THÈSE

présentée à l'École Normale Supérieure de Cachan

pour obtenir le grade de docteur

par Sergiu Bursuc

Spécialité: Informatique

Contraintes de déductibilité dans une algèbre quotient: réduction de modèles et applications à la sécurité

Composition du Jury :

• Hubert Comon-Lundh	directeur de thèse
• Stéphanie Delaune	directrice de thèse
• Claude Kirchner	examinateur
• Yassine Lakhnech	président du jury
• Michäel Rusinowitch	rapporteur
• Ralf Treinen	rapporteur

Thèse préparée au sein du Laboratoire Spécification et Vérification (LSV)

Table des matières

Chapitre 1

Remerciements

"On n'enseigne pas ce que l'on sait ou ce que l'on croit savoir: on n'enseigne et on ne peut enseigner que ce que l'on est" (Jean Jaures). Je remercie mon directeur de thèse et ma directrice de thèse, le LSV, ses membres et ses amis, les membres de mon jury, la nouvelle vague, faculty of computer science Iasi et Raymond Aron.

Chapitre 2

Introduction

2.1 Opportunités et défis de la numérisation

Le monde devient numérique [Ber07b]: de plus en plus d'activités - économiques, culturelles ou sociales - se situent desormais avec prépondérance dans un espace informatique, dont la manifestation la plus prenante est l'internet. Ainsi, la communication, le commerce, les services bancaires, les services audio-visuels, les élections, la cartographie, les transports et maintes autres activités ont tendance à devenir électroniques.

En même temps qu'une évidente amélioration des services concernés et de leur accessibilité, les possibilités ouvertes par la numérisation posent aussi beaucoup de problèmes subreptices. Comment s'assurer que la communication a lieu avec la personne voulue ? Comment payer le bon montant, pour le bon bien, au bon commerçant ? Comment établir une relation de confiance entre une banque et ses clients sur internet ? Comment protéger les droits d'auteur ? Et l'anonymat ? Peut-on vérifier que le résultat d'une élection correspond bien à la somme des intentions des individus ? Comment éviter les mauvais endroits et les accidents sur les routes ? Pourquoi la ligne 14 du métro parisien, sans conducteur, est-elle sûre ? Pourquoi un utilisateur malhonnête ne peut pas voler tous les vélos dans le réseau Velib de Paris ?

Ce sont des questions de vérification [Ber07a], où on essaye de rendre visibles les problèmes potentiels lors du passage à l'internet ou, plus généralement, à un reseau informatisé. Notons que ces propriétés sont non seulement assurées malgré le numérique, mais maintes fois elles sont, ou seront, possibles seulement grâce à lui. Par exemple, la verifiabilité du résultat des élections ou la fiabilité du réseau Velib, voire son existence, sont pratiquement impossibles sans l'aide d'un système informatique.

De manière générale, la vérification s'intéresse à déterminer si un système satisfait les propriétés qu'il est censé satisfaire. Étant donnée l'importance critique des questions posées, la réponse doit être basée sur des arguments rigoureux, mathématiques. Pour cela, les éléments qui sont en jeu (les agents, les

messages, les opérations, le réseau, le protocole qui régit l'application), ainsi que les propriétés désirées, doivent être représentés dans un modèle mathématique, dit aussi formel. Les questions de vérification se traduisent alors dans des questions mathématiques et leurs réponses (positives ou négatives) nous permettent un retour vers l'application, en nous donnant une certaine garantie de bon fonctionnement ou en pointant un problème de conception.

2.2 Vérification des protocoles de sécurité

Dans ce contexte, nous nous intéressons à la vérification des protocoles de sécurité. Ce sont des programmes dont le but est d'établir une communication sûre entre plusieurs agents. Les protocoles de sécurité sont parfois appelés cryptographiques, car dans la plupart des cas ils utilisent la cryptographie: la technique et la science de protéger des messages, souvent à l'aide des clefs, assurant leur confidentialité, authenticité et intégrité.

Quasiment toutes les applications citées dans la section précédente ont besoin des protocoles de sécurité pour assurer leur bon fonctionnement. Cette importance des protocoles de sécurité dans des applications critiques rend d'autant plus importante la question de leur correction. Elle repose, d'une part, sur la correction des primitives cryptographiques utilisées et, d'une autre, sur la correction des programmes qui régissent le protocole. La première fait l'objet de la cryptologie, discipline à part: nous allons abstraire les primitives cryptographiques, en parlant seulement de leur propriétés, et non pas des détails de leur implémentation.

Dans le cas d'une cryptographie correcte, une exécution normale du protocole devrait satisfaire par conception les propriétés de sécurité, mais une attaque est possible à cause de la logique réactive du protocole et de la non-sécurité du réseau: un intrus peut intercepter les messages, les modifier et faire exécuter le protocole d'une manière imprévue [CJ97, CLS02].

Exemple 1 *Voyons comment un intrus peut exploiter les failles logiques d'un protocole, pour l'attaquer même quand il utilise une cryptographie parfaite. Supposons les opérations de:*

- chiffrement symétrique $\mathtt{enc}(m, k)$ - *le chiffrement du message m avec la clef k. Seul quelqu'un qui connaît k peut récuperer m à partir de $\mathtt{enc}(m, k)$.*
- chiffrement public $\mathtt{enc}(m, \mathtt{pub}(a))$ - *le chiffrement du message m avec la clef publique de a, $\mathtt{pub}(a)$. Seul quelqu'un qui connaît $\mathtt{priv}(a)$, la clef privée de a, peut récuperer m à partir de $\mathtt{enc}(m, \mathtt{pub}(a))$. Tout le monde connaît $\mathtt{pub}(a)$ et peut faire le chiffrement.*

Imaginons que B *est une banque, qui détient des secrets concernant ses clients. À un certain moment, l'un de ses clients,* A*, veut obtenir de* B *une information secrète qui le concerne,* $\mathsf{secret}(\mathsf{A}, \mathsf{B})$*. Le but d'un protocole de sécurité dans ce contexte serait de communiquer* $\mathsf{secret}(\mathsf{A}, \mathsf{B})$ *à* A *de manière à ce qu'il soit le seul à l'apprendre. Nous pouvons imaginer le protocole suivant:*

- *Supposons que* B *a une clef publique,* $\mathit{pub}(\mathsf{B})$*. Le chiffrement public étant coûteux, on ne peut pas fournir une clef publique à chaque client. Par contre, ils peuvent génerer des nouvelles clefs symétriques.*

- A *va demander à* B *de lui envoyer le secret chiffré avec une nouvelle clef symétrique* k*. Pour cela, il va générer* k *et va envoyer à la banque* $\mathit{enc}(\langle \mathsf{A}, \mathsf{k} \rangle, \mathit{pub}(\mathsf{B}))$ *- ce message à la fois identifie* A *et communique à* B *la clef* k *qui va être utilisée pour communiquer avec* A*.*

- *La banque* B *attend des demandes. Quand elle reçoit un message de la forme* $\mathit{enc}(\langle \mathsf{x}, \mathsf{y} \rangle, \mathit{pub}(\mathsf{B}))$*, la logique du protocole lui dit que c'est le client avec le nom* x *qui veut qu'on lui communique son secret chiffré avec la clef* y*.* B *va donc répondre en envoyant* $\mathit{enc}(\mathsf{secret}(\mathsf{x}, \mathsf{B}), \mathsf{y})$ *à* x*.*

À priori, aucun autre agent, à part A *et* B*, ne peut connaître* k*: le message* $\mathit{enc}(\mathsf{secret}(\mathsf{A}, \mathsf{B}), \mathsf{k})$ *renvoyé par* B *en réponse à* $\mathit{enc}(\langle \mathsf{A}, \mathsf{k} \rangle, \mathit{pub}(\mathsf{B}))$ *ne peut pas être dechiffré par un intrus pour apprendre le secret.*

Pourtant, il y a un moyen pour l'intrus d'apprendre $\mathsf{secret}(\mathsf{A}, \mathsf{B})$ *en exploitant une faiblesse dans la logique du protocole. En effet, il suffit qu'il envoye* $\mathit{enc}(\langle \mathsf{A}, \mathsf{k}' \rangle, \mathit{pub}(\mathsf{B}))$ *à* B*, où* k' *est une clef qu'il connaît et on suppose qu'il connaît aussi le nom de l'agent qu'il veut attaquer. Suivant la spécification,* B *va répondre avec* $\mathit{enc}(\mathsf{secret}(\mathsf{A}, \mathsf{B}), \mathsf{k}')$ *et donc divulguer le secret à l'intrus, sans qu'il ait à casser le chiffrement.*

Nous avons donc besoin d'un modèle formel pour exprimer précisément le protocole, les capacités de l'intrus et les propriétés qu'on veut prouver. Les méthodes formelles (e.g. basées sur la logique, les techniques de réécriture, etc) peuvent être ensuite utilisées pour prouver que la modèlisation du protocole satisfait les propriétés de sécurité. De plus, à cause de la multitude de protocoles existants et de l'apparition permanente de nouveaux protocoles, l'automatisation de la vérification formelle est un objectif pratique important, en plus d'être une question intéressante en soi. Le but est de concevoir des méthodes génériques s'appliquant à une classe de protocoles, en s'appuyant sur des techniques de démonstration automatique.

2.3 État de l'art et contributions de la thèse

Les efforts de vérification de protocoles de sécurité commencent avec les travaux de D. Dolev, A. Yao et autres [DY83, DEK82]. Le modèle formel consideré comporte une représentation des messages, des opérations sur les messages et des règles du protocole. De plus, les possibilités de l'intrus sont prises en compte: il peut espionner le réseau, modifier les messages en utilisant les opérations disponibles et contrôler leur transmission. Le problème du secret d'une donnée est alors formalisé comme une question d'atteignabilité d'un état dans un système de transitions, determiné par les règles du protocole et les capacités de l'intrus. Il est montré décidable en temps polynomial.

Même si ce modèle de départ prend en compte des aspects importants des protocoles de sécurité, il est pourtant assez restricif: les clefs de chiffrement sont atomiques et la classe de protocoles est particulière (dite "ping-pong"). Depuis, en dépit du fait que le problème est indécidable en toute généralité [MDL+99], maints travaux ont proposé des algorithmes pour décider la sécurité dans des modèles plus réalistes: e.g. [Hui99, AL00, Bor01, RT01, MS01]. Une restriction typiquement faite pour obtenir la décidabilité est le fait de ne considérer qu'un nombre borné de sessions. Même pour un nombre non-borné de sessions on obtient des résultats de décidabilité dans certains cas particuliers [CLC03, CKR+03].

Tous ces travaux utilisent l'algèbre libre de termes pour modèliser les messages du protocole. Un système de déduction permet ensuite de spécifier les actions autorisées sur les messages. Mais cette approche est basée sur l'hypothèse du chiffrement parfait: e.g. on peut obtenir des informations sur un message chiffré seulement si on possède la clef de déchiffrement. En fait, il s'avère que cette hypothèse est trop forte. Beaucoup de protocoles utilisent des primitives cryptographiques qui ont des propriétés algébriques: associativité, commutativité, inverse, nilpotence... Ces propriétés s'ajoutent au pouvoir de l'intrus et lui permettent d'obtenir plus d'informations: des attaques impossibles dans le modèle de l'algèbre libre ont été trouvées dans ce modèle étendu [Sim94, BGW01]. Pour traiter ces cas, on considère une théorie équationelle exprimant les propriétés algébriques des primitives cryptographiques et on augmente le système de déduction de l'intrus avec une règle d'inférence équationelle [Del06, Laf06].

Depuis que la nécessité de prendre en compte les propriétés algébriques a été reconnue, beaucoup de résultats ont été obtenus pour des théories particulières: commutativité du chiffrement [CKRT05], "ou" exclusif [CLS03, CKRT03b], groupe abélien [MS05], théories avec un symbole d'homomorphisme [DLLT06], substitution des clefs [CK07], ...

Par contre, il existe peu d'approches traitant d'une classe générale de théories: [DLLT08, Bau07, CLT03, BC08]. D'autres résultats qui vont dans le sens d'une extension des classes décidables sont les résultats de combinaison: disjointe [CR05] ou hiérarchique [CR06].

Dans cette thèse, motivés à la fois par des besoins pratiques et des intérêts théoriques, nous suivons cette piste et nous étudions les techniques générales pour la vérification des propriétés de sécurité de protocoles modulo une théorie équationnelle. Nous allons d'abord rappeler, dans la partie I, que, dans le cadre d'un nombre borné de sessions, qui nous intéresse ici, ces propriétés peuvent se formuler à l'aide des systèmes de contraintes, suivant e.g. [Del06, Laf06, Bau07, CLCZ10, MS01, RT01].

Ensuite (partie II), suivant une étude de cas de porte-monnaie électronique, dont la théorie équationelle est une combinaison de théories, mais où les techniques de combinaison existantes ne s'appliquent pourtant pas, nous étudions le problème de décomposition d'une théorie. Nous introduisons un schéma de combinaison qui généralise les combinaisons disjointes [CR05] et hiérarchiques [CR06] et nous permet de dériver un algorithme de décision pour la théorie du

porte-monnaie électronique.

Parfois la théorie n'est pas décomposable, même dans le cadre des combinaisons non-hiéarchiques. C'est le cas, par exemple, pour la théorie qui modélise les signatures en aveugle. Ceci nous amène, dans la partie III, à considerer une autre approche pour la résolution de systèmes de contraintes, qui ne simplifie plus la théorie, mais les systèmes de contraintes. Pour cela, nous montrons qu'il est suffisant que la théorie soit locale, i.e. saturée par résolution ordonnée ([BG01]), en utilisant une notion de redondance bien-choisie. Nous obtenons ainsi un premier résultat général qui relie directement la notion de localité à la décidabilité des systèmes de contraintes. De plus, comme dans [CLCZ10] pour une théorie particulière, nous avons une représentation symbolique de toutes les solutions, préservant ainsi les propriétés de toute trace du protocole. Ce résultat nous permet ensuite de dériver un algorithme pour la théorie des signatures en aveugle, qui n'est ni sous-terme convergente [Bau07], ni monoidale [DLLT06]. Ainsi on généralise un résultat de [BC08], car nous avons une représentation de toutes les solutions.

Partie I

Modélisation

Chapitre 3

Termes, unification, réécriture

Une signature $\mathcal{F}$ est un ensemble de symboles de fonction. Par $\mathcal{X}$ on va noter un ensemble (infini) de variables. L'ensemble des termes construits en utilisant les symboles de $\mathcal{F}$ et $\mathcal{X}$ est noté par $\mathcal{T}(\mathcal{F}, \mathcal{X})$ et l'ensemble des termes clos par $\mathcal{T}(\mathcal{F})$. On va parfois appeler contextes les termes avec des variables. Pour un terme t, on va noter par $top(t)$ le symbole de sa racine. Les termes avec des symboles associatifs et commutatifs (AC) sont aplatis: si f est AC et $top(t) = f$, alors $t = f(t_1, \ldots, t_n)$ et, pour tout $1 \leq i \leq n$, $top(t_i) \neq f$. Pour tout terme t, $\mathsf{Var}(t)$ est l'ensemble des variables qui apparaissent dans t.

Definition 1 (sous-terme syntaxique, taille d'un terme) *Soit $t \in \mathcal{T}(\mathcal{F}, \mathcal{X})$. L'ensemble $\mathsf{St}_s(t)$ des* sous-termes syntaxiques *de t est le plus petit ensemble tel que*

- $t \in \mathsf{St}_s(t)$
- *si $f(t_1, \ldots, t_n) \in \mathsf{St}_s(t)$, alors $t_1, \ldots, t_n \in \mathsf{St}_s(t)$.*

$\mathsf{St}_s(t) \setminus \{t\}$ est l'ensemble des sous-termes (syntaxiques) stricts de t.

La taille $|t|$ de t est définie par $|t| = 1 + |t_1| + \ldots + |t_n|$, si $t = f(t_1, \ldots, t_n)$.

Une *théorie équationelle* $\mathcal{E}$ est donnée par un ensemble fini d'équations. On va noter par $=_\mathcal{E}$ la plus petite relation de congruence sur $\mathcal{T}(\mathcal{F}, \mathcal{X})$ telle que $u\sigma =_\mathcal{E} v\sigma$, pour toute paire $u = v \in \mathcal{E}$ et toute substitution σ.

Le domaine d'une substitution σ, noté par $\mathsf{dom}(\sigma)$, est défini par $\{x \mid x \in \mathcal{X} \text{ et } x\sigma \neq x\}$.

Definition 2 ($\mathcal{E}$-unification) *Deux termes s et t sont dits $\mathcal{E}$-unifiables s'il existe une substitution σ telle que $s\sigma =_\mathcal{E} t\sigma$. Une telle substitution est appelée $\mathcal{E}$-unificateur de s et t.*

Une substitution σ est plus générale qu'une substitution θ s'il existe une substitution τ telle que $x\theta = x\sigma\tau$, pour tout $x \in \mathsf{dom}(\theta)$. S'il existe un unificateur le plus général de s et t, il va être noté par $\mathsf{mgu}(s, t)$.

Un système de réécriture est un ensemble fini de règles de réécriture $l \to r$ avec $l \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ et $r \in \mathcal{T}(\mathcal{F}, \mathsf{Var}(l))$. Un terme $s \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ se réécrit (resp. modulo une théorie équationelle $\mathcal{E}$) en t par un système de réécriture $\mathcal{R}$, noté $s \to_{\mathcal{R}} t$, s'il existe une règle $l \to r \in \mathcal{R}$, un contexte C et une substitution σ tels que $s = C[l\sigma]$ (resp. $s =_{\mathcal{E}} C[l\sigma]$) et $t = C[r\sigma]$ (resp. $t =_{\mathcal{E}} C[r\sigma]$). $l\sigma$ est alors appelé un radical de s. Un terme est *en forme normale* (par rapport à $\mathcal{R}$), s'il ne se réécrit pas par $\mathcal{R}$.

Definition 3 (stratégie de réécriture innermost) *Un radical est dit profond si tous ses sous-termes stricts sont en forme normale. Une séquence de réduction suit la* stratégie innermost *si lors de chacune des étapes de réécriture le radical remplacé est profond.*

Dans la suite, toutes nos séquences de réécriture sont innermost et les systèmes de réécriture sont (AC)-convergents: tout terme a une unique forme normale (modulo AC). De plus, le réductions innermost considérées sont toujours de longueur minimale.

Un pas de *surréduction* d'un terme avec variables t est donné par une paire (θ, t') telle que θ est une substitution et t' un terme tel que $t\theta \to_{\mathcal{R}} t'$. La substitution θ est nommée substitution de la surréduction, tandis que t' est son résultat.

La propriété des *variants finis* a été introduite dans [CD05]. Elle exprime que les réductions possibles d'un terme t peuvent être anticipées: ce sont des instances des *variants finis* de t, qui sont obtenus par surréduction:

Definition 4 (Variants finis) *Un système de réécriture* AC*-convergent $\mathcal{R}$ a la* propriété des variants finis*, si, pour tout terme t, il existe un ensemble fini des termes $\mathcal{V}(t) = \{t\theta_1\downarrow, \ldots, t\theta_k\downarrow\}$ tel que $\forall\sigma.\exists\theta, \exists u \in \mathcal{V}(t).t\sigma\downarrow = u\theta$. $\mathcal{V}(t)$ est nommé l'ensemble des variants de t.*

Un exemple typique pour lequel on a les variants finis est le cas des théories pour lesquels la surréduction termine.

Soit E un ensemble d'objets. Un multi-ensemble de E est une fonction qui associe à chaque élément de E un entier. Pour un ordre $\prec$ sur E, son extension aux multi-ensembles de E est définie par $M \prec_m M' \Leftrightarrow M \neq M' \;\&\; \forall y.(M(y) > M'(y) \implies \exists x.\ y \prec x \;\&\; M(x) < M'(x))$.

Chapitre 4

Systèmes de contraintes et théories de l'intrus

4.1 Théories de l'intrus

Les théories de l'intrus spécifient les opérations autorisées sur les messages et les propriétés algébriques des symboles de fonction:

Definition 5 (Théorie de l'intrus) *Une* théorie de l'intrus *est une paire* $(\mathcal{I}, \mathcal{E})$*, où*

- $\mathcal{I}$ *est un ensemble de règles d'inférence. Une* regle d'inférence *consiste en un ensemble fini de termes* $\{u_1, \ldots, u_n\}$*, ses* prémisses*, et un terme* u*, sa* conclusion*, tels que* $\mathsf{Var}(u) \subseteq \mathsf{Var}(\{u_1, \ldots, u_n\})$*.*
- $\mathcal{E}$ *est une théorie équationelle.*

Dans la suite, on notera

$$\frac{u_1 \quad \cdots \quad u_n}{u}$$

ou encore $I(u_1), \ldots, I(u_n) \rightarrow I(u)$ une règle d'inférence ayant pour prémisses $u_1, \ldots, u_n$ et pour conclusion le terme u.

La signature $(\mathcal{F})$ qui modélise les messages contient des symboles pour les données atomiques de protocoles et pour les fonctions qui permettent de construire des nouveaux messages. Certains de ses éléments sont publics $(\mathcal{F}_{\mathsf{pub}})$, tels certains algorithmes, d'autres sont privés $(\mathcal{F}_{\mathsf{priv}})$, tels les mots de passe ou les clefs privées. L'intrus peut toujours appliquer une fonction publique à des messages qu'il connaît. Le type des théories de l'intrus le plus pertinent pour l'application aux protocoles de sécurité est par conséquent le suivant:

Definition 6 (Théorie de l'intrus équationnelle) *Soit* $\mathcal{F} = \mathcal{F}_{\mathsf{pub}} \uplus \mathcal{F}_{\mathsf{priv}}$ *une signature et* $\mathcal{E}$ *une théorie équationnelle. La* théorie de l'intrus équationnelle

associée à $(\mathcal{F}, \mathcal{E})$ *est* $(\mathcal{I}_\mathcal{F}, \mathcal{E})$*, où les regles de* $\mathcal{I}_\mathcal{F}$ *sont*

$$\frac{x_1 \quad \cdots \quad x_n}{f(x_1, \ldots, x_n)}$$

pour chaque $f \in \mathcal{F}_{\mathsf{pub}}$ *d'arité* n*.*

Exemple 2 *Soit* $\mathcal{F}_{\mathsf{DY}} = \{\mathtt{enc}/3, \mathtt{dec}/2, \mathtt{pub}/1, \mathtt{priv}/1, \langle,\rangle/2, \mathtt{proj_1}/1, \mathtt{proj_2}/1\}$*, où tous les symboles sauf* $\mathtt{priv}$ *sont publics. Soit* $\mathcal{E}_{\mathsf{DY}}$ *la théorie équationnelle suivante:*

$$\begin{aligned} \mathtt{dec}(\mathtt{enc}(\mathtt{pub}(y), x, z), \mathtt{priv}(y)) &= x \\ \mathtt{proj_1}(\langle x, y\rangle) &= x \\ \mathtt{proj_2}(\langle x, y\rangle) &= y \end{aligned}$$

La théorie de l'intrus associée à $(\mathcal{F}_{\mathsf{DY}}, \mathcal{E}_{\mathsf{DY}})$*, denotée par* $(\mathcal{I}_{\mathsf{DY}}, \mathcal{E}_{\mathsf{DY}})$*, modélise le chiffrement symétrique probabiliste et l'appariement.*

Definition 7 (Preuve et déductibilité) *Soit* $(\mathcal{I}, \mathcal{E})$ *une théorie de l'intrus,* H *un ensemble de termes et* t *un terme. Une* preuve*, avec des hypothèses* H *et conclusion* t *est un arbre, dont les noeuds sont étiquetés avec des termes tels que,*

- *la racine est étiquetée par* t
- *chaque feuille est étiquetée avec un terme* s *pour lequel il existe un terme* $s' \in H$ *tel que* $s =_\mathcal{E} s'$
- *si un noeud est étiqueté avec* s *et ses fils sont étiquetés avec* $s_1, \ldots, s_n$ *(*$n \geq 1$*), il existe une règle d'inférence dont les prémisses sont* $u_1, \ldots, u_n$ *et la conclusion est* u *et une substitution* θ *telle que* $u\theta =_\mathcal{E} s$ *et, pour tout* i*,* $u_i\theta = s_i$*.*

On dit que t *est déductible de* H*, et on écrit* $H \vdash t$*, quand il existe une preuve avec les hypothèses* H *et la conclusion* t*.*

Exemple 3 *L'arbre suivant est une preuve de* $\mathtt{enc}(\mathtt{pub}(k), a, r), \langle \mathtt{priv}(k), a\rangle \vdash a$ *dans* $(\mathcal{I}_{\mathsf{DY}}, \mathcal{E}_{\mathsf{DY}})$

$$\dfrac{\mathtt{enc}(\mathtt{pub}(k), a, r) \qquad \dfrac{\langle \mathtt{priv}(k), a\rangle}{\mathtt{priv}(k)}}{a}$$

Dans tout les cas d'application, la théorie $\mathcal{E}$ est presentée par un ensemble fini d'équations, qui peut (par complétion [JK86]) être transformé dans un système de réécriture $\mathcal{R}$ convergent (modulo AC). De plus, nous supposons dans cette thèse que le système de réécriture ainsi obtenu est fini. C'est une hypothèse naturelle, étant donné que ce système correspond, d'une certaine manière, au calcul effectué par les agents. Les arbres de preuve sont alors étiquetés avec des termes en forme normale. Si la théorie de l'intrus est équationnelle, on obtient une formulation utile de la notion de preuve à l'aide de la réécriture (e.g. [ANR07]):

Lemme 1 (Recettes) *Soit $(\mathcal{I}_\mathcal{F}, \mathcal{E})$ une théorie de l'intrus équationnelle. Identifions un ensemble de termes $T = \{t_1, \ldots, t_n\}$ avec une substitution $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ et soit t un terme en forme normale. Il existe une preuve π de $T \vdash t$ si, et seulement si, il existe un contexte $\zeta \in \mathcal{T}(\mathcal{F}_{\mathsf{pub}}, \{x_1, \ldots, x_n\})$ tel que $\zeta[T]{\downarrow} = t$. ζ est le terme de preuve de π, qu'on va nommer sa* recette.

