vdash P\alpha, \Delta}\ (\text{ car } \mathcal{RE} = \emptyset\ )}{\Gamma, \bigwedge_{x\in V} x\alpha \approx x\beta \vdash P\alpha, P\beta, \Delta}\ w$$

Π_3 :

$$\dfrac{\Pi_1 \qquad \Pi_2}{\Gamma, \bigwedge_{x\in V} x\alpha \approx x\beta \vdash P\beta, \Delta}\ cut$$

Π_4 :

$$\dfrac{\dfrac{\Gamma \vdash_{\mathcal{RE}} \bigwedge_{x\in V} x\alpha \approx x\beta \quad (\text{ par hypothèse })}{\Gamma \vdash \bigwedge_{x\in V} x\alpha \approx x\beta \quad (\text{ par hypothèse })}\ \text{car } \mathcal{RE} = \emptyset}{\Gamma \vdash \bigwedge_{x\in V} x\alpha \approx x\beta, P\beta, \Delta}\ w$$

Π :

$$\dfrac{\dfrac{\Pi_3 \qquad \Pi_4}{\Gamma \vdash P\beta, \Delta}\ cut}{\Gamma \vdash_{\mathcal{RE}\beta} P\beta, \Delta}\ (\text{ car } \mathcal{RE} = \emptyset)$$

Cas de récurrence :

Soit κ l'une des règles de $\mathcal{RE}$, P_κ la proposition canonique associée, et $\mathcal{RE}' = \mathcal{RE} \setminus \{\kappa\}$
Considérons la dérivation suivante :

Π :

$$\begin{array}{c}
\dfrac{\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta \quad (\text{ par hypothèse })}{\Gamma \vdash_{\mathcal{RE}'\alpha\cup\{\kappa\alpha\}} P\alpha, \Delta}\ \text{par définition} \\
\dfrac{\Gamma, P_\kappa\alpha \vdash_{\mathcal{RE}'\alpha} P\alpha, \Delta}{\Gamma \vdash_{\mathcal{RE}'\alpha} (P_\kappa \Rightarrow P)\alpha, \Delta}\ pop \\
\dfrac{\Gamma \vdash_{\mathcal{RE}'\beta} (P_\kappa \Rightarrow P)\beta, \Delta}{\Gamma, P_\kappa\beta \vdash_{\mathcal{RE}'\beta} P\beta, \Delta}\ \Rightarrow_r \\
\dfrac{\Gamma \vdash_{\mathcal{RE}'\beta\cup\{\kappa\beta\}} P\beta, \Delta}{\Gamma \vdash_{\mathcal{RE}\beta} P\beta, \Delta}\ \text{par récurrence} \\
\text{lemme A.7} \\
push \\
\text{par définition}
\end{array}$$

□

Notation: On obtient ainsi la nouvelle règle :

$$\dfrac{\Gamma \vdash_{\mathcal{RE}} \bigwedge_{x\in V} x\alpha \approx x\beta \quad \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha}{\Gamma \vdash_{\mathcal{RE}\beta} P\beta}\ \text{lemme } A.13\ (\text{si } \mathcal{V}ar(P) \cup \mathcal{V}ar(\mathcal{RE}) \subseteq V)$$

Corollaire A.2 Pour tout contexte Γ tel que $L(\approx) \subseteq \Gamma$, pour tout contexte Δ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, pour toutes substitutions α, β, et pour tout ensemble de variables V tel que $\mathcal{V}ar(P) \cup \mathcal{V}ar(\mathcal{RE}) \subseteq V$, si les séquents $\Gamma, x \in V \vdash_{\mathcal{RE}} x\alpha \approx x\beta$ et $\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta$ ont une preuve dans le calcul des séquents modulo, alors on peut construire une preuve de $\Gamma \vdash_{\mathcal{RE}\beta} P\beta, \Delta$

Preuve. Soit $v \in V$, on peut alors poser $\Gamma \vdash_{\mathcal{RE}} v \in V$. Considérons alors les dérivations suivantes : Π_v :

$$\frac{\Gamma, v \in V \vdash_{\mathcal{RE}} v\alpha \approx v\beta \text{ (supposé)} \quad \Gamma \vdash_{\mathcal{RE}} v \in V}{\Gamma \vdash_{\mathcal{RE}} v\alpha \approx v\beta} \; cut$$

Π_1 :

$$\frac{\overset{\bullet}{v \in V}\,\Pi_v}{\Gamma \vdash_{\mathcal{RE}} \bigwedge_{x \in V} x\alpha \approx x\beta} \; \wedge_r$$

Π :

$$\frac{\Pi_1 \quad \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha \text{ (supposé)}}{\Gamma \vdash_{\mathcal{RE}\beta} P\beta} \text{ lemme } A.13 \text{ (si } \mathcal{V}ar(P) \cup \mathcal{V}ar(\mathcal{RE}) \subseteq V)$$

□

Notation: On obtient ainsi la nouvelle règle :

$$\frac{\Gamma, x \in V \vdash_{\mathcal{RE}} x\alpha \approx x\beta \quad \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha}{\Gamma \vdash_{\mathcal{RE}\beta} P\beta} \text{ corollaire } A.2 \text{ (si } \mathcal{V}ar(P) \cup \mathcal{V}ar(\mathcal{RE}) \subseteq V)$$

A.2 Lemmes ensemblistes

A.2.1 Ensembles finis

Lemme A.14 Pour tout ensemble E, si E est fini, on a :

$$\forall x \; x \in E \Leftrightarrow \bigvee_{e \in E} x \approx e$$

Corollaire A.3 Pour tout ensemble E, si E est fini, on a le séquent :

$$x \in E \vdash \bigvee_{e \in E} x \approx e$$

Proof :

C'est une conséquence du lemme $A.1$.
Notation: On obtient ainsi la règle suivante :

$$\frac{}{x \in E \vdash \bigvee_{e \in E} x \approx e} \text{ corollaire } A.3 \text{ (si E est fini.)}$$

A.2.2 Définition et caractérisation de $\mathcal{T}(\Sigma)$

Notation: Une signature Σ, un ensemble $\mathcal{X}$ de variables, et un entier m étant donnés, on note :
- $Tr_\Sigma(\mathcal{X})$ l'ensemble de tous les arbres construits à partir de Σ et de $\mathcal{X}$
- $FTr_\Sigma(\mathcal{X})$ l'ensemble de tous les arbres *finis* construits à partir de Σ et de $\mathcal{X}$
- $\mathcal{T}(\Sigma)$ l'ensemble $FTr_\Sigma(\emptyset)$
- $\mathcal{T}_m(\Sigma)$ l'ensemble $\{t \in T_\Sigma \mid |t| \leq m\}$ ($|t|$ désignant la profondeur de t, i.e la longueur de la plus grande chaîne contenue dans t).

Lemme A.15

$$\forall x\ x \in \mathcal{T}(\Sigma) \Rightarrow \exists m\ x \in \mathcal{T}_m(\Sigma)$$

Preuve. Par construction □

Lemme A.16 Une signature Σ et un entier m étant donnés, $\mathcal{T}_m(\Sigma)$ est fini.

Preuve. Immédiat □

Lemme A.17 Une signature Σ étant donnée, on a le séquent :

$$x \in \mathcal{T}(\Sigma) \vdash \bigvee_{y \in \mathcal{T}_m(\Sigma)} x \approx y$$

Preuve. C'est une conséquence du lemme $A.16$ et du lemme $A.14$ □

Corollaire A.4 Une signature Σ et un entier n étant donnés, on a les séquents :

1. $$x \in \mathcal{T}(\Sigma) \vdash \exists m\ x \in \mathcal{T}_m(\Sigma)$$
2. $$\overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash \exists m\ \overrightarrow{x} \in \mathcal{T}_m(\Sigma)^n$$

Preuve. 1. est une conséquence du lemme $A.1$ et du lemme $A.15$.
2. est une conséquence de 1. □

Notation: On obtient ainsi les deux nouvelles règles :

1. $$\frac{}{x \in \mathcal{T}(\Sigma) \vdash \exists m\ x \in \mathcal{T}_m(\Sigma)}\ \text{corollaire } A.4\ (1)$$
2. $$\frac{}{\overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash \exists m\ \overrightarrow{x} \in \mathcal{T}_m(\Sigma)^n}\ \text{corollaire } A.4\ (2)$$

A.3 Lemmes concernant les formes closes

Lemme A.18

Pour tous contextes Γ et Δ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, et pour tout vecteur de variables $\overrightarrow{x} \in \mathcal{X}^n$, si $\overrightarrow{x}$ est le vecteur des variables libres de $\{P\} \cup \mathcal{RE}$, et si le séquent $\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P, \Delta$ est dérivable, alors, pour toute substitution close α de $\overrightarrow{x}$, le séquent $\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta$ est dérivable.

Preuve. Considérons d'abord la dérivation suivante :

Π_1

$$\dfrac{\dfrac{\Gamma \vdash_{\mathcal{RE}} \overrightarrow{x}\alpha \in \mathcal{T}(\Sigma)^n \quad \text{(par hypothèse)}}{\Gamma \vdash_{\mathcal{RE}} \overrightarrow{x}\alpha \in \mathcal{T}(\Sigma)^n, \Delta}\, w}{\Gamma \vdash_{\mathcal{RE}\alpha} \overrightarrow{x}\alpha \in \mathcal{T}(\Sigma)^n, \Delta}\ \text{(lemme } A.8 \text{ et car } \overrightarrow{x}\alpha\alpha = \overrightarrow{x}\alpha)$$

Π :

$$\dfrac{\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P, \Delta \quad \text{(supposé)} \qquad \Pi_1}{\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta}\ \text{lemme } A.9$$

□

Lemme A.19

Pour tous contextes Γ et Δ, pour tout système de réécriture $\mathcal{RE}$, pour toute proposition P, et pour tout vecteur de variables $\overrightarrow{x} \in \mathcal{X}^n$, si $\overrightarrow{x}$ est le vecteur des variables libres de $\{P\} \cup \mathcal{RE}$, et si, pour toute substitution close α de $\overrightarrow{x}$, le séquent $\Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta$ est dérivable, alors il existe une preuve de $\Gamma, \overrightarrow{x} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P, \Delta$

Preuve. Par récurrence sur le nombre de règles de $\mathcal{RE}$.

Cas de base : si $\mathcal{RE} = \emptyset$

Introduisons tout d'abord les notations suivantes :

m	représente	un entier.
S_m	représente	l'ensemble $\{\alpha \in Subst^{\Sigma} \mid \overrightarrow{x}\alpha \in T_m(\Sigma)^n\})$

Considérons à présent les dérivations suivantes :

$\Pi_{1,m}$:

$$\dfrac{\dfrac{\underset{\alpha \in S_m}{\bullet}\ \Gamma \vdash P\alpha, \Delta}{\Gamma \vdash \bigwedge\limits_{\alpha \in S_m} P\alpha, \Delta}\ \wedge - r}{\Gamma, \underline{\overrightarrow{x}} \in T_m(\Sigma)^n \vdash \bigwedge\limits_{\alpha \in S_m} P\alpha, P, \Delta}\ w$$

Choisissons $\beta \in S_m$, et considérons la dérivation suivante :

$\Pi_{1,m,\beta}$:

$$\dfrac{\dfrac{\Gamma, \underline{\overrightarrow{x}} \approx \overrightarrow{x}\beta, P\beta \vdash P \quad \text{(lemme } A.10)}{\Gamma, \underline{\overrightarrow{x}} \approx \overrightarrow{x}\beta, \bigwedge\limits_{\alpha \in S_m} P\alpha, P\beta \vdash P}\ w \qquad \dfrac{\dfrac{\dfrac{\overline{P\beta \vdash P\beta}\ Ax}{\Gamma, P\beta \vdash P\beta}\ w}{\Gamma, \bigwedge\limits_{\alpha \in S_m} P\alpha \vdash P\beta}\ w,\ \wedge_l}{\Gamma, \underline{\overrightarrow{x}} \approx \overrightarrow{x}\beta, \bigwedge\limits_{\alpha \in S_m} P\alpha \vdash P, P\beta}\ w}{\Gamma, \underline{\overrightarrow{x}} \approx \overrightarrow{x}\beta, \bigwedge\limits_{\alpha \in S_m} P\alpha \vdash P}\ cut$$

On peut alors construire la preuve de la façon suivante :

$\Pi_{2,m}$:

$$\dfrac{\dfrac{\underset{\beta\in S_m}{\bullet}\ \Pi_{1,m,\beta}}{\Gamma,\ \bigvee_{\beta\in S_m} \overrightarrow{\underline{x}} \approx \overrightarrow{x}\beta,\ \bigwedge_{\alpha\in S_m} P\alpha \vdash P}\ \vee_l}{\Gamma,\ \bigvee_{\beta\in S_m} \overrightarrow{\underline{x}} \approx \overrightarrow{x}\beta, \overrightarrow{x} \in \mathcal{T}_m(\Sigma)^n,\ \bigwedge_{\alpha\in \mathcal{T}_m(\Sigma)} P\alpha \vdash P}\ w$$

$\Pi_{3,m}$:

$$\dfrac{\dfrac{}{\Gamma,\ \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n \vdash \bigvee_{\beta\in S_m} \overrightarrow{\underline{x}} \approx \overrightarrow{x}\beta}\ \text{corollaire } A.3 \text{ (car } \mathcal{T}_m(\Sigma) \text{ est fini et par définition de } S_m)}{\Gamma, \overrightarrow{x} \in \mathcal{T}_m(\Sigma)^n,\ \bigwedge_{\alpha\in S_m} P\alpha \vdash P,\ \bigvee_{\beta\in S_m} \overrightarrow{\underline{x}} \approx \overrightarrow{x}\beta}\ w$$

$\Pi_{4,m}$:

$$\dfrac{\dfrac{\Pi_{2,m} \quad \Pi_{3,m}}{\Gamma, \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n,\ \bigwedge_{\alpha\in \mathcal{T}_m(\Sigma)^n} P\alpha \vdash P}\ cut}{\Gamma, \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n,\ \bigwedge_{\alpha\in \mathcal{T}_m(\Sigma)^n} P\alpha \vdash P, \Delta}\ w$$

$\Pi_{5,m}$:

$$\dfrac{\dfrac{\dfrac{\Pi_{1,m} \quad \Pi_{4,m}}{\Gamma, \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n \vdash P, \Delta}\ cut}{\Gamma, \exists m\ \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n \vdash P, \Delta}\ \exists_l}{\Gamma, \overrightarrow{\underline{x}} \in \mathcal{T}(\Sigma)^n, \exists m\ \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n \vdash P, \Delta}\ w$$

Π_6 :

$$\dfrac{\dfrac{}{\Gamma, \overrightarrow{\underline{x}} \in \mathcal{T}(\Sigma)^n \vdash \exists m\ \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n \text{ (lemme } A.4)}\ \text{corollaire } A.4\ (2)}{\Gamma, \overrightarrow{\underline{x}} \in \mathcal{T}(\Sigma)^n \vdash \exists m\ \overrightarrow{\underline{x}} \in \mathcal{T}_m(\Sigma)^n, P, \Delta}\ w$$

A ce niveau, il ne reste plus qu'une étape de dérivation pour conclure :

$$\dfrac{\Pi_{5,m} \quad \Pi_6}{\Gamma, \overrightarrow{\underline{x}} \in \mathcal{T}(\Sigma)^n \vdash P, \Delta}\ cut$$

Cas de récurrence : si $\mathcal{RE}$ contient $p+1$ règles.