Preuve: Immédiate par définitions. □

Exemple 4 *Soit $T = \{x_1 \mapsto \mathtt{enc}(\mathtt{pub}(k), a, r), x_2 \mapsto \langle \mathtt{priv}(k), a\rangle\}$ et $\mathcal{R}_{\mathsf{DY}}$ le système de réécriture convergent pour $\mathcal{E}_{\mathsf{DY}}$ (toutes les équations sont orientées de gauche à droite). La recette pour la preuve dans $(\mathcal{I}_{\mathsf{DY}}, \mathcal{R}_{\mathsf{DY}})$ de l'exemple 3 est $\mathtt{dec}(x_1, \mathtt{proj}_1(x_2))$.*

4.2 Systèmes de contraintes

Un système de contraintes représente les exécutions possibles d'un protocole, étant donné un entrelacement d'actions des agents (e.g, [Del06, MS01, CR05]). Il peut aussi exprimer la capacité de l'intrus d'apprendre un secret après n interactions avec le protocole. De plus, toutes les traces possibles d'un nombre borné des sessions du protocole peuvent être représentées par un ensemble fini de systèmes de contraintes [Bau07, CLCZ10].

Exemple 5 *Considérons un protocole jouet qui contient le seul rôle:*

$$R_a = \nu b.\ \mathsf{in}(x_0).\mathsf{in}(x_1).\mathsf{out}(\mathtt{enc}(\mathtt{proj}_1(x_0), \langle b, x_1\rangle, r)).\mathsf{out}(\mathtt{priv}(a))$$

νb représente la création d'un nouveau nonce b, in *la réception et* out *l'envoi d'un message. Le processus ci-dessus correspond à un agent* a *qui reçoit deux messages, x_0 et x_1, et envoie sur le réseau $\langle b, x_1\rangle$, chiffré avec la première partie du message x_0, et sa clef privée. Ensuite, on demande pour quelles valeurs de ces messages l'intrus peut apprendre b:*

$$D := \left\{ \begin{array}{llll} & & & \\ & a \overset{?}{\vdash} x_0 & \wedge & \\ & a \overset{?}{\vdash} x_1 & \wedge & \\ \mathtt{enc}(\mathtt{proj}_1(x_0), \langle b, x_1\rangle, r),\ \mathtt{priv}(a),\ a & \overset{?}{\vdash} x_2 & \wedge & x_2 = b \end{array} \right.$$

À gauche de ces contraintes est représentée la connaissance de l'intrus. Initiallement, il connaît le nom de l'agent qui exécute le protocole, ensuite il intercepte les messages envoyés par cet agent. À droite sont représentées les données attendues par l'agent avant d'envoyer sa réponse. La dernière contrainte exprime la capacité de l'intrus d'apprendre b.

Plus généralement, si T_1 est l'ensemble de termes représentant la connaissance initiale de l'intrus, alors $T_1 \overset{?}{\vdash} x_1 \wedge x_1 = s_1$ éxprime la capacité de construire à partir de T_1 un message x_1, qui doit être de la forme s_1, le motif attendu par

un agent A. Ensuite, l'attaquant intercepte la réponse t_1 de A, laquelle, avec T_1, produit une connaisance augmentée T_2, et ainsi de suite... Des variables prennent la place des sous-messages qui sont traités comme des boites noires par les participants, et donc peuvent être remplacées par des messages arbitraires:

Definition 8 (Système de contraintes) *Un* système de contraintes $\mathcal{C}$ *est un ensemble fini d'équations entre les termes de* $\mathcal{T}(\mathcal{F}, \mathcal{X})$ *et une séquence finie de contraintes de déductibilité* $\{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_n \stackrel{?}{\vdash} x_n\}$*, où chaque* T_i *est un ensemble fini de termes dans* $\mathcal{T}(\mathcal{F}, \mathcal{X})$*, et* $x_1, \ldots, x_n$ *sont des variables. De plus, les propriétés suivantes doivent être satisfaites:*

Origination: $\mathsf{Var}(T_i) \subseteq \{x_1, \ldots, x_{i-1}\}$*, pour tout* $1 \leq i \leq n$*.*

Monotonie: $T_i \subseteq T_{i+1}$*, pour* $1 \leq i < n$*.*

Le fait de considérer la monotonie et l'origination dans le modèle formel n'est pas restrictif, compte tenu de l'application visée. En effet, la monotonie est due au fait que l'intrus n'oublie jamais des messages. L'origination exprime que les messages envoyés par les agents sur le réseau sont construits à partir des messages reçus précédemment, et non pas à partir des messages arbitraires. Notons que les variables qui apparaissent dans les équations de $\mathcal{C}$ peuvent ne pas appartenir à $\{x_1, \ldots, x_n\}$.

Exemple 6 *L'ensemble de contraintes ci-dessous n'est pas un système de contraintes, puisqu'il ne satisfait pas l'origination:*

$$\begin{cases} a, b \stackrel{?}{\vdash} z \quad \wedge \quad z = \mathtt{enc}(x, y) \\ a, b, x \stackrel{?}{\vdash} y \end{cases}$$

En revanche, D dans l'exemple 5 est un système de contraintes.

Definition 9 (Solution) *Étant donnée une théorie de l'intrus* $(\mathcal{I}, \mathcal{E})$ *et un système de contraintes* $\mathcal{C}$*, une substitution* σ *est une* solution *de* $\mathcal{C}$ *si: son domaine contient toutes les variables de* $\mathcal{C}$*, et*

- *pour chaque* $s = t \in \mathcal{C}$*,* $s\sigma =_{\mathcal{E}} t\sigma$
- *pour chaque* $T \stackrel{?}{\vdash} s \in \mathcal{C}$*,* $T\sigma \vdash s\sigma$*.*

Exemple 7 $\sigma = \{x_0 \mapsto \langle \mathtt{pub}(a), a\rangle, x_1 \mapsto \langle a, a\rangle\}$ *est une solution de* D*, de l'exemple 5, dans* $(\mathcal{I}_{\mathsf{DY}}, \mathcal{E}_{\mathsf{DY}})$*.*

Dans la suite, nous nous intéressons à la question suivante: un système de contraintes a-t-il une solution ? Peut-on le simplifier de manière à ce que la représentation de ses solutions soit triviale? Ce qui nous intéresse sont les techniques générales pour apporter des réponses à ces questions, basées sur des propriétés de la théorie de l'intrus. Notre fil conducteur sera la réduction de modèles: nous identifions des concepts pour simplifier d'abord la théorie de l'intrus (partie II) et ensuite les systèmes de contraites (partie III).

Partie II

Théories pures et porte-monnaie électronique

Des problèmes de plus en plus complexes se posent pour la vérification formelle des propriétés de sécurité. Elles viennent, d'une part, des propriétés nouvelles désirables en pratique (e.g. dans les protocoles de vote) et, d'une autre, d'un souci d'avoir un modèle formel au plus proche de la réalité. Ces deux sources de complexité conduisent à des théories équationnelles toujours plus riches, pour exprimer dans le modèle formel les propriétés algébriques des opérations utilisées.

D'où la nécessité de concepts et de techniques pour diminuer cette complexité, diviser le problème en sous-problèmes plus simples. Les techniques de combinaison des théories sont des réponses naturelles à ce défi: si les théories se combinent d'une manière disjointe ([CR05]) ou hiérarchique ([CR06]), le problème pour la théorie résultat peut être décomposé en sous-problèmes correspondant à chaque théorie de départ.

Mais, étant donnée la profusion mentionnée ci-dessus, il est parfois, voire souvent, le cas que la théorie considérée combine les symboles d'une manière plus intriquée. Nous allons voir dans ce chapitre une étude de cas où les combinaisons disjointes et hiérarchiques ne s'appliquent pas. Pour effectuer quand même une simplification, qui nous amène ensuite à une procédure de décision, nous allons proposer un schéma de combinaison plus souple, qui ne suppose ni des théories disjointes ni une hiérarchie, mais qui est une généralisation des deux. La portée de ce résultat est à la fois pratique, car on déplace la frontière des théories combinables, et conceptuelle, car il permet de choisir la décomposition en fonction des équations de la théorie, au lieu de partir de combinaisons préetablies, comme l'union disjointe ou hiérarchique.

Contributions. Nous allons proposer un algorithme de résolution des systèmes de contraintes en deux parties: un pas général de réduction et, pour résoudre les systèmes simplifiés obtenus, une procédure spécifique pour la théorie de notre étude de cas EP (chapitre 5).

Dans un premier temps, nous allons montrer comment le problème général peut se réduire à un problème plus élémentaire: la résolution de systèmes de contraintes purs, où les preuves à chercher sont atomiques. La notion de système pur et le résultat de réduction sont determinés par une notion de sous-terme sémantique, qui décompose un terme selon les sous-théories du système de réécriture et détermine la complexité des preuves qui restent à chercher (chapitre 6). Ce chapitre établit donc un résultat géneral de combinaison.

De plus, pour purifier encore plus le problème, nous allons montrer que la théorie des termes dans les systèmes obtenus peut être stabilisée. Ça nous permettra de remplacer les termes par des constantes et, en nous appuyant sur des propriétés AG, réduire les systèmes de contraintes modulo EP vers des systèmes diophantiens linéaires (chapitre 7).

En mettant les deux réductions ensemble, nous allons dériver une procédure de décision pour notre étude de cas EP (chapitre 8).

Chapitre 5

Étude de cas: le porte-monnaie électronique

Dans ce chapitre nous allons présenter notre étude de cas: un protocole de porte-monnaie électronique developpé par France Telecom et modelisé par Stéphanie Delaune [Del06].

5.1 Le protocole

Le protocole étudié comporte trois agents: le porte-monnaie électronique EP, un serveur S et une autorité de confiance A. On ne va pas condisérer ici l'autorité A, qui participe au protocole seulement dans le cas d'une dispute des deux parties. On dénote par b et r deux entiers positifs, qui sont publics. La clé publique de EP est $b^s \mod r$, tandis que s est sa clé privée.

Il y a d'abord une phase d'authentification de S. On ne la considère pas ici, car elle ne repose pas sur des propriétés algébriques. Après cette phase, S et EP s'accordent sur un nonce de session N_s et S possède la clé publique $b^s \mod r$ de EP. Alors,

1. Le porte-monnaie EP génère un nonce N, construit un message M (qui est utilisé seulement en cas de conflit et n'est pas relevant ici) et envoie au serveur S: $\mathsf{hash}(b^N \mathsf{mod}\ r, S, N_s, M, X)$, où X est le montant à payer.
2. Le serveur S envoie un challenge à EP, sous la forme d'un nonce N_c.
3. Le porte-monnaie EP renvoie $N - s \times N_c$ et soustrait X de son compte.
4. Le serveur S vérifie que le message reçu au pas 1 est en concordance avec le message reçu au pas 3 et rajoute X à son compte.

Le pas important et difficile est le dernier: S doit être capable d'effectuer la vérification. Voici les operations qui lui permettent de la faire:

$$\mathsf{hash}((b^s)^{N_c} \times b^{N-s\times N_c} \mathsf{mod}\ r, S, N_s, M, X) = \mathsf{hash}(b^N \mathsf{mod}\ r, S, N_s, M, X)$$

Le serveur S elève b^s à la puissance N_c (b^s est public et N_c est connu par S), elève b à la puissance $N - s \times N_c$ (qui est le message envoyé au pas 3), et multiplie les deux résultats.

On peut voir que les propriétés suivantes de l'exponentielle sont utilisées dans la vérification ci-dessus:

$$exp(exp(b, y), z) = exp(b, y \times z) \quad exp(b, x) \times exp(b, y) = exp(b, y + z)$$

ainsi que les propriétés de groupe abélien pour $\times$ et $+$.

5.2 La théorie équationnelle

Le problème est que, si on prend en compte toutes ces propriétés dans notre modèle formel, on peut dériver la distributivité de $\times$ par rapport à $+$:

$$exp(b, x \times (y + z)) = exp(exp(b, x), y) \times exp(exp(b, x), z) = exp(b, x \times y + x \times z)$$

Dans ce cas, l'unification ([Nar96]) et donc la sécurité ([CDL06]) devient indécidable. C'est pourquoi nous introduisons un nouveau symbole de fonction h, dont le sens est de représenter l'exponentiation avec une base fixe b: $h(x) = exp(b, x)$. Nous utilisons aussi deux symboles pour la multiplication: $\bullet$ et $\star$, avec les axiomes EP suivantes: $\mathsf{AG}(+, J_+, e_+)$, $\mathsf{AG}(\star, J_\star, e_\star)$, $\mathsf{AG}(\bullet, J_\bullet, e_\bullet)$ (où AG sont les axiomes de groupe abélien, qui vont être presentées ci-dessus), ainsi que:

$$\begin{array}{rcl} exp(\mathsf{h}(x), y) & = & \mathsf{h}(x \star y) \\ exp(exp(x, y), z) & = & exp(x, y \star z) \end{array} \qquad \mathsf{h}(x) \bullet \mathsf{h}(y) = \mathsf{h}(x + y)$$

Ces équations suffisent pour la vérification effectuée dans le dernier pas du protocole. L'utilisation des deux symboles pour la multiplication nous aide dans la théorie que nous allons developper, mais notre conjecture, pour les travaux futurs, est que ce partage n'est pas nécessaire.

À cette théorie équationnelle, on associe un système de réécriture convergent. Pour chaque $\circ \in \{+, \star, \bullet\}$, $\mathcal{R}_{\mathsf{AG}(\circ)}$ est le système de réécriture modulo AC pour $\circ$:

$$\begin{array}{rcl} x \circ e_\circ & \to & x \\ J_\circ(x) \circ J_\circ(y) & \to & J_\circ(x \circ y) \\ J_\circ(J_\circ(x)) & \to & x \\ J_\circ(x) \circ J_\circ(y) \circ z & \to & J_\circ(x \circ y) \circ z \\ J_\circ(x \circ y) \circ x \circ z & \to & J_\circ(y) \circ z \end{array} \qquad \begin{array}{rcl} x \circ J_\circ(x) & \to & e_\circ \\ J_\circ(e_\circ) & \to & e_\circ \\ J_\circ(x) \circ x \circ y & \to & y \\ J_\circ(x \circ y) \circ x & \to & J_\circ(y) \\ J_\circ(J_\circ(x) \circ y) & \to & x \circ J_\circ(y) \end{array}$$

L'orientation inhabituelle des règles pour les inverses est pour obtenir la propriété des variants finis [CD05, Del06], dont on va voir par la suite le rôle essentiel dans notre procédure de décision.

Les propriétés de l'exponentiation se traduisent dans les règles suivantes:

$$\mathcal{R}_0 = \left\{ \begin{array}{rcl@{\qquad}rcl} exp(h(x), y) & \to & h(x \star y) & J_\bullet(h(x)) & \to & h(J_+(x)) \\ exp(exp(x, y), z) & \to & exp(x, y \star z) & h(e_+) & \to & e_\bullet \\ h(x) \bullet h(y) & \to & h(x + y) & J_\bullet(h(x) \bullet y) & \to & h(J_+(x)) \bullet J_\bullet(y) \\ h(x) \bullet h(y) \bullet z & \to & h(x + y) \bullet z & exp(e_\bullet, x) & \to & h(e_+ \star x) \\ exp(x, e_\star) & \to & x & & & \end{array} \right.$$

On prend alors $\mathcal{R}_{EP} = \mathcal{R}_0 \cup \mathcal{R}_{AG(+)} \cup \mathcal{R}_{AG(\star)} \cup \mathcal{R}_{AG(\bullet)}$.

Lemme 2 *$\mathcal{R}_{\mathsf{EP}}$ est convergent modulo $\mathsf{AC}(+, \star, \bullet)$.*

Preuve: Ceci a été prouvé automatiquement avec CiME [CM96]. □

Lemme 3 ([Del06]) *$\mathcal{R}_{\mathsf{EP}}$ a la propriété des variants finis.*

Exemple 8 *Considerons le terme $x + J_+(a)$. Ses variants finis (pour EP) sont $e_+ = (a + J_+(a))\downarrow$, $y = (y + a + J_+(a))\downarrow$,$J_+(a) = (e_+ + J_+(a))\downarrow$, $J_+(y + a) = (J_+(y) + J_+(a))\downarrow$, $x + J_+(a)$.*

Notons que, en orientant la règle pour l'inverse dans l'autre direction, on obtient bien un système de réécriture AC-convergent pour AG, mais qui n'a pas la propriété des variants finis [CD05].

Chapitre 6

Théories pures et décomposition des contraintes de déductibilité

Nous allons voir que les résultats existants sur la combinaison de théories [CR05, CR06] ne s'appliquent pas à notre étude de cas EP. Un nouveau résultat de combinaison, que nous mettons en place dans ce chapitre et qui généralise les combinaisons disjointes et bien-modées, est utilisé pour effectuer une réélle réduction du problème pour EP.

L'idée sera d'identifier une notion de sous-terme compatible avec la théorie équationnelle (section 6.1), d'où deux propriétés fondamentales découlent: la localité (section 6.3) et la conservativité (section 6.4). Avec la propriété des variants finis, elles permettent d'effectuer la réduction vers des systèmes de contraintes purs (section 6.5), où les preuves à chercher sont élémentaires (i.e. sans sous-termes).

Ensuite, pour les systèmes de contraintes purs, nous montrons qu'on peut stabiliser la théorie des termes qui apparaissent dans les contraintes: par instanciation et normalisation, la théorie d'un terme ne change pas (section 6.6). Cette deuxième réduction nous permet de fixer les termes étrangers à une contrainte et les abstraire par des constantes. Ceci achève le processus de combinaison et va être utilisé dans le chapitre 7 pour obtenir des équations diophantiennes linéaires.

6.1 Sous-termes sémantiques et théories pures

Le problème de résolution des contraintes consiste à chercher des contextes (les recettes) qui temoignent de la déductibilité de certains termes. Pour réduire le problème, nous allons définir une manière de décomposer les termes (et donc les contextes). Ce qui nous intéresse, c'est une manière de choisir un sous-

ensemble de $\mathsf{St}_s(t)$, les sous-termes syntaxiques de t, qui a du sens par rapport à la théorie équationnelle. Ce sont les sous-termes sémantiques $\mathsf{St}(t) \subseteq \mathsf{St}_s(t)$, que nous appelerons simplement sous-termes dans la suite. La définition des sous-termes peut être donnée en définissant d'abord les facteurs, qui sont les sous-termes aux positions minimales:

Definition 10 (Facteurs, Sous-termes) *Les* facteurs *sont définis par une fonction* Fact*, qui associe à chaque terme t un contexte linéaire et non-vide ζ et une substitution ρ tels que $t = \zeta\rho$. Les facteurs de t sont alors $\mathsf{Fact}(t) = \mathsf{img}(\rho)$ - un sous-ensemble de ses sous-termes syntaxiques stricts. Souvent, nous appelerons ζ* le contexte qui définit les facteurs de t. $\mathsf{Fact}(t)$ *sera, selon nos besoins, soit un ensemble, soit un multi-ensemble.*

Les sous-termes sont définis à partir des facteurs, $\mathsf{St}(t) = \{t\} \cup \mathsf{St}(\mathsf{Fact}(t))$.

Pour être utile, la définition des sous-termes doit dépendre de la théorie équationnelle. Prenons d'abord les exemples de théories disjointes et de théories bien-modées.

Théories disjointes [CR05]. Soit $\mathcal{R}$ un système de réécriture sur une signature $\mathcal{F}$. Supposons que $\mathcal{F}$ est une union disjointe $\mathcal{F} = \mathcal{F}_0 \uplus \mathcal{F}_1$, avec la propriété que pour toute règle $l \to r \in \mathcal{R}$, on a ou bien $l, r \in \mathcal{T}(\mathcal{F}_0, \mathcal{X})$, ou bien $l, r \in \mathcal{T}(\mathcal{F}_1, \mathcal{X})$. On pose $\top(t) = i$, si $top(t) \in \mathcal{F}_i$, et $\top(t) = t$, quand t est une constante qui n'apparaît pas dans $\mathcal{F}$ ou une variable.

Les facteurs de t sont alors définis par le contexte minimal, linéaire et non-vide ζ, pour lequel il existe $t_1, \ldots, t_n$, tels que:

- $t = \zeta[t_1, \ldots, t_n]$
- $\forall 1 \leq i \leq n.\ \top(t_i) \neq \top(\zeta)$

Alors $\mathsf{Fact_d}(t) = \{t_1, \ldots, t_n\}$. Autrement dit, $\mathsf{Fact_d}(t)$ sont les sous-termes syntaxiques maximaux de t qui ne sont pas dans la même théorie que t.

Par exemple, si $\mathcal{F} = \{+, J_+, e_+, \star, J_\star, e_\star\}$, alors $\mathcal{R} = \mathcal{R}_{\mathsf{AG}(+)} \cup \mathcal{R}_{\mathsf{AG}(\star)}$ est une combinaison des théories disjointes. Pour $t = J_\star(a \star b + c) \star (a + b) \star c$, on a $\mathsf{St_d}(t) = \{t, a \star b + c, a + b, a \star b, a, b, c\}$.

$\mathcal{R}_{\mathsf{EP}}$ n'est pas une combinaison des théories disjointes, car tous les symboles sont en interaction. Mais on peut observer une hiérarchie des symboles, avec $\mathcal{F}_+$ et $\mathcal{F}_\star$ au premier niveau et les autres symboles à un niveau supérieur. Ceci suggère qu'une combinaison hiérarchique est peut-être possible. Nous introduisons donc maintenant les théories bien modées de [CR06] dont le but est précisément la combinaison hiérarchique.

Théories bien-modés [CR06]. Notre présentation ci-dessous est légèrement différente de celle de [CR06], pour mettre en évidence les points communs entre ces différentes méthodes de combinaison. Mais les définitions correspondent exactement aux définitions de [CR06].

Soit M un ensemble fini et ordonné de modes. Pour chaque symbole de fonction f, $m(f) \in M^+$ est une séquence de modes telle que $m(f) = m_1, \ldots, m_n \to m$, où n est l'arité de f et $m \geq m_i$ pour tout i. Quand $f \in \mathcal{F}_{ac}$, $m(f) = m_1^* \to m$ et $m \geq m_1$: le même mode m_1 est affecté à tous les arguments. Pour tout terme t, on pose $\top(t) = m$ si $top(t) = f$ pour un f tel que ci-dessus, et $\top(t) = t$, si t est une constante qui n'apparaît pas dans $\mathcal{R}$ ou une variable.

Pour tout contexte ζ et toute variable $x \in \mathsf{Var}(\zeta)$, on associe alors les modes (i.e. les théories) attendues aux positions determinées par x:

$$\begin{array}{ll} \mathsf{th}(\zeta, x) = \{m_i \mid & \exists \zeta', \zeta_1, \ldots, \zeta_n, f. \\ & \zeta = \zeta'[f(\zeta_1, \ldots, \zeta_{i-1}, x, \zeta_{i+1}, \ldots, \zeta_n)] \\ & \& \ m(f) = m_1, \ldots, m_n \to m\} \end{array}$$

Les *facteurs* d'un terme t, $\mathsf{Fact}_{\mathsf{BM}}(t)$, sont définis par le contexte minimal, linéaire et non-vide ζ pour lequel il existe $t_1, \ldots, t_n$ tels que

- $t = \zeta[t_1, \ldots, t_n]$
- $\forall 1 \leq i \leq n.\ \top(t_i) \notin \mathsf{th}(\zeta, x_i)$

Exemple 9 *Considérons $\mathcal{R}_{\mathsf{EP}}$, avec $M = \{0, 1\}$, $1 > 0$ et la fonction m définie par $h : 0 \to 1, exp : 1, 0 \to 1$, $+, \star : 0^* \to 0$, $\bullet : 1^* \to 1$.*

Pour le contexte $C = x \bullet exp(y, z)$, on a $\mathsf{th}(C, x) = \mathsf{th}(C, y) = 1$ et $\mathsf{th}(C, z) = 0$. Donc, pour $t = (a \star b) \bullet exp(a \bullet b, c \bullet d)$, on a $\mathsf{Fact}_{\mathsf{BM}}(t) = \{a \star b, a, b, c \bullet d\}$.

Un terme t est *bien-modé* si $\mathsf{Fact}(t) \subseteq \mathcal{X}$. Un système de réécriture $\mathcal{R}$ est bien-modé si (il existe un choix des modes tel que), pour chaque règle $l \to r \in \mathcal{R}$,

1. l et r sont bien-modés
2. - ou bien $\top(l) = \top(r)$ et
     - $\forall x \in \mathsf{Var}(l).\ |\mathsf{th}(l, x)| = 1$ & $\forall x \in \mathsf{Var}(r).\ |\mathsf{th}(r, x)| = 1$
     - $\forall x \in \mathsf{Var}(r).\ \mathsf{th}(r, x) = \mathsf{th}(l, x)$
   - ou bien $r \in \mathsf{Var}(l)$ et $\mathsf{th}(l, r) = \{\top(l)\}$

La première condition assure que la réécriture est compatible avec la hiérarchie, tandis que la deuxième nous dit que les théories ne sont pas confondues par la réécriture.

La théorie peut alors se décomposer en $\mathcal{R} = \mathcal{R}_1 \cup \ldots \cup \mathcal{R}_{|M|}$, où chaque $\mathcal{R}_i$ contient les règles $l \to r$ telles que $\top(l) = m_i$.

L'étude de cas. Pour le choix des modes donné dans l'exemple 9, $\mathcal{R}_{\mathsf{EP}}$ est bien-modé. Il n'existe pas d'autre choix (en dehors du choix trivial, où on a un mode unique) pour lequel le système est bien modé.

Même si $\mathcal{R}_{\mathsf{EP}}$ est une théorie bien-modée, on ne peut pas appliquer le résultat de [CR06], parce que la théorie maximale, i.e. $\{exp, h, \bullet, J_\bullet, e_\bullet\}$, est trop complexe. L'hypothèse principale de [CR06] (nommée hypothèse 1 dans [CR06]) sur la théorie $\mathcal{R} = \mathcal{R}_1 \cup \mathcal{R}_0$ est la suivante:

Pour toutes recettes ζ_1 et ζ_2 de théorie 1, pour tout ensemble de termes T, on a $\zeta_1[\zeta_2[T], T]\downarrow = t$ & $\zeta_2[T]\downarrow \notin \mathsf{St}(T, t) \implies \exists \zeta', \zeta''.\ |\zeta'| \leq |\zeta_1|$ & $\top(\zeta'') = 0$ & $\zeta'[\zeta''[T], T]\downarrow = t$.

Autrement dit, tout enchainement de symboles de mode 1, dont les termes intermédiaires ne sont pas des sous-termes de la conclusion ou des hypothèses, peut être simplifié en descendant dans la hiérarchie. Comme montré dans [CR06], cette hypothèse est satisfaite par la théorie de l'exponentiation quand on prend en compte seulement la propriété $exp(exp(x, y), z) = exp(x, y \star z)$: exp est le seul symbole de mode 1 et un enchainement des exp peut etre simulé par un seul exp plus un contexte qui n'utilise que $\star$, de mode 0:

$$exp(\ldots, exp(t_0, t_1), \ldots, t_n)\ldots) = exp(t_0, t_1 \star \ldots \star t_n)$$

Dans l'exemple qui suit, on voit que cette propriété n'est pas satisfaite par $\mathcal{R}_{\mathsf{EP}}$.

Exemple 10 *Soit* $T = \{h(a), h(b), c\}$ *et* $t = h((a + b) \star c)$. *On a* $exp(h(a) \bullet h(b), c)\downarrow = t$, $h(a) \bullet h(b)\downarrow \notin \mathsf{St}(T, t)$ *et pourtant il n'existe pas une autre preuve, plus simple, de* $T \vdash t$

Une autre hypothèse importante pour l'effectivité de la réduction dans [CR06] est la finitude du nombre des contextes $f(x_1, \ldots, x_n)$, où f est de mode 1, qu'on utilise dans les preuves (i.e. les recettes). Ce n'est jamais le cas pour une théorie dont un symbole AC (ou AG) est de mode maximal, car, pour la recherche des preuves (e.g. pour la localité ou l'hypothèse 1 de [CR06]) on est obligés de ne considérer que des preuves aplaties. En particulier, cette condition n'est pas satisfaite par $\mathcal{R}_{\mathsf{EP}}$, car on a un symbole de type AG, $\bullet$, de mode 1.

Les deux exemples de combinaison que nous venons de voir sont en fait des instances du schéma de combinaison que nous allons introduire dans le paragraphe suivant. L'idée sera la même: avoir une définition de sous-termes sémantiques compatible avec la théorie équationnelle. Mais cette compatibilité sera basée sur d'autres criteres, plus souples que la disjonction ou la hiérarchie de symboles.

Théories pures. Soit $\mathcal{F} = \mathcal{F}_1 \uplus \ldots \uplus \mathcal{F}_\ell \uplus \{\mathsf{h}\}$. Les ensembles $\mathcal{F}_1, \ldots, \mathcal{F}_\ell$ définissent les sous-théories de $\mathcal{R}$ et h est un symbole d'interface, qui va déterminer l'interaction entre les sous-théories.

On considère une fonction partielle $\mathsf{H} : \{1, \ldots, \ell\} \mapsto \{1, \ldots, \ell\}$, qui définit l'interaction entre les théories, en associant une théorie d'arrivée pour l'application du symbole d'interface à une théorie de départ. L'idée, on le verra formellement dans la suite, est qu'un terme de théorie th' est consideré comme étant de théorie th, si $\mathsf{H}(\mathsf{th}') = \mathsf{th}$ et il utilise h comme chapeau. Si Th est un ensemble de théories, on définit $\mathsf{H}(\mathsf{Th}) = \{\mathsf{th} \mid \exists \mathsf{th}_0 \in \mathsf{Th}.\ \mathsf{H}(\mathsf{th}_0) = \mathsf{th}\}$ et $\mathsf{H}^{-1}(\mathsf{Th}) = \{\mathsf{th} \mid \exists \mathsf{th}_0 \in \mathsf{Th}.\ \mathsf{H}(\mathsf{th}) = \mathsf{th}_0\}$. On pose $\mathsf{thdest}(\mathsf{h}) = \mathsf{H}(\{1, \ldots, \ell\})$ et $\mathsf{thsource}(\mathsf{h}) = \mathsf{H}^{-1}(\{1, \ldots, \ell\})$.

Dans la suite, nous allons appeler "théorie" ou bien un des ensembles $\mathcal{F}_{\mathsf{th}}$, ou bien un des indices $\mathsf{th} \in \{1, \ldots, \ell\}$.

Exemple 11 *Nous avons* $\mathcal{F}_{\mathsf{EP}} = \mathcal{F}_{exp} \uplus \mathcal{F}_\bullet \uplus \mathcal{F}_\star \uplus \mathcal{F}_+ \uplus \{\mathsf{h}\}$, *avec*

- $\mathcal{F}_{exp} = \{exp\}$
- $\mathcal{F}_\bullet = \{\bullet, J_\bullet, e_\bullet\}$
- $\mathcal{F}_\star = \{\star, J_\star, e_\star\}$
- $\mathcal{F}_+ = \{+, J_+, e_+\}$
- $\mathsf{h} = h$, $\mathsf{H}(+) = \bullet$ *et* $\mathsf{H}(\star) = exp$

On a $\mathsf{thdest}(\mathsf{h}) = \{\bullet, exp\}$ *et* $\mathsf{thsource}(\mathsf{h}) = \{+, \star\}$.

Pour simplifier la présentation et les preuves dans la suite, on suppose que H est injective, que $\mathsf{thsource}(\mathsf{h}) \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$ et que h est un symbole public.

Nous avons choisi de ne considérer ici qu'un seul symbole d'interface et prendre les restrictions ci-dessus. En effet ces restrictions sont satisfaites dans notre exemple et, bien que les lever ne semble pas poser de problème majeur, cela conduirait à alourdir considérablement la rédaction.

Definition 11 (Théorie d'un terme) *On définit* $\top : \mathcal{T}(\mathcal{F}, \mathcal{X}, \mathcal{C}) \mapsto \{1, \ldots, \ell, \mathsf{h}\} \cup \mathcal{C} \cup \mathcal{X}$ *comme suit:*

- $\top(t) = t$, *si* $t \in \mathcal{C} \cup \mathcal{X}$
- $\top(f(t_1, \ldots, t_n)) = \mathsf{th}$, *si* $f \in \mathcal{F}_{\mathsf{th}}, \mathsf{th} \in \{1, \ldots, \ell\}$.
- $\top(\mathsf{h}(t)) = \mathsf{H}(\top(t))$, *si* $\top(t) \in \mathsf{thsource}(\mathsf{h})$, $\top(\mathsf{h}(t)) = \mathsf{h}$, *sinon.*

Exemple 12 *On obtient la définition suivante de* $\top$ *pour EP:* $\top(t) = \circ$, *si* $top(t) \in \{\circ, e_\circ, J_\circ\}$, $\top(h(t)) = \bullet$, *si* $\top(t) = +$ *et* $\top(h(t)) = exp$, *si* $\top(t) = \star$. *Dans tous les autres cas, on a* $\top(t) = top(t)$.

Comme dans le cas des théories bien-modées, nous associons une théorie pour chaque argument des symboles de fonction. La différence est que nous ne supposons pas de hiérarchie de théories. De plus, comme nous l'avons vu dans la définition de $\top$, le symbole d'interface permet de greffer un terme d'une théorie dans une autre.

Pour chaque symbole $f \in \mathcal{F}_1 \uplus \ldots \uplus \mathcal{F}_\ell$ et tout i, $1 \leq i \leq ar(f)$, on associe une théorie pour le i-eme argument de f, $\mathsf{th}(f, i) \in \{1, \ldots, \ell\}$. Si f est AC, on a une seule théorie pour tous ses arguments. L'idée de ces choix est d'identifier, dans la théorie équationnelle, avec quelles théories interagit une fonction à travers l'un de ses arguments fixé. Evidemment, si un symbole de fonction apparaît plusieurs fois dans les équations de la théorie, tous les symboles avec lesquels il interagit par cet argument doivent appartenir à la même théorie. L'alternative serait de considérer $\mathsf{th}(f, i)$ comme étant un ensemble de théories.

Exemple 13 *Pour* EP *on a*

- $\mathsf{th}(exp, 1) = exp$, *à cause de e.g.* $exp(exp(x, y), z) \to exp(x, y \star z)$. *Notons que* x *dans* $exp(h(x), y) \to h(x \star y)$ *est moralement (et formellement ci-dessous) dans la théorie* $\star$*, d'où* $h(x)$ *est dans la théorie exp, par définition de* H. *L'interaction de exp avec* $\bullet$ *(à travers* $e_\bullet$ *dans la dernière équation) est facilement contrôlable, car elle s'effectue par l'intermediaire d'une constante.*
- $\mathsf{th}(exp, 2) = \star$, *à cause de e.g.* $exp(exp(x, y), z) \to exp(x, y \star z)$.
- *Pour tout* $\circ \in \{+, \star, \bullet\}$, $\mathsf{th}(J_\circ, 1) = \circ$, $\mathsf{th}(\circ) = \circ$, *à cause des équations* AG.

On étend cette notion de théorie(s) attendue(s) à l'ensemble de positions d'un terme, qui peut contenir des symboles d'interface. Puisqu'on a des symboles AC, les ensembles de positions sont specifiées par des contextes. Pour la définition ci-dessus, on étend H^{-1} par $\mathsf{H}^{-1}(\{\mathsf{h}\}) = \mathsf{thsource}(\mathsf{h})$.

Definition 12 (Théories d'une position) *Pour tout contexte non-variable* $\zeta \in \mathcal{T}(\mathcal{F}, \mathcal{X})$ *et toute variable* x*, on définit* $\mathsf{th}(\zeta, x)$ *par*

- $\mathsf{th}(\zeta, x) = \emptyset$, *si* $x \notin \mathsf{Var}(\zeta)$,
- $\mathsf{th}(\zeta, x) = \mathsf{th}(\zeta, x, \{\top(\zeta)\})$, *sinon*

où $\mathsf{th}(.,.,.)$ *est défini comme suit, pour* $\mathsf{Th}_0 \subseteq \{1, \ldots, \ell, \mathsf{h}\}$,

- $\mathsf{th}(x, x, \mathsf{Th}_0) = \mathsf{Th}_0$
- $\mathsf{th}(\zeta, x, \mathsf{Th}_0) = \emptyset$, *si* $x \notin \mathsf{Var}(\zeta)$
- $\mathsf{th}(f(\zeta_1, \ldots, \zeta_n), x, \mathsf{Th}_0) = \mathsf{th}(\zeta_1, x, \mathsf{th}(f, 1)) \cup \ldots \cup \mathsf{th}(\zeta_n, x, \mathsf{th}(f, n))$, *si* $f \neq \mathsf{h}$
- $\mathsf{th}(\mathsf{h}(\zeta'), x, \mathsf{Th}_0) = \mathsf{th}(\zeta', x, \mathsf{H}^{-1}(\mathsf{Th}_0))$

Le rôle de $\mathsf{th}(\zeta, x)$ est de collecter les théories avec lesquelles ζ (e.g. une règle de réécriture) interagit à travers la variable x. Le but est de pouvoir ensuite raisonner selon le rapport entre la théorie d'une instance de x et cet ensemble de théories.

Le sous-termes syntaxiques maximaux qui ne sont pas dans la théorie attendue sont les facteurs:

Definition 13 (Facteurs) *Pour tout terme* t*, soit* $\zeta[x_1, \ldots, x_n]$ *le contexte linéaire minimal en taille et non-vide pour lequel il existe* $t_1, \ldots, t_n$ *tels que*

- $t = \zeta[t_1, \ldots, t_n]$
- $\forall i.\ \top(t_i) \notin \mathsf{th}(\zeta, x_i)$
- $\forall i.\ top(t_i) = \mathsf{h} \implies \mathsf{thdest}(\mathsf{h}) \cap \mathsf{th}(\zeta, x_i) = \emptyset$

On définit alors $\mathsf{Fact}(t) = \{t_1, \ldots, t_n\}$.

La troisième condition nous dit que le symbole d'interface est transparent s'il est greffé dans une des théories qui sont dans sa destination.

Pour ne pas alourdir la notation, les fonctions Fact et St pour les combinaisons pures ne sont pas indicées, contrairement à ceux pour les théories disjointes, bien-modées ou, ci-dessous, EP. Dans la suite, les notations Fact et St se réfèrent à l'instantiation des notions générales de facteurs et sous-termes pour les combinaisons pures, sauf dans les cas où leur généralité sera claire à partir du contexte.

Exemple 14 *En corroborant nos choix de* $\mathsf{th}(f, i)$ *et les définitions 12,13, on déduit la définition suivante des facteurs* EP*: si* $t \in \mathcal{C} \cup \mathcal{X}$, $\mathsf{Fact}_{\mathsf{EP}}(t) = \emptyset$. *Sinon,* $\mathsf{Fact}_{\mathsf{EP}}(t) = \mathsf{Fact}_{\top(t)}(t)$, *où*

- $\mathsf{Fact}_\circ(C_\circ[t_1, \ldots, t_n]) = \bigcup_{i=1}^n \mathsf{Fact}_\circ(t_i)$ *si* $C_\circ \in \mathcal{T}(\{\circ, J_\circ, e_\circ\}, \mathcal{X})$
- $\mathsf{Fact}_\bullet(h(t)) = \mathsf{Fact}_+(t)$ *et, dans tous les autres cas,* $\mathsf{Fact}_\circ(t) = \{t\}$ *si* $\top(t) \neq \circ$
- $\mathsf{Fact}_{exp}(exp(u, v)) = \mathsf{Fact}_{exp}(u) \cup \mathsf{Fact}_\star(v)$, $\mathsf{Fact}_{exp}(h(u)) = \mathsf{Fact}_\star(u)$ *et, dans tous les autres cas,* $Fact_{exp}(t) = \{t\}$ *si* $\top(t) \neq exp$
- *Pour les autres symboles de fonction* f, $\mathsf{Fact}_f(f(t_1, ..., t_n)) = \{t_1, \ldots, t_n\}$ *et* $\mathsf{Fact}_f(t) = \{t\}$ *si* $top(t) \neq f$.

Par exemple,

- $\mathsf{Fact}_{\mathsf{EP}}(exp(h(a \star b), c \star d)) = \{a, b, c, d\}$
- $\mathsf{Fact}_{\mathsf{EP}}(exp(h(a + b), c + d)) = \{a + b, c + d\}$
- $\mathsf{Fact}_{\mathsf{EP}}(h(a \star b) \bullet h(a + c) \bullet (a + d)) = \{a \star b, a, c, a + d\}$
- $\mathsf{Fact}_{\mathsf{EP}}(h(a \star b)) = \mathsf{Fact}_{\mathsf{EP}}(h(a + b)) = \{a, b\}$ *et* $\mathsf{Fact}_{\mathsf{EP}}(h(a \bullet b)) = \{a \bullet b\}$

Notons aussi que $\mathsf{St}_{\mathsf{EP}}$ *est une notion plus fine que lorsque* EP *est traité comme une théorie bien-modée: pour* t *dans l'exemple 9, on a* $\mathsf{Fact}_{\mathsf{EP}}(t) = \{a \star b, exp(a \bullet b, c \bullet d)\}$ *et* $\mathsf{St}_{\mathsf{EP}}(t) = \mathsf{St}_{\mathsf{BM}}(t) \cup \{exp(a \bullet b, c \bullet d), a \bullet b\}$.

Pour deux termes t, u, la définition suivante formalise l'existence d'une occurrence de u dans $\mathsf{St}(t)$ qui n'est pas sous une occurrence du symbole d'interface h. Elle sera utile pour avoir une information plus fine sur les occurrences de facteurs d'un terme.

Definition 14 (Sous-terme de surface) *Soient* t, u *deux termes. On définit le predicat* $\mathsf{SSt}(t, u)$ *par*

- $\mathsf{SSt}(t, u) = false$, *si* $u \notin \mathsf{St}(t)$

- $\mathsf{SSt}(t,t) = true$
- $\mathsf{SSt}(f(t_1,\ldots,t_n),u) = \mathsf{SSt}(t_1,u) \vee \ldots \vee \mathsf{SSt}(t_n,u)$, *si* $f \neq \mathsf{h}$
- $\mathsf{SSt}(\mathsf{h}(t'),u) = false$

Soit le contexte $\zeta[x_1,\ldots,x_n]$ *tel que* $t = \zeta[t_1,\ldots,t_n]$ *et* $\mathsf{Fact}(t) = \{t_1,\ldots,t_n\}$. *Alors on définit* $\mathsf{SSt}_f(t,u) = \bigvee_{t_i=u} \mathsf{SSt}(\zeta,x_i)$.

Dans SSt_f on s'intéresse à certains occurences précises d'un sous-terme: celles où il est un facteur.

Exemple 15 *Pour* $t = x \bullet (x+y) \bullet h(x+y)$, *on a*

- $\mathsf{SSt}(t,x) = \mathsf{SSt}_f(t,x) = true$, *car on a deux occurences de* x *qui ne sont pas couverts par un* h, *dont une est un facteur de* t.
- $\mathsf{SSt}(t,y) = true$ *et* $\mathsf{SSt}_f(t,y) = false$, *car, même si on a une occurence de* y *qui n'est pas couverte par un* h, *l'occurence qui est un facteur est sous un* h.

Definition 15 (Terme pur) *On va noter par* $\mathcal{C}_\mathcal{R}$ *les constantes de* $\mathcal{R}$ *en forme normale. Un terme* t *dont tous les facteurs sont des variables ou des constantes en forme normale, i.e.* $\mathsf{Fact}(t) \subseteq \mathsf{Var}(t) \cup \mathcal{C}_\mathcal{R}$, *est nommé* pur.

Exemple 16 *Chaque terme dans les règles de* EP *est pur. En fait, les facteurs de toute règle sont des variables, sauf pour la dernière,* $exp(e_\bullet, x) \to \mathsf{h}(e_+ \star x)$, *qui contient des facteurs dans* $\mathcal{C}_\mathcal{R}$.

La définition suivante généralise les combinaisons disjointes et bien-modées de théories:

Definition 16 (Théorie pure) *Un système de réécriture* $\mathcal{R}$ *est* pur *si, pour toute règle* $l \to r \in \mathcal{R}$, *on a*

1. l *et* r *sont purs.*
2. - *ou bien* $\top(l) = \top(r)$ *et*
     - $\forall x \in \mathsf{Var}(l).\ |\mathsf{th}(l,x)| = 1$ & $\forall x \in \mathsf{Var}(r).\ |\mathsf{th}(r,x)| = 1$
     - $\forall x \in \mathsf{Var}(r).\ \mathsf{th}(r,x) = \mathsf{th}(l,x)$
   - *ou bien* $r \in \mathsf{Var}(l)$ *et* $\mathsf{th}(l,r) = \{\top(l)\}$
3. - *Pour tout* $x \in \mathsf{Var}(r)$, $\mathsf{SSt}(r,x) \Rightarrow \mathsf{SSt}(l,x)$
   - *Si* $l = \mathsf{h}(l')$, *alors* $l', r \in \mathcal{C}_\mathcal{R}$

Les deux premières conditions sont les mêmes que pour les combinaisons disjointes ou bien-modées, les notions de théorie d'un terme, théorie d'une position, facteurs étant differentes. Le troisème point énonce des conditions techniques sur l'occurrence du symbole d'interface dans les règles de réécriture ($\Rightarrow$ est

l'implication logique). Elles sont trivialement satisfaites par les autres types des combinaisons (disjointes, bien-modées), ou il n'existe pas de symbole d'interface.

Cette classe contient évidemment les combinaisons disjointes, bien-modées et, il est facile de le voir, notre étude de cas EP. Au fait, pour toute théorie, on peut calculer un partage en sous-théories qui la fait pure:

Exemple 17 (Rechiffrement) *Soit* $\mathcal{F}_{renc} = \{\mathtt{enc}, \mathtt{dec}, \mathtt{renc}, f\}$ *et considérons le système de réécriture* $\mathcal{R}_{renc}$ *qui modélise la théorie du rechiffrement du chiffrement randomisé:*

$$\begin{array}{rcl} \mathtt{dec}(\mathtt{enc}(x,y,z),x) & \to & y \\ \mathtt{renc}(\mathtt{enc}(x,y,z),z') & \to & \mathtt{enc}(x,y,f(z,z')) \end{array}$$

Considérons les sous-théories suivantes $\mathcal{F}_{enc} = \{\mathtt{enc}, \mathtt{renc}\}$, $\mathcal{F}_{dec} = \{\mathtt{dec}\}$, $\mathcal{F}_f = \{f\}$, $\mathcal{F}_\perp = \emptyset$ *et le choix des arguments*

- $\mathsf{th}(\mathtt{enc}, 1) = \perp$, $\mathsf{th}(\mathtt{enc}, 2) = \mathtt{dec}$, $\mathsf{th}(\mathtt{enc}, 3) = f$

  $\mathsf{th}(\mathtt{enc}, 1) = \perp$ *exprime que le premier argument de* enc *n'a pas d'interaction: toute théorie est étrangere pour cet argument. Notons que* $\perp$ *est juste la théorie vide. Le deuxième argument de* enc *est dans la théorie* dec *à cause de la première règle: la variable* y *doit être dans la théorie* dec*. La théorie du troisième argument est déterminée par le membre droit de la deuxième règle.*

  Des raisons similaires expliquent les autres choix:

- $\mathsf{th}(\mathtt{dec}, 1) = \mathtt{enc}$, $\mathsf{th}(\mathtt{dec}, 2) = \perp$
- $\mathsf{th}(\mathtt{renc}, 1) = \mathtt{enc}$, $\mathsf{th}(\mathtt{renc}, 2) = f$
- $\mathsf{th}(f, 1) = \mathsf{th}(f, 2) = f$

Avec ces choix, la théorie est pure.