Soit κ l'une des règles de l'ensemble $\mathcal{RE}$, $\mathcal{RE}' = \mathcal{RE} \setminus \{\kappa\}$, et P_κ la proposition canonique associée à κ.

Considérons la dérivation suivante :

Π :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\bullet_{\alpha\in Subst^{\Sigma}}\ \Gamma \vdash_{\mathcal{RE}\alpha} P\alpha, \Delta}{\bullet_{\alpha\in Subst^{\Sigma}}\ \Gamma \vdash_{\mathcal{RE}'\alpha\cup\{\kappa\alpha\}} P\alpha, \Delta}\ \text{(par définition)}}{\bullet_{\alpha\in Subst^{\Sigma}}\ \Gamma, P_\kappa\alpha \vdash_{\mathcal{RE}'\alpha} P\alpha, \Delta}\ pop}{\bullet_{\alpha\in Subst^{\Sigma}}\ \Gamma \vdash_{\mathcal{RE}'\alpha} (P_\kappa \Rightarrow P)\alpha, \Delta}\ \Rightarrow_r}{\Gamma, \underline{\overrightarrow{x}} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}'} P_\kappa \Rightarrow P, \Delta}\ \text{(par récurrence, avec } \underline{\overrightarrow{x}} = \overrightarrow{\mathcal{V}ar(\mathcal{RE}' \cup \{P_k \Rightarrow P\})} = \overrightarrow{\mathcal{V}ar(\mathcal{RE} \cup \{P\})}\text{)}}{\Gamma, P_\kappa, \underline{\overrightarrow{x}} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}'} P, \Delta}\ \text{(lemme A.7)}}{\Gamma, \underline{\overrightarrow{x}} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}'\cup\{\kappa\}} P, \Delta}\ push}{\Gamma, \underline{\overrightarrow{x}} \in \mathcal{T}(\Sigma)^n \vdash_{\mathcal{RE}} P, \Delta}\ \text{(par définition)}$$

□

A.4 Lemmes inductifs

Notation: Pour tout terme $t \in \mathcal{T}(\Sigma, \mathcal{X})$ et pour tout vecteur de variables $\overrightarrow{x}$, $t\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}$ va désigner le terme obtenu par remplacement de chaque occurence d'une composante x_i de $\overrightarrow{x}$ dans t par un $\underline{x}_i$ correspondant. On étend cette notation aux propositions, pour lesquelles seules les variables libres peuvent être remplacées. Pour toute proposition P, pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ de variables libres de P, pour tout ensemble τ, et pour toute relation binaire R sur τ^n, introduisons les notations suivantes :

$$\begin{array}{lcl}
\mathcal{RE}_{ind}(P, R, \tau) & : & \underline{\overrightarrow{x}} \in \tau^n \wedge R(\overrightarrow{x}, \underline{\overrightarrow{x}}) \Rightarrow P\{\underline{\overrightarrow{x}}/\overrightarrow{x}\} \\
\mathcal{PE}_{ind}(P, R, \tau) & : & \forall \underline{\overrightarrow{x}}\ \underline{\overrightarrow{x}} \in \tau \wedge R(\overrightarrow{x}, \underline{\overrightarrow{x}}) \Rightarrow P\{\underline{\overrightarrow{x}}/\overrightarrow{x}\} \\
Ind(P, R, \tau) & : & \forall \overrightarrow{x}\ \overrightarrow{x} \in \tau \Rightarrow (\mathcal{PE}_{ind}(P, R, \tau) \Rightarrow P) \\
True(P, \tau) & : & \forall \overrightarrow{x}\ x \in \tau \Rightarrow P(x) \\
NoethInd(P, R, \tau) & : & Ind(P, R, \tau) \Rightarrow True(P, \tau) \\
NI & : & \forall P\ \forall R\ \forall \tau\ Noeth(R, \tau) \Rightarrow NoethInd(P, R, \tau)
\end{array}$$

Lemme A.20 Pour tout contexte Γ, pour tout système de réécriture $\mathcal{RE}$, pour tout ensemble τ, et pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ des variables libres de P,

$$\text{Si } \begin{cases} \mathcal{V}ar(\mathcal{RE}) = \emptyset \\ \Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\mathcal{RE}_{ind}(P,R,\tau)} P \end{cases}$$

alors, on peut construire une preuve du séquent suivant :

$$\Gamma \vdash_{\mathcal{RE}} Ind(P, R, \tau)$$

Preuve.

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\mathcal{RE}_{ind}(P,R,\tau)} P}{\Gamma, \overrightarrow{x} \in \tau^n, \mathcal{PE}_{ind}(P, R, \tau) \vdash_{\mathcal{RE}} P}\ pop}{\Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}} \mathcal{P}_{ind}(P, R, \tau) \Rightarrow P}\ \Rightarrow_r}{\Gamma \vdash_{\mathcal{RE}} \overrightarrow{x} \in \tau^n \Rightarrow (\mathcal{P}_{ind}(P, R, \tau) \Rightarrow P)}\ \Rightarrow_r}{\Gamma \vdash_{\mathcal{RE}} Ind(P, R, \tau)}\ \forall_r$$

Notation 1

On appellera lemme $A.20$ la nouvelle règle :

$$\frac{\Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE} \cup \mathcal{RE}_{ind}(P,R,\tau)} P}{\Gamma \vdash_{\mathcal{RE}} Ind(P,R,\tau)} \text{ si } \mathcal{V}ar(P) = \overrightarrow{x} \text{ et} \mathcal{V}ar(\mathcal{RE}) = \emptyset$$

□

Lemme A.21 Si le séquent $\Gamma \vdash_{\mathcal{RE}} Noeth(R,\tau)$ est dérivable, alors, pour toute proposition P, on peut prouver le séquent :

$$\Gamma, NI, Ind(P,R,\tau) \vdash_{\mathcal{RE}} True(P,\tau)$$

Preuve. Π_1 :

$$\cfrac{\cfrac{\cfrac{\cfrac{}{Noeth(R,\tau) \Rightarrow NoethInd(P,R,\tau) \vdash_{\mathcal{RE}} Noeth(R,\tau) \Rightarrow NoethInd(P,R,\tau)}\ Ax}{Noeth(R,\tau) \Rightarrow NoethInd(P,R,\tau), Noeth(R,\tau) \vdash_{\mathcal{RE}} NoethInd(P,R,\tau)}\ \text{lemme A.7}}{Noeth(R,\tau) \Rightarrow NoethInd(P,R,\tau), Noeth(R,\tau), Ind(P,R,\tau) \vdash_{\mathcal{RE}} True(P,\tau)}\ \text{lemme A.7}}{\cfrac{NI, Noeth(R,\tau), Ind(P,R,\tau) \vdash_{\mathcal{RE}} True(P,\tau)}{\Gamma, NI, Noeth(R,\tau), Ind(P,R,\tau) \vdash_{\mathcal{RE}} True(P,\tau)}\ w}\ \forall_l$$

Π_2 :

$$\frac{\Gamma \vdash_{\mathcal{RE}} Noeth(R,\tau) \quad \text{(supposé)}}{\Gamma, NI, Ind(P,R,\tau) \vdash_{\mathcal{RE}} Noeth(R,\tau), True(P,\tau)}\ w$$

Π :

$$\frac{\Pi_1 \quad \Pi_2}{\Gamma, NI, Ind(P,R,\tau) \vdash_{\mathcal{RE}} True(P,\tau)}\ cut$$

□

Notation 2

On notera lemme $A.21$ la nouvelle règle :

$$\frac{\Gamma \vdash_{\mathcal{RE}} Noeth(R,\tau)}{\Gamma, NI, Ind(P,R,\tau) \vdash_{\mathcal{RE}} True(P,\tau)}\ \text{lemma } A.21$$

Lemme A.22 Pour tout contexte Γ, pour tout système de réécriture $\mathcal{RE}$, pour tout ensemble τ, et pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ des variables libres de P, si $\mathcal{V}ar(\mathcal{RE}) = \emptyset$, et les séquents $\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE} \cup \mathcal{RE}_{ind}(P)} P$ et $\Gamma \vdash_{\mathcal{RE}} Noeth(R,\tau)$ sont dérivables, alors on peut dériver le séquent $\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}} P$

Preuve. Π_1 :

$$\cfrac{\cfrac{\Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE} \cup \mathcal{RE}_{ind}(P)} P \quad \text{(supposé)}}{\Gamma \vdash_{\mathcal{RE}} Ind(P,R,\tau)}\ \text{lemme } A.20}{\Gamma, NI \vdash_{\mathcal{RE}} Ind(P,R,\tau), True(P,\tau)}\ w$$

Π_2 :

$$\frac{\Gamma \vdash_{\mathcal{RE}} Noeth(R,\tau) \quad \text{(supposé)}}{\Gamma, NI, Ind(P,R,\tau) \vdash_{\mathcal{RE}} True(P,\tau)}\ \text{lemme } A.21$$

Π :

$$\dfrac{\dfrac{\Pi_1 \quad \Pi_2}{\Gamma, NI \vdash_{\mathcal{RE}} True(P,\tau)}\ cut}{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}} P}\ \text{lemme } A.1$$

□

Notation: On note lemme $A.22$ la nouvelle règle :

$$\dfrac{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\mathcal{RE}_{ind}(P)} P \quad \Gamma \vdash_{\mathcal{RE}} Noeth(R,\tau)}{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}} P} \quad \begin{array}{l}\text{si } \mathcal{V}ar(P) = \overrightarrow{x} \text{ et} \\ \mathcal{V}ar(\mathcal{RE}) = \emptyset\end{array}$$

On suppose à partir de maintenant que $<$ est un ordre de réduction défini sur un ensemble τ.
Pour tout entier n, $<_n$ désignera alors l'extension cartésienne de $<$ à τ^n (cf définition 2.5 de la partie I)

Notation: Pour toute proposition P, pour tout ensemble τ, pour tout entier n, et pour tout vecteur $\overrightarrow{x} \in \tau^n$ des variables libres de P, $\mathcal{RE}_{ind}(P,<,\tau)$ et $\mathcal{PE}_{ind}(P,<,\tau)$ sont définis de la façon suivante :

$$\begin{array}{l}\mathcal{RE}_{ind}(P,<,\tau) \triangleq (\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}) \Rightarrow P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\} \\ \mathcal{PE}_{ind}(P,<,\tau) \triangleq \forall \overrightarrow{\underline{x}}\ (\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}) \Rightarrow P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}\end{array}$$

Lemme A.23 Supposons que $\mathcal{RE}$ est un système de réécriture contenant des règles de l'un des types suivants :

$$\begin{array}{ll}\text{Type 1 :} & \kappa = l \to r \text{ ou } \kappa = l \approx r \\ \text{Type 2 :} & \kappa = (\overrightarrow{\underline{x_\kappa}} \in \tau^{n_\kappa} \wedge \overrightarrow{\underline{x_\kappa}} <_n \overrightarrow{t_\kappa}) \Rightarrow (l \to r)\{\overrightarrow{\underline{x_\kappa}}/\overrightarrow{x_\kappa}\} \text{ ou} \\ & (\overrightarrow{\underline{x_\kappa}} \in \tau^{n_\kappa} \wedge \overrightarrow{\underline{x_\kappa}} <_n \overrightarrow{t_\kappa}) \Rightarrow (l \approx r)\{\overrightarrow{\underline{x_\kappa}}/\overrightarrow{x_\kappa}\} \\ & \text{avec } \overrightarrow{x_\kappa} \in \tau^{n_\kappa} \text{ le vecteur des variables de } l \approx r\end{array}$$

Alors, pour tout contexte Γ, pour tout ensemble τ, et pour tout vecteur $\overrightarrow{x} \in \mathcal{X}^n$ des variables libres de $\mathcal{RE} \cup \{P\}$, si les séquents $\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\mathcal{RE}_{ind}(P,<,\tau)} P$ et $\Gamma \vdash Noeth(<,\tau)$ sont dérivables, alors on peut montrer le séquent $\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}} P$.
On obtient ainsi la règle suivante :

$$\dfrac{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\mathcal{RE}_{ind}(P,<,\tau)} P \quad \Gamma \vdash Noeth(<,\tau)}{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}} P} \quad \begin{array}{l}\text{si } \mathcal{V}ar(\mathcal{RE} \cup \{P\}) = \overrightarrow{x} \text{ et} \\ < \text{ est un ordre de réduction} \\ \text{sur } \tau\end{array}$$

Preuve. Par récurrence sur le nombre de règles de type 2 dans $\mathcal{RE}$.

Cas de base : si $\mathcal{RE}$ ne contient aucune règle de type 2, on applique le lemme $A.22$

Cas de récurrence : si $\mathcal{RE}$ contient $p+1$ règles de type 2.

Supposons que $\overrightarrow{x} \in \mathcal{X}^n$ soit tel que $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE} \cup \{P\})}$. Posons κ l'une des règles de type 2 de $\mathcal{RE}$, et $\mathcal{RE}' = \mathcal{RE} \setminus \{\kappa\}$.
Supposons que κ soit défini de la façon suivante :

$$\begin{array}{l}\kappa = (\overrightarrow{\underline{x_\kappa}} \in \tau^{n_\kappa} \wedge \overrightarrow{\underline{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\overrightarrow{\underline{x_\kappa}}/\overrightarrow{x_\kappa}\} \\ \text{avec } Q_\kappa = l_\kappa \rightarrow r_\kappa \text{ ou } l_\kappa \approx r_\kappa \text{ et } \overrightarrow{x_\kappa} = \overrightarrow{\mathcal{V}ar(Q_\kappa)}\end{array}$$

et posons P_κ la proposition canonique associée :

$$\forall \overrightarrow{\underline{x_\kappa}}\ (\overrightarrow{\underline{x_\kappa}} \in \tau^{n_\kappa} \wedge \overrightarrow{\underline{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\overrightarrow{\underline{x_\kappa}}/\overrightarrow{x_\kappa}\}$$

On a :
Π_1 :

$$\frac{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\{\mathcal{RE}_{ind}(P_\kappa \Rightarrow P, <, \tau)\}} P}{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}'\cup\{\kappa, \mathcal{RE}_{ind}(P_\kappa \Rightarrow P, <, \tau)\}} P}\ \text{Définition}$$

Donc, sachant que P_κ est la théorie canonique associée à κ, cela entraîne :

Π_2 :

$$\frac{\Pi_1}{\Gamma, NI, \overrightarrow{x} \in \tau^n, P_\kappa \vdash_{\mathcal{RE}'\cup\{\mathcal{RE}_{ind}(P_\kappa \Rightarrow P, <, \tau)\}} P}\ pop$$

Π_3 :

$$\frac{\Pi_2}{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}'\cup\{\mathcal{RE}_{ind}(P_\kappa \Rightarrow P, <, \tau)\}} P_\kappa \Rightarrow P}\ \Rightarrow_r$$

Et puisque $\mathcal{RE}'$ contient p règles de type 2, on peut écrire :

Π_4 :

$$\frac{\Pi_3 \quad \Gamma \vdash Noeth(R, \tau) \text{ supposé}}{\Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}'} P_\kappa \Rightarrow P}\ \text{par récurrence}$$

Π_5 :

$$\frac{\Pi_4}{\Gamma, \overrightarrow{x} \in \tau^n, P_\kappa \vdash_{\mathcal{RE}'} P}\ \text{lemme logique}$$

Π_6 :

$$\frac{\Pi_5}{\Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}'\cup\{\kappa\}} P}\ push$$

Et sachant que $\mathcal{RE} = \mathcal{RE}' \cup \{\kappa\}$, on peut écrire :

Π_7 :

$$\frac{\Pi_6}{\Gamma, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}} P}\ \text{Définition}$$

Donc, pour démontrer le lemme, il suffit de prouver la règle suivante :

$$\frac{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\{\mathcal{RE}_{ind}(P, <, \tau)\}} P}{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE}\cup\{\mathcal{RE}_{ind}(P_\kappa \Rightarrow P, <, \tau)\}} P} \tag{A.1}$$

Du fait que $<$ est un ordre de réduction, on a le séquent suivant :

$S_1 : \Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}) \vdash \underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_\kappa \overrightarrow{t}_\kappa$

Considérons maintenant les dérivations suivantes :

Π_8 :

$$\frac{S_1}{\Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}), P_\kappa \vdash (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_\kappa \overrightarrow{t_\kappa}), Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}$$

Π_9 :

$$\frac{}{(\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\} \vdash (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ Ax$$

Π_{10} :

$$\frac{\Pi_9}{\forall \underline{\overrightarrow{x_\kappa}}\ (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t}_\kappa) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\} \vdash (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ \forall_l$$

Donc, comme :

$$P_\kappa \triangleq \forall \underline{\overrightarrow{x_\kappa}}\ (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}$$

on peut écrire :

Π_{10} :

$$\frac{\Pi_9}{P_\kappa \vdash (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ \forall_l$$

Π_{11} :

$$\frac{\Pi_{10}}{\Gamma, NI, P_\kappa \vdash (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ w$$

Π_{12} :

$$\frac{\Pi_{11}}{\Gamma, NI, P_\kappa, (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \vdash Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ \text{logical lemma}$$

Π_{13} :

$$\frac{\Pi_{12}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), \quad (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}), P_\kappa, (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}) \vdash Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ w$$

Π_{14} :

$$\frac{\Pi_8 \quad \Pi_{13}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}), P_\kappa \vdash Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ cut$$

Π_{15} :

$$\frac{\Pi_{14}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), P_\kappa \vdash (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ \Rightarrow_r$$

Π_{16} :

$$\frac{\Pi_{15}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), P_\kappa \vdash \forall \underline{\overrightarrow{x_\kappa}}\ (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}}\ \forall_r$$

Et, comme :

$$P_\kappa\{\underline{\overrightarrow{x}}/\overrightarrow{x}\} \triangleq \forall \underline{\overrightarrow{x_\kappa}}\ (\underline{\overrightarrow{x_\kappa}} \in \tau^{n_\kappa} \wedge \underline{\overrightarrow{x_\kappa}} <_{n_\kappa} \overrightarrow{t_\kappa}\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}) \Rightarrow Q_\kappa\{\underline{\overrightarrow{x_\kappa}}/\overrightarrow{x_\kappa}\}$$

on peut écrire :

Π_{16} :

$$\frac{\Pi_{15}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), P_\kappa \vdash P_\kappa\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}}\ \forall_r$$

Π_{17}

$$\frac{\Pi_{16}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}), P_\kappa, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P) \vdash P_\kappa\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}, P\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}}\ w$$

Let $p \in \mathbb{N}$ et $\overrightarrow{y} \in \mathcal{X}^p$, such that $\overrightarrow{y} = \overrightarrow{\mathcal{V}ar(P_\kappa \Rightarrow P)}$

Π_{18} :

$$\frac{}{(\underline{\overrightarrow{y}} \in \tau^p \wedge \underline{\overrightarrow{y}} <_p \overrightarrow{y}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\} \vdash (\underline{\overrightarrow{y}} \in \tau^p \wedge \underline{\overrightarrow{y}} <_p \overrightarrow{y}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\}}\ Ax$$

Π_{19} :

$$\frac{\Pi_{18}}{\forall \underline{\overrightarrow{y}}\ (\underline{\overrightarrow{y}} \in \tau^p \wedge \underline{\overrightarrow{y}} <_p \overrightarrow{y}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\} \vdash (\underline{\overrightarrow{y}} \in \tau^p \wedge \underline{\overrightarrow{y}} <_p \overrightarrow{y}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\}}\ \forall_l$$

Et comme :

$$\mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \triangleq (\underline{\overrightarrow{y}} \in \tau^p \wedge \underline{\overrightarrow{y}} <_p \overrightarrow{y}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\}$$

On a :

Π_{19} :

$$\frac{\Pi_{18}}{\mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \vdash (\underline{\overrightarrow{y}} \in \tau^p \wedge \underline{\overrightarrow{y}} <_p \overrightarrow{y}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\}}\ \forall_l$$

Maintenant, du fait que P_κ est la proposition canonique associée à κ, $\mathcal{V}ar(P_\kappa) = \mathcal{V}ar(\kappa)$, alors, du fait que $\kappa \in \mathcal{RE}$, $\mathcal{V}ar(P_\kappa \Rightarrow P) \subset \mathcal{V}ar(\mathcal{RE} \cup \{P\})$, et comme $\overrightarrow{y} = \overrightarrow{\mathcal{V}ar(P_\kappa \Rightarrow P)}$, $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE} \cup \{P\})}$, et par définition de $<_n$, on a l'égalité suivante :

$$((\underline{\overrightarrow{y}} \in \tau^p \wedge \underline{\overrightarrow{y}} <_p \overrightarrow{y}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{y}}/\overrightarrow{y}\}) = ((\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{x}}/\overrightarrow{x}\})$$

On peut donc ainsi écrire : Π_{19} :

$$\frac{\Pi_{18}}{\mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \vdash (\underline{\overrightarrow{x}} \in \tau^n \wedge \underline{\overrightarrow{x}} <_n \overrightarrow{x}) \Rightarrow (P_\kappa \Rightarrow P)\{\underline{\overrightarrow{x}}/\overrightarrow{x}\}}\ \forall_l$$

Π_{20} :

$$\frac{\Pi_{19}}{\mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau), (\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_p \overrightarrow{x}), P_\kappa\{\overrightarrow{\underline{x}}/\overrightarrow{x}\} \vdash P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}} \text{ lemme logique}$$

Π_{21} :

$$\frac{\Pi_{20}}{\Gamma, NI, P_\kappa, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau), (\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}), P_\kappa\{\overrightarrow{\underline{x}}/\overrightarrow{x}\} \vdash P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}} \; w$$

Π_{22} :

$$\frac{\Pi_{17} \quad \Pi_{21}}{\Gamma, NI, P_\kappa, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau), (\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}) \vdash P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}} \; cut$$

Π_{23} :

$$\frac{\Pi_{22}}{\Gamma, NI, P_\kappa, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \vdash (\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}) \Rightarrow P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}} \; \Rightarrow_r$$

Π_{24} :

$$\frac{\Pi_{23}}{\Gamma, NI, P_\kappa, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \vdash \forall \overrightarrow{\underline{x}} \; (\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}) \Rightarrow P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\}} \; \forall_r$$

Posons maintenant $q \in \mathbb{N}$ et $\overrightarrow{z} \in \mathcal{X}^q$ tels que $\overrightarrow{z} = \overrightarrow{\mathcal{V}ar(P)}$, on a :

$$\mathcal{PE}_{ind}(P, <, \tau) \triangleq \forall \overrightarrow{\underline{z}} \; (\overrightarrow{\underline{z}} \in \tau^q \wedge \overrightarrow{\underline{z}} <_q \overrightarrow{z}) \Rightarrow P\{\overrightarrow{\underline{z}}/\overrightarrow{z}\}$$

Comme $\overrightarrow{z} = \overrightarrow{\mathcal{V}ar(P)}$, $\overrightarrow{x} = \overrightarrow{\mathcal{V}ar(\mathcal{RE} \cup \{P\})}$, et par définition de $<_n$, on obtient l'égalité :

$$((\overrightarrow{\underline{z}} \in \tau^q \wedge \overrightarrow{\underline{z}} <_q \overrightarrow{z}) \Rightarrow P\{\overrightarrow{\underline{z}}/\overrightarrow{z}\}) = ((\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}) \Rightarrow P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\})$$

Donc :

$$\mathcal{PE}_{ind}(P, <, \tau) = ((\overrightarrow{\underline{x}} \in \tau^n \wedge \overrightarrow{\underline{x}} <_n \overrightarrow{x}) \Rightarrow P\{\overrightarrow{\underline{x}}/\overrightarrow{x}\})$$

D'où :

Π_{24} :

$$\frac{\Pi_{23}}{\Gamma, NI, P_\kappa, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \vdash \mathcal{PE}_{ind}(P, <, \tau)} \; \forall_r$$

Π_{25} :

$$\frac{\Pi_{24}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau), P_\kappa \vdash \mathcal{PE}_{ind}(P, <, \tau), P} \; w$$

Π_{26} :

$$\frac{\Pi_{25}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \vdash_\kappa \mathcal{PE}_{ind}(P, <, \tau), P} \; push$$

Et puisque $\mathcal{RE} = \mathcal{RE}' \cup \{\kappa\}$, on a :

Π_{27} :

$$\frac{\Pi_{26}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, \mathcal{PE}_{ind}(P_\kappa \Rightarrow P, <, \tau) \vdash_{\mathcal{RE}} \mathcal{PE}_{ind}(P, <, \tau), P} \; \begin{matrix} w \\ + \\ push \end{matrix}$$

Π_{28} :

$$\frac{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE} \cup \{\mathcal{RE}_{ind}(P,<,\tau)\}} P \quad \text{(supposé)}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, \mathcal{PE}_{ind}(P,<,\tau) \vdash_{\mathcal{RE}} P \quad \text{(supposé)}} \; pop$$

Π_{29}

$$\frac{\Pi_{28}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, \mathcal{PE}_{ind}(P,<,\tau), \mathcal{P}_{ind}(P_\kappa \Rightarrow P,<,\tau) \vdash_{\mathcal{RE}} P} \; w$$

Π_{30} :

$$\frac{\Pi_{27} \quad \Pi_{29}}{\Gamma, NI, \overrightarrow{x} \in \tau^n, \mathcal{P}_{ind}(P_\kappa \Rightarrow P,<,\tau) \vdash_{\mathcal{RE}} P} \; cut$$

Π_{31} :

$$\frac{\Pi_{30}}{\Gamma, NI, \overrightarrow{x} \in \tau^n \vdash_{\mathcal{RE} \cup \mathcal{RE}_{ind}(P_\kappa \Rightarrow P,<,\tau)} P} \; push$$

ce qui achève la preuve. $\square$

B

Théorème de compatibilité principale

On suppose que $\leqslant$ est un quasi-ordre du type de celui défini à la section 1.2 de la partie II

Notation: Deux termes $t,\ t' \in \mathcal{T}(\Sigma, \mathcal{X})$ étant donnés, on écrira $t\#t'$ si t et t' ne peuvent être comparés avec le quasi-ordre $\leqslant$.

Lemme B.24 Pour tous $t_1,\ t_2,\ t_3,\ t_4 \in T_\Sigma[X]$, on a :

$$(t_1 < t_3 \wedge t_2 < t_3) \Rightarrow t_1 \approx t_2 <_e t_3 \approx t_4$$

Lemme B.25 Pour tous $t_1,\ t_2,\ t_3,\ t_4 \in T_\Sigma[X]$, on a :

$$(t_1 \leqslant t_3 \wedge t_2 < t_4) \Rightarrow t_1 \approx t_2 <_e t_3 \approx t_4$$

Lemme B.26 Pour tous $t_1,\ t_2,\ t_3,\ t_4 \in T_\Sigma[X]$, on a :

$$(t_1 \gtrless t_2 \wedge t_1 \gtrless t_3 \wedge t_1 \approx t_2 <_e t_3 \approx t_4) \Rightarrow t_1 < t_4$$

Preuve. La preuve des trois lemmes ci-dessus est laissée au lecteur, elle se fait au moyen d'une simple analyse par cas. □

Lemme B.27 Pour tous $t_1,\ t,\ t_3,\ t' \in T_\Sigma[X]$, on a :

$$(t_3 \approx t <_e t_1 \approx t' \wedge t \gtrless t') \Rightarrow (t_1 > t_3 \vee t > t_3)$$

Preuve. Considérons les différents cas dans le tableau suivant :

$C(t_3 \approx t)$ / $C(t_1 \approx t')$	$(\{\{[t_3]\}\}, \{\{[t]\}\})$	$(\{\{[t]\}\}, \{\{[t_3]\}\})$	$(\{\{[t], [t_3]\}\}, \emptyset)$
$(\{\{[t_1]\}\}, \{\{[t']\}\})$	$t < t_3 < t_1$	$t_3 < t < t_1$	$(t < t_1 \wedge t_3 < t_1 \wedge$ $(t\#t_3 \vee t \gtrless t_3)$
$(\{\{[t']\}\}, \{\{[t_1]\}\})$	$\emptyset$	$t_3 < t_1 < t$	$\emptyset$
$(\{\{[t'], [t_1]\}\}, \emptyset)$	$\emptyset$	$(t\#t_1 \vee t \gtrless t_1) \wedge$ $t_3 < t$	$(t\#t_1 \vee t \gtrless t_1) \wedge$ $(t\#t_3 \vee t \gtrless t_3) \wedge t_3 < t_1$

□

Lemme B.28 Considérons une égalité $\alpha_1 \approx \alpha_2$, une égalité $t_1 \approx t_2$, une substitution σ, et deux termes $\alpha'_1[t_1\sigma]_{\omega'}$ et α'_2 tels que : $\alpha_1 \approx \alpha_2 >_e \alpha'_1[t_1\sigma]_{\omega'} \approx \alpha'_2 \geqslant_e \alpha'_1[t_2\sigma]_{\omega'} \approx \alpha'_2$

On a :

1) $\alpha_2 \gtrless \alpha'_2 \Rightarrow (t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$

2) $\alpha_1 \gtrless \alpha'_1[t_1\sigma]_{\omega'} \Rightarrow \left((t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2 \Leftrightarrow \vee \begin{cases} a)\ \omega' \neq \varepsilon \\ b)\ (\alpha_2 \not< \alpha_1) \wedge \neg(t_1\sigma \gtrless t_2\sigma) \\ c)\ t_2\sigma \nleqslant t_1\sigma \\ d)\ t_2\sigma < \alpha_2 \end{cases} \right)$

Preuve. 1) Supposons $\alpha_2 \gtrless \alpha'_2$.

Observons l'inégalité $\alpha_1 \approx \alpha_2 >_e \alpha'_1[t_1\sigma]_{\omega'} \approx \alpha'_2$. Puisque $\alpha_2 \gtrless \alpha'_2$, on peut considérer les deux cas du lemme B.27.

Cas 1 : $\alpha_1 > \alpha'_1[t_1\sigma]_{\omega'}$

Alors $\alpha_1 > t_1\sigma$ (car $<$ est un ordre de simplification)

Sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} \gtrless \alpha'_1[t_2\sigma]_{\omega'}$

Alors $\alpha_1 > t_2\sigma$, et on conclut $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ d'après le lemme B.24

Sous-cas 2 : $\alpha'_1[t_1\sigma]_{\omega'}$ et $\alpha'_1[t_2\sigma]_{\omega'}$ ne sont pas équivalents pour $\leqslant$

En combinant avec l'hypothèse du lemme $B.28$, on obtient :

$\alpha'_1[t_1\sigma]_{\omega'} \approx \alpha'_2 >_e \alpha'_1[t_2\sigma]_{\omega'} \approx \alpha'_2$, et on peut considérer une nouvelle fois les deux différents cas du lemme $B.27$.