Notons que la seule décomposition (hiérarchique) non-triviale qui fait la théorie bien-modée est $\{\mathtt{enc}, \mathtt{dec}, \mathtt{renc}\} \uplus \{f\}$*. La restriction d'avoir une hiérarchie pousse* enc, dec, *et* renc *dans la même théorie. Or, l'intérêt est de maximiser le nombre des sous-théories, pour que le problème obtenu après simplifications soit plus élémentaire.*

Discussion. Contrairement à leur ressemblance (voulue), les différences entre les systèmes de réécriture purs et notre résultat de réduction d'un côté, et les systèmes de réécriture bien-modés et le résultat de [CR06] de l'autre, sont importantes et multiples:

- dans notre résultat de réduction, nous ne demanderons aucune hypothèse supplémentaire sur la théorie, à part une propriété de variants finis. C'est une différence fondamentale par rapport à [CR06], où ils ont à la fois une forte hypothèse supplémentaire (hypothèse 1), et une condition sur la réductibilité de la théorie (hypothèse 3), qui joue le role des nos variants finis. Nous allons montrer que tout système pur satisfait en fait une propriété plus générale que l'hypothèse 1 de [CR06]: la localité.

- la théorie de mode maximal doit etre finie pour l'effectivité de la réduction dans [CR06]. Cette restriction enlève les théories avec des symboles AC (e.g. $\mathcal{F}_\bullet$). Pour notre résultat, nous n'avons aucune hypothèse de ce genre.

- nos sous-théories ne sont pas hiérarchiques: aucune restriction n'est posée dans le choix des théories des arguments d'un symbole de fonction. C'est pourquoi, par exemple, nous avons un partage plus fin dans les exemples du rechiffrement et de la théorie EP.

- les symboles d'interface permettent plus de souplesse dans la définition de facteurs: la théorie d'un argument de h dépend de son contexte. Il est possible qu'un tel passage au contexte ouvre des portes pour des résultats futurs ou fasse la liaison avec d'autres domaines de la réécriture.

6.2 Lemmes préliminaires: facteurs, réécriture et remplacement

Nous allons montrer dans cette section quelques propriétés de systèmes purs. Nous allons voir comment les facteurs d'un terme sont déterminés après l'application d'une substitution (section 6.2.1), des pas de réécriture (section 6.2.2) ou des remplacements (6.2.3). À chaque fois, le système de réécriture considéré est pur.

Notation. Soit S une fonction qui associe un ensemble de termes à un terme (e.g. $\mathsf{St}, \mathsf{Fact}, G$). Par convention, pour un ensemble T, nous avons $\mathsf{S}(T) = \bigcup_{t\in T} \mathsf{S}(t)$ et $\mathsf{S}(M_1, \ldots, M_n) = \mathsf{S}(M_1) \cup \ldots \cup \mathsf{S}(M_n)$, où M_i sont des termes ou des ensembles de termes. Nous allons parfois identifier une substitution avec l'ensemble de termes dans son image: $\mathsf{S}(\sigma) = \mathsf{S}(\mathsf{img}(\sigma))$.

6.2.1 Propriétés des facteurs

Dans les combinaisons disjointes ou bien-modées, quand on instancie une variable $x \in \mathsf{Fact}(t)$ avec une substitution σ, ou bien l'instance est dans une théorie étrangère, et alors les facteurs de $t\sigma$ s'arretent à $x\sigma$, ou bien l'instance appartient à une théorie attendue, et alors les facteurs descendent dans $\mathsf{Fact}(x\sigma)$. Dans notre cas, la présence du symbole d'interface nous oblige à descendre dans $x\sigma$, juste en enlevant le h, même quand $x\sigma$ est un terme alien. C'est le role de la fonction $\overline{G}$. La fonction $\overline{F}$, au contraire, rajoute un h au dessus d'un terme: un terme t qui change de théorie par normalisation peut être égal à l'un de ses facteurs (comme pour les combinaisons sans symbole d'interface), mais aussi à un terme dans $\overline{F}(\mathsf{Fact}(t))$.

Definition 17 (Fonctions $F, \overline{F}, \overline{\overline{F}}, G, \overline{G}, \overline{\overline{G}}$) *Pour tout terme t, soit $F(t) = \{t\}$, si $\top(t) \notin \mathsf{thsource}(\mathsf{h})$, et $F(t) = \{t, \mathsf{h}(t)\}$, si $\top(t) \in \mathsf{thsource}(\mathsf{h})$. Notons que, puisque $\mathsf{thsource}(\mathsf{h}) \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$, F est idempotente. Pour tout t, on définit:*

- $\overline{F}(t) = \{t, \mathsf{h}(t)\}$
- $G(t) = \{u \mid t \in F(u)\}$
- $\overline{G}(t) = \{u \mid t \in \overline{F}(u)\}$
- $\overline{\overline{F}}(t) = \overline{F}(\overline{F}(t))$ *et* $\overline{\overline{G}}(t) = \overline{G}(\overline{G}(t))$

Notons que $F(\overline{F}(t)) = \overline{F}(t)$.

Exemple 18 *Pour EP, on a* $F(t) = \{t, h(t)\}$, *si* $top(t) \in \{+, J_+, e_+, \star, J_\star, e_\star\}$. *Dans tous les autres cas, on a* $F(t) = \{t\}$.

Le rôle de la définition suivante est de calculer les facteurs d'un terme par rapport à un contexte: c'est la vue qu'on a d'un terme en venant d'une autre théorie. Ici nous voyons quand on va descendre dans l'instance en enlevant le symbole d'interface.

Definition 18 (Facteurs par rapport à une théorie) *Soit* t *un terme et* Th *un ensemble de théories. On définit* $\mathsf{Fact}_{\mathsf{Th}}(t)$ *de la manière suivante,*

- *Si* $\top(t) \in \mathsf{Th}$, $\mathsf{Fact}_{\mathsf{Th}}(t) = \mathsf{Fact}(t)$.
- *Sinon,*
  - *Si* $top(t) \neq \mathsf{h} \vee \mathsf{Th} \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$, $\mathsf{Fact}_{\mathsf{Th}}(t) = \{t\}$
  - *Si* $t = \mathsf{h}(t')$ & $\mathsf{Th} \cap \mathsf{thdest}(\mathsf{h}) \neq \emptyset$, $\mathsf{Fact}_{\mathsf{Th}}(t) = \{t'\}$

Exemple 19 *Pour EP, on a* $\mathsf{Fact}_{\{exp,\bullet\}}(h(a + b)) = \mathsf{Fact}_{\{exp,\bullet\}}(h(a \star b)) = \{a, b\}$. *Par contre,* $\mathsf{Fact}_{\{exp,\bullet\}}(h(a \bullet b)) = \{a \bullet b\}$ *et* $\mathsf{Fact}_{\star}(h(a \bullet b)) = \{h(a \bullet b)\}$.

Quand on descend dans un terme étranger, ce n'est jamais trop profondement:

Lemme 4 *Soit* t *un terme et* Th *un ensemble de théories avec* $\top(t) \notin \mathsf{Th}$. *On a*

- $\mathsf{Fact}_{\mathsf{Th}}(t) \subseteq \overline{G}(t)$
- *Si* $\mathsf{Fact}_{\mathsf{Th}}(t) = \{t\}$, *alors* $\forall t'.top(t') = top(t) \implies \mathsf{Fact}_{\mathsf{Th}}(t') = \{t'\}$

Preuve: Le premier point est immediat par les définitions de $\overline{G}$ et de $\mathsf{Fact}_{\mathsf{Th}}$. Pour le deuxième point, il suffit de remarquer que, pour tout u, $\mathsf{Fact}_{\mathsf{Th}}(u) = \{u\}$ si et seulement si $top(u) \neq \mathsf{h}$ ou $top(u) = \mathsf{h}$ & $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{Th} = \emptyset$. □

La proposition 1 montre un résultat simple, mais crucial dans la suite: pour calculer les facteurs d'une instance d'un terme, il suffit de calculer les facteurs de la substitution par rapport aux théories de positions où elle s'applique. D'abord, nous montrons un lemme sur le comportement de $\mathsf{th}(.,.)$ lors de la composition des contextes.

Lemme 5 *Soit ζ un contexte linéaire, $\zeta' \in \mathsf{St}_s(\zeta)$ et $x \in \mathsf{Var}(\zeta')$. Si $\zeta' \neq \mathsf{h}(x)$, alors $\mathsf{th}(\zeta, x) = \mathsf{th}(\zeta', x)$.*

Preuve: Si $top(\zeta') \neq \mathsf{h}$, on conclut facilement par définition. Si $\zeta' = \mathsf{h}(\zeta'')$, soit $\zeta_0[y]$ une recette telle que $\zeta = \zeta_0[\zeta']$. On a

- $\mathsf{th}(\zeta, x) = \mathsf{th}(\zeta'', x, \mathsf{H}^{-1}(\mathsf{th}(\zeta_0, y)))$.
- $\mathsf{th}(\zeta', x) = \mathsf{th}(\zeta'', x, \mathsf{H}^{-1}(\{\top(\zeta')\}))$.

On utilisant le fait que $\mathsf{thsource}(\mathsf{h}) \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$, les deux ensembles sont égaux à $\mathsf{th}(\zeta'', x, \emptyset)$ et on peut conclure. □

Nous sommes prets à prouver le résultat principal de cette sous-section:

Proposition 1 *Soit $\zeta[x_1, \ldots, x_n]$ un contexte linéaire tel que $\mathsf{Fact}(\zeta) = \{x_1, \ldots, x_n\}$. Pour toute substitution $\theta = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$, on a*

$$\mathsf{Fact}(\zeta\theta) = \mathsf{Fact}_{\mathsf{th}(\zeta, x_1)}(t_1) \cup \ldots \cup \mathsf{Fact}_{\mathsf{th}(\zeta, x_n)}(t_n)$$

Et, par le lemme 4, $\mathsf{Fact}(\zeta\theta) \subseteq \mathsf{Fact}(\theta) \cup \overline{G}(\theta)$.

Preuve: Soit χ le contexte qui définit les facteurs de $\zeta[t_1, \ldots, t_n]$, i.e. il existe $u_1, \ldots, u_m$ tels que $\zeta[t_1, \ldots, t_n] = \chi[u_1, \ldots, u_m]$ et $\mathsf{Fact}(\zeta[t_1, \ldots, t_n]) = \{u_1, \ldots, u_m\}$.

Montrons d'abord que $\chi = \zeta\gamma$, pour une substitution γ. Ce n'est pas le cas seulement s'il existe un $y \in \mathsf{Var}(\chi)$ et un $\zeta' \in \mathsf{St}_s(\zeta) \setminus \mathcal{X}$ tels que $\top(\zeta'\theta) \notin \mathsf{th}(\chi, y)$ et, si $top(\zeta'\theta) = \mathsf{h}$, $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{th}(\chi, y) = \emptyset$. Si $top(\zeta'\theta) \neq \mathsf{h}$, on obtient $\top(\zeta') = \top(\zeta'\theta)$ et $\zeta' \in \mathsf{Fact}(\zeta)$, en contradiction avec l'hypothèse sur ζ. Si $top(\zeta'\theta) = \mathsf{h}$, comme on a dans ce cas $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{th}(\chi, y) = \emptyset$, on déduit $\top(\zeta') \notin \mathsf{th}(\chi, y)$ et donc $\zeta' \in \mathsf{Fact}(\zeta)$, contredisant encore la pureté de ζ.

Par conséquent, pour chaque $1 \leq i \leq n$, il existe un contexte linéaire ζ_i et une substitution σ_i tels que $t_i = \zeta_i\sigma_i$ et $\mathsf{Fact}(\zeta[t_1, \ldots, t_n]) = \mathsf{Var}(\zeta_1)\sigma_1 \cup \ldots \cup \mathsf{Var}(\zeta_n)\sigma_n$.

Considérons d'abord un i tel que $\top(t_i) \in \mathsf{th}(\zeta, x_i)$. Montrons que dans ce cas $\mathsf{Var}(\zeta_i)\sigma_i = \mathsf{Fact}(t_i) = \mathsf{Fact}_{\mathsf{th}(\zeta, x_i)}(t_i)$. Soit χ_i le contexte qui définit les facteurs de t_i, i.e. $t_i = \chi_i[\mathsf{Fact}(t_i)]$. Il suffit de montrer que $\chi_i = \zeta_i$. On considère deux cas:

- $\zeta_i \notin \overline{F}(\mathcal{X})$ et $\chi_i \notin \overline{F}(\mathcal{X})$. Dans ce cas, $\zeta_i = \chi_i$ résulte immédiatement par le lemme 5 et la définition des facteurs.
- $\zeta_i \in \overline{F}(\mathcal{X})$ ou $\chi_i \in \overline{F}(\mathcal{X})$. Montrons que ce cas n'est pas possible. $\chi_i \notin \mathcal{X}$ est immédiat par la définition des facteurs. $\zeta_i \notin \mathcal{X}$ aussi, puisque $\top(\zeta_i\sigma_i) \in \mathsf{th}(\zeta, x_i)$ et donc $\zeta_i\sigma_i \notin \mathsf{Fact}(\zeta\sigma)$. Supposons maintenant par absurde que $\zeta_i = \mathsf{h}(w)$, avec $w \in \mathcal{X}$. Puisque $w\sigma_i \in \mathsf{Fact}(\zeta[t_1, \ldots, t_n])$, on a $\top(w\sigma_i) \notin \mathsf{H}^{-1}(\mathsf{th}(\zeta, x_i))$. Puisque $\top(\mathsf{h}(w\sigma_i)) = \top(\zeta_i\sigma_i) = \top(t_i) \in \mathsf{th}(\zeta, x_i)$, on a au contraire $\top(w\sigma_i) \in \mathsf{H}^{-1}(\mathsf{th}(\zeta, x_i))$. On conclut donc $\zeta_i \notin \overline{F}(\mathcal{X})$. De même, si $\chi = \mathsf{h}(w)$, on doit avoir $\top(w\sigma_i) \notin \mathsf{thsource}(\mathsf{h})$ et donc $\top(t_i) = \mathsf{h}$, ce qui est en contradiction avec $\top(t_i) \in \mathsf{th}(\zeta, x_i)$. Par conséquence, on conclut l'impossibilité de ce cas.

Nous venons de conclure le cas $\top(t_i) \in \mathsf{th}(\zeta, x_i)$.

Considérons maintenant un i tel que $\top(t_i) \notin \mathsf{th}(\zeta, x_i)$. Par la définition des facteurs, on a

- ou bien $top(t_i) \notin \mathsf{h} \vee \mathsf{thdest}(\mathsf{h}) \cap \mathsf{th}(\zeta, x_i) = \emptyset$, et alors $t_i \in \mathsf{Fact}(\zeta[t_1, \ldots, t_n])$
- ou bien $t_i = \mathsf{h}(u_i)$ & $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{th}(\zeta, x_i) \neq \emptyset$, et alors, par l'idempotence de H et les définitions de $\top$ et th, $u_i \in \mathsf{Fact}(\zeta[t_1, \ldots, t_n])$.

C'est exactement la définition de $\mathsf{Fact}_{\mathsf{th}(\zeta,x_i)}(t_i)$, quand $\top(t_i) \notin \mathsf{th}(\zeta, x_i)$ et on peut conclure le lemme.

□

Exemple 20 *Pour* EP*, reprenant l'exemple 14,*

- *Pour* $\zeta = exp(x, y)$*, on a* $\mathsf{th}(\zeta, x) = \{exp\}, \mathsf{th}(\zeta, y) = \{\star\}$ *et*
  - *Si* $x\theta = h(a \star b)$ *et* $y\theta = c \star d$*, on a* $\mathsf{Fact}(\zeta\theta) = \{a, b, c, d\} = \mathsf{Fact}(x\theta, y\theta)$*, puisque* $\top(x\theta) \in \{exp\}$ *et* $\top(y\theta) \in \{\star\}$
  - *Si* $x\theta = h(a + b)$ *et* $y\theta = c + d$*, on a* $\mathsf{Fact}(\zeta\theta) = \{a + b, c + d\} \subseteq \overline{G}(x\theta, y\theta)$*, puisque* $\top(x\theta) \notin \{exp\}$ *et* $\top(y\theta) \notin \{\star\}$
- *Pour* $\zeta = x \bullet y \bullet z$*, on a* $\mathsf{Fact}(h(a \star b) \bullet h(a+c) \bullet (a+d)) = \{a \star b, a, c, a+d\} = \mathsf{Fact}_\bullet(h(a \star b)) \cup \mathsf{Fact}_\bullet(h(a \star b)) \cup \mathsf{Fact}_\bullet(a + d)$.
- *Pour* $\zeta = h(x)$*, on a* $th(\zeta, x) = \mathsf{thsource}(h) = \{+, \star\}$ *et*
  - *si* $x\theta \in \{a + b, a \star b\}$, $\mathsf{Fact}(h(x\theta)) = \{a, b\} = \mathsf{Fact}_{\{+,\star\}}(x\theta)$
  - *si* $x\theta = h(a \bullet b)$, $\mathsf{Fact}(h(x\theta)) = \{a \bullet b\} = \mathsf{Fact}_{\{+,\star\}}(x\theta)$

Le lemme suivant nous permet d'utiliser la proposition 1 pour le contexte qui définit les facteurs d'un terme.

Lemme 6 *Pour tout terme* t*, soit* ζ *le contexte linéaire qui définit les facteurs de* t*, i.e.* $t = \zeta\sigma, \mathsf{img}(\sigma) = \mathsf{Fact}(t)$*. Alors* ζ *est tel que* $\mathsf{Fact}(\zeta) = \mathsf{Var}(\zeta)$.

Preuve: Soit, par l'absurde, $\zeta' \in \mathsf{Fact}(\zeta) \setminus \mathcal{X}$ et $\zeta''[x]$ tel que $\zeta = \zeta''[\zeta']$. Par définitions, on a $\top(\zeta') \notin \mathsf{th}(\zeta'', x)$ et

- ou bien $top(\zeta') \neq \mathsf{h}$.
- ou bien $top(\zeta') = \mathsf{h}$ & $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{th}(\zeta'', x) = \emptyset$.

Dans les deux cas, par définitions, on déduit $\top(\zeta'\sigma) \notin \mathsf{th}(\zeta'', x)$ et $\zeta'\sigma \in \mathsf{Fact}(t)$. Contradiction avec la définition de ζ. □

Pour étudier le comportement d'un facteur, nous allons collecter toutes les théories dans lesquelles il occure:

Definition 19 (Occurrences d'un facteur) *Soit* t, u *deux termes et* ζ *le contexte linéaire qui définit les facteurs de* t*, i.e.* $t = \zeta\sigma$*,* $\mathsf{img}(\sigma) = \mathsf{Fact}(t)$*. On définit* $\mathsf{th}_f(t, u) = \bigcup_{x \in \mathsf{Var}(\zeta), x\sigma = u} \mathsf{th}(\zeta, x)$.

Exemple 21 $\mathsf{th}_f(h(a\star b)\bullet(a\star b), a\star b) = \{+, \bullet\}$ *et* $\mathsf{th}_f(h(a\bullet b)\bullet(a\bullet b), a\bullet b) = \{+\}$.

Le lemme suivant a une grande importance par la suite. Étant une généralisation de la proposition 1, il explique comment sont obtenus les facteurs et les sous-termes d'un terme après l'application d'une substitution. Encore une foi, la situation est comme dans la combinaison disjointe: la substitution est ou bien étrangère et les facteurs sont à la surface de σ (i.e dans $\overline{G}(\sigma)$), ou bien indigène, et les facteurs sont dans $\mathsf{Fact}(\sigma)$.

Lemme 7 *Pour tout terme u et substitution σ,*

$$\mathsf{Fact}(u\sigma) = (\mathsf{Fact}(u) \setminus \mathcal{X})\sigma \cup M_\sigma$$

et

$$\mathsf{St}(u\sigma) = (\mathsf{St}(u) \setminus \mathcal{X})\sigma \cup \mathsf{St}(M_\sigma),$$

où $M_\sigma \subseteq \mathsf{Fact}(\sigma) \cup \overline{G}(\sigma)$.

Plus précisément, $M_\sigma = \bigcup_{x\in\mathsf{Var}(u)\cap\mathsf{Fact}(u)} M_{x\sigma}$, où, pour tout $x \in \mathsf{Var}(u) \cap \mathsf{Fact}(u)$, $M_{x\sigma} \subseteq \mathsf{Fact}(x\sigma) \cup \overline{G}(x\sigma)$.

De plus, quand $|\mathsf{th}_f(u, x)| = 1$ ou bien x a une seule occurrence dans $\mathsf{Fact}(u)$, on a $M_{x\sigma} = \mathsf{Fact}_{\mathsf{th}_f(u,x)}(x\sigma)$.

Preuve: Soit ζ le contexte linéaire et (par le lemme 6) tel que $\mathsf{Fact}(\zeta) = \mathsf{Var}(\zeta)$ qui définit les facteurs de u, i.e. $\exists\theta.\ u = \zeta\theta, \mathsf{img}(\theta) = \mathsf{Fact}(u)$.

Soit $\mathsf{Var}(\zeta) = V_1 \uplus V_2$, avec V_1, V_2 tels que $\forall x \in V_1.\ x\theta \notin \mathsf{Var}(u)$ et $\forall x \in V_2.\ x\theta \in \mathsf{Var}(u)$. Par la proposition 1 et le lemme 4, on obtient $\mathsf{Fact}(u\sigma) = \mathsf{Fact}(\zeta(\theta\sigma)) = V_1\theta\sigma \cup \bigcup_{y\in V_2} \mathsf{Fact}_{\mathsf{th}(\zeta,y)}(y\theta\sigma) = (\mathsf{Fact}(u) \setminus \mathcal{X})\sigma \cup M_\sigma$, avec $M_\sigma \subseteq \mathsf{Fact}(\sigma) \cup \overline{G}(\sigma)$ (on a aussi utilisé le lemme 4).

Plus précisément, $M_\sigma = \bigcup_{x\in\mathsf{Fact}(u)\cap\mathsf{Var}(u)} M_{x\sigma}$, avec $M_{x\sigma}$ défini comme suit: pour chaque $x \in \mathsf{Fact}(u) \cap \mathsf{Var}(u)$, $M_{x\sigma} = \bigcup_{y\in V_x} \mathsf{Fact}_{\mathsf{th}(\zeta,y)}(x\sigma)$, avec $V_x = \{y \in \mathsf{Var}(\zeta) \mid y\theta = x\}$. Si, de plus, $|\mathsf{th}_f(u, x)| = 1$ ou si x a une seule occurrence dans $\mathsf{Fact}(u)$ (i.e. $|V_x| = 1$), on a $\forall y, y' \in V_x.\ \mathsf{th}(\zeta, y) = \mathsf{th}(\zeta, y') = \mathsf{th}_f(u, x)$. Donc $M_{x\sigma} = \mathsf{Fact}_{\mathsf{th}_f(u,x)}(x\sigma)$. □

Exemple 22 *Pour $u = exp(h(x), x)$ on a $|\mathsf{th}_f(u, x)| = |\{\star\}| = 1$ et $\mathsf{Fact}(u\sigma) = \mathsf{Fact}_\star(x\sigma)$, pour tout σ. Par contre, pour $u = exp(x, x)$, on a $|\mathsf{th}_f(u, x)| = |\{exp, \star\}| = 2$ et $\mathsf{Fact}(exp(exp(a, b), exp(a, b))) = \{a, b, exp(a, b)\} = \mathsf{Fact}(exp(a, b)) \cup \overline{G}(exp(a, b))$.*

La localité (section 6.3) et la conservativité (section 6.4) sont des propriétés qui assurent que les sous-termes des preuves ou des solutions à considérer sont dans un ensemble borné par le problème de départ. Pour les prouver, nous allons montrer que tout terme trop grand dans une preuve ou une solution peut être *remplacé* par un terme plus petit. Le remplacement est, bien sûr, dans les sous-termes sémantiques:

Definition 20 (Remplacement) *Soit $v = \zeta[v_1, \ldots, v_n]$ un terme tel que $\mathsf{Fact}(v) = \{v_1, \ldots, v_n\}$. Pour deux termes u, t, on définit le remplacement de t par u dans v par*

- $v\{t \mapsto u\} = u$, *si* $v = t$
- $v\{t \mapsto u\} = \zeta[v_1\{t \mapsto u\}, \ldots, v_n\{t \mapsto u\}]$, *sinon*

Exemple 23 *Pour* $t = a \star b$ *et* $u = c$, *on a*

- $(exp(a \star b, a \star b) \bullet (a \star b))\{t \mapsto u\} = exp(c, a \star b) \bullet c$
- $(exp(h(a \star b), h(a \star b)) \star a \star b)\{t \mapsto u\} = exp(h(a \star b), h(a \star b)) \star a \star b$

Comme dans le lemme 7, mais cette fois-ci pour les notions de remplacement et sous-terme de surface, le rôle du lemme 8 est d'étudier le comportement d'un terme t lors de l'application d'une substitution. Un sous-terme de $x\sigma$ sera de surface dans $t\sigma$ s'il est de surface dans $x\sigma$ et si x et de surface dans t. Le remplacement est propagé aussi dans la substitution.

Lemme 8 *Soit* u *un terme tel que, pour tout* $x \in \mathsf{Var}(u) \cap \mathsf{Fact}(u)$, $|\mathsf{th}_f(u, x)| = 1$ *ou bien* x *a une seule occurrence dans* $\mathsf{Fact}(u)$. *Considérons une substitution* σ *et un terme* $t \neq u\sigma$. *Soit la partition* $F_1 \uplus F_2 \uplus F_3$ *de* $\mathsf{Var}(u) \cap \mathsf{Fact}(u)$ *telle que:*

- $\forall x \in F_1 . \top(x\sigma) \in \mathsf{th}_f(u, x)$
- $\forall x \in F_2 . \top(x\sigma) \notin \mathsf{th}_f(u, x)$ & $\mathsf{Fact}_{\mathsf{th}_f(u,x)}(x\sigma) = \{t\}$
- $F_3 = (\mathsf{Var}(u) \cap \mathsf{Fact}(u)) \setminus (F_1 \cup F_2)$

1. *Soit* $M = \{v \in \mathsf{Fact}(u) \setminus \mathcal{X} \mid v\sigma = t\}$. *Alors*

$$\mathsf{SSt}_f(u\sigma, t) = \mathsf{SSt}_f(u, M) \vee \bigvee_{x \in F_1} (\mathsf{SSt}_f(u, x) \wedge \mathsf{SSt}_f(x\sigma, t)) \vee \bigvee_{x \in F_2} (\mathsf{SSt}_f(u, x) \wedge \mathsf{SSt}(x\sigma, t))$$

2. *Soit* $C[w_1, \ldots, w_l, F_1, F_2, F_3]$ *un contexte tel que* $u = C[u_1, \ldots, u_l, F_1, F_2, F_3]$ *et* $\{u_1, \ldots, u_l\} = \mathsf{Fact}(u) \setminus \mathcal{X}$. *Pour un terme* v, *soit* σ' *la substitution définie comme suit:*

   - $\forall x \in F_1$,
     - $x\sigma' = x\sigma\{t \mapsto v\}$, *si* $x\sigma \neq t$
     - $x\sigma' = x\sigma$, *si* $x\sigma = t$
   - $\forall x \in F_2$, $x\sigma' = \chi[v]$, *si* $x\sigma = \chi[t] \in \overline{F}(t)$
   - $\forall x \in F_3$, $x\sigma' = x\sigma\{t \mapsto v\}$

   Alors

$$u\sigma\{t \mapsto v\} = C[u_1\sigma\{t \mapsto v\}, \ldots, u_l\sigma\{t \mapsto v\}, F_1\sigma', F_2\sigma', F_3\sigma']$$

3. *Si* M *est un ensemble de termes clos et* $t \notin \mathsf{Var}(u)$,

$$u\sigma\{t \mapsto v\} = u\{t \mapsto v\}\sigma'$$

Preuve: Soit $F_1 = \{x_1, \ldots, x_n\}$, $F_2 = \{y_1, \ldots, y_m\}$, $F_3 = \{z_1, \ldots, z_s\}$ et C comme dans le point 2 du lemme. Par la proposition 1, le lemme 4 et le lemme 7, ils existent $\zeta_1, \ldots, \zeta_n, \chi_1, \ldots, \chi_m, \pi_1, \ldots, \pi_s$ tels que

- pour $C' = C[w_1, \ldots, w_l, \zeta_1, \ldots, \zeta_n, \chi_1, \ldots, \chi_m, \pi_1, \ldots, \pi_s]$, on a $u\sigma = C'[\mathsf{Fact}(u\sigma)]$, i.e. C' est le contexte qui définit les facteurs de $u\sigma$
- pour tout $1 \leq i \leq n$, on a $x_i\sigma = \zeta_i[\mathsf{Fact}(x_i\sigma)]$
- pour tout $1 \leq i \leq m$, on a $y_i\sigma = \chi_i[t]$, avec $\chi_i \in \overline{F}(\mathcal{X})$
- pour tout $1 \leq i \leq s$, on a $z_i\sigma = \pi_i[t']$, avec $t' \neq t$ et $\pi_i \in \overline{F}(\mathcal{X})$

On conclut le premier point, par la définition de SSt_f, et le deuxième point, par la définition du remplacement. Le troisième point est un cas particulier du deuxième. □

Exemple 24 *Soit* $u = x \bullet h(y) \bullet (z + b)$.

1. *Si* $x\sigma = a + b$, *on a* $\mathsf{SSt}_f(u\sigma, a + b) = true$, *grâce à* $\mathsf{SSt}_f(u, x) = true$. *Si* $z\sigma = a$, *on a* $\mathsf{SSt}_f(u\sigma, a + b) = true$, *grâce à* $\mathsf{SSt}_f(u, z + a) = true$. *Si* $x\sigma = a \bullet b$ *et* $y\sigma = a \bullet b$, *on a* $\mathsf{SSt}_f(u\sigma, a \bullet b) = false$, *car* $top(x\sigma) \in \mathsf{th}_f(u, x)$ *et* $\mathsf{SSt}_f(u, y) = false$.
2. - *Si* $t = a + b$, $x\sigma = h(a + b)$, $y\sigma = a + b$ *et* $z\sigma = a$, *on a* $x, y \in F_1$ *et* $u\sigma\{t \mapsto v\} = h(a+b) \bullet h(a+b) \bullet c$, *car* $h(a+b)\{a+b \mapsto c\} = h(a+b)$.
   - *Si* $t = a \star b$, $x\sigma = h(a \star b)$, $z\sigma = a \star b$, *on a* $x, y \in F_2$ *et* $u\sigma\{t \mapsto v\} = h(c) \bullet h(c) \bullet (c + b)$

6.2.2 Facteurs et réécriture

Dans cette sous-section nous allons montrer que, dans la forme normale d'un terme t (par rapport à un système de réécriture pur),

- ou bien l'ensemble des facteurs est inclus dans l'ensemble des formes normales de facteurs de t, ou bien on change de théorie lors de la réduction en rentrant dans un des facteurs de t (proposition 2).
- l'ensemble des sous-termes stricts est inclus dans les sous-termes des formes normales de facteurs de t (corollaire 2)

Pour cela, nous allons d'abord montrer deux lemmes préliminaires concernant l'instance d'une règle (lemme 9) et son application dans un pas de réécriture (lemme 10).

Le lemme suivant montre que, ou bien une instance d'une règle préserve la théorie et les facteurs, ou bien le changement de la théorie a lieu à travers une règle dont le membre droit est une variable et son instance un facteur de l'instance du membre gauche. Le deuxième item dans le premier point nous dit que les facteurs qui ne sont pas déjà à la surface de $l\theta$ ne peuvent pas surgir à la surface de $r\theta$.

Lemme 9 *Soit $\mathcal{R}$ un système de réécriture pur. Pour toute règle $l \to r \in \mathcal{R}$ et toute substitution θ, on a*

- *Si $\top(r\theta) = \top(l\theta)$, alors*
  - $\mathsf{Fact}(r\theta) \subseteq \mathsf{Fact}(l\theta) \cup \mathcal{C}_\mathcal{R}$ *et*
  - $\forall t \in \mathsf{Fact}(r\theta) \setminus \mathcal{C}_\mathcal{R}.\ \mathsf{SSt}_f(r\theta, t) \Rightarrow \mathsf{SSt}_f(l\theta, t)$
- *Si $\top(r\theta) \neq \top(l\theta)$, alors*
  - $r \in \mathcal{X}$ *et*
  - $\exists t \in \mathsf{Fact}(l\theta).\ \mathsf{Fact}_{\mathsf{th}(l,r)}(r\theta) = \{t\}\ \&\ \mathsf{SSt}(r\theta, t) \Rightarrow \mathsf{SSt}_f(l\theta, t)$
- $\top(l\theta) = \top(l)$ *et, si $r \notin \mathcal{X}$, $\top(r\theta) = \top(r)$*

Preuve: On commence par prouver le dernier point. On a $\top(l\theta) \neq \top(l)$, seulement si $l = \mathsf{h}(x)$. Par la définition des systèmes purs, c'est impossible. De même, si $r = \mathsf{h}(x)$, puisque $\top(l) = \top(r) = \mathsf{h}$, on aurait $l = \mathsf{h}(l')$ et donc r est une constante - contradiction. Avant de prouver les deux premiers points, notons aussi que, pour tout $x \in \mathsf{Var}(l)$, $\mathsf{th}(l, x) = \mathsf{th}_f(l, x)$ et $\mathsf{th}(r, x) = \mathsf{th}_f(r, x)$, grâce au fait que l et r sont des termes purs.

Admettons maintenant que $\top(r\theta) = \top(l\theta)$. Soit $\mathsf{Var}(r) = \{x_1, \ldots, x_n\}$. Par le lemme 7, puisque $\forall i.|\mathsf{th}(r, x_i)| = 1$ et r est pur, on a $\mathsf{Fact}(r\theta) = \mathsf{Fact}_{\mathsf{th}(r,x_1)}(x_1\theta) \cup \ldots \cup \mathsf{Fact}_{\mathsf{th}(r,x_n)}(x_n\theta) \cup (\mathsf{Fact}(r) \cap \mathcal{C}_\mathcal{R})$. D'autre part, $\mathsf{Var}(l) = \{x_1, \ldots, x_n, y_1, \ldots, y_m\}$, pour certaines variables $y_1, \ldots, y_m$. Comme pour r, grâce au fait que $\mathcal{R}$ est pur et $\forall i.\mathsf{th}(l, x_i) = \mathsf{th}(r, x_i)$, on a $\mathsf{Fact}(l\theta) = \mathsf{Fact}_{\mathsf{th}(r,x_1)}(x_1\theta) \cup \ldots \cup \mathsf{Fact}_{\mathsf{th}(r,x_n)}(x_n\theta) \cup M_1 \cup \ldots \cup M_m \cup (\mathsf{Fact}(l) \cap \mathcal{C}_\mathcal{R})$, pour certains ensembles $M_1, \ldots, M_m$. On conclut donc $\mathsf{Fact}(r\theta) \subseteq \mathsf{Fact}(l\theta) \cup \mathcal{C}_\mathcal{R}$.

Pour la deuxième partie du premier cas, on utilise le lemme 8. Fixons un $t \in \mathsf{Fact}(r\theta) \setminus \mathcal{C}_\mathcal{R}$. Soit $F_1^r, F_2^r \subseteq \mathsf{Var}(r)$ et, respectivement, $F_1^l, F_2^l \subseteq \mathsf{Var}(l)$ les ensembles des variables donnés par le lemme 8 appliqué à r, θ, t et, respectivement, l, θ, t. Par la définition des systemes purs, on a $F_1^r \subseteq F_1^l$ et $F_2^r \subseteq F_2^l$. Puisque $\mathsf{Fact}(r) \setminus \mathsf{Var}(r), \mathsf{Fact}(l) \setminus \mathsf{Var}(l) \subseteq \mathcal{C}_\mathcal{R}$, on déduit par le lemme 8 et par le fait que $\mathcal{R}$ est pur (qui nous donne $\forall x \in \mathsf{Var}(r).\mathsf{SSt}(r, x) \Rightarrow \mathsf{SSt}(l, x)$, $\mathsf{SSt}_f(r, x) = \mathsf{SSt}(r, x), \mathsf{SSt}_f(l, x) = \mathsf{SSt}(l, x)$), que

$$\begin{array}{rcl} \mathsf{SSt}_f(r\theta, t) & = & \bigvee_{x \in F_1^r}(\mathsf{SSt}(r, x) \wedge \mathsf{SSt}_f(x\theta, t)) \vee \bigvee_{x \in F_2^r}(\mathsf{SSt}(r, x) \wedge \mathsf{SSt}(x\theta, t)) \\ & \Rightarrow & \bigvee_{x \in F_1^l}(\mathsf{SSt}(l, x) \wedge C\mathcal{I}_f(x\theta, t)) \vee \bigvee_{x \in F_2^l}(\mathsf{SSt}(l, x) \wedge \mathsf{SSt}(x\theta, t)) \\ & = & \mathsf{SSt}_f(l\theta, t) \end{array}$$

Supposons maintenant $\top(r\theta) \neq \top(l\theta)$. Si, par absurde, $r \notin \mathcal{X}$, on a $\top(l\theta) = \top(l) = \top(r) = \top(r\theta)$ - contradiction. On a donc que $r \in \mathsf{Var}(l)$. De plus, $\top(r\theta) \notin \{\top(l)\} = \mathsf{th}(l, r)$. Par le lemme 7, on déduit qu'il existe un t tel que $\mathsf{Fact}_{\mathsf{th}(l,r)}(r\theta) = \{t\}$ et $t \in \mathsf{Fact}(l\theta)$. De plus, si $\mathsf{SSt}(r\theta, t) = \mathit{true}$, on a $r\theta = t$. Puisque $\mathsf{SSt}(r, r) = \mathit{true}$, on a, par la définition des systèmes purs, $\mathsf{SSt}(l, r) = \mathit{true}$. Par le lemme 8, on obtient $\mathsf{SSt}(l, r) \wedge \mathsf{SSt}(r\theta, t) \Rightarrow \mathsf{SSt}_f(l\theta, t)$ et on conclut le deuxième point. □

Nous relevons maintenant le lemme précédent à un pas de réécriture:

Lemme 10 *Pour tout terme t, avec $\mathsf{Fact}(t)$ en forme normale, et terme t' tel que $t \to t'$ on a*

- *ou bien*
  - $\mathsf{Fact}(t') \subseteq \mathsf{Fact}(t) \cup \mathcal{C}_{\mathcal{R}}$ *et*
  - $\forall u \in \mathsf{Fact}(t') \setminus \mathcal{C}_{\mathcal{R}}.\ \mathsf{SSt}_f(t', u) \Rightarrow \mathsf{SSt}_f(t, u)$
- *ou bien* $\exists u \in \mathsf{Fact}(t).\ t' \in \overline{\overline{F}}(u)$. *De plus, si* $t' = u$, $\mathsf{SSt}_f(t, u)$.

où $\mathcal{C}_{\mathcal{R}}$ est l'ensemble des constantes de $\mathcal{R}$ en forme normale.
De plus, si $\top(t') \neq \top(t)$, on est dans le deuxième cas du lemme.

Preuve: Soit $l \to r$ la règle de réécriture appliqée, i.e. il existe un contexte $C[x]$ et une susbtitution θ, tels que $t = C[l\theta]$ et $t' = C[r\theta]$. Si $C = x$, on conclut par le lemme 9. Sinon, par le lemme 7, on a $\mathsf{Fact}(t) = (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \mathsf{Fact}_{\mathsf{th}(C,x)}(l\theta)$. On fait quelques disjonctions de cas:

- $\top(l\theta) \in \mathsf{th}(C, x)$. Dans ce cas, $\mathsf{Fact}(t) = (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \mathsf{Fact}(l\theta)$. On a deux cas:
  - Si $\top(r\theta) = \top(l\theta)$, on a évidemment $\top(t') = \top(t)$. Par le lemme 7, on déduit

    $$\begin{array}{rcl} \mathsf{Fact}(t') & = & (\mathsf{Fact}(C)\backslash\{x\})\theta \cup \mathsf{Fact}(r\theta) \\ & \subseteq & (\mathsf{Fact}(C)\backslash\{x\})\theta \cup \mathsf{Fact}(l\theta) \ = \ \mathsf{Fact}(t) \end{array}$$

    en utilisant le lemme 9 pour l'inclusion.

    Il nous reste à montrer la deuxième propriété pour le premier cas du lemme. En utilisant le lemme 8 et le lemme 9, pour tout $u \in \mathsf{Fact}(t') \setminus \mathcal{C}_{\mathcal{R}}$, on a

    $$\begin{array}{rcll} \mathsf{SSt}_f(C[r\theta], u) & = & \mathsf{SSt}_f(C, u) \vee (\mathsf{SSt}_f(C, x) \wedge \mathsf{SSt}_f(r\theta, u)) & \text{lemme 8} \\ & \Rightarrow & \mathsf{SSt}_f(C, u) \vee (\mathsf{SSt}_f(C, x) \wedge \mathsf{SSt}_f(l\theta, u)) & \text{lemme 9} \\ & = & \mathsf{SSt}_f(C[l\theta], u) & \text{lemme 8} \end{array}$$

    d'où on conclut la propriété voulue.
  - Si $\top(r\theta) \neq \top(l\theta)$, en utilisant le lemme 9, on a $r \in \mathsf{Var}(l)$ et $r\theta \in \overline{F}(\mathsf{Fact}(l\theta))$. On considère deux cas:
    - $C \notin \overline{F}(\mathcal{X})$. Alors $\top(t') = \top(t)$. De plus, par l'injectivité de H et $\mathsf{thsource}(\mathsf{h}) \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$, on a $|\mathsf{th}(C, x)| = 1$ et donc $\mathsf{th}(C, x) = \{\top(l\theta)\}$. Par la définition des systèmes purs, on a $\top(l\theta) = \{\top(l)\} = \mathsf{th}(l, r)$ et on déduit $\mathsf{th}(C, x) = \mathsf{th}(l, r)$. D'où on a

      $$\begin{array}{rcll} \mathsf{Fact}(C[r\theta]) & = & (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \mathsf{Fact}_{\mathsf{th}(C,x)}(r\theta) & \text{lemme 8} \\ & = & (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \mathsf{Fact}_{\mathsf{th}(l,r)}(r\theta) & \text{puisque } \mathsf{th}(C, x) = \mathsf{th}(l, r) \\ & \subseteq & (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \mathsf{Fact}(l\theta) & \text{lemme 8} \\ & = & \mathsf{Fact}(C[l\theta]) & \end{array}$$

Il nous reste à prouver la propriété supplémentaire pour le premier cas de l'énoncé. Notons qu'on a $\top(r\theta) \notin \mathsf{th}(l,r) = \mathsf{th}(C[x],x)$, car $\{\top(r\theta)\} \neq \{\top(l\theta)\} = \{\top(l)\} = \mathsf{th}(l,r)$. Pour tout $u \in \mathsf{Fact}(C[r\theta]) \setminus \mathcal{C}_\mathcal{R}$, on a donc

$$\begin{array}{lcll} \mathsf{SSt}_f(C[r\theta],u) & = & \mathsf{SSt}_f(C,u) \vee (\mathsf{SSt}_f(C,x) \wedge \mathsf{SSt}(r\theta,u)) & \text{lemme 8} \\ & \Rightarrow & \mathsf{SSt}_f(C,u) \vee (\mathsf{SSt}_f(C,x) \wedge \mathsf{SSt}_f(l\theta,u)) & \text{lemme 9} \\ & = & \mathsf{SSt}_f(C[l\theta],u) & \text{lemme 8} \end{array}$$

* $C \in \overline{F}(\mathcal{X})$. Dans ce cas, en utilisant l'injectivité de H, on a
  - ou bien $\top(t') \neq \top(t)$: on doit être dans le deuxième cas de l'énoncé. En effet, comme dans le cas précédent, on a $\exists u.\ \mathsf{Fact}_{\mathsf{th}(l,r)}(r\theta) = \{u\} \subseteq \mathsf{Fact}(l\theta) \subseteq \mathsf{Fact}(C[l\theta])$. Donc, en utilisant aussi le lemme 4 et les définitions de $\overline{F}, \overline{G}$, on a $t' \in \overline{\overline{F}}(u)$, pour un $u \in \mathsf{Fact}(t)$. De plus, si $t' = u$, on a $C \in \mathcal{X}$ &$r\theta = u$ et, puisque $\mathsf{SSt}(r,r) = true$ et $\mathsf{SSt}(r,r) \Rightarrow \mathsf{SSt}(l,r)$, on obtient $\mathsf{SSt}_f(t,u) = true$.
  - ou bien $t = \mathsf{h}(l\theta)$, $t' = \mathsf{h}(r\theta)$ et $\top(l\theta) \notin \mathsf{thsource}(\mathsf{h})$. Alors on a $l\theta \in \mathsf{Fact}(t)$, en contradiction avec le fait que les facteurs de t sont en forme normale. Ce cas est donc impossible.

- $\top(l\theta) \notin \mathsf{th}(C,x)$. Dans ce cas, par le lemme 7, on a $\mathsf{Fact}(t) = (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \{w\}$, avec $w \in \overline{G}(l\theta)$. Donc $l\theta \in \overline{F}(\mathsf{Fact}(t))$. Puisque tous les facteurs de t sont en forme normale, on a $w \neq l\theta$. Donc $top(l) = \mathsf{h}$ et $l\theta = \mathsf{h}(w)$. Par la définition des systèmes purs, ça implique que r est une constante en forme normale. En particulier, r n'est pas une variable. On a donc $\top(r) = \top(l) = \top(l\theta)$ et on conclut:

$$\mathsf{Fact}(t') = (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \{r\} \subseteq \mathsf{Fact}(t) \cup \mathcal{C}_\mathcal{R}$$

  De plus, $\top(t') = \top(t)$. Il nous reste donc a verifier la deuxième propriété pour le premier cas de l'ennoncé du lemme. On a $\mathsf{Fact}(t') \setminus \mathcal{C}_\mathcal{R} \subseteq (\mathsf{Fact}(C) \setminus \{x\})\theta$ et donc $\forall u \in \mathsf{Fact}(t') \setminus \mathcal{C}_\mathcal{R}.\mathsf{SSt}_f(t',u) \Rightarrow \mathsf{SSt}_f(C,u) \Rightarrow \mathsf{SSt}_f(t,u)$, en utilisant le lemme 8.

□

Exemple 25 *Dans cet exemple, nous voyons comment le lemme précédent fonctionne itérativement. Pour* $\mathcal{R}_{\mathsf{EP}}$*,*

- *si* $t = exp(h(a \star b), J_\star(b) \star c)$*, on a* $t \to^* h(a \star c)$ *et* $\mathsf{Fact}(h(a \star c)) = \{a,c\} \subseteq \mathsf{Fact}(t)$*.*
- *si* $t = exp(h(a \star b), J_\star(b))$*, on a* $t \to^* h(a) \in \overline{F}(\mathsf{Fact}(t))$*.*
- *nous n'avons pas d'exemple pour le cas* $t' \in \overline{\overline{F}}(\mathsf{Fact}(t)) \setminus \overline{F}(\mathsf{Fact}(t))$*. Pour garder la classe de systèmes purs simple, nous avons preferé de payer le prix de la double barre, même si elle n'est pas nécessaire pour l'étude de cas.*

Lemme 11 *Pour tout terme t en forme normale, on a $\overline{F}(t)\downarrow \subseteq \overline{F}(t) \cup \mathcal{C}_{\mathcal{R}}$.*

Preuve: Par la définition des systèmes purs,

- Si $t \notin \mathcal{C}_{\mathcal{R}}$, $\mathsf{h}(t)$ est en forme normale, donc $\overline{F}(t)\downarrow = \overline{F}(t)$
- Si $t \in \mathcal{C}_{\mathcal{R}}$, $\mathsf{h}(t)\downarrow \in \{\mathsf{h}(t)\} \cup \mathcal{C}_{\mathcal{R}} \subseteq \overline{F}(t) \cup \mathcal{C}_{\mathcal{R}}$. □

Dans la proposition suivante nous caractérisons l'interaction des théories lors de la normalisation d'un terme: ou bien la théorie ne change pas, et les facteurs de la forme normale sont les mêmes que les facteurs avant la normalisation, ou bien on change de théorie, en obtenant un facteur et, eventuellement, le décomposant pour obtenir la forme normale.