Sous-sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} > \alpha'_1[t_2\sigma]_{\omega'}$

Alors $\alpha_1 > t_2\sigma$ et on conclut $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ d'après le lemme B.24

Sous-sous-cas 2 : $\alpha'_2 > \alpha'_1[t_2\sigma]_{\omega'}$

Alors $\alpha_2 > t_2\sigma$ (car $\alpha_2 \gtrless \alpha'_2$ par hypothèse) et on conclut $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme

Cas 2 : $\alpha'_2 > \alpha'_1[t_1\sigma]_{\omega'}$

Alors $\alpha_2 > t_1\sigma$ (car $\alpha_2 \gtrless \alpha'_2$)

Sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} \gtrless \alpha'_1[t_2\sigma]_{\omega'}$

Alors $\alpha_2 > t_2\sigma$ et on conclut $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme B.24

Sous-cas 2 : $\alpha'_1[t_1\sigma]_{\omega'}$ et $\alpha'_1[t_2\sigma]_{\omega'}$ non équivalents pour $\leqslant$.

On considère une nouvelle fois les deux différents cas du lemme $B.27$.

Sous-sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} > \alpha'_1[t_2\sigma]_{\omega'}$

Alors $\alpha_2 > t_2\sigma$ et on conclut $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemma B.24

Sous-sous-cas 2 : $\alpha'_2 > \alpha'_1[t_2\sigma]_{\omega'}$

Alors $\alpha_2 > t_2\sigma$ et on conclut $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemma B.24

2)

Supposons : $\alpha_1 \gtrless \alpha'_1[t_1\sigma]_{\omega'}$

A)Montrons :

$$(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2 \Rightarrow \vee \begin{cases} a)\ \omega' \neq \varepsilon \\ b)\ \alpha_2 \not< \alpha_1 \wedge \neg(t_1\sigma \gtrless t_2\sigma) \\ c)\ t_2\sigma \nleqslant t_1\sigma \\ d)\ t_2\sigma < \alpha_2 \end{cases}$$

Cela revient à montrer l'implication suivante :

$$(((t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2) \wedge (\omega' = \varepsilon) \wedge (\alpha_2 < \alpha_1 \vee t_1\sigma \gtrless t_2\sigma) \wedge (t_2\sigma \leqslant t_1\sigma)) \Rightarrow t_2\sigma < \alpha_2$$

Donc, supposons :

$$((t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2) \wedge (\omega' = \varepsilon) \wedge ((\alpha_2 < \alpha_1) \vee (t_1\sigma \gtrless t_2\sigma) \wedge (t_2\sigma \leqslant t_1\sigma)$$

Cas 1 : $t_1\sigma \gtrless t_2\sigma$

On a :

$$((t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2) \wedge (\omega' = \varepsilon) \wedge (t_1\sigma \gtrless t_2\sigma)$$

Du fait que $\omega' = \varepsilon$, $\alpha_1 \gtrless \alpha'_1[t_1\sigma]_{\omega'}$, et $t_1\sigma \gtrless t_2\sigma$, on a $t_1\sigma \gtrless t_2\sigma \gtrless \alpha_1$. Donc, la proposition $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ devient équivalente à $\alpha_1 \approx \alpha_1 <_e \alpha_1 \approx \alpha_2$. Ainsi, par le lemme B.26, $\alpha_1 < \alpha_2$, d'où $t_2\sigma < \alpha_2$.

Cas 2 : $\neg(t_1\sigma \gtrless t_2\sigma)$

On a :

$$((t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2) \wedge (\omega' = \varepsilon) \wedge (\alpha_2 < \alpha_1) \wedge (t_2\sigma < t_1\sigma)$$

car $\omega' = \varepsilon$ et $\alpha_1 \gtrless \alpha'_1[t_1\sigma]_{\omega'}$, $\alpha_1 \gtrless t_1\sigma$. En outre, du fait que $t_2\sigma < t_1\sigma$ et $\alpha_1 \gtrless t_1\sigma$, on a (cf notations de la section 1.3 du chapitre 1 de la partie II) :

$$C((t_1 \approx t_2)\sigma) = (\{\{[\alpha_1]\}\}, \{\{[t_2\sigma]\}\}) \tag{B.1}$$

Et puisque $\alpha_2 < \alpha_1$, on a également :

$$C(\alpha_1 \approx \alpha_2) = (\{\{[\alpha_1]\}\}, \{\{[\alpha_2]\}\}) \tag{B.2}$$

Maintenant, (B.1), (B.2) et l'hypothèse $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ conduisent à $t_2\sigma < \alpha_2$

B) Montrons :

$$(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2 \Leftarrow \vee \begin{cases} a)\ \omega' \neq \varepsilon \\ b)\ (\alpha_2 \not< \alpha_1) \wedge \neg(t_1\sigma \gtrless t_2\sigma) \\ c)\ t_2\sigma \nleqslant t_1\sigma \\ d)\ \alpha_2 > t_2\sigma \end{cases}$$

Premièrement, en considérant l'équivalence $\alpha_1 \gtrless \alpha'_1[t_1\sigma]_{\omega'}$, on a : $t_1\sigma \leqslant \alpha_1$

Sachant que, par hypothèse, $\alpha_1 \approx \alpha_2 >_e \alpha'_1[t_1\sigma]_{\omega'} \approx \alpha'_2$ et $\alpha_1 \gtrless \alpha'_1[t_1\sigma]_{\omega'}$, on peut considérer les deux cas du lemme $B.27$.

Cas 1 : $\alpha_2 > \alpha'_2$

Sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} \gtrless \alpha'_1[t_2\sigma]_{\omega'}$

Alors $t_1\sigma \not< t_2\sigma$, $t_1\sigma \not> t_2\sigma$ (car $<$ est stable par contexte) et $t_2\sigma \leqslant \alpha_1$

a : $\omega' \neq \varepsilon$:

Alors $t_1\sigma < \alpha_1$, $t_2\sigma < \alpha_1$ et on conclut $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme B.24

b : $(\alpha_2 \not< \alpha_1) \wedge \neg(t_1\sigma \gtrless t_2\sigma)$:

On peut facilement vérifier que l'on a :

- $C(\alpha_1 \approx \alpha_2) \in \{(\{\{[\alpha_2]\}\}, \{\{[\alpha_1]\}\}),\ (\{\{[\alpha_1], [\alpha_2]\}\}, \emptyset)\}$
- $C(t_1\sigma \approx t_2\sigma) = (\{\{[t_1\sigma], [t_2\sigma]\}\}, \emptyset)$

On peut de nouveau considérer les différents cas dans le tableau suivant :

$(\{\{[\alpha_2]\}\}, \{\{[\alpha_1]\}\})$	$(\{\{[\alpha_1], [\alpha_2]\}\}, \emptyset)$
$t_1\sigma \leqslant \alpha_1$ alors $t_1\sigma < \alpha_2$ $t_2\sigma \leqslant \alpha_1$ alors $t_2\sigma < \alpha_2$ d'où le résultat	$t_1\sigma \leqslant \alpha_1$ et $t_2\sigma \leqslant \alpha_1$ si $t_1\sigma \gtrless \alpha_1$ alors $t_2\sigma < \alpha_1$(car $\neg(t_1\sigma \gtrless t_2\sigma)$) donc $t_2\sigma < t_1\sigma$ et contradiction (voir plus haut) donc $t_1\sigma < \alpha_1$ et $t_2\sigma < \alpha_1$ (symétrique) d'où le résultat

c : $t_2\sigma \nleqslant t_1\sigma$

On peut facilement montrer que l'on a :

- $C(t_1\sigma \approx t_2\sigma) = (\{\{[t_1\sigma], [t_2\sigma]\}\}, \emptyset)$

On considère les différents cas dans le tableau suivant :

$(\{\{[\alpha_1]\}\},\{\{[\alpha_2]\}\})$	$(\{\{[\alpha_2]\}\},\{\{[\alpha_1]\}\})$	$(\{\{[\alpha_1],[\alpha_2]\}\},\emptyset)$
$t_1\sigma \leqslant \alpha_1$et $t_2\sigma \leqslant \alpha_1$ si $t_1\sigma \gtrless \alpha_1$then $t_2\sigma \leqslant t_1\sigma_1$ donc contradiction si $t_2\sigma \gtrless \alpha_1$ alors $t_1\sigma \leqslant t_2\sigma$ et, comme $t_1\sigma \not< t_2\sigma,\ t_1\sigma \gtrless t_2\sigma$ (voir plus haut) donc $t_2\sigma \leqslant t_1\sigma$ et contradiction donc $t_1\sigma < \alpha_1$ et $t_2\sigma < \alpha_1$ d'où le résultat	$t_1\sigma \leqslant \alpha_1$ donc $t_1\sigma < \alpha_2$ $t_2\sigma \leqslant \alpha_1$ donc $t_2\sigma < \alpha_2$ d'où le résultat	voir la première colonne

d : $\alpha_2 > t_2\sigma$

En observant que $\alpha_2 > t_2\sigma$ et $\alpha_1 \geqslant t_1\sigma$, on déduit l'inégalité $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme B

Sous-cas 2 : $\neg(\alpha'_1[t_1\sigma]_{\omega'} \gtrless \alpha'_1[t_2\sigma]_{\omega'})$

En combinant avec l'hypothèse du lemme B.28, on a :
$\alpha'_1[t_1\sigma]_{\omega'} \approx \alpha'_2 >_e \alpha'_1[t_2\sigma]_{\omega'} \approx \alpha'_2$, et on considère une nouvelle fois les deux différents cas du lemme $B.27$.

Sous-sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} > \alpha'_1[t_2\sigma]_{\omega'}$

Alors $t_1\sigma \not< t_2\sigma$ et $t_2\sigma < \alpha_1$

a : $\omega' \neq \varepsilon$

Alors $t_1\sigma < \alpha_1$, $t_2\sigma < \alpha_1$ et on conclut l'inégalité $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme B.24

b : $\alpha_2 \not< \alpha_1$ et $\neg(t_1\sigma \gtrless t_2\sigma)$

On peut facilement vérifier que l'on a :
– $C(\alpha_1 \approx \alpha_2) \in \{(\{\{[\alpha_2]\}\},\{\{[\alpha_1]\}\}),\ (\{\{[\alpha_1],[\alpha_2]\}\},\emptyset)\}$
– $C(t_1\sigma \approx t_2\sigma) \in \{(\{\{[t_1\sigma]\}\},\{\{[t_2\sigma]\}\}),\ (\{\{[t_1\sigma],[t_2\sigma]\}\},\emptyset)\}$
Considérons les différents cas dans le tableau suivant :

	$(\{\{[\alpha_2]\}\},\{\{[\alpha_1]\}\})$	$(\{\{[\alpha_1],[\alpha_2]\}\},\emptyset)$
$(\{\{[t_1\sigma]\}\},\{\{[t_2\sigma]\}\})$	$t_1\sigma \leqslant \alpha_1$ alors $t_1\sigma < \alpha_2$ d'où la conclusion.	$t_1\sigma \leqslant \alpha_1$ d'où la conclusion.
$(\{\{[t_1\sigma],[t_2\sigma]\}\},\emptyset)$	$t_1\sigma \leqslant \alpha_1$ alors $t_1\sigma < \alpha_2$ $t_2\sigma < \alpha_1$ et $t_2\sigma < \alpha_2$ d'où la conclusion.	$t_1\sigma \leqslant \alpha_1,\ t_2\sigma < \alpha_1$ d'où la conclusion.

c : $t_2\sigma \nleqslant t_1\sigma$

On a :

– $C(t_1\sigma \approx t_2\sigma) \in \{(\{\{[t_2\sigma]\}\}, \{\{[t_1\sigma]\}\}),\ (\{\{[t_1\sigma], [t_2\sigma]\}\}, \emptyset)\}$
Considérons les différents cas dans le tableau suivant :

	$(\{\{[\alpha_1]\}\}, \{\{[\alpha_2]\}\})$	$(\{\{[\alpha_2]\}\}, \{\{[\alpha_1]\}\})$	$(\{\{[\alpha_1], [\alpha_2]\}\}, \emptyset)$
$(\{\{[t_2\sigma]\}\}, \{\{[t_1\sigma]\}\})$	$t_2\sigma < \alpha_1$ d'où le résultat.	$t_2\sigma < \alpha_1$ donc $t_2\sigma < \alpha_2$ d'où la conclusion.	$t_2\sigma < \alpha_1$ d'où la conclusion.
$(\{\{[t_1\sigma], [t_2\sigma]\}\}, \emptyset)$	$t_1\sigma \leqslant \alpha_1$ $t_2\sigma < \alpha_1$ $t_2\sigma \nleqslant t_1\sigma$ donc $t_1\sigma < \alpha_1$ d'où la conclusion.	$t_1\sigma \leqslant \alpha_1$ donc $t_1\sigma < \alpha_2$ $t_2\sigma < \alpha_1$ donc $t_2\sigma < \alpha_2$ d'où la conclusion.	voir la première colonne

d : $\alpha_2 > t_2\sigma$

Sachant que $\alpha_2 > t_2\sigma$ et $\alpha_1 \geqslant t_1\sigma$, on déduit l'inégalité $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme B

Sous-sous-cas 2 : $\alpha'_2 > \alpha'_1[t_2\sigma]_{\omega'}$

En observant l'inégalité $\alpha_2 > \alpha'_2$ (cas 1), on obtient $\alpha_2 > t_2\sigma$
Maintenant, sachant que $\alpha_2 > t_2\sigma$ et $\alpha_1 \geqslant t_1\sigma$, on déduit $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme B

Cas 2 : $\alpha_1 > \alpha'_2$

Sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} \gtrless \alpha'_1[t_2\sigma]_{\omega'}$: voir Cas 1/Sous-cas 1.

Sous-cas 2 : $\neg(\alpha'_1[t_1\sigma]_{\omega'} \gtrless \alpha'_1[t_2\sigma]_{\omega'})$

En combinant avec l'hypothèse du lemme B.28, on a
$\alpha'_1[t_1\sigma]_{\omega'} \approx \alpha'_2 >_e \alpha'_1[t_2\sigma]_{\omega'} \approx \alpha'_2$, et on considère une fois de plus les deux différents cas du lemme $B.27$.

Sous-sous-cas 1 : $\alpha'_1[t_1\sigma]_{\omega'} > \alpha'_1[t_2\sigma]_{\omega'}$

On a $t_1\sigma \not< \sigma(t_2)$ et $\alpha_1 > t_2\sigma$, donc cf Cas 1/Sous-cas 2/Sous-sous-cas 1

Sous-sous-cas 2 : $\alpha'_2 > \alpha'_1[t_2\sigma]_{\omega'}$

$\alpha_1 > \alpha'_2$ (cas 2), et $\alpha'_2 > \alpha'_1[t_2\sigma]_{\omega'}$ (sous-sous-cas 2), donc $\alpha_1 > t_2\sigma$.

Sous-sous-sous-cas 1 : $\alpha_1 \gtrless t_1\sigma$

$\alpha_1 > t_2\sigma$, et $\alpha_1 \gtrless t_1\sigma$, donc $t_1\sigma > t_2\sigma$. On en déduit alors $t_1\sigma \not< t_2\sigma$, et on peut se référer au cas 1/sous-cas 2/sous-sous-cas 1

Sous-sous-sous-cas 2 : $\alpha_1 > t_1\sigma$

Alors $t_1\sigma < \alpha_1$, $t_2\sigma < \alpha_1$ et on obtient $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$ par le lemme B.24 □

Lemme B.29 Considérons deux égalités $\alpha_1 \approx \alpha_2$, $t_1 \approx t_2$, une substitution σ, et trois termes α'_1, $\alpha''_1[t_1\sigma]_{\omega''}$ et α''_2, tels que :

$$\begin{cases} 1)\alpha_1 > \alpha'_1 \\ 2)\alpha'_1 \approx \alpha_2 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \text{ ou } \alpha_2 \approx \alpha'_1 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \\ 3)\alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \geqslant_e \alpha''_1[t_2\sigma]_{\omega''} \approx \alpha''_2 \\ 4)\neg(\alpha_2 \gtrless \alpha''_1[t_1\sigma]_{\omega''}) \end{cases}$$

Alors $(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha_2$

Preuve. Cas 1 : $\alpha'_1 \approx \alpha_2 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2$.