Proposition 2 *Pour tout terme $v = \zeta[t_1, \ldots, t_n]$, avec $\mathsf{Fact}(v) = \{t_1, \ldots, t_n\}$, on a:*

- *ou bien* $\mathsf{Fact}(v\downarrow) \subseteq \mathsf{Fact}(\zeta[t_1\downarrow, \ldots, t_n\downarrow]) \cup \mathcal{C}_{\mathcal{R}}$
- *ou bien il existe un s, $1 \leq s \leq n$, tel que*
  - *ou bien* $v\downarrow \in \overline{\overline{F}}(t_s\downarrow)$
  - *ou bien* $v\downarrow \in \overline{\overline{F}}(\mathsf{Fact}(t_s\downarrow))$ & $\top(t_s\downarrow) \neq \top(t_s)$ & $top(t_s\downarrow) \neq top(t_s)$.
- *ou bien* $v\downarrow \in \overline{\overline{F}}(\mathcal{C}_{\mathcal{R}})$

De plus, si $\top(v\downarrow) \neq \top(v)$, on est toujours dans les deux derniers cas.

Preuve: Notons d'abord que, en utilisant la proposition 1, on a $\mathsf{Fact}(\zeta[t_1\downarrow, \ldots, t_n\downarrow]) = F_1 \cup \ldots \cup F_n$, où, pour tout $1 \leq i \leq n$

- $\top(t_i\downarrow) \in \mathsf{th}(\zeta, x_i) \implies F_i = \mathsf{Fact}(t_i\downarrow)$
- $\top(t_i\downarrow) \notin \mathsf{th}(\zeta, x_i) \implies F_i = \{u_i\} \subseteq \overline{G}(t_i\downarrow)$

Puisque, par la définition des facteurs, pour tout $1 \leq i \leq n$, on a $\top(t_i) \notin \mathsf{th}(\zeta, x_i)$, on déduit que $F_i \not\subseteq \overline{G}(t_i\downarrow)$ implique $\top(t_i\downarrow) \neq \top(t_i)$ et, en utilisant en plus le fait que $\mathsf{Fact}_{th(\zeta,x_i)}(t_i) = \{t_i\}$ et le lemme 4, aussi $top(t_i\downarrow) \neq top(t_i)$.

Pour chaque i tel que $F_i = \{u_i\} \subseteq \overline{G}(t_i\downarrow)$, écrivons $t_i\downarrow = C_i[u_i]$, avec $C_i \in \overline{F}(\mathcal{X})$.

Soit $v_0 = \zeta[t_1\downarrow, \ldots, t_n\downarrow]$. Considérons une dérivation $v_0 \rightarrow^* v_1 \rightarrow^* v\downarrow$ telle que chaque étape de réduction de $v_0 \rightarrow^* v_1$ satisfait les conditions du premier cas du lemme 10, tandis que la première étape de réduction de $v_1 \rightarrow^* v\downarrow$ satisfait le deuxième cas du lemme 10.

Notons que $\mathsf{Fact}(v_0) \cup \mathcal{C}_{\mathcal{R}}$ est un ensemble de termes en forme normale, par la proposition 1 et le lemme 6. Donc, par une simple récurrence sur la longueur de $v_0 \rightarrow^* v_1$ et le lemme 10, on a que $\mathsf{Fact}(v_1) \subseteq \mathsf{Fact}(v_0) \cup \mathcal{C}_{\mathcal{R}}$. Si la dérivation $v_1 \rightarrow^* v\downarrow$ est vide, on conclut le lemme. Sinon, soit $\mathsf{Fact}(v_0) = G \uplus B$, où $G = \{u \mid \exists s.F_s \not\subseteq \overline{G}(t_s\downarrow)$ & $u \in F_s\}$ et $B = \mathsf{Fact}(v_0) \setminus G$. Par le lemme

10 appliqué à v_1, on a $v_1 \to v' \in \overline{\overline{F}}(\mathsf{Fact}(v_1))$ et donc $v{\downarrow} \in \overline{\overline{F}}(\mathsf{Fact}(v_1)){\downarrow} \subseteq \overline{\overline{F}}(\mathsf{Fact}(v_0) \cup \mathcal{C}_\mathcal{R}){\downarrow} \subseteq \overline{\overline{F}}(\mathsf{Fact}(v_0) \cup \mathcal{C}_\mathcal{R})$, en utilisant le lemme 11 pour la dernière inclusion. Soit $t \in \mathsf{Fact}(v_0) \cup \mathcal{C}_\mathcal{R}$ tel que $v{\downarrow} \in \overline{\overline{F}}(t)$. Si on est dans l'un des cas suivants, on conclut:

- $t \in G \cup \mathcal{C}_\mathcal{R}$
- $t \in \{t_1{\downarrow}, \ldots, t_n{\downarrow}\}$
- $v{\downarrow} \in \overline{\overline{F}}(t) \setminus \{t\}$

Le cas problématique est quand $v{\downarrow} = t \in \overline{G}(\{t_1{\downarrow}, \ldots, t_n{\downarrow}\}) \setminus (\{t_1{\downarrow}, \ldots, t_n{\downarrow}\} \cup G \cup \mathcal{C}_\mathcal{R})$. On montre que ce cas n'est pas possible. En effet, par récurrence sur la longueur de $v_0 \to^* v_1$ et le lemme 10, on a d'abord $\mathsf{SSt}_f(v_1, t) \Rightarrow \mathsf{SSt}_f(v_0, t)$. D'autre part, par le deuxième cas de ce lemme appliqué a v_1 et puisque $v_1{\downarrow} = t$, on a $\mathsf{SSt}_f(v_1, t) = \mathit{true}$. On obtient donc $\mathsf{SSt}_f(v_0, t) = \mathit{true}$. D'autre part, si $t \in \overline{G}(\{t_1{\downarrow}, \ldots, t_n{\downarrow}\}) \setminus (\{t_1{\downarrow}, \ldots, t_n{\downarrow}\} \cup G \cup \mathcal{C}_\mathcal{R})$, chaque occurence de t dans $\mathsf{Fact}(v_0)$ et dans un t_i tel que $t_i{\downarrow} = \mathsf{h}(t)$. Par définition, ça implique $\mathsf{SSt}_f(v_0, t) = \mathit{false}$ - contradiction.

De plus, si $\top(v{\downarrow}) \neq \top(v)$, on est dans les deux derniers cas, par le lemme 10. □

Exemple 26 *Pour* EP*, on a*

- *Si* $v = exp(h((a + b) \star J_\star(c)), c \star d)$, $\mathsf{Fact}(v) = \{a + b, c, d\}$ *et*

$$\mathsf{Fact}(v{\downarrow}) = \{a + b, d\} \subseteq \mathsf{Fact}(v)$$

- *Si* $v = exp(h((a+b+J_+(b)) \star J_\star(c)), c)$, $\mathsf{Fact}(v) = \{a+b+J_+(b), J_\star(c), c\}$ *et*

$$v{\downarrow} = h(a) \in \overline{F}((a + b + J_+(b){\downarrow})$$

- *Si* $v = (a \star b + b + J_+(b)) \star J_\star(b)$, $\mathsf{Fact}(v) = \{a \star b + b + J_+(b), J_\star(b)\}$ *et*

$$v{\downarrow} = a \in \mathsf{Fact}((a \star b + b + J_+(b)){\downarrow})$$

  avec $\top(a \star b + b + J_+(b))$ *et* $\top(a \star b + b + J_+(b))$ *qui changent par normalisation.*

Pour un terme t, on va noter par $ld(t)$ la longueur d'une normalisation innermost de t de longueur minimale. Quand la théorie d'un terme t change, nous montrons que soit la théorie de l'un des ses facteurs change, soit il est egal à l'un des ses facteurs u tel que $ld(u) < ld(t)$:

Corollaire 1 (de la preuve) *Pour tout terme* $v = \zeta[t_1, \ldots, t_n]$*, avec* $\mathsf{Fact}(v) = \{t_1, \ldots, t_n\}$ *et* $\top(v{\downarrow}) \neq \top(v)$*, ou bien* $v{\downarrow} \in \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$*, ou bien il existe un* i *tel que*

- *ou bien* $\top(t_i{\downarrow}) \neq \top(t_i)$ *&* $top(t_i{\downarrow}) \neq top(t_i)$

- *ou bien* $\exists w \in \overline{\overline{F}}(t_i).\ v{\downarrow} = w{\downarrow}\ \&\ ld(w) < ld(v)$

Preuve: Supposons que $v{\downarrow} \notin \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$ et, pour tout $1 \leq i \leq n$, $\top(t_i{\downarrow}) = \top(t_i) \vee top(t_i{\downarrow}) = top(t_i)$. Considérons à nouveau la derivation $v_0 \rightarrow^* v_1 \rightarrow^* v{\downarrow}$ définie dans la preuve de la proposition 2. Puisque $\top(v{\downarrow}) \neq \top(v)$, on a $v_1 \rightarrow^+ v{\downarrow}$. Comme on a vu, on a $v_1{\downarrow} = C[C_s[t]]$, pour des contextes C, C_s tels que $C[C_s] \in \overline{\overline{F}}(\mathcal{X})$ et un t tel que $C_s[t] = t_s{\downarrow}$, pour un $1 \leq s \leq n$. On peut donc choisir $i = s$ et $w = C[t_s]$, car $ld(C[t_s]) = ld(t_s) < ld(v)$, où on utilise $v{\downarrow} \notin \mathcal{C}_\mathcal{R}$, pour avoir $C[t_s{\downarrow}] = C[t_s]{\downarrow}$, et $v_1 \rightarrow^+ v{\downarrow}$, pour l'inégalité. □

Lemme 12 *Pour tout terme t,*

1. $\mathsf{St}(\overline{F}(t)) \subseteq \overline{F}(t) \cup \mathsf{St}(t) \subseteq \overline{F}(\mathsf{St}(t))$
2. $\mathsf{St}(\overline{G}(t)) \subseteq \overline{G}(t) \cup \mathsf{St}(t) \subseteq \overline{G}(\mathsf{St}(t))$
3. $\overline{G}(\overline{F}(t)) \subseteq \overline{F}(\overline{G}(t))$ *et* $G(\overline{F}(t)) \subseteq \overline{F}(G(t))$
4. $\overline{G}(\overline{G}(t)) \subseteq \overline{G}(\mathsf{St}(t))$
5. $\mathsf{St}_<(\overline{F}(t)) \subseteq \mathsf{St}(t)$, *où* $\mathsf{St}_<(t) = \mathsf{St}(t) \setminus \{t\}$

Preuve:

1. $\mathsf{St}(\overline{F}(t)) = \mathsf{St}(t) \cup \mathsf{St}(\mathsf{h}(t)) = \mathsf{St}(t) \cup \{\mathsf{h}(t)\} \cup \mathsf{St}(\mathsf{Fact}(\mathsf{h}(t))) \subseteq \mathsf{St}(t) \cup \{\mathsf{h}(t)\} \cup \mathsf{St}(\mathsf{Fact}_{\mathsf{thsource}(\mathsf{h})}(t))$. En utilisant $\mathsf{thsource}(\mathsf{h}) \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$, on déduit $\mathsf{Fact}_{\mathsf{thsource}(\mathsf{h})}(t) \subseteq \mathsf{Fact}(t) \cup \{t\}$ et on conclut $\mathsf{St}(\overline{F}(t)) \subseteq \overline{F}(t) \cup \mathsf{St}(t)$.

2. Si $\overline{G}(t) = \{t\}$ le résultat est immédiat. Si $t = \mathsf{h}(t')$, on a $\mathsf{St}(\overline{G}(t)) = \mathsf{St}(t) \cup \mathsf{St}(t') = \mathsf{St}(t) \cup \{t'\} \cup \mathsf{St}(\mathsf{Fact}(t'))$. Pour conclure, il suffit de montrer que $\mathsf{Fact}(t') \subseteq \mathsf{St}(t)$. Si $t' \in \mathsf{St}(t)$, c'est immédiat. Sinon, on a $\top(t') \in \mathsf{thsource}(\mathsf{h})$ et donc $\mathsf{Fact}(\mathsf{h}(t')) = \mathsf{Fact}(t')$, par la proposition 1.

3. Cette propriété est immédiate par définition.

4. Cette propriété n'est pas évidente seulement si $t = \mathsf{h}(\mathsf{h}(t'))$. Dans ce cas, il suffit de remarquer que $\mathsf{Fact}(t) = \{\mathsf{h}(t')\}$, en utilisant la proposition 1 et le fait que $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{thsource}(\mathsf{h}) = \emptyset$.

5. On a $\mathsf{St}_<(\overline{F}(t)) = \mathsf{St}(\mathsf{Fact}(\overline{F}(t)))$. Comme vu dans le premier point, $\mathsf{Fact}(\overline{F}(t)) \subseteq \mathsf{Fact}(t) \cup \{t\}$. On conclut donc $\mathsf{St}_<(\overline{F}(t)) \subseteq \mathsf{St}(t)$.

□

Les sous-termes d'un terme v sont, par définition, v en union avec les sous-termes de ses facteurs. Au fait, grâce à nos relations de compatibilité entre sous-termes et réécriture, ç'est une propriété qui reste vraie après la normalisation de v, modulo les éventuelles descentes dans $\overline{G}$:

Corollaire 2 *Pour tout terme $v = \zeta[t_1, \ldots, t_n]$, avec $\mathsf{Fact}(v) = \{t_1, \ldots, t_n\}$,*

$$\mathsf{St}(v\downarrow) \subseteq \overline{G}(v\downarrow) \cup \mathsf{St}(t_1\downarrow, \ldots, t_n\downarrow) \cup \overline{G}(t_1\downarrow, \ldots, t_n\downarrow) \cup \mathcal{C}_{\mathcal{R}}$$

Preuve: Par définition, $\mathsf{St}(v\downarrow) = \{v\downarrow\} \cup \mathsf{St}(\mathsf{Fact}(v\downarrow))$. Par les propositions 2 et 1, on a

- ou bien $\mathsf{Fact}(v\downarrow) \subseteq \mathsf{Fact}(\zeta[t_1\downarrow, \ldots, t_n\downarrow]) \cup \mathcal{C}_{\mathcal{R}} \subseteq \mathsf{Fact}(t_1\downarrow, \ldots, t_n\downarrow) \cup \overline{G}(t_1\downarrow, \ldots, t_n\downarrow) \cup \mathcal{C}_{\mathcal{R}}$ et alors $\mathsf{St}(v\downarrow) \subseteq \{v\downarrow\} \cup \mathsf{St}(t_1\downarrow, \ldots, t_n\downarrow) \cup \overline{G}(t_1\downarrow, \ldots, t_n\downarrow) \cup \mathcal{C}_{\mathcal{R}}$, par définitions et le lemme 12.
- ou bien il existe un s, $1 \leq s \leq n$, tel que
  - ou bien $v\downarrow \in \overline{\overline{F}}(t_s\downarrow)$. Par le lemme 12, on a $\mathsf{St}(\overline{\overline{F}}(t_s\downarrow)) \subseteq \overline{\overline{F}}(t_s\downarrow) \cup \mathsf{St}(t_s\downarrow)$. On conclut, car $\overline{\overline{F}}(t_s\downarrow) \subseteq \overline{G}(v\downarrow) \cup \{t_s\downarrow\}$.
  - ou bien $v\downarrow \in \overline{\overline{F}}(\mathsf{Fact}(t_s\downarrow))$. Ce cas est similaire du précédent.
- ou bien $v\downarrow \in \overline{\overline{F}}(\mathcal{C}_{\mathcal{R}})$ et on conclut immédiatement.

□

Exemple 27 *Donnons un exemple qui montre la nécessité de $\overline{G}$ pour* EP*. Prenons $v = exp(h(a), b) \bullet c$, avec $\mathsf{Fact}(v) = \{exp(h(a), b), c\}$. On a $v\downarrow = h(a \star b) \bullet c$ et $\mathsf{St}(v\downarrow) = \{v\downarrow\} \cup \{a \star b, a, b, c\}$. On a $a \star b \in \overline{G}(exp(h(a), b))\downarrow) \setminus \{exp(h(a), b))\downarrow\}$.*

6.2.3 Remplacement et réécriture

Notre dernier résultat préliminaire est la proposition 3, qui montre que la réécriture commute avec le remplacement de facteurs. Comme dans la sous-section précédente, nous considérons d'abord les cas d'une regle de réécriture (lemme 13) et d'un seul pas de réécriture (lemme 14).

Lemme 13 *Pour toute règle $l \to r$ d'un système de réécriture pur, toute substitution θ et tous termes en forme normale t, v, on a $(l\theta)\{t \mapsto v\} = l\theta'$ et, si $r \notin \mathcal{X}$, $(r\theta)\{t \mapsto v\} = r\theta'$, où, pour tout $x \in \mathsf{Var}(l)$,*

- *$x\theta' = x\theta\{t \mapsto v\}$, si*
  - *$\top(x\theta) \in \mathsf{th}(l, x)$ & $x\theta \neq t$ ou*
  - *$\top(x\theta) \notin \mathsf{th}(l, x)$ & $\mathsf{Fact}_{\mathsf{th}(l,x)}(x\theta) \neq \{t\}$*
- *$x\theta' = x\theta$, si $\top(x\theta) \in \mathsf{th}(l, x)$ & $x\theta = t$*
- *$x\theta' = \chi[v]$, si $\top(x\theta) \notin \mathsf{th}(l, x)$ & $\mathsf{Fact}_{\mathsf{th}(l,x)}(x\theta) = \{t\}$ & $x\theta = \chi[t]$*

Preuve: $l\theta\{t \mapsto v\} = l\theta'$ suit immédiatement par le deuxième point du lemme 8, car, pour tout x, $\mathsf{th}_f(l, x) = \mathsf{th}(l, x)$ et le système est pur. Si $r \notin \mathcal{X}$, $r\theta\{t \mapsto v\} = r\theta'$ suit de la même manière, car alors $\mathsf{th}(r, x) = \mathsf{th}(l, x)$, pour tout $x \in \mathsf{Var}(r)$. □

Quand on remplace un sous-terme t d'un terme u qui contient un radical, par la pureté du système, le remplacement peut se faire soit à coté, soit dans la partie substitution, soit plus haut que le radical. Si, en plus, le terme a tous ses facteurs en forme normale (ou si t est en forme normale), le dernier cas n'est pas possible: le remplacement peut alors commuter avec le pas de réduction. Sauf dans le cas special quand le résultat de la réduction est t (ou dans $F(t)$), car alors t peut être construit par la réduction et non pas obtenu d'une de ses occurences dans $\mathsf{St}(u)$:

Exemple 28 *Soit $t = a + b$.*

- *Pour $u = (a + b) \star b \star J_\star(b)$, on a $u \to t$ et $u\{t \mapsto c\} \to t\{t \mapsto c\} = c$.*
- *Pour $u = a + b + c + J_+(c)$, on a $u \to t$ et $u\{t \mapsto c\} \to t$.*
- *Pour $u = h(a) \bullet h(b)$, on a $u \to h(t)$ et $u\{t \mapsto c\} \to h(t)$. Cependant, dans ce cas, $h(t)\{t \mapsto c\} = h(t)$.*

Lemme 14 *Pour tout terme u, avec $\mathsf{Fact}(u)$ en forme normale, tout terme u', tel que $u \to u'$, et termes en forme normale $t \notin \mathcal{C}_\mathcal{R}, v$, on a*

- *ou bien $u' \notin F(t)$ et $u\{t \mapsto v\} \to u'\{t \mapsto v\}$*
- *ou bien $u' = \zeta[t] \in F(t)$ et $u\{t \mapsto v\} \to u'' \in \{u', \zeta[v]\}$*

Preuve: Notons que, si $u' = \zeta[t] \in F(t)$, $u'\{t \mapsto v\} \in \{u', \zeta[v]\}$.

Soit $C[x]$ et $l \to r \in \mathcal{R}$ tels que $u = C[l\theta]$ et $u' = C[r\theta]$. Si $C = x$, on conclut par le lemme 13. Sinon, on considère quelques cas.

- Cas $\top(l\theta) \in \mathsf{th}(C, x)$. Par le lemme 8 ($x$ a une seule occurrence dans C) et puisque t est en forme normale et donc $l\theta \neq t$, on obtient $C[l\theta]\{t \mapsto v\} = C\{t \mapsto v\}[l\theta\{t \mapsto v\}]$. Par le lemme 13, on a $l\theta\{t \mapsto v\} = l\theta'$ et, si $r \notin \mathcal{X}$, $r\theta\{t \mapsto v\} = r\theta'$, pour une substitution θ' definie dans l'énoncé du lemme. Si $r \notin \mathcal{X}$, on a aussi $\top(l\theta) = \top(r\theta)$, par le lemme 9, et, si $r\theta = t$, $r\theta' = r\theta$. Donc on conclut $C\{t \mapsto v\}[r\theta'] = C\{t \mapsto v\}[r\theta\{t \mapsto v\}] = C[r\theta]\{t \mapsto v\}$ et $u\{t \mapsto v\} \to u'\{t \mapsto v\}$.

  Si $r \in \mathsf{Var}(l)$, on considère deux cas:

  - $\top(r\theta) = \top(l\theta)$. Dans ce cas, on a $\top(r\theta) \in \mathsf{th}(l, r)$ et, par le lemme 13, $r\theta' = r\theta\{t \mapsto v\}$, si $r\theta \neq t$, et $r\theta' = r\theta$, si $r\theta = t$. D'autre part, par le lemme 8, on a $C[r\theta]\{t \mapsto v\} = C\{t \mapsto v\}[r\theta\{t \mapsto v\}]$, si $r\theta \neq t$, et $C[r\theta]\{t \mapsto v\} = C\{t \mapsto v\}[r\theta\{t \mapsto v\}]$, si $r\theta = t$. On conclut donc $u\{t \mapsto v\} \to u'\{t \mapsto v\}$.
  - $\top(r\theta) \neq \top(l\theta)$. Notons que, par le lemme 9, on a $r \in \mathsf{Var}(l)$ et $\top(l) = \top(l\theta)$. Par la définition des systèmes purs, on obtient $\top(r\theta) \notin \mathsf{th}(l, r)$. Par le lemme 13 et la definition de $\mathsf{Fact}_{th(l,r)}$, on a alors deux cas
    - Si $\mathsf{Fact}_{\mathsf{th}(l,r)}(r\theta) = \{t\}, r\theta = \chi[t] \in \overline{F}(t)$, on a $r\theta' = \chi[v]$.

· Si $C \neq \mathsf{h}(x)$, on a, par l'injectivité de H et $\mathsf{thsource}(\mathsf{h}) \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$, $|\mathsf{th}(C,x)| = 1$ et donc $\mathsf{th}(C,x) = \{l\theta\}$, $\top(r\theta) \notin \mathsf{th}(C,x)$. Ce qui nous donne $C[r\theta]\{t \mapsto v\} = C\{t \mapsto v\}[\chi[v]] = C\{t \mapsto v\}[r\theta']$. On a donc $u\{t \mapsto v\} \to u'\{t \mapsto v\}$.

· Si $C = \mathsf{h}(x)$ et $\top(r\theta) \notin \mathsf{th}(C,x)$, on conclut comme dans le cas precedent. Si $\top(r\theta) \in \mathsf{th}(C,x)$, on a $C[r\theta] \in F(r\theta)$, $top(r\theta) \neq \mathsf{h}$ et $r\theta = t$. On obtient donc $u' = \mathsf{h}(t) \in F(t)$ et $u\{t \mapsto v\} \to \mathsf{h}(v)$.