Alors, par définition de $\geq_e$, $\alpha'_1 \geqslant \alpha''_1[t_1\sigma]_{\omega''}$. Maintenant, en considérant l'hypothèse 4, $\alpha_1 > \alpha'_1$, ce qui conduit à $\alpha_1 > \alpha''_1[t_1\sigma]_{\omega''}$. Or, d'après le lemme B, on a $\alpha_1 \approx \alpha''_2 >_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2$, et en combinant avec l'hypothèse 3, on obtient :

$$\alpha_1 \approx \alpha''_2 >_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \geqslant_e \alpha''_1[t_2\sigma]_{\omega''} \approx \alpha''_2$$

Par le lemme B.28, cela conduit à :

$$(t_1 \approx t_2)\sigma <_e \alpha_1 \approx \alpha''_2 \tag{B.3}$$

Maintenant, du fait de l'inégalité $\alpha'_1 \approx \alpha_2 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2$, et par définition de $\leq_e$, on a $\alpha_2 \geqslant \alpha''_2$, donc, d'après le lemme B, et par définition de $\leqslant_e$, on a :

$$\alpha_1 \approx \alpha''_2 \leqslant_e \alpha_1 \approx \alpha_2 \tag{B.4}$$

(B.3) et (B.4) permettent alors de conclure.

Cas 2 : $\alpha_2 \approx \alpha'_1 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2$

Par définition de $\geq_e$, $\alpha_2 \geqslant \alpha''_1[t_1\sigma]_{\omega''}$, et, en combinant avec l'hypothèse 4, on obtient $\alpha_2 > \alpha''_1[t_1\sigma]_{\omega''}$. Or, d'après le lemms B, on a $\alpha_2 \approx \alpha''_2 >_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2$. En considérant l'hypothèse 3, cela nous conduit à écrire :

$$\alpha_2 \approx \alpha''_2 >_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \geqslant_e \alpha''_1[t_2\sigma]_{\omega''} \approx \alpha''_2$$

D'après le lemme B.28, on obtient alors :

$$(t_1 \approx t_2)\sigma <_e \alpha_2 \approx \alpha''_2 \tag{B.5}$$

Or, en considérant l'inégalité $\alpha_2 \approx \alpha'_1 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2$, et par définition de $\leq_e$, on a $\alpha'_1 \geqslant \alpha''_2$. En considérant maintenant l'hypothèse 1, on a $\alpha_1 > \alpha''_2$, donc, d'après le lemme B, et par symétrie de $<_e$, on a :

$$\alpha_2 \approx \alpha''_2 \leqslant_e \alpha_1 \approx \alpha_2 \tag{B.6}$$

(B.5) et (B.6) permettent alors de conclure. □

Lemme B.30 Considérons deux égalités $\alpha_1 \approx \alpha_2$, $t_1 \approx t_2$, une substitution σ, et trois termes $\alpha''_1[t_1\sigma]_{\omega''}$, α'_1 et α''_2 tels que :

$$\begin{cases} 1)\alpha_1 > \alpha'_1 \\ \\ 2)\alpha'_1 \approx \alpha_2 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \text{ or } \alpha_2 \approx \alpha'_1 \geq_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \\ \\ 3)\alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \geqslant_e \alpha''_1[t_2\sigma]_{\omega''} \approx \alpha''_2 \end{cases}$$

On a :

$$\begin{array}{l} (t_1 \approx t_2)\sigma < \alpha_1 \approx \alpha_2 \Leftrightarrow \\ \neg(\alpha_2 \gtrless \alpha''_1[t_1\sigma]_{\omega''}) \vee \neg(\omega'' = \varepsilon) \vee \neg(\alpha_1 < \alpha_2 \vee t_1\sigma \gtrless t_2\sigma) \vee \neg(t_2\sigma \leqslant t_1\sigma) \vee (t_2\sigma < \alpha_1) \end{array}$$

Preuve. D'après l'hypothèse 2, puisque $\leq_e \subset \leqslant_e$, et par symétrie de $\leqslant_e$, on a $\alpha'_1 \approx \alpha_2 \geqslant_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2$. Maintenant, sachant que $\alpha_1 > \alpha'_1$, et d'après le lemme B, $\alpha_1 \approx \alpha_2 >_e \alpha'_1 \approx \alpha_2$. En combinant avec l'hypothèse 3), cela entraîne :

$$\alpha_1 \approx \alpha_2 >_e \alpha''_1[t_1\sigma]_{\omega''} \approx \alpha''_2 \geqslant_e \alpha''_1[t_2\sigma]_{\omega''} \approx \alpha''_2$$

$\Rightarrow$:
Supposons : $(t_1 \approx t_2)\sigma < \alpha_1 \approx \alpha_2$
Si $\alpha_2 \gtrless \alpha''_1[t_1\sigma]_{\omega''}$, on a, d'après le lemme B.28 :

$$\neg(\omega'' = \varepsilon) \vee \neg(\alpha_1 < \alpha_2 \vee t_1\sigma \gtrless t_2\sigma) \vee \neg(t_2\sigma \leqslant t_1\sigma) \vee (t_2\sigma < \alpha_1)$$

$\Leftarrow$:
Supposons :

$$\neg(\omega'' = \varepsilon) \vee \neg(\alpha_1 < \alpha_2 \vee t_1\sigma \gtrless t_2\sigma) \vee \neg(t_2\sigma \leqslant t_1\sigma) \vee (t_2\sigma < \alpha_1)$$

Cas 1 : $\alpha_2 \gtrless \alpha''_1[t_1\sigma]_{\omega''}$.

Alors, d'après le lemme B.28, on a $(t_1 \approx t_2)\sigma < \alpha_1 \approx \alpha_2$

Cas 2 : $\neg(\alpha_2 \gtrless \alpha''_1[t_1\sigma]_{\omega''})$

Alors, en considérant les hypothèses et le lemme B.29, $(t_1 \approx t_2)\sigma < \alpha_1 \approx \alpha_2$. □

Théorème B.4 (Théorème de compatibilité principale) *Considérons des égalités Q_1, Q_2, Q_3 et Q_4, $l \to r$ une règle de réécriture telle que $l > r$, κ_0 une règle de réécriture $l_{\kappa_0} \to r_{\kappa_0}$, ou une égalité $l_{\kappa_0} \approx r_{\kappa_0}$. Considérons l'inégalité I : $(l_{\kappa_0} \approx r_{\kappa_0})\sigma <_e Q_1$ et supposons :*

1. $Q_1 \rightarrow_{l \rightarrow r,\ j.\omega_j,\ \theta} Q_2$
2. $Q_2 \geq Q_3$
3. $Q_3 \rightarrow_{\kappa_0,\ i.\omega_i,\ \sigma} Q_4$
4. $Q_3 \geqslant_e Q_4$

Alors :

1. *I est vérifiée si $\omega_i \neq \varepsilon$ ou $i = j$*
2. *Si $\omega_i = \varepsilon$ et $i \neq j$:*
   (a) *si $l_{\kappa_0} > r_{\kappa_0}$, alors :*
   $$I \Leftrightarrow ((Q_{1|i} \gtrless l_{\kappa_0}\sigma) \wedge (Q_{1|j} < Q_{1|i}) \Rightarrow (Q_{1|j} > r_{\kappa_0}\sigma))$$
   (b) *si $l_{\kappa_0} \gtrless r_{\kappa_0}$, alors :*
   $$I \Leftrightarrow ((Q_{1|i} \gtrless l_{\kappa_0}\sigma) \Rightarrow (Q_{1|j} > r_{\kappa_0}\sigma))$$
   (c) *sinon :*
   $$\begin{aligned} I \Leftrightarrow (\quad & ((Q_{1|i} \gtrless l_{\kappa_0}\sigma) \wedge ((Q_{1|j} < Q_{1|i}) \vee (l_{\kappa_0}\sigma \gtrless r_{\kappa_0}\sigma)) \wedge (r_{\kappa_0}\sigma \leqslant l_{\kappa_0}\sigma)) \\ & \Rightarrow (Q_{1|j} > r_{\kappa_0}\sigma)\ ) \end{aligned}$$

Preuve. Soit $\overline{j} \in \{1, 2\}$ tel que $\overline{j} \neq j$.

On peut facilement montrer, en utilisant le lemme B.30, que l'on a :

$$\begin{aligned} & (l_{\kappa_0} \approx r_{\kappa_0})\sigma <_e Q_1 \Leftrightarrow \\ & \neg(Q_{1|\overline{j}} \gtrless l_{\kappa_0}\sigma) \vee \neg(\omega_i = \varepsilon) \vee \neg(Q_{1|j} < Q_{1|\overline{j}} \vee l_{\kappa_0}\sigma \gtrless r_{\kappa_0}\sigma) \vee \neg(r_{\kappa_0}\sigma \leqslant l_{\kappa_0}\sigma) \vee (Q_{1|j} > r_{\kappa_0}\sigma) \end{aligned} \tag{B.7}$$

Cas 1 : $\omega_i \neq \varepsilon$ or $i = j$.

Sous-cas 1 : $\omega_i \neq \varepsilon$. On conclut en considérant (B.7).

Sous-cas 2 : $\omega_i = \varepsilon$ et $i \neq j$.

On peut facilement vérifier que l'on a $Q_{1|j} > Q_{2|j} \geqslant Q_{4|j}$. Or, sachant que $i = j$, $Q_{4|j} = Q_{4|i}$, et que $Q_{4|i} \geqslant r_{\kappa_0}\sigma$, on obtient $Q_{1|j} > r_{\kappa_0}\sigma$, et on conclut d'après (B.7).

Cas 2 : $i \neq j$.

Alors $\overline{j} = i$, et on conclut d'après (B.7). □

C

Formes plates

Dans cette annexe, on se place dans une algèbe de termes variadiques. + désignera un symbole variadique quelconque et, pour tout terme aplati $\overline{s} = \overline{s_1} + \ldots \overline{s_n} \in \mathcal{T}V(\Sigma, \mathcal{X})$, on identifiera $\varepsilon \in \mathcal{D}om(\overline{s})$ et $\{1, \ldots, n\}$.

Lemme C.31 Un terme aplati $\overline{s}$ et une position $\omega \in \mathcal{D}om(s)$ étant donnés, il existe p et ω_0, tels que :

1. $\omega = p.\omega_0$
2. $p = \varepsilon$ ou il existe $i_1, \ldots i_k \in \mathbb{N}$ tels que $p = i_1 \ldots . i_k$
3. $\omega_0 \subset \mathbb{N} \setminus \{0\}$

Preuve. par récurrence immédiate □

Lemme C.32 Pour tous $t,\ t' \in \mathcal{T}(\Sigma, \mathcal{X})$, $t =_A t'$ si, et seulement si $\overline{t} = \overline{t'}$.

Preuve. Immédiat. □

Proposition C.5 Etant donné un terme s, une règle $l \to r$, une position $\omega \in \mathcal{D}om(s)$, et une substitution σ, on a l'égalité suivante :

$$\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{flat_s(\omega)}$$

Preuve. Soit $s \in \mathcal{T}(\Sigma, \mathcal{X})$ et $\omega \in \mathcal{D}om(s)$. On va faire une preuve par récurrence sur la profondeur de ω.

- Cas de base : $\omega = \varepsilon$:
  On a alors $\overline{s[t]_\varepsilon} = \overline{t}$, et $\overline{s}[\overline{t}]_{flat_s(\varepsilon)} = \overline{s}[\overline{t}]_\varepsilon = \overline{t}$
- Cas de récurrence : $\omega = i.\omega_i$:
  - Cas 1 : $s(\varepsilon) \notin \mathcal{V}$.

    Posons $f = s(\varepsilon)$. On peut alors écrire :

$$\begin{array}{rcl} \overline{s} & = & f\overline{s_1} \ldots \overline{s_n} \\ \overline{s[t]_\omega} & = & f\overline{s_1} \ldots \overline{s_{i-1}}\,\overline{s_i[t]_{\omega_i}}\,\overline{s_{i+1}} \ldots \overline{s_n} \end{array} \tag{C.1}$$

    L'application de l'hypothèse de récurrence permet maintenant d'écrire l'égalité $\overline{s_i[t]_{\omega_i}} = \overline{s_i}[\overline{t}]_{flat_{s_i}(\omega_i)}$, ce qui implique, d'après (C.1), $\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{i.flat_{s_i}(\omega_i)}$. On conclut alors en observant que $flat_s(\omega) = i.flat_{s_i}(\omega_i)$

– Cas 2 : $s(\varepsilon) \in \mathcal{V}$.
En posant $+ = s(\varepsilon)$, on a $s = u + v$ et $\omega = 1.\omega_1$ ou $\omega = 2.\omega_1$
– Sous cas 1 : $\omega = 1.\omega_1$

On a alors

$$\begin{array}{rcl} \overline{s[t]_\omega} & = & \overline{u[t]_{\omega_1} + v} \\ & = & \overline{\overline{u[t]_{\omega_1}} + \overline{v}} \end{array} \tag{C.2}$$

L'application de l'hypothèse de récurrence permet maintenant d'écrire l'égalité $\overline{u[t]_{\omega_1}} = \overline{u}[\overline{t}]_{flat_u(\omega_1)}$, ce qui implique, d'après (C.2), $\overline{s[t]_\omega} = \overline{\overline{u}[\overline{t}]_{flat_u(\omega_1)} + \overline{v}}$.
Posons maintenant $\overline{u} = \overline{u_1} + \ldots + \overline{u_n}$ et $\overline{v} = \overline{v_1} + \ldots + \overline{v_m}$

Si $n = 1$, on peut ainsi écrire :

$$\overline{s[t]_\omega} = \overline{\overline{u}_1[\overline{t}]_{flat_{u_1}(\omega_1)} + \overline{v_1} + \ldots \overline{v_m}}$$

Mais comme $\overline{s} = \overline{u_1} + \overline{v_1} + \ldots + \overline{v_m}$, on en déduit, par définition :

$$\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{1.flat_{u_1}(\omega_1)}$$

et on conclut en observant que $flat_s(\omega) = 1.flat_{u_1}(\omega_1)$.

Si $n \geqslant 2$, on peut ainsi écrire :

$$\overline{s[t]_\omega} = \overline{(\overline{u}_1 + \ldots + \overline{u_n})[\overline{t}]_{flat_u(\omega_1)} + \overline{v_1} + \ldots + \overline{v_m}}$$

Si $\omega_1 = \varepsilon$, l'égalité ci-dessus devient :

$$\begin{array}{rcl} \overline{s[t]_\omega} & = & \overline{\overline{t} + \overline{v_1} + \ldots + \overline{v_m}} \\ & = & \overline{(\overline{u_1} + \ldots + \overline{u_n} + \overline{v_1} + \ldots + \overline{v_m})[\overline{t}]_{\{1,\ldots,n\}}} \end{array}$$

Mais comme $\overline{s} = \overline{u_1} + \ldots + \overline{u_n} + \overline{v_1} + \ldots + \overline{v_m}$, on en déduit, par définition :

$$\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{\{1,\ldots,n\}}$$

et on conclut en observant que $flat_s(\omega) = \{1, \ldots, n\}$.