∗ Sinon, on a $r\theta' = r\theta\{t \mapsto v\}$

· Si $C \notin \overline{F}(\mathcal{X})$, on a $|\mathsf{th}(C,x)| = 1$ et donc $\top(r\theta) \notin \mathsf{th}(C,x)$. Ce qui nous donne $C[r\theta]\{t \mapsto v\} = C\{t \mapsto v\}[r\theta\{t \mapsto v\}] = C\{t \mapsto v\}[r\theta']$. On a donc $u\{t \mapsto v\} \to u'\{t \mapsto v\}$.

· Si $C = \mathsf{h}(x)$ et $\top(r\theta) \notin \mathsf{th}(C,x)$, on conclut comme dans le cas precedent. Si $\top(r\theta) \in \mathsf{th}(C,x)$ et $r\theta \neq t$, on a $u\{t \mapsto v\} \to C[r\theta'] = C[r\theta]\{t \mapsto v\} = u'\{t \mapsto v\}$. Si $r\theta = t$, on a $\mathsf{h}(r\theta) \in F(t)$ et $r\theta' = v$. Donc $u' = \mathsf{h}(t) \in F(t)$ et $u\{t \mapsto v\} \to \mathsf{h}(v)$.

- Cas $\top(l\theta) \notin \mathsf{th}(C,x)$. Dans ce cas, $\mathsf{Fact}(u) = (\mathsf{Fact}(C) \setminus \{x\})\theta \cup \{w\}$, avec $w \in \overline{G}(l\theta)$. Donc $l\theta \in \overline{F}(\mathsf{Fact}(u))$. Puisque tous les facteurs de u sont en forme normale, on a $w \neq l\theta$. Donc $top(l) = \mathsf{h}$. Par la définition des systèmes purs, ça implique que w et r sont des constantes et donc $t \notin \mathsf{St}(\{w,r\}) \cup \overline{G}(w,r)$. On a donc $C[l\theta]\{t \mapsto v\} = C\{t \mapsto v\}[l\theta]$ et $C[r\theta]\{t \mapsto v\} = C\{t \mapsto v\}[r\theta]$ et on conclut $u\{t \mapsto v\} \to u'\{t \mapsto v\}$.

□

Proposition 3 *Pour tout terme u, avec tous les facteurs en forme normale, et termes $t \notin \mathcal{C}_\mathcal{R}, v$ en forme normale*

- *ou bien $u\downarrow \notin F(t)$ et $u\{t \mapsto v\}\downarrow = u\downarrow\{t \mapsto v\}\downarrow$*
- *ou bien $u\downarrow = \chi[t] \in F(t)$ et $u\{t \mapsto v\}\downarrow \in \{u\downarrow, \chi[v]\downarrow\}$*

Preuve: Supposons d'abord que pour tout terme dans la derivation de u vers sa forme normale, tous les facteurs sont en forme normale. Des lors, on peut appliquer le lemme 14 à chaque pas. Notons que $F(t)$ est en forme normale, par la définition des systèmes purs. Donc, si la normalisation de u est $u \to^* u_1 \to u\downarrow$, dans chaque pas de la derivation $u \to^* u_1$ on est dans le premier cas du lemme 14. Par une simple recurrence, on déduit $u\{t \mapsto v\} \to^* u_1\{t \mapsto u\}$. Si dans le dernier pas $u_1 \to u\downarrow$ on est aussi dans le premier cas, on obtient $u\{t \mapsto v\} \to^* u\downarrow\{t \mapsto u\}$. Par convergence, on conclut $u\{t \mapsto v\}\downarrow = u\downarrow\{t \mapsto v\}\downarrow$.

Si dans $u_1 \to u\downarrow$ on est dans le deuxième cas du lemme 14, on obtient $u\{t \mapsto u\} \to^* u'' \in \{u\downarrow, \chi[v]\}$. Par convergence, on conclut $u\{t \mapsto v\}\downarrow \in \{u\downarrow, \chi[v]\downarrow\}$.

Soit maintenant u_1 le premier terme dans la normalisation de u qui n'a pas ses facteurs en forme normale. Par le lemme 10, on a $u_1 \in \overline{\overline{F}}(\mathsf{Fact}(u))$. Donc u_1 n'a pas ses facteurs en forme normale seulement si $u_1 = \mathsf{h}(\mathsf{h}(w))$,

avec $w \in \mathsf{Fact}(u)$, $\mathsf{Fact}(u_1) = \{\mathsf{h}(w)\}$ et $\mathsf{h}(w)$ n'est pas en forme normale. Ça implique $u_1 \notin F(t)$ et $t \notin \mathsf{St}(u_1)$. Puisque la derivation $u \to^* u_1$ est comme dans le cas précédent, on obtient $u\{t \mapsto v\} \to^* u_1\{t \mapsto v\} = u_1$. Comme, par la définition des systèmes purs, $u_1{\downarrow} \in \overline{F}(\mathcal{C}_{\mathcal{R}}) \setminus \overline{F}(t)$, on a $u_1{\downarrow}\{t \mapsto v\} = u_1{\downarrow}$. On conclut $u{\downarrow} \notin F(t)$ et $u\{t \mapsto v\}{\downarrow} = u{\downarrow}\{t \mapsto v\}{\downarrow}$. □

Exemple 29

- *Soit* $u = exp(h((a+b) \star J_\star(c+d)), c+d)$. *Alors,*
  - *pour* $t = c+d$, *on a* $u{\downarrow} \notin F(t)$ *et* $u\{t \mapsto v\}{\downarrow} = u{\downarrow}\{t \mapsto v\}$
  - *pour* $t = a+b$, *on a* $u{\downarrow} \in F(t)$ *et* $u\{t \mapsto v\}{\downarrow} = h(v) \neq h(t) = u{\downarrow}\{t \mapsto v\}$. *Dans ce cas, on a par contre* $u\{t \mapsto v\}{\downarrow} = h(v)$, *quand* $u{\downarrow} = h(t)$.
- *Pour* $u = h(a \star b + c + d) \bullet h(J_+(a \star b))$ *et* $t = a \star b$, *on a* $u{\downarrow} = h(t) \in F(t)$ *et* $u\{t \mapsto v\}{\downarrow} = h(v)$.
- *Pour* $u = h(a \star b + c + d) \bullet h(J_+(c+d))$ *et* $t = c+d$, *on a* $u{\downarrow} = h(t) \in F(t)$ *et* $u\{t \mapsto v\}{\downarrow} = u{\downarrow}$.

6.3 Localité

La localité est une propriété de systèmes de déduction introduite par McAllester [McA93], qui s'est averée d'une grande importance dans les applications à la sécurité. Elle nous dit que, s'il existe une preuve de $T \vdash v$, il existe une preuve ζ *locale*, i.e. dont tous les termes intérmédiaires sont des sous-termes de T et v. Dans nos notations: $\forall \zeta' \in \mathsf{St}(\zeta).\ \zeta'[T]{\downarrow} \in \mathsf{St}(T, v)$. Maints résultats de décidabilité pour la récherche d'attaques (passifs - recherche de preuve dans un ensemble clos - ou actifs - résolution des contraintes de déductibilité) sont basées sur un résultat de localité: e.g. [CLS03, CKRT03a, CKRT03b, CLT03, DLLT08, Shm04, LLT07].

Pour traîter des théories encore plus complexes, des résultats de combinaison pour la localité sont désirables. Cependant, il n'existe pas des travaux dans cette direction, à part [ACD07]. Même pour les théories bien-modées, les auteurs de [CR06] prennent une hypothèse plus forte que la localité (l'ainsi nommée hypothèse 1), ne montrant pas de manière satisfaisante quand elle est satisfaite.

Le résultat principal de cette section est une généralisation de [ACD07] et [BCD07b], au dela des théories bien-modées. Nous allons montrer que, pour les systèmes de réécriture purs, toute preuve est equivalente à une preuve locale, dont la décomposition selon les sous-termes sémantiques nous donne des termes intermédiaires qui sont des sous-termes des hypothèses ou de la conclusion.

La localité repose sur la notion de sous-termes sémantiques et sur une fonction $\mathcal{D}$ qui rajoute ou enlève des contextes à un terme t. $\mathcal{D}$ est défini par un ensemble fini de règles de remplacement $u_i \mapsto v_i$, $1 \leq i \leq n$: pour tout t, $\mathcal{D}(t) = \{v_i\theta \mid t = u_i\theta,\ 1 \leq i \leq n\}$.

Definition 21 (Localité) *Un système de réécriture* $\mathcal{R}$ *est* local *par rapport à* $\mathcal{D}$ *et* St *si, pour toute recette* C *et tout ensemble fini* T *de termes en forme*

normale, il existe une recette ζ *telle que* $\zeta[T]\downarrow = C[T]\downarrow$ *et,* $\forall \zeta' \in \mathsf{St}(\zeta)$, $\zeta'[T]\downarrow \in \mathcal{D}(\mathsf{St}(T)) \cup \mathcal{D}(\mathsf{St}(\zeta[T]\downarrow))$

La localité nous dit que, si t peut être déduit de T, alors il existe une recette ζ telle que $\zeta[T]\downarrow = t$ et, si on décompose ζ selon ses sous-termes, en les normalisant on reste proches des sous-termes de T et t. La notion de "proximité" est donnée par $\mathcal{D}$.

Pour EP, $\mathcal{D}$ doit par exemple rajouter un h au dessus des termes:

Exemple 30 *Soit* $T = \{h(h(a) + J_+(b)), b\}$ *et* $t = h(h(a) \star b)$. *Pour* $\mathcal{R}_{\mathsf{EP}}$, *la recette* ζ *la plus simple telle que* $\zeta[T]\downarrow = t$ *est* $\zeta = exp(x_1 \bullet h(x_2), x_2)$. *Si on considère* $\zeta' = x_1 \bullet h(x_2) \in \mathsf{St}_{\mathsf{EP}}(\zeta)$, *on obtient* $\zeta'[T]\downarrow = h(h(a))$, *qui n'est pourtant pas dans* $\mathsf{St}_{\mathsf{EP}}(T) \cup \mathsf{St}_{\mathsf{EP}}(t)$. *On a* $\zeta'[T]\downarrow = h(u)$, *avec* $u \in \mathsf{St}_{\mathsf{EP}}(T)$, *d'où la necessite du remplacement* $\{x \mapsto h(x)\}$ *dans* $\mathcal{D}_{\mathsf{EP}}$.

Passons à la preuve de localité pour les systèmes purs, par rapport à St et $\mathcal{D} = \{x \mapsto \mathsf{h}(x), x \mapsto \mathsf{h}(\mathsf{h}(x)), \mathsf{h}(x) \mapsto x\} \cup \{x \mapsto u \mid u \in \overline{\overline{F}}(\mathcal{C}_{\mathcal{R}})\}$. Nous allons d'abord montrer un résultat qui parle de la décomposition (l'acces à ses sous-termes) d'un terme: si la théorie d'un terme change par normalisation, il est égal à l'un des ses sous-termes stricts. Nous utiliserons ceci plus tard pour identifier des preuves qu'on va appeler des *décompositions*.

Lemme 15 *Pour tout terme* u *et système de réécriture pur* $\mathcal{R}$, *on a*

$$\top(u\downarrow) \neq \top(u) \ \&\ top(u\downarrow) \neq top(u) \implies u\downarrow \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u)))\downarrow \cup \overline{\overline{F}}(\mathcal{C}_{\mathcal{R}})$$

où $\mathsf{St}_<(u) = \mathsf{St}(u)\backslash\{u\}$.

Preuve: Nous faisons la preuve par récurrence sur $ld(u)$ - la longueur d'une derivation innermost de u vers sa forme normale. Nous allons prouver une propriété plus forte: $\exists w \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u)))$. $u = w\downarrow$ & $ld(w) < ld(u)$. Si u est en forme normale, $\top(u\downarrow) = \top(u)$ et on conclut facilement. Si u est une constante ou une variable, l'ensemble de ses facteurs est vide. Puisque $\top(u\downarrow) \neq \top(u)$, on a, par la proposition 2, $u\downarrow \in \emptyset$ - contradiction.

Supposons maintenant que $u = \zeta[u_1, \ldots, u_k]$, tel que $\mathsf{Fact}(u) = \{u_1, \ldots, u_k\}$.

Notons d'abord que, pour chaque u_i, on peut supposer que $ld(u_i) < ld(u)$. En effet, puisque $\top(u) \neq \top(u\downarrow)$, ceci n'est pas le cas seulement si $u = \mathsf{h}(u_i)$ et $u\downarrow = \mathsf{h}(u_i\downarrow)$. Donc $top(u) = top(u\downarrow)$ et on peut conclure par contradiction. Admettons donc que $ld(u_i) < ld(u)$, pour tout $1 \leq i \leq k$. De plus, on peut supposer que $\zeta \notin F(\mathcal{X})$, sinon $u\downarrow \in \mathcal{C}_{\mathcal{R}}$.

Maintenant, puisque $\top(u\downarrow) \neq \top(u)$, par le corollaire 1, ou bien $u\downarrow \in \overline{\overline{F}}(\mathcal{C}_{\mathcal{R}})$ - et on conclut - ou bien il existe un i, $1 \leq i \leq k$, tel que

- ou bien $\top(u_i\downarrow) \neq \top(u_i)$ & $top(u_i\downarrow) \neq top(u_i)$
- ou bien $\exists w \in \overline{\overline{F}}(u_i)$. $w\downarrow = v\downarrow$ & $ld(w) < ld(u)$.

Dans le deuxième cas, on conclut.

Dans le premier cas, puisque $ld(u_i) < ld(u)$, on peut appliquer l'hypothèse de récurrence: il existe un $w \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u_i)))$ tel que $w{\downarrow} = u_i{\downarrow}$ et $ld(w) < ld(u_i)$.

Considérons le terme $u' = \zeta[u_1, \ldots, u_{i-1}, w, u_{i+1}, \ldots, u_k]$. On a $ld(u') < ld(u)$ et $u'{\downarrow} = u{\downarrow}$. Pour appliquer l'hypothèse de récurrence à u', notons que $\top(u') = \top(u)$, car $\zeta \notin \overline{F}(\mathcal{X})$, et donc $\top(u'{\downarrow}) \neq \top(u')$ & $top(u'{\downarrow}) \neq top(u')$. Par l'hypothèse de récurrence, on obtient $u{\downarrow} \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u')){\downarrow} \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$.

Pour conclure, il suffit de montrer que $\overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u')){\downarrow} \subseteq \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u)){\downarrow}$. Notons que $\overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u')){\downarrow} = \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u'){\downarrow}){\downarrow}$, par convergence.

Par le lemme 7, on a $\mathsf{St}_<(u') \subseteq \mathsf{St}_<(u) \cup \mathsf{St}(\mathsf{Fact}(w) \cup \overline{G}(w))$. Puisque $w \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u_i)))$, on a $w \in \{w', \mathsf{h}(w'), \mathsf{h}(\mathsf{h}(w'))\}$, pour un $w' \in \overline{G}(\mathsf{St}_<(u_i)) \subseteq \overline{G}(\mathsf{St}_<(u))$. On considère chaque cas:

- Si $w = w'$, $\mathsf{St}_<(u') \subseteq \overline{G}(\mathsf{St}_<(u))$ et $\overline{G}(\mathsf{St}_<(u')) \subseteq \overline{G}(\overline{G}(\mathsf{St}_<(u))) \subseteq \overline{G}(\mathsf{St}_<(u))$, en utilisant le lemme 12. On obtient donc $\overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u')){\downarrow} \subseteq \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u)){\downarrow}$ et on conclut.

- Si $w = \mathsf{h}(w')$, $\mathsf{St}_<(u') \subseteq \overline{G}(\mathsf{St}_<(u)) \cup \{\mathsf{h}(w')\}$ et $\overline{G}(\mathsf{St}_<(u')) \subseteq \overline{G}(\overline{G}(\mathsf{St}_<(u))) \cup \{\mathsf{h}(w')\} \subseteq \overline{G}(\mathsf{St}_<(u)) \cup \{\mathsf{h}(w')\}$, par le lemme 12. Puisque $\mathsf{h}(w'){\downarrow} = u_i{\downarrow} \in \mathsf{St}_<(u){\downarrow}$, on obtient $\overline{G}(\mathsf{St}_<(u')){\downarrow} \subseteq \overline{G}(\mathsf{St}_<(u)){\downarrow}$ et on conclut $\overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u')){\downarrow} \subseteq \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u)){\downarrow}$.

- Si $w = \mathsf{h}(\mathsf{h}(w'))$, $\mathsf{St}_<(u') \subseteq \overline{G}(\mathsf{St}_<(u)) \cup \{\mathsf{h}(w'), \mathsf{h}(\mathsf{h}(w'))\}$. Puisque $\mathsf{h}(\mathsf{h}(w')){\downarrow} = u_i{\downarrow} \in \mathsf{St}_<(u){\downarrow}$, on obtient $\overline{G}(\mathsf{St}_<(u')){\downarrow} \subseteq \overline{G}(\mathsf{St}_<(u)){\downarrow} \cup \{h(w'){\downarrow}\}$ et donc $\overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u')){\downarrow} \subseteq \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u)){\downarrow} \cup \overline{\overline{F}}(\{\mathsf{h}(w')\}){\downarrow}$. On conclut, en observant

  $$\overline{\overline{F}}(\{\mathsf{h}(w')\}){\downarrow} = \{\mathsf{h}(w'){\downarrow}, \mathsf{h}(\mathsf{h}(w')){\downarrow}, \mathsf{h}(\mathsf{h}(\mathsf{h}(w'))){\downarrow}\} = \{\mathsf{h}(w'){\downarrow}, u_i{\downarrow}, \mathsf{h}(u_i){\downarrow}\} \subseteq \overline{F}(\overline{G}(\mathsf{St}_<(u))){\downarrow}$$

On conclut donc $u{\downarrow} \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u)){\downarrow} \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$ □

Exemple 31 *Pour* EP*,*

- *Si* $u = exp(h((a + b) \star c), J_\star(c))$*, on a*
  - $\top(u) = top(u) = exp$
  - $u{\downarrow} = h(a + b)$, $\top(u{\downarrow}) = \bullet$, $top(u{\downarrow}) = h$
  - $u{\downarrow} \in \overline{F}(\mathsf{St}_<(u))$
- *Si* $u = h((a \star b) + c) \bullet h(J_+(c))$*, on a*
  - $\top(u) = top(u) = \bullet$
  - $u{\downarrow} = h(a \star b)$, $\top(u{\downarrow}) = exp$, $top(u{\downarrow}) = h$
  - $u{\downarrow} \in \overline{F}(\mathsf{St}_<(u))$

La double barre au-dessus du F n'est pas nécessaire pour EP*. Donnons un exemple artificiel, où elle l'est. Considérons un système de réécriture qui contient (parmi d'autres) les règles $f(x \bullet y, y) \to x, x \bullet y \bullet J_\bullet(y) \to x$ et supposons que $\bullet \in \mathsf{thdest}(\mathsf{h})$ et $\top(f) = \bullet$. Notons aussi que, pour que $f(x \bullet y, y)$ soit pur, on doit avoir $\mathsf{th}(f, 1) = \bullet$. Pour $u = f(\mathsf{h}(a) \bullet b, b) \bullet a \bullet J_\bullet(a)$, on obtient*

- $\mathsf{St}_<(u) = \{a, b\}$
- $\top(u) = top(u) = \bullet$
- $u{\downarrow} = \mathsf{h}(\mathsf{h}(a)), \top(u{\downarrow}) = top(u{\downarrow}) = \mathsf{h}$
- $u{\downarrow} \in \overline{\overline{F}}(\mathsf{St}_<(u)) \smallsetminus \overline{F}(\mathsf{St}_<(u))$

Nous allons montrer que, pour toute preuve qui n'est pas locale, il existe une preuve plus petite de la même chose, où l'ordre entre les preuves est donné par:

Definition 22 (Mesure sur les recettes) *Pour une recette ζ, soit $|\zeta|_{\neq \mathsf{h}}$ le nombre d'occurrences dans ζ de symboles qui ne sont pas dans $\{\mathsf{h}\} \cup \mathcal{C}_\mathcal{R}$, et $|\zeta|$ la taille de ζ. Alors on definit $m(\zeta) = (|\zeta|_{\neq \mathsf{h}}, |\zeta|)$ et on compare les recettes suivant la composition lexicographique des deux ordres sur les entiers naturels: $m(\zeta') < m(\zeta)$ ssi ou bien $|\zeta'|_{\neq \mathsf{h}} < |\zeta|_{\neq \mathsf{h}}$ ou bien $|\zeta'|_{\neq \mathsf{h}} = |\zeta|_{\neq \mathsf{h}}$ & $|\zeta'| < |\zeta|$.*

Nous allons montrer dans la suite qu'un ζ avec $m(\zeta)$ minimal est local.

Le lemme suivant identifie deux possibilités pour les preuves avec un ensemble des hypothèses T: ou bien leur conclusion est un sous-terme de T, ou bien leur théorie est identique à celle de leur conclusion (ce sont les preuves par *composition*). Plus précisément, le lemme montre que, si $T \vdash v$ et v n'est pas un sous-terme de T, alors il existe une preuve par composition dont la mesure est minimale. Cette classification adapte celle (déjà) classique (e.g. [RT01, CKRT03a, ACD07]) au cas des combinaisons pures.

Dans la suite, nous allons parfois identifier une substitution σ avec l'ensemble de termes dans son image, $\mathsf{img}(\sigma)$. Ceci justifie la notation $\sigma \vdash v$ utilisée dans le lemme suivant et ensuite.

Lemme 16 (Décomposition/Composition) *Soit σ une substitution en forme normale et v un terme tel que $\sigma \vdash v$. Alors,*

Décomposition: *ou bien $v \in \overline{\overline{F}}(\overline{G}(\mathsf{St}(\sigma))) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$*

Composition: *ou bien il existe une recette ζ telle que:*

- $\zeta\sigma{\downarrow} = v$
- $\forall \zeta'. \zeta'\sigma{\downarrow} = v \implies m(\zeta) \leq m(\zeta')$
- $\top(\zeta\sigma{\downarrow}) = \top(\zeta\sigma) \vee top(\zeta\sigma{\downarrow}) = top(\zeta\sigma)$.

Preuve: Soit ζ une recette telle que $\zeta\sigma\downarrow = v$ et $m(\zeta)$ minimal. Si $\top(\zeta\sigma\downarrow) = \top(\zeta\sigma) \ \vee top(\zeta\sigma\downarrow) = top(\zeta\sigma)$, on conclut. Supposons que ce n'est pas le cas. Par le lemme 15, on a $v \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(\zeta\sigma)))\downarrow \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$. Si $v \in \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$, on conclut. Sinon, par le lemme 7, on obtient ou bien $v \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(\zeta))\sigma)\downarrow$ ou bien $v \in \overline{\overline{F}}(\overline{G}(\mathsf{St}(\sigma)))\downarrow$. Si on est dans le deuxième cas, on conclut, en utilisant aussi le lemme 11.