Si $\omega_1 = j.\omega_2$, l'égalité ci-dessus devient :

$$\begin{array}{rcl} \overline{s[t]_\omega} & = & \overline{(\overline{u_1} + \ldots + \overline{u_n})[\overline{t}]_{flat_u(j.\omega_2)} + \overline{v_1} + \ldots + \overline{v_m}} \\ & = & \overline{(\overline{u_1} + \ldots + \overline{u_n} + \overline{v_1} + \ldots + \overline{v_m})[\overline{t}]_{flat_u(j.\omega_2)}} \end{array}$$

Mais comme $\overline{s} = \overline{u_1} + \ldots + \overline{u_n} + \overline{v_1} + \ldots + \overline{v_m}$, on en déduit, par définition :

$$\overline{s[t]_\omega} = \overline{s}[\overline{t}]_{flat_u(j.\omega_2)}$$

et on conclut en observant que $flat_s(\omega) = flat_u(j.\omega_2)$
– Sous-cas 2 : se traite de la même façon que le Sous-cas 1.
□

Lemme C.33 Soient s, s' deux termes, et $\omega \in \mathcal{D}om(\overline{s})$. Si $\overline{s} \equiv_p \overline{s'}$, il existe $\omega' \in \mathcal{D}om(\overline{s'})$, tel que :

- $\overline{s}_{|\omega} \equiv_p \overline{s'}_{|\omega'}$
- Pour tout terme t, $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{|\omega'}$

Preuve. Par récurrence sur $Level(\omega)$.

- Si $Level(\omega) = 0$
  - Si $\omega = \varepsilon$, la conclusion est immédiate.
  - Sinon.
    Il existe $n,\ k,\ i_1, \ldots,\ i_k \in \mathbb{N}$, $s_1,\ \ldots,\ s_n \in \mathcal{T}(\Sigma, \mathcal{X})$ et une permutation τ, tels que
    1. $\overline{s} = \overline{s_1} + \ldots + \overline{s_n}$
    2. $\overline{s}' = \overline{s'_1} + \ldots + \overline{s'_n}$
    3. Pour tout i, $\overline{s'}_i \equiv_p \overline{s_{\tau(i)}}$.
    4. $\omega \subset \{1, \ldots n\}$

    Maintenant, si on pose $\omega' = \tau(\omega)$, on a clairement $\overline{s}_{|\omega} \equiv_p \overline{s'}_{|\omega'}$ et $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{\omega'}$
- Si $Level(\omega) = l + 1$. Soit $i \in \mathbb{N}^*$ et ω_i tels que $\omega = i.\omega_i$.
  - Cas 1. $\overline{s} = f\overline{s_1} \ldots \overline{s_n}$, avec $f \notin \Sigma_V$. Par définition de $\equiv_p$ (définition 1.12), Il existe $s'_1, \ldots, s'_n$, tels que $\overline{s'} = f\overline{s'_1} \ldots \overline{s'_n}$ et $\overline{s_j} \equiv_p \overline{s'}_j$ pour tout j. Observons maintenant que, puisque $\overline{s_i} \equiv_p \overline{s'}_i$, et puisque $\omega_i \in \mathcal{D}om(\overline{s_i})$, par récurrence , on peut affirmer qu'il existe $\omega'_i \in \mathcal{D}om(\overline{s'_i})$ tel que $\overline{s_i}_{|\omega_i} \equiv_p \overline{s'_i}_{|\omega'_i}$ et $\overline{s_i}[\overline{t}]_{\omega_i} \equiv_p \overline{s'_i}[\overline{t}]_{\omega'_i}$. Posons $\omega' = i.\omega'_i$, on a clairement $\overline{s}_{|\omega} \equiv_p \overline{s'}_{|\omega'}$ et $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{\omega'}$
  - Cas 2. $\overline{s} = \overline{s_1} + \ldots + \overline{s_n}$. Par définition de $\equiv_p$, il existe $s'_1, \ldots, s'_n \in \mathcal{T}(\Sigma, \mathcal{X})$ et une permutation τ de $\{1, \ldots, n\}$, tels que $\overline{s'} = \overline{s'_1} + \ldots + \overline{s'_n}$ et $\overline{s'_i} \equiv_p \overline{s_{\tau(i)}}$ pour tout i. Observons alors que, puisque $\overline{s'_{\tau^{-1}(i)}} \equiv_p \overline{s_i}$, et puisque $\omega_i \in \mathcal{D}om(\overline{s_i})$, par récurrence, on peut affirmer qu'il existe $\omega'_i \in \mathcal{D}om(\overline{s'_{\tau^{-1}(i)}})$ tel que $\overline{s_i}_{|\omega_i} \equiv_p \overline{s'_{\tau^{-1}(i)}}_{|\omega'_i}$ et $\overline{s_i}[\overline{t}]_{\omega_i} \equiv_p \overline{s'_{\tau^{-1}(i)}}[\overline{t}]_{\omega'_i}$. Si on pose maintenant $\omega' = \tau^{-1}(i).\omega'_i$, on obtient clairement $\overline{s}_{|\omega} \equiv_p \overline{s'}_{\omega'}$ et $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{\omega'}$.

□

Lemme C.34 Les propositions 1 et 2 ci-dessous sont équivalentes :

1. $(\omega \in \mathcal{D}om(\overline{s}))$ et il existe $i, k \in \mathbb{N}$ et une séquence d'entiers ω_0, tels que $\omega = \omega_0.i$ ou $\omega = \omega_0.\{i, i+1, \ldots, i+k\}$
2. Il existe $s' \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega' \in \mathcal{D}om(s')$, tels que $\overline{s'} = \overline{s}$ et $\omega = flat_{s'}(\omega')$

Preuve. – $1 \Rightarrow 2$: par récurrence sur $Level(\omega)$

1. Si $Level(\omega) = \varepsilon$:
   (a) Si $\omega = \varepsilon$: on a $\varepsilon = flat_s(\varepsilon)$
   (b) Sinon : $\omega = i$ ou $\omega = \{i, i+1, \ldots, i+k\}$, et il existe n, s_1, $\ldots s_n$, tels que $\overline{s} = \overline{s_1} + \ldots \overline{s_n}$ et $\{i, i+1, \ldots, i+k\} \subsetneq \{1, \ldots n\}$
   On va effectuer un raisonnement par récurrence sur n.
   - $n = 2$
     - Cas 1 : $i = 1$: alors $\omega = 1$, car $\omega \subsetneq \{1, 2\}$. On a alors $flat_s(\omega) = flat_s(1) = 1 = \omega$ et la preuve est terminée.
     - Cas 2 : $i = 2$: alors $\omega = 2$, car $\omega \subsetneq \{1, 2\}$. Donc $flat_s(\omega) = flat_s(2) = 2 = \omega$ et la preuve est terminée.

- $n > 2$
  - Cas 1 : $i = 1$. Alors $i+k < n$, car $\{i, i+1, \ldots, i+k\} \subsetneq \{1, \ldots n\}$. Posons u tel que $\overline{u} = \overline{s_1} + \ldots + \overline{s_{n-1}}$, et $s' = u + s_n$. $\omega \in \mathcal{D}om(\overline{u})$, donc, par récurrence, il existe $u' \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega'' \in \mathcal{D}om(u')$, tels que $\overline{u'} = \overline{u}$ et $\omega = flat_{u'}(\omega'')$. Soit $\omega' = 1.\omega''$.
    $\rightarrow$ Sous-cas 1 : $\omega'' = \varepsilon$. On a alors $flat_{s'}(\omega') = \{1, \ldots, n-1\}$, et, puisque $\omega = flat_{u'}(\omega'')$, on peut écrire $\omega = flat_{u'}(\varepsilon)$, et, du fait que $\overline{u'} = \overline{s_1} + \ldots \overline{s_{n-1}}$, cela nous conduit à $\omega = \{1, \ldots, n-1\}$. Donc $\omega = flat_{s'}(\omega')$
    $\rightarrow$ Sous-cas 2 : $\omega'' \neq \varepsilon$. Alors $flat_{s'}(\omega') = flat_{u'}(\omega'') = \omega$
  - Cas 2 : $i > 1$. Soit v tel que $\overline{v} = \overline{s_2} + \ldots + \overline{s_n}$, et $s' = s_1 + v$.
    $\{i-1, \ldots, i+k-1\} \in \mathcal{D}om(\overline{v})$, donc, par récurrence, il existe $v' \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega'' \in \mathcal{D}om(v')$, tels que $\overline{v'} = \overline{v}$ et $flat_{v'}(\omega'') = \{i-1, \ldots, i+k-1\}$. Posons $\omega' = 2.\omega''$.
    $\rightarrow$ Sous-cas 1 : $\omega'' = \varepsilon$. Alors $flat_{s'}(\omega') = \{2, \ldots, n\}$, et, étant donné que $flat_{v'}(\omega'') = \{i-1, \ldots, i+k-1\}$, on a $flat_{v'}(\varepsilon) = \{i-1, \ldots, i+k-1\}$, et, du fait que $\overline{v'} = \overline{s_2} + \ldots \overline{s_n}$, cela entraîne $\{1, \ldots, n-1\} = \{i-1, \ldots, i+k-1\}$. On a donc $\omega = flat_{s'}(\omega')$
    $\rightarrow$ Sous-cas 2 : $\omega'' \neq \varepsilon$. Alors $flat_{s'}(\omega') = \{1+i-1, \ldots, 1+i+k-1\} = \omega$

2. Si $Level(\omega) = l + 1$. Posons $i \in \mathbb{N}^*$ et ω' tels que $\omega = i.\omega_i$.
   (a) Cas 1 : $s = fs_1 \ldots s_n$. Alors $\overline{s} = f\overline{s_1} \ldots \overline{s_n}$. $\omega_i \in \mathcal{D}om(\overline{s_i})$, donc, par récurrence, il existe $s'_i \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega'_i \in \mathcal{D}om(s'_i)$, tels que $\overline{s'_i} = \overline{s_i}$ et $\omega_i = flat_{s'_i}(\omega'_i)$. Posons $s' = fs_1 \ldots s_{i-1}s'_i s_{i+1} \ldots s_n$ et $\omega' = i.\omega'_i$. On a $\overline{s'} = \overline{s}$, et, par définition de l'application $flat_s$ (définition 2.4), $flat_{s'}(\omega') = i.flat_{s'_i}(\omega'_i) = i.\omega_i = \omega$.
   (b) Cas 2 : $\overline{s} = \overline{s_1} + \ldots \overline{s_n}$
   - Sous-cas 1 : $i = 1$. Posons v tel que $\overline{v} = \overline{s_2} + \ldots + \overline{s_n}$.
     $\omega_1 \in \mathcal{D}om(s_1)$, donc, par récurrence, il existe $s'_1 \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega'_1 \in \mathcal{D}om(s'_1)$, tels que $\overline{s'_1} = \overline{s_1}$ et $flat_{s'_1}(\omega'_1) = \omega_1$. Posons $s' = s'_1 + v$ et $\omega' = 1.\omega'_1$. On a $\overline{s'} = \overline{s}$, et $flat_{s'}(\omega') = 1.flat_{s'_1}(\omega'_1) = 1.\omega_1 = \omega$.
   - Sous-cas 2 : $i = n$. Soit u tel que $\overline{u} = \overline{s_1} + \ldots + \overline{s_{n-1}}$.
     $\omega_n \in \mathcal{D}om(s_n)$, donc, par récurrence, il existe $s'_n \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega'_n \in \mathcal{D}om(s'_n)$, tels que $\overline{s'_n} = \overline{s_n}$ et $flat_{s'_n}(\omega'_n) = \omega_n$. Posons $s' = u + s'_n$ et $\omega' = 2.\omega'_n$. On a $\overline{s'} = \overline{s}$, et $flat_{s'}(\omega') = n.flat_{s'_n}(\omega'_n) = n.\omega_n = \omega$.
   - Sous-cas 3 : $1 < i < n$. Soit v tel que $\overline{v} = \overline{s_{i+1}} + \ldots + \overline{s_n}$.
     $\omega_i \in \mathcal{D}om(s_i)$, donc, par récurrence, il existe $s'_i \in \mathcal{T}(\Sigma, \mathcal{X})$, $\omega'_i \in \mathcal{D}om(s'_i)$, tels que $\overline{s'_i} = \overline{s_i}$ et $flat_{s'_i}(\omega'_i) = \omega_i$.
     Soit $s' = (\ldots(s_1 + s_2) + s_3) + \ldots + s_{i-1}) + s'_i) + v$ et $\omega' = 1.2.\omega'_i$. On a $\overline{s'} = \overline{s}$, et $flat_{s'}(\omega') = flat_{u'}(2.\omega'_i) = i.flat_{s'_i}(\omega'_i) = i.\omega_i = \omega$.

- $2 \Rightarrow 1$: il suffit de montrer que, pour tout $s' \in \mathcal{T}(\Sigma, \mathcal{X})$, il existe $i, k \in \mathbb{N}$ et une séquence d'entiers ω_0, tels que $flat_{s'}(\omega') = \omega_0.i$ ou $flat_{s'}(\omega') = \omega_0.\{i, i+1, \ldots, i+k\}$.
  On effectue une preuve par récurrence sur la profondeur de s'.
  - Cas 1 : $s' = fs'_1 \ldots s'_n$ avec $f \notin \Sigma_V$.
    - Sous-cas 1 : $\omega' = \varepsilon$. $flat_{s'}(\varepsilon) = \varepsilon$.
    - Sous-cas 2 : Sinon, il existe $j \in \{1, \ldots, n\}$ et $\omega'_j \in \mathcal{D}om(s'_j)$ tels que $\omega' = j.\omega'_j$. Par récurrence, on peut affirmer qu'il existe $i, k \in \mathbb{N}$ et une séquence d'entiers ω_1 tels que

$$flat_{s'_j}(\omega'_j) = \omega_1.i \text{ ou } flat_{s'_j}(\omega'_j) = \omega_1.\{i, i+1, \ldots, i+k\} \tag{C.3}$$

Par définition de $flat_{s'_j}$, et comme $\omega' = j.\omega'_j$ et $f \notin \mathcal{V}$, on a :

$$flat_{s'}(\omega') = j.flat_{s'_j}(\omega'_j) \tag{C.4}$$

On peut alors clairement affirmer que ω a la forme souhaitée à partir des égalités (C.3) et (C.4).

- Cas 2 : $s' = u' + v'$. Soit $m, n \in \mathbb{N}$, $s'_1, \ldots, s'_n, s'_{n+1}, \ldots, s'_{n+m}$, tels que $\overline{u'} = \overline{s'_1} + \ldots + \overline{s'_n}$ et $\overline{v'} = \overline{s'_{n+1}} + \ldots + \overline{s'_{n+m}}$.
  - Sous-cas 1 : $\omega' = \varepsilon$. $flat_{s'}(\varepsilon) = \varepsilon$.
  - Sous-cas 2 : $\omega' = 1$. $flat_{s'}(\omega') = \{1, \ldots, n\}$.
  - Sous-cas 3 : $\omega' = 2$. $flat_{s'}(\omega') = \{n+1, \ldots, n+m\}$.
  - Sous-cas 4 : $\omega' = 1.j.q$. $j.q \in \mathcal{D}om(u')$, donc, par récurrence, il existe $i, k \in \mathbb{N}$, tels que $flat_{u'}(j.q) = \omega_0.\{i, i+1, \ldots, i+k\}$.
    $\rightarrow$ Sous-sous-cas 1 : $n = 1$, alors $flat_{s'}(\omega') = 1.p.\{i, i+1, \ldots, i+k\}$.
    $\rightarrow$ Sous-sous-cas 2 : $n > 1$, alors $flat_{s'}(\omega') = p.\{i, i+1, \ldots, i+k\})$.
  - Sous-cas 5 : $\omega' = 2.j.q$. $j.q \in \mathcal{D}om(v')$, alors, par récurrence, il existe $i, k \in \mathbb{N}$, tels que $flat_{v'}(j.q) = \omega_0.\{i, i+1, \ldots, i+k\}$.
    $\rightarrow$ Sous-sous-cas 1 : $m = 1$, alors $flat_{s'}(\omega') = (n+1).p.\{i, i+1, \ldots, i+k\})$.
    $\rightarrow$ Sous-sous-cas 2 : $m > 1$
    $\longrightarrow$ Sous-sous-sous-cas 1 : $\omega_0 = \varepsilon$ Alors $flat_{s'}(\omega') = \{n+i, \ldots, n+i+k\}$.
    $\longrightarrow$ Sous-sous-sous-cas 2 : il existe $k \in \mathbb{N}$ et une suite d'entiers ω_1 tels que $\omega_0 = k.\omega_1$. Alors $flat_{s'}(\omega') = (n+k).\omega_1.\{i, i+1, \ldots, i+k\}$.

□

Lemme C.35 Pour tous termes $s,\ t \in \mathcal{T}(\Sigma, \mathcal{X})$ et pour tout $\omega \in \mathcal{D}om(\overline{s})$, il existe $s' \in \mathcal{T}(\Sigma, \mathcal{X})$ et $\omega' \in \mathcal{D}om(s')$, tels que $\overline{s} \equiv_p \overline{s'}$, $\overline{s}_{|\omega} = \overline{s'}_{|flat_{s'}(\omega')}$, et $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{flat_{s'}(\omega')}$

Preuve. On note $\overline{s} = f\overline{s_1} \ldots \overline{s_n}$ et on effectue un raisonnement par récurrence sur la profondeur de ω.

- Si $\omega = \varepsilon$, il suffit de poser $s' = s$ et $\omega' = \varepsilon$.
- Si $\omega \subset \mathbb{N}$, posons $\mathbb{N} - \omega = \{i_1, \ldots, i_k\}$, et $\overline{u} = \overline{s_{i_1}} + \ldots + \overline{s_{i_k}} + \sum_{j \in \omega} \overline{s_j}$. D'après le lemme C.34, il existe $s' \in \mathcal{T}(\Sigma, \mathcal{X})$ et $\omega' \in \mathcal{D}om(s')$, tels que $\overline{s'} = \overline{u}$ et $\{k+1, \ldots, n\} = flat_{s'}(\omega')$. On a donc $\overline{s'} \equiv_p \overline{s}$, $\overline{s'}_{|flat_{s'}(\omega')} \equiv_p \overline{s}_{|\omega}$, et $\overline{s'}[\overline{t}]_{flat_{s'}(\omega')} = \overline{u}[\overline{t}]_{\{k+1,\ldots,n\}}$, ce qui entraîne clairement $\overline{s'}[\overline{t}]_{flat_{s'}(\omega')} \equiv_p \overline{s}[\overline{t}]_\omega$.
- Si il existe $i \in \{1, \ldots, n\}$ tel que $\omega = i.\omega_i$. D'après l'hypothèse de récurrence, il existe $s'_i \in \mathcal{T}(\Sigma, \mathcal{X})$ et $\omega'_i \in \mathcal{D}om(s'_i)$, tels que $\overline{s_i} \equiv_p \overline{s'_i}$, $\overline{s_i}_{|\omega_i} = \overline{s'_i}_{|flat_{s'_i}(\omega'_i)}$, et $\overline{s_i}[\overline{t}]_{\omega_i} \equiv_p \overline{s'_i}[\overline{t}]_{flat_{s'_i}(\omega'_i)}$. Observons maintenant que l'égalité $\overline{s} = f\overline{s_1} \ldots \overline{s_n}$ implique $s_i(\varepsilon) \neq f$, et, en considérant la congruence $\overline{s_i} \equiv_p \overline{s'_i}$, on en déduit $s'_i(\varepsilon) \neq f$. On peut ainsi poser $\overline{s''} = f\overline{s_1} \ldots \overline{s_{i-1}}\overline{s'_i}\overline{s_{i+1}} \ldots \overline{s_n}$. Remarquons que $flat_{s'_i}(\omega'_i)$ est une position aplatie dans $\mathcal{D}om(\overline{s'_i})$, donc $\omega'' = i.flat_{s'_i}(\omega'_i)$ est aplatie dans $\mathcal{D}om(\overline{s''})$. D'après le lemme C.34, on peut alors affirmer qu'il existe $s' \in \mathcal{T}(\Sigma, \mathcal{X})$ et $\omega' \in \mathcal{D}om(s')$, tels que $\overline{s'} = \overline{s''}$ et $flat_{s'}(\omega') = \omega''$. De la congruence $\overline{s'_i} \equiv_p \overline{s_i}$ on déduit la congruence $\overline{s''} \equiv_p \overline{s}$, de l'égalité $\overline{s_i}_{|\omega_i} = \overline{s'_i}_{|flat_{s'_i}(\omega'_i)}$ on déduit l'égalité $\overline{s}_{|\omega} = \overline{s'}_{|flat_{s'}(\omega')}$, et de la congruence $\overline{s_i}[\overline{t}]_{\omega_i} \equiv_p \overline{s'_i}[\overline{t}]_{flat_{s'_i}(\omega'_i)}$ on déduit la congruence $\overline{s}[\overline{t}]_\omega \equiv_p \overline{s'}[\overline{t}]_{flat_{s'}(\omega')}$.

□

Index

$(\mathcal{R}, E)$, **36**
$(u_1, \ldots, u_n)$, **6**
$(\mathcal{T}(\Sigma, \mathcal{X}), \Sigma)$, **18**
., **5**
$<$, **68**, **87**
$<_e$, **68**
$<_n$, **68**
$=_E$, **23**
$=_{AC}$, **71**
$A/\sim$, **22**
$A(+, *)$, **144**
$\mathcal{V}$, **136**
$AC(+, *)$, **141**
A^*, **5**
$CSU_E(s, t)$, **33**
$Cons$, **98**
E-terminante, **36**
E-unifiables, **33**
E-unificateur, **33**
Eq, **98**
Id, **11**
$L_i(\mathcal{RE})$, **114**
$Level(\omega)$, **136**
$Mod(\Gamma)$, **20**
$Mod(\Sigma, \Gamma)$, **20**
NI, **57**
$NoSmallHead(s)$, **72**
$Q\{u/x\}$, **54**
$Subst^{\Sigma}$, **12**
$Th_{\mathcal{RE}}$, **56**
$Th_{\mathcal{RE}_2\sigma}$, **97**
$[t]$, **68**
$\Gamma \models P$, **21**
$\Gamma \models_i P$, **21**
$\Pi(\mathcal{RE})$, **112**
Σ-termes, **18**
$\Sigma-$termes finis, **6**
$\Sigma = \Sigma_f \cup \Sigma_r$, **12**
$\alpha\beta$, **11**
$\mathcal{A} \models \Gamma$, **20**
$\mathcal{A} \models P$, **20**
$\mathcal{M}\ (E)$, **27**
$\gtrless$, **68**
$\leq_e$, **68**
$\leqslant$, **26**
$\leqslant_E$, **33**
$\leqslant_e$, **68**
$\lneq_e$, **68**
$\sim$, **66**
$\sim^{\Gamma}$, **54**
$\sim_E$, **23**
$\trianglelefteq$, **66**
$\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, **11**
$f : s_1 \times \ldots \times s_n \to s$, **8**
$flat(s, f)$, **72**
$flat_s$, **137**
$l \to r$, **28**
$mgu(s, t)$, **32**
nil, **xi**
$s \downarrow$, **26**
$s \to_{\mathcal{R}} t$, **29**
$s \to_{\mathcal{R},E}\ t$, **38**
$s \to_{[l \to r, \omega, \sigma]} t$, **29**
$s \to t$, **29**
$s[t]_\omega$, **11**
$t_1 \downarrow_{\mathcal{R}^E} t_2$, **37**
$x\sigma$, **11**
$x : s$, **8**
$\mathcal{D}om(t)$, **67**
$\mathcal{T}(\Sigma, \mathcal{X})$, **12**
$\mathcal{T}(\Sigma, \mathcal{X})_s$, **9**
$[\![t]\!]^{\mathcal{A},\sigma}$, **19**
$\mathcal{T}(\Sigma)$, **9**
$\mathcal{PE}_{ind}(Q, <)$, **93**
$\mathcal{RE}_{ind}(Q)$, **88**
$\mathcal{RE}_{ind}(Q, <)$, **88**
$\mathcal{RE}_{ind}(Q, \prec, \tau)(X)$, **57**
$\mathcal{RE}_{ind}(Q, <)\sigma : j$, **114**
$(\mathcal{R}\ ,E)$, **36**
(quasi-) ordre bien-fondé, **27**

(quasi-) ordre noethérien, **27**
(quasi-)ordre de réduction, **64**
(quasi-)précédence, **66**
égalité conditionnelle, **16**

algèbre de Herbrand, **19**
algèbre des termes variadiques, **71**
algèbre initiale, **19**
algèbre multi-sortée, **17**
algèbre triviale, **17**
aplatissement, **71**
aplatissement de tête, **72**
arité, **6**
atomes, **12**
axiome, **7**
axiomes, **15**, **30**

Binôme, **148**

calcul des séquents LK, **14**
calcul des séquents modulo, **54**
canonique, **32**
Church-Rosser, **28**
Church-Rosser modulo, **37**
classique, **14**
clausale, **13**
clause équationnelle, **16**
clause de Horn, **13**
cohérence, **37**
cohérente modulo, **37**
compatible, **65**
complétude, **17**
complet, **32**
composition, **11**
concaténation, **5**
conclusion, **7, 8**, **16**
confluent modulo, **37**
confluente, **28**
confluente modulo, **37**
congruence, **21**
congruence de permutation, **71**
conséquence inductive, **21**
conséquence sémantique, **21**
consistance, **16**
contracté, **29**
convergent, **32**
convergente, **28**
conversion, **26**, **31**

décidabilité, **17**
définie-maximale, **35**, **140**
définie-maximales, **133**, **142–144**, **149**, **151–153**
dérivation, **7**
dérivation strictement décroissante, **100**
descente infinie, **42**, **58**
domaine, **10, 11**, **19**

ensemble S-sorté, **8**
ensemble complet de E-unificateurs, **33**
ensemble test, **xv**
extension cartésienne, **27**
extension lexicographique, **27**

fertilisation, **48**
filtre modulo, **37**
finitaire, **33**
forme normale, **26**
formules atomiques, **12**

greffe, **10**

Hauptsatz, **15**

idempotente, **12**
idempotentes, **12**
identité, **14**
image, **11**
instance, **11**
instance test, **xv**
instanciation close , **11**
interprétation, **17**, **19**
isomorphisme, **18**

jugements, **7**

liée, **13**
linéaire, **9**
littéral, **13**
localement cohérente modulo, **37**
localement confluente modulo, **37**
longueur, **5**

membre gauche, **29**
minimal, **33**
modèle de Herbrand, **21**
monoïde libre, **6**
multi-ensemble, **27**

négation, **13**

niveau, **136**
noethérien, **29**
nom, **7**

occurences, **10**
ordre, **26**
ordre de réduction, **64**
ordre de simplification, **65**
ordre lexicographique sur les chemins, **66**
ordre récursif sur les chemins, **66**
ordre strict, **26**
ouverte, **13**
ouverts, **9**

par induction, **x**
permutation, **12**
position aplatie, **138**
positions, **10**
précédence, **66**
précondition, **16**
préfixe, **27**
prémisses, **7**
principe de récurrence bien fondé, **xiii**
principe de récurrence noethérien, **xiii**
principe de récurrence simple, **x**
profil, **8**
profondeur, **10**
propriété d'effacement, **71**

quasi-ordre, **26**
quasi-ordre de réduction, **64**

réécriture syntaxique, **36**
récurrence implicite, **50**
récurrence par réécriture, **50**
récurrence sur les termes, **6**
réductibilité inductive, **35**
réductible, **26**, **29**
règle, **28**
règle de réécriture conditionnelle, **29**
règles logiques, **14**
règles structurelles, **14**
raisonnement équationnel, **31**
rang, **11**
redex, **29**
relation d'équivalence, **26**
remplacement, **10**
restriction, **11**

séquents, **14**
satisfait, **20**
schéma de récurrence, **42**
schémas de récurrence, **xv**
signature fonctionnelle, **12**
signature relationnelle, **12**
solution, **32**
sorte, **8**
spécification, **30**
spécification équationnelle conditionnelle, **30**
stabilité par contextes, **64**
stabilité par substitutions, **64**
structuré, **39**
structure libre, **41**
substitution, **11**, **19**
substitution irréductible, **26**
substitution normalisée, **26**
support, **19**
surréductible, **76**
symbole de constante, **6**
symbole de tête, **10**
symbole fonctionnel, **6**
système d'inférence, **7**
système de réécriture non conditionnel, **28**

terme aplati, **71**
termes clos, **9**
terminante modulo, **36**
théorie, **15**
théorie engendrée, **21**

unifiables, **32**
unificateur, **32**
unificateur plus général, **32**
unificateurs constructeurs, **79**
unification finie, **32**

valide, **32**
variable de récurrence, **42**
variables, 9
variables de récurrence, **xv**
variables libres, **13**
vecteur, **13**

Bibliographie

[AOT 06] Takahito Aoto. "Dealing with Non-orientable Equations in Rewriting Induction". In F. Pfenning, éditeur, *Proceedings of the 17th International Conference on Rewriting Techniques and Applications*, volume 4098, pages 242–256, Nara (Japan), apr 2006. LNCS.

[BAC 88] L. Bachmair. "Proof by consistency in equational theories". Technical report, SUNY at Stony Brook USA, June 1988.

[BAR 03] Gilles Barthe, Horatiu Cirstea, Claude Kirchner, and Luigi Liquori. "Pure Patterns Type Systems". In *Principles of Programming Languages - POPL2003, New Orleans, USA*. ACM, January 2003.

[BER 95] N. Berregeb, A. Bouhoula, and M. Rusinowitch. "Extending SPIKE to associative and commutative theories". In *Seminar on Automation of proof by induction*, Dagstuhl seminar, Germany, July 1995.

[BER 96a] Narjes Berregeb, Adel Bouhoula, and Michaël Rusinowitch. "Automated verification by induction with associative-commutative operators.". In Rajeev Alur, and Thomas A. Henzinger, éditeurs, *CAV*, volume 1102 de *Lecture Notes in Computer Science*, pages 220–231. Springer, 1996.

[BER 96b] Narjes Berregeb, Adel Bouhoula, and Michaël Rusinowitch. "Spike-ac : A system for proofs by induction in associative-commutative theories". In Harald Ganzinger, éditeur, *RTA*, volume 1103 de *Lecture Notes in Computer Science*, pages 428–431. Springer, 1996.

[BER 97] Narjès Berregeb. "*Preuves par induction implicite : cas des théories associatives-commutatives et observationnelles*". Thèse de Doctorat d'Université, Université Henri Poincaré – Nancy 1, June 1997.

[BOU 92a] Adel Bouhoula, E. Kounalis, and M. Rusinowitch. "Automated mathematical induction". Technical report 1636, INRIA, 1992.

[BOU 92b] Adel Bouhoula, E. Kounalis, and M. Rusinowitch. "Spike : An automatic theorem prover". In *Proceedings of the 1st International Conference on Logic Programming and Automated Reasoning, St. Petersburg (Russia)*, volume 624 de *Lecture Notes in Artificial Intelligence*, pages 460–462. Springer-Verlag, July 1992.

[BOU 95] Adel Bouhoula, and Michaël Rusinowitch. Implicit induction in conditional theories. *Journal of Automated Reasoning*, 14(2) :189–235, 1995.

[BOU 97] A. Bouhoula. Automated theorem proving by test set induction. *Journal of Symbolic Computation*, 23 :47–77, May 1997.

[BRO 06] James Brotherston. "*Sequent Calculus Proof Systems for Inductive Definitions*". PhD thesis, University of Edinburgh, November 2006.

[BUN 93] A. Bundy, F. van Harmelen, C. Horn, and A. Smaill. Rippling : A heuristic for guiding inductive proofs. *Artificial Intelligence*, 62 :183–253, 1993.

[BUN 99] Alan Bundy. "The automation of proof by mathematical induction". In Alan Robinson, and Andrei Voronkov, éditeurs, *Handbook of automated reasonning*. Elsevier Science Publishers B. V. (North-Holland), 1999.

[BUN 05] Alan Bundy, David Basin, Dieter Hutter, and Andrew Ireland. "*Rippling : Meta-Level Guidance for Mathematical Reasoning*". Cambridge University Press, 2005.

[COM 86] H. Comon. "Sufficient completeness, term rewriting system and anti-unification". In J. Siekmann, éditeur, *Proceedings 8th International Conference on Automated Deduction, Oxford (UK)*, volume 230 de *Lecture Notes in Computer Science*, pages 128–140. Springer-Verlag, 1986.

[COM 97] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. Tree automata techniques and applications. Available on : `http://www.grappa.univ-lille3.fr/tata`, 1997. Released October 1st, 2002.

[COM 00] Hubert Comon, and Robert Nieuwenhuis. Induction=i-axiomatization+first-order consistency. *Inf. Comput.*, 159(1-2) :151–186, 2000.

[COM 01] Hubert Comon. "Inductionless induction". In A. Robinson, and A. Voronkov, éditeurs, *Handbook of Automated Reasoning*, volume I, chapter 14, pages 914–959. Elsevier Science, 2001.

[DEP 02] Eric Deplagne. "*Système de preuve modulo récurrence*". Thèse de doctorat, Université Nancy 1, November 2002.

[DEP 03] Eric Deplagne, Claude Kirchner, Hélène Kirchner, and Quang-Huy Nguyen. "Proof search and proof check for equational and inductive theorems". In Franz Baader, éditeur, *Proceedings of CADE-19*, Miami, Florida, July 2003. Springer-Verlag.

[DEP 04] Eric Deplagne, and Claude Kirchner. "Induction as deduction modulo". Rapport de recherche, LORIA, Nov 2004.

[DER 79a] N. Dershowitz. A note on simplification orderings. *Information Processing Letters*, 9 :212–215, 1979.

[DER 79b] Nachum Dershowitz, and Zohar Manna. Proving termination with multiset orderings. *Communications of the ACM*, 22(8) :465–476, 1979.

[DER 82] N. Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17 :279–301, 1982.

[DER 83] N. Dershowitz. "Well-founded orderings". In *Information Science Research Office, ATR-83-8478-3*, The Aerospace Corporation, El Segundo, California USA, 1983.

[DER 01] Nachum Dershowitz, and David A. Plaisted. "Rewriting". In A. Robinson, and A. Voronkov, éditeurs, *Handbook of Automated Reasoning*, volume I, chapter 9, pages 535–610. Elsevier Science, 2001.

[DOW 98] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. "HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic". Rapport de Recherche 3556, Institut National de Recherche en Informatique et en Automatique, November 1998. `http://pauillac.inria.fr/~dowek/RR-3556.ps.gz`.

[DOW 99] G. Dowek. "*La part du Calcul*". Université de Paris 7, 1999. Mémoire d'habilitation.

[DOW 01] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science*, 11(1) :21–45, 2001.

[DOW 03a] Gilles Dowek. "Confluence as a cut elimination property.". In Robert Nieuwenhuis, éditeur, *RTA*, volume 2706 de *Lecture Notes in Computer Science*, pages 2–13. Springer, 2003.

[DOW 03b] Gilles Dowek, Thérèse Hardin, and Claude Kirchner. Theorem proving modulo. *Journal of Automated Reasoning*, 31(1) :33–72, Nov 2003.

[ECH 05] Mnacho Echenim. "*Déduction et unification dans les théories permutatives*". Thèse de Doctorat d'Université, Institut National Polytechnique de Grenoble, dec 2005.

[EHR 85] H. Ehrig, and B. Mahr. "*Fundamentals of Algebraic Specification 1. Equations and initial semantics*", volume 6 de *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1985.

[FAY 79] M. Fay. "First order unification in equational theories". In *Proceedings 4th Workshop on Automated Deduction, Austin (Tex., USA)*, pages 161–167, 1979.

[FER 95] M.C.F. Ferreira. "*Termination of Rewriting*". PhD thesis, Universiteit Utretch, Utretch, 1995.

[FRI 86] L. Fribourg. "A strong restriction of the inductive completion procedure". In *Proceedings 13th International Colloquium on Automata, Languages and Programming*, volume 226 de *Lecture Notes in Computer Science*, pages 105–115. Springer-Verlag, 1986.

[GAN 92] Harald Ganzinger, and Jurgen Stuber. "Inductive theorem proving by consistency for first-order clauses". In *Conditional Term Rewriting Systems*, pages 226–241, 1992.

[GIR 89] J.-Y. Girard, Y. Lafont, and P. Taylor. "*Proofs and Types*", volume 7 de *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1989.

[GOG 80] J. A. Goguen. "How to prove algebraic inductive hypotheses without induction, with applications to the correctness of data type implementation". In W. Bibel, and R. Kowalski, éditeurs, *Proceedings 5th International Conference on Automated Deduction, Les Arcs (France)*, volume 87 de *Lecture Notes in Computer Science*, pages 356–373. Springer-Verlag, 1980.

[GUT 75] J. Guttag. "*The specification and Application to Programming of Abstract Data Types*". Thèse de Doctorat d'Université, University of Toronto, 1975.

[GUT 78a] John V Guttag. "Notes on type abstraction". In Friedrich L.Bauer, and Manfred Broy, éditeurs, *Program Construction*, volume 69 de *Lecture Notes in Computer Science*, pages 593–616. Springer-Verlag, 1978.

[GUT 78b] John V. Guttag, and James J. Horning. The algebraic specification of abstract data types. *Acta Informatica*, 10 :27–52, 1978.

[HUE 72] G. Huet. "*Constrained Resolution : A Complete Method for Higher Order Logic*". PhD thesis, Case Western Reserve University, 1972.

[HUE 76] G. Huet. "*Résolution d'equations dans les langages d'ordre 1,2, ...,ω*". Thèse de Doctorat d'Etat, Université de Paris 7 (France), 1976.

[HUE 80] Gérard Huet. Confluent reductions : Abstract properties and applications to term rewriting systems. *Journal of the ACM*, 27(4) :797–821, October 1980. Preliminary version in 18th Symposium on Foundations of Computer Science, IEEE, 1977.

[HUE 82] G. Huet, and J.-M. Hullot. Proofs by induction in equational theories with constructors. *Journal of Computer and System Sciences*, 25(2) :239–266, October 1982. Preliminary version in Proceedings 21st Symposium on Foundations of Computer Science, IEEE, 1980.

[HUL 80a] J.-M. Hullot. "Canonical forms and unification". In *Proceedings 5th International Conference on Automated Deduction, Les Arcs (France)*, pages 318–334, July 1980.

[HUL 80b] Jean-Marie Hullot. "*Compilation de Formes Canoniques dans les Théories équationelles*". Thèse de Doctorat de Troisième Cycle, Université de Paris Sud, Orsay (France), 1980.

[JOU 82] Jean-Pierre Jouannaud, Pierre Lescanne, and Fernand Reinig. "Recursive decomposition ordering". In D. Bjørner, éditeur, *Formal Description of Programming Concepts 2*, pages 331–348, Garmisch-Partenkirchen, Germany, 1982. Elsevier Science Publishers B. V. (North-Holland).

[JOU 83] J.-P. Jouannaud. "Confluent and coherent equational term rewriting systems. Applications to proofs in abstract data types". In G. Ausiello, and M. Protasi, éditeurs, *Proceedings of the 8th Colloquium on Trees in Algebra and Programming, L'Aquila (Italy)*, volume 159 de *Lecture Notes in Computer Science*, pages 269–283. Springer-Verlag, 1983.

[JOU 86a] J.-P. Jouannaud, and E. Kounalis. "Automatic proofs by induction in theories without constructors". Technical report 295, Université de Paris-Sud, Centre d'Orsay (France), September 1986.

[JOU 86b] Jean-Pierre Jouannaud, and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15(4) :1155–1194, 1986.

[KAM 82] S. Kamin, and J.-J. Lévy. Attempts for generalizing the recursive path ordering. *Inria, Rocquencourt*, 1982.

[KAP 84a] S. Kaplan. Conditional rewrite rules. *Theoretical Computer Science*, 33 :175–193, 1984.

[KAP 84b] S. Kaplan. "Fair conditional term rewriting systems : Unification, termination, and confluence". Technical report, University of Paris Sud, Orsay (France), Orsay, 1984.

[KAP 85] D. Kapur, P. Narendran, and G. Sivakumar. "A path ordering for proving termination of term rewriting systems". In H. Ehrig, C. Floyd, M. Nivat, and J. Thatcher, éditeurs, *Proceedings of the 6th Conference on Automata, Algebra and Programming*, volume 185. Springer-Verlag, 1985.

[KAP 87a] S. Kaplan. Simplifying conditional term rewriting systems : Unification, termination and confluence. *Journal of Symbolic Computation*, 4(3) :295–334, December 1987.

[KAP 87b] D. Kapur, P. Narendran, and H. Zhang. On sufficient completeness and related properties of term rewriting systems. *Acta Informatica*, 24 :395–415, 1987.

[KAP 90] D. Kapur, G. Sivakumar, and H. Zhang. "A new method for proving termination of AC-rewrite systems". In *Proceedings 10th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 472 de *Lecture Notes in Computer Science*, pages 133–148. Springer-Verlag, 1990.

[KAP 91] D. Kapur, P. Narendran, D. J. Rosenkrantz, and H. Zhang. Sufficient completeness, ground-reducibility and their complexity. *Acta Informatica*, 28 :311–350, 1991.

[KAP 95] Deepak Kapur, and Hantao Zhang. An overview of rewrite rule laboratory (RRL). *J. Computer and Mathematics with Applications*, 29(2) :91–114, 1995.

[KIR] C. Kirchner, and H. Kirchner. Rewriting, solving, proving. To appear.

[KIR 90] Claude Kirchner, Hélène Kirchner, and M. Rusinowitch. Deduction with symbolic constraints. *Revue d'Intelligence Artificielle*, 4(3) :9–52, 1990. Special issue on Automatic Deduction.

[KIR 99] Claude Kirchner, and Hélène Kirchner. Rewriting, solving, proving. A preliminary version of a book available at `www.loria.fr/~ckirchne/rsp.ps.gz`, 1999.

[KNU 70] Donald E. Knuth, and P. B. Bendix. "Simple word problems in universal algebras". In J. Leech, éditeur, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[KOI 00] H Koike, and Y. Toyama. Inductionless induction and rewriting induction. *Computer Software*, 17(6) :1–12, 2000.

[KOU 85] E. Kounalis. "Completeness in data type specifications". In B. Buchberger, éditeur, *Proceedings EUROCAL Conference, Linz (Austria)*, volume 204 de *Lecture Notes in Computer Science*, pages 348–362. Springer-Verlag, 1985.

[KOU 90] E. Kounalis, and M. Rusinowitch. "Mechanizing inductive reasoning". In *Proceedings of the American Association for Artificial Intelligence Conference, Boston*, pages 240–245. AAAI Press and MIT Press, July 1990.

[LAL 90] René Lalement. "*Logique, réduction, résolution*". Masson, 1990.

[LAN 79] D.S. Lankford. "A unification algorithm for abelian group theory". Technical report, Memo MTP-1, Mathematics Depart., Louisiana Tech. Univ., Ruston, LA, 1979.

[LAZ 90] A. Lazrek, P. Lescanne, and J.-J. Thiel. Tools for proving inductive equalities, relative completeness and ω-completeness. *Information and Computation*, 84(1) :47–70, January 1990.

[MAR 93] C. Marché. "*Réécriture modulo une théorie présentée par un système convergent et décidabilité du problème du mot dans certaines classes de théories équationnelles*". Thèse de Doctorat d'Université, Université de Paris-Sud, Orsay (France), October 1993.

[MUS 80a] D. R. Musser. "On proving inductive properties of abstract data types". In *Proceedings 7th ACM Symp. on Principles of Programming Languages*, pages 154–162. ACM, 1980.

[MUS 80b] David Musser. "On proving inductive properties of abstract data types". In *Proceedings, Symposium on Principles of Programming Languages*, volume 7. Association for Computing Machinery, 1980.

[NEW 42] M. H. A. Newman. “On theories with a combinatorial definition of equivalence”. In *Annals of Math*, volume 43, pages 223–243, 1942.

[PET 81] G. Peterson, and M. E. Stickel. Complete sets of reductions for some equational theories. *Journal of the ACM*, 28 :233–264, 1981.

[PLA 78] D. Plaisted. A recursively defined ordering for proving termination of term rewriting sytems. *Dept. of Computer Science Report 78-943*, 1978.

[PLO 72] Gordon Plotkin. Building-in equational theories. *Machine Intelligence*, 7 :73–90, 1972.

[RED 90] Udday Reddy. “Term rewriting induction”. In M. E. Stickel, éditeur, *Proceedings 10th International Conference on Automated Deduction, Kaiserslautern (Germany)*, volume 449 de *Lecture Notes in Computer Science*, pages 162–177. Springer-Verlag, 1990.

[RUB 95] Albert Rubio, and Robert Nieuwenhuis. A total AC-compatible ordering based on RPO. *Theoretical Computer Science*, 142(2) :209–227, 1995.

[RUB 02] Alberto Rubio. A fully syntactic ac-rpo. *Information and Computation*, 178(2) :515–533, November 2002.

[SEL 72] A. Selman. Completeness of calculii for axiomatically defined classes of algebras. *Algebra Universalis*, 2 :20–32, 1972.

[SLA 74] J. R. Slagle. Automated theorem-proving for theories with simplifiers, commutativity and associativity. *Journal of the ACM*, 21(4) :622–642, 1974.

[STE] Graham Steel. Proof by consistency - a literature survey.

[STE 89] J. Steinbach. “Extensions and comparisons of simplification orderings”. In N. Dershowitz, éditeur, *Proceedings 3rd Conference on Rewriting Techniques and Applications, Chapel Hill (N.C., USA)*, volume 355 de *Lecture Notes in Computer Science*, pages 434–448. Springer-Verlag, April 1989.

[WAC 05] Benjamin Wack. “*Typage et déduction dans le calcul de réécriture*”. Thèse de doctorat, Université Henri Poincaré - Nancy I, October, 7- 2005.

[WEC 92] Wolfgang Wechler. “*Universal Algebra for Computer Scientists*”, volume 25 de *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1992.

[WIR 04] Claus-Peter Wirth. Descente infinie+deduction. *Logic journal of the IGPL*, 12(1) :1–96, 2004.