Si on est dans le premier cas, soit $\zeta_0 \in \overline{G}(\mathsf{St}_<(\zeta))$ la recette telle que $v \in \{\zeta_0\sigma\downarrow, \mathsf{h}(\zeta_0\sigma)\downarrow, \mathsf{h}(\mathsf{h}(\zeta_0\sigma))\downarrow\}$. Pour un $\zeta' \in \{\zeta_0, \mathsf{h}(\zeta_0), \mathsf{h}(\mathsf{h}(\zeta_0))\}$, on a $\zeta'\sigma\downarrow = v$. Si $top(\zeta) \neq \mathsf{h}$, on contredit la minimalité de ζ, car $|\zeta'|_{\neq\mathsf{h}} < |\zeta|_{\neq\mathsf{h}}$. Si $top(\zeta) = \mathsf{h}$, on a, par le lemme 11, ou bien $\zeta\sigma\downarrow \in \mathcal{C}_\mathcal{R}$, et on conclut, ou bien $top(\zeta\sigma\downarrow) = \mathsf{h}$, et on contredit notre supposition sur ζ. Ce qui nous permet de conclure le lemme. □

Maintenant, si une preuve donnée de $T \vdash v$ n'est pas locale, le lemme précédent nous assure que les gros termes intermediaires sont obtenus par composition. Nous allons montrer que dans ce cas ils peuvent être remplacés par une constante publique c. Les lemmes des sections précédentes assurent que ce remplacement est compatible avec la réécriture: en normalisant après le remplacement, on obtient la conclusion v ou le gros terme est remplacé par c. Puisque le gros terme n'apparaît pas dans les sous-termes de v, ce dernier remplacement est vide et la conclusion de la nouvelle preuve est ainsi v. On obtient ainsi une preuve locale de $T \vdash v$, car chaque remplacement d'un gros terme diminue la mesure m:

Théorème 1 (Localité) *Soit $\mathcal{R}$ un système de réécriture pur, σ une substitution normalisée et v un terme tel que $\sigma \vdash v$. Alors il existe une recette ζ telle que*

- $\zeta\sigma\downarrow = v$
- $\forall\zeta' \in \mathsf{St}(\zeta).\zeta'\sigma\downarrow \in \overline{\overline{F}}(\overline{G}(\mathsf{St}(\sigma, v)) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$

Preuve: Soit ζ la recette avec $m(\zeta)$ minimal et telle que $\zeta\sigma\downarrow = v$. On montre par récurrence sur la taille de ζ qu'elle satisfait une propriété plus précise que celle requise: $\forall\zeta' \in \mathsf{St}(\zeta).\zeta'\sigma\downarrow \in S(\sigma, v)$, avec

$$\begin{array}{rcl} S(\sigma, v) & = & \overline{\overline{F}}(\overline{G}(\mathsf{St}(\sigma))) \\ & \cup & \overline{F}(G(\mathsf{St}_<(v))) \\ & \cup & F(G(v)) \\ & \cup & \overline{\overline{F}}(\mathcal{C}_\mathcal{R}) \end{array}$$

Si ζ est une variable, on conclut imméditatement, car alors $v \in \mathsf{St}(\sigma)$. Sinon, soit $\zeta = \zeta_0[\zeta_1, \ldots, \zeta_k]$, avec $\mathsf{Fact}(\zeta) = \{\zeta_1, \ldots, \zeta_k\}$. Soient $v_1 = \zeta_1\sigma\downarrow, \ldots, v_k = \zeta_k\sigma\downarrow$. Notons que, pour tout i, $m(\zeta_i)$ est minimal parmi les recettes qui permettent de déduire $\sigma \vdash v_i$. Par l'hypothèse de récurrence, on a que $\forall i.\forall\zeta' \in \mathsf{St}(\zeta_i)$: $\zeta'\sigma\downarrow \in S(\sigma, v_i)$.

On montre maintenant que $\forall i.\ v_i \in S(\sigma, v)$. Soit $\mathsf{Big} = \{v_{i_1}, \ldots, v_{i_n}\} \subseteq \{v_1, \ldots, v_k\}$ l'ensemble des termes tels que $\forall j : v_{i_j} \notin S(\sigma, v)$ et supposons par contradiction que cet ensemble n'est pas vide. Par le lemme 16, pour chaque $v_i \in \mathsf{Big}$, on peut supposer que $\top(v_i) = \top(\zeta_i\sigma)$ ou $top(v_i) = top(\zeta_i\sigma)$. En conséquence, puisque, pour tout $i \in \{i_1, \ldots, i_n\}$, $\zeta_i \in \mathsf{Fact}(\zeta) \setminus \mathcal{X}$, on déduit par la proposition 1 et le lemme 4, que $\top(v_i) \notin \mathsf{th}(\zeta_0, x_i)$ et $\mathsf{Fact}_{\mathsf{th}(\zeta_0,x_i)}(v_i) = \{t_i\} \in \overline{G}(v_i)$.

Soit $m \in \{i_1, \ldots, i_n\}$ tel que t_m est maximal (en taille) parmi $\bigcup_{j=1}^{n} \mathsf{Fact}_{\mathsf{th}(\zeta_0,x_{i_j})}(v_{i_j})$.

Pour chaque $1 \leq j \leq k$, voyons quelle valeur $v_j\{t_m \mapsto c\}$ peut prendre, pour une constante $c \in \mathcal{C}_\mathcal{R}$. On considère trois cas:

- $v_j \in S(\sigma, v) \setminus F(G(v))$. On montre que $t_m \notin \mathsf{St}_<(v_j)$. Si, par contradiction, $t_m \in \mathsf{St}_<(v_j)$, on a $t_m \in \mathsf{St}_<(S(\sigma, v) \setminus F(G(v))) \subseteq \overline{F}(\overline{G}(\mathsf{St}(\sigma))) \cup G(\mathsf{St}_<(v)) \cup \overline{F}(\mathcal{C}_\mathcal{R})$, en utilisant le lemme 12 pour l'inclusion. On déduit $v_m \in \overline{F}(t) \subseteq \overline{F}(\overline{F}(\overline{G}(\mathsf{St}(\sigma)))) \cup \overline{F}(G(\mathsf{St}_<(v))) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R}) \subseteq S(\sigma, v)$, en contradiction avec $v_m \in \mathsf{Big}$. Donc $t_m \notin \mathsf{St}_<(v_j)$ et $v_j\{t_m \mapsto c\} \in \{v_j, c\}$.

- $v_j \in \mathsf{Big}$. Si $v_j \in \{t_m, \mathsf{h}(t_m)\}$, on a $v_j\{t_m \mapsto c\} \in \{c, \mathsf{h}(c), v_m\}$. Sinon, on a $v_j \in \{t_j, \mathsf{h}(t_j)\}$, pour $\{t_j\} = \mathsf{Fact}_{\mathsf{th}}(\zeta_0, x_j)(v_j)$ et $t_m \notin \{t_j, \mathsf{h}(t_j)\}$. De plus, par la maximalité de t_m, on a $t_m \notin \mathsf{St}_s(t_j)$. Donc $v_j\{t_m \mapsto c\} = v_j$.

- $v_j \in F(G(v))$. Supposons par contradiction que $t_m \in \mathsf{St}_<(v_j) \subseteq G(v) \cup \mathsf{St}_<(v)$. Si $t_m \in \mathsf{St}_<(v)$, on a $v_m \in \overline{F}(\mathsf{St}_<(v))$, contredisant le fait que $v_m \in \mathsf{Big}$. Le seul cas qui reste a considerer est $t_m \in G(v)$. Par la minimalité de ζ, on obtient $v_m = \mathsf{h}(t_m)$ et $v = t_m$. Alors $top(\zeta_m) \neq \mathsf{h}$, car autrement on aurait une preuve plus petite de v, contredisant la minimalité de ζ. Puisque $v_m \in \mathsf{Big}$, on obtient $\top(v_m) = \top(\zeta_m\sigma)$. Puisque $top(\zeta_m\sigma) \neq \mathsf{h}$ (on a utilisé aussi que $\zeta_m \notin \mathcal{X}$), on obtient $\top(v_m) \neq \mathsf{h}$. Par définitions, on déduit $v_m \in F(t_m)$. On obtient donc $v_m \in F(G(v)) \subseteq S(\sigma, v)$, en contradiction avec $v_m \in \mathsf{Big}$. Donc $t_m \notin \mathsf{St}_<(v_j)$ et $v_j\{t_m \mapsto c\} \in \{v_j, c\}$.

On conclut que, pour tout $1 \leq j \leq k$, $v_j\{t_m \mapsto c\} \in \{c, \mathsf{h}(c), v_j\}$

Notons que $v \notin F(t_m)$. Dans le cas contraire, $t_m \in G(v)$, contredisant la minimalité de ζ. Par conséquent, puisque $v = \zeta_0[v_1, \ldots, v_k]\downarrow$ et $\zeta_0[v_1, \ldots, v_k]$ est avec tous les facteurs en forme normale, par la proposition 3, on a $v\{t_m \mapsto c\} = \zeta_0[v_1, \ldots, v_k]\{t_m \mapsto c\}\downarrow$. De plus, $t_m \notin \mathsf{St}_<(v)$, sinon contredisant $v_m \in \mathsf{Big}$, donc $v\{t_m \mapsto c\} = v$.

En utilisant le lemme 8, on obtient: $\zeta_0[v_1, \ldots, v_k]\{t_m \mapsto c\} = \zeta_0[v'_1, \ldots, v'_k]$, où:

- si $\top(v_i) \notin \mathsf{th}(\zeta_0, x_i)$ & $\mathsf{Fact}_{\mathsf{th}(\zeta_0,x_i)}(v_i) = \{t_m\}$, alors $v'_i \in \{c, \mathsf{h}(c)\}$. En particulier, $v'_m \in \{c, \mathsf{h}(c)\}$

- sinon, $v'_i \in \{v_i\{t_m \mapsto c\}, v_i\}$

Puisque $v_i\{t_m \mapsto c\} \in \{v_i, c, \mathsf{h}(c)\}$, pour tout i, on déduit qu'il existe une preuve ζ' de $\sigma \vdash \zeta_0[v'_1, \ldots, v'_k]\downarrow = \zeta_0[v_1, \ldots, v_k]\{t_m \mapsto c\}\downarrow = v$. Soient $\zeta'_1, \ldots, \zeta'_k$ les preuves correspondantes de $v'_1, \ldots, v'_k$ et $\zeta' = \zeta_0[\zeta'_1, \ldots, \zeta'_k]$. Notons que:

- pour tout i, $m(\zeta'_i) \leq m(\zeta_i)$. En effet, ou bien $v'_i = v_i$, et alors $\zeta'_i = \zeta_i$, ou bien $v_i \in \{t_m, \mathsf{h}(t_m)\}$ & $v'_i \in \{c, \mathsf{h}(c)\}$ et alors $\zeta'_i = v'_i$. Dans ce cas, $|\zeta'_i|_{\neq h} = 0$. Montrons que $|\zeta'_i| \leq |\zeta_i|$ ou $|\zeta_i|_{\neq \mathsf{h}} > 0$. Puisque $t \notin \mathsf{St}(\sigma)$, et donc $\zeta_i \notin \mathcal{X}$, ce n'est pas le cas seulement si ζ_i est une constante. Puisque $v_i \in \{t_m, \mathsf{h}(t_m)\}$ et $t_m \notin \mathcal{C}_\mathcal{R}$, on déduit que $\zeta_i \notin \mathcal{C}_\mathcal{R}$. Par conséquent, $|\zeta_i|_{\neq h} > 0$.

- $m(\zeta'_m) < (\zeta_m)$. En effet, on a $v'_m \in \{c, \mathsf{h}(c)\}$ et donc $\zeta'_m \in \{c, \mathsf{h}(c)\}, |\zeta'_m|_{\neq \mathsf{h}} = 0$. On a $|\zeta'_m| = |\zeta_m|$ seulement si $\zeta_m \in \{a, x, g(a), g(x)\}$, pour $g, a \in \mathcal{F}$. Dans tous les cas, on a $|\zeta_m|_{\neq \mathsf{h}} \geq 1$, en utilisant $v_m \notin \overline{F}(\mathsf{St}(\sigma) \cup \mathcal{C}_\mathcal{R})$.

On conclut donc que $m(\zeta') < m(\zeta)$, en contradiction avec la minimalité de ζ. Donc Big est vide et on déduit que $\{v_1, \ldots, v_k\} \subseteq S(\sigma, v)$. (**)

Soit $\zeta' \in \mathsf{St}(\zeta)$. Par definition, $\zeta' \in \{\zeta\} \cup \mathsf{St}(\zeta_1, \ldots, \zeta_k)$. Si $\zeta' = \zeta$, on a immédiatement $\zeta'\sigma\downarrow \in S(\sigma, v)$. Si $\zeta' \in \mathsf{St}(\zeta_1, \ldots, \zeta_k)$, on a, par (*) et (**), $\zeta'\sigma\downarrow \in \overline{\overline{F}}(\overline{G}(\mathsf{St}(\sigma))) \cup \overline{F}(G(\mathsf{St}_<(S(\sigma, v)))) \cup F(G(S(\sigma, v))) \cup S(\sigma, v) \subseteq S(\sigma, v)$, en utilisant le lemme 12 et l'idempotence de F, G. □

En d'autres termes, le théoreme précédent nous dit que tout système pur $\mathcal{R}$ est local par rapport à St et $\mathcal{D} = \{x \mapsto \mathsf{h}(x), x \mapsto \mathsf{h}(\mathsf{h}(x)), \mathsf{h}(x) \mapsto x\} \cup \{x \mapsto u \mid u \in \overline{\overline{F}}(\mathcal{C}_\mathcal{R})\}$.

Exemple 32 *Pour* EP*, (on va noter $J_+(t)$ par $-t$)*

1. *Soit $\sigma = \{x_1 \mapsto h(a \star b - b), x_2 \mapsto b\}$ et $v = h(a \star b \star b)$. On a que $\zeta = exp(x_1 \bullet h(x_2), x_2)$ est une preuve locale de $\sigma \vdash v$: son seul terme intermediaire est $\zeta' = x_1 \bullet h(x_2)$ et on a $\zeta'\sigma\downarrow = h(a \star b) \in \overline{F}(\mathsf{St}(\sigma))$. Notons qu'il n'existe pas de preuve de $\sigma \vdash v$ qui evite $\zeta'\sigma\downarrow$.*

2. *Soit $\sigma = \{x_1 \mapsto h(a), x_2 \mapsto b\}$ et $v = h(a \star b + b)$. On a que $exp(x_1, x_2) \bullet h(x_2)$ est une preuve locale de $\sigma \vdash v$: son seul terme intermediaire est $\zeta' = exp(x_1, x_2)$ et $\zeta'\sigma\downarrow = h(a \star b) \in \overline{F}(\mathsf{St}(v))$. Notons qu'il n'existe pas de preuve de $\sigma \vdash v$ qui evite $\zeta'\sigma\downarrow$.*

6.4 Conservativité

Dans le cas de contraintes closes, la localité est suffisante pour réduire le problème à ce qu'on appelle la déductibilité dans un pas (e.g. [BCD07b, LLT07, ACD07, CLT03]): on devine les étapes intermédiaires de la preuve dans l'ensemble de termes fourni par la localité.

Quand le système de contraintes $\mathcal{C}$ contient des variables, l'ensemble de termes intermédiaires ($\mathsf{St}(\mathcal{C}\sigma)$) est inconnu, car il dépend de la substitution considérée σ. Nous allons montrer dans cette section qu'il peut être factorisé: il suffit de considérer des solutions σ pour lesquelles $\mathsf{St}(\mathcal{C}\sigma) \subseteq \mathsf{St}(\mathcal{C})\sigma$. Pour cela, comme dans la preuve de localité, nous allons remplacer chaque gros terme d'une solution par une constante publique. La compatibilité de la réécriture avec le remplacement nous assure qu'on obtient ainsi toujours une solution,

plus petite. Cette idée (l'existence de "petite attaque") a été deja utilisée dans e.g. [RT03, CR05, CR06, DLLT08, MS05], notre contribution dans cette section étant sa généralisation aux théories pures.

On considère, ici aussi, une fonction $\mathcal{D}$ (qui n'est pas nécessairement la même que pour la localité) définie par un ensemble fini de règles de remplacement, qui rajoute ou enlève des contextes à un terme.

Definition 23 (Conservativité) *Une théorie E est conservative (par rapport à $\mathcal{D}$ et St) si, pour tout système de contraintes satisfaisable $\mathcal{C}$, il existe une solution σ de $\mathcal{C}$ telle que $\mathsf{St}(\mathcal{C}\sigma\downarrow) \subseteq \mathcal{D}(\mathsf{St}(\mathcal{C})\sigma\downarrow)$.*

Pour EP, $\mathcal{D}$ doit par exemple enlever un h:

Exemple 33

$$\mathcal{C} = \begin{cases} h(a+b) \bullet c & \stackrel{?}{\vdash} \quad x \\ h(a+b) \bullet c, exp(x \bullet J_\bullet(c), c \star d) & \stackrel{?}{\vdash} \quad y \qquad y = h(z \star d) \end{cases}$$

$\sigma = \{x \mapsto h(a+b) \bullet c, y \mapsto h((a+b) \star c \star d), z \mapsto (a+b) \star c\}$ est une solution minimale de $\mathcal{C}$. $t = a + b \in \mathsf{St}_{\mathsf{EP}}(\mathcal{C}\sigma\downarrow) \backslash \mathsf{St}_{\mathsf{EP}}(\mathcal{C})\sigma\downarrow$. Puisque $h(a+b) = (x \bullet J_\bullet(c))\sigma\downarrow$, σ est conservative si le remplacement $\{h(x) \mapsto x\}$ fait partie de la définition de $\mathcal{D}_{\mathsf{EP}}$.

Passons à la preuve de conservativité pour les systèmes de réécriture purs, par rapport à St et $\mathcal{D} = \{h(x) \mapsto x\} \cup \{x \mapsto u \mid u \in \overline{\overline{F}}(\mathcal{C}_\mathcal{R})\}$. Nous allons d'abord montrer que le remplacement d'un "gros" terme descend toujours dans la substitution:

Lemme 17 *Soit u un terme, t, v des termes en forme normale et σ une substitution normalisée telle que $t \notin \overline{G}(\mathsf{St}(u)\sigma\downarrow) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$. Alors:*

1. $(u\sigma)\{t \mapsto v\} = u(\sigma\{t \mapsto v\})$ *et* $t \in \mathsf{St}(u\sigma) \implies t \in \mathsf{St}(\sigma)$.
2. $(u\sigma)\{t \mapsto v\}\downarrow = u\sigma\downarrow\{t \mapsto v\}\downarrow = u(\sigma\{t \mapsto v\})\downarrow$.
3. $t \in \mathsf{St}(u\sigma\downarrow) \implies t \in \mathsf{St}(\sigma)$.

Preuve: On fait la preuve par récurrence sur la taille de u. Si u est une variable ou une constante, le résultat est immédiat pour tous les 3 points. Sinon, soit $u = \zeta[u_1, \ldots, u_k]$ tel que $\mathsf{Fact}(u) = \{u_1, \ldots, u_k\}$.

1. D'abord, par la proposition 1, on a:

$$\mathsf{Fact}(\zeta[u_1\sigma, \ldots, u_k\sigma]) \subseteq \mathsf{Fact}(u_1\sigma, \ldots, u_k\sigma) \cup \overline{G}(u_1\sigma, \ldots, u_k\sigma)$$

De plus, par le lemme 12, $\mathsf{St}(\overline{G}(u_i\sigma)) \subseteq \mathsf{St}(u_i\sigma) \cup \overline{G}(u_i\sigma)$, pour tout $1 \leq i \leq k$. Ainsi, on a $t \in \mathsf{St}(u\sigma) \implies t \in \{u\sigma\} \cup \mathsf{St}(u_1\sigma, \ldots, u_k\sigma) \cup \overline{G}(u_1\sigma, \ldots, u_k\sigma)$.

Par les hypothèses du lemme, on a $t \notin \overline{G}(u\sigma\downarrow, u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow)$. On en déduit $t \notin \overline{G}(u\sigma, u_1\sigma, \ldots, u_k\sigma)$, en utilisant que t est en forme normale, $t \notin \mathcal{C}_\mathcal{R}$,

le lemme 11 et le dernier point dans la définition de systèmes purs. Il s'ensuit que $t \in \mathsf{St}(u\sigma) \implies t \in \mathsf{St}(u_1\sigma, \ldots, u_k\sigma)$. De plus, puisque, pour tout i, $u_i\sigma \notin \overline{F}(t)$, par le lemme 8, on obtient:

$$(u\sigma)\{t \mapsto v\} = \zeta[(u_1\sigma)\{t \mapsto v\}, \ldots, (u_k\sigma)\{t \mapsto v\}]$$

Par l'hypothèse de récurrence, pour tout i, $(u_i\sigma)\{t \mapsto v\} = u_i(\sigma\{t \mapsto v\})$ et $t \in \mathsf{St}(u_i\sigma) \implies t \in \mathsf{St}(\sigma)$. Ainsi on conclut $(u\sigma)\{t \mapsto v\} = u(\sigma\{t \mapsto v\})$ et $t \in \mathsf{St}(u\sigma) \implies t \in \mathsf{St}(\sigma)$.

2. Prouvons d'abord la première egalité. On a

$$u\sigma\downarrow = \zeta[u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow]\downarrow$$

Notons que, pour chaque i, u_i satisfait les hypothèses du lemme et on peut appliquer l'hypothèse de récurrence: $(u_i\sigma)\{t \mapsto v\}\downarrow = u_i\sigma\downarrow\{t \mapsto v\}\downarrow = u_i(\sigma\{t \mapsto v\})\downarrow$.

Puisque $t \notin \overline{G}(\mathsf{St}(u)\sigma\downarrow)$, on a $t \notin \overline{G}(u_i\sigma\downarrow)$. Par la proposition 1, il resulte que $\mathsf{Fact}(\zeta[u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow]) \cap \{t\} \subseteq \mathsf{Fact}(u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow)$. Puisque, en plus, $\zeta[u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow] \neq t$ (car t est en forme normale et $(u\sigma)\downarrow \neq t$), on obtient par le lemme 8

$$\zeta[u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow]\{t \mapsto v\} = \zeta[u_1\sigma\downarrow\{t \mapsto v\}, \ldots, u_k\sigma\downarrow\{t \mapsto v\}]$$

En outre, $\mathsf{Fact}(\zeta[u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow]) \subseteq \mathsf{Fact}(u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow) \cup \overline{G}(u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow)$: tous les facteurs de $\zeta[u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow]$ sont en forme normale et, en plus, $u\sigma\downarrow \notin F(t)$. Par la proposition 3, on a

$$\zeta[u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow]\{t \mapsto v\}\downarrow = u\sigma\downarrow\{t \mapsto v\}\downarrow$$

En rassemblant toutes ces identités et en utilisant la convergence de $\mathcal{R}$, on obtient:

$$\zeta[u_1\sigma\downarrow\{t \mapsto v\}\downarrow, \ldots, u_k\sigma\downarrow\{t \mapsto v\}\downarrow]\downarrow = u\sigma\downarrow\{t \mapsto v\}\downarrow$$

Par l'hypothèse de récurrence, $(u_i\sigma)\downarrow\{t \mapsto v\}\downarrow = (u_i\sigma)\{t \mapsto v\}\downarrow$. De plus, on a vu dans le point **1** ci-dessus que $\zeta[u_1\sigma, \ldots, u_k\sigma]\{t \mapsto v\} = \zeta[u_1\sigma\{t \mapsto v\}, \ldots, u_k\sigma\{t \mapsto v\}]$. Alors,

$$\begin{array}{rcl}
(u\sigma)\{t \mapsto v\}\downarrow & = & \zeta[u_1\sigma, \ldots, u_k\sigma]\{t \mapsto v\}\downarrow \\
& = & \zeta[(u_1\sigma)\{t \mapsto v\}, \ldots, (u_k\sigma)\{t \mapsto v\}]\downarrow \\
& = & \zeta[(u_1\sigma)\{t \mapsto v\}\downarrow, \ldots, (u_k\sigma)\{t \mapsto v\}\downarrow]\downarrow \\
& = & \zeta[u_1\sigma\downarrow\{t \mapsto v\}\downarrow, \ldots, u_k\sigma\downarrow\{t \mapsto v\}\downarrow]\downarrow \\
& = & (u\sigma)\downarrow\{t \mapsto v\}\downarrow
\end{array}$$

La deuxième egalité requise suit de $(u\sigma)\{t \mapsto v\} = u(\sigma\{t \mapsto v\})$, le premier point du lemme.

3. Par le corollaire 2, on a

$$\mathsf{St}(u\sigma\downarrow) \subseteq \overline{G}(u\sigma\downarrow) \cup \mathsf{St}(u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow) \cup \overline{G}(u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow) \cup \mathcal{C}_{\mathcal{R}}$$

Puisque $t \notin \overline{G}(u\sigma\downarrow, u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow) \cup \mathcal{C}_\mathcal{R}$, on obtient $t \in \mathsf{St}(u\sigma\downarrow) \implies t \in \mathsf{St}(u_1\sigma\downarrow, \ldots, u_k\sigma\downarrow)$. Par l'hypothèse de récurrence, on conclut $t \in \mathsf{St}(\sigma)$.

□

En mettant ensemble le lemme 17 et la proposition 3 on obtient:

Corollaire 3 *Soient u un terme, $t \notin \mathcal{C}_\mathcal{R}, v$ des termes en forme normale et σ une substitution normalisée. Supposons que ou bien tous les facteurs de $u\sigma$ sont en forme normale, ou bien $t \notin \overline{G}(\mathsf{St}(u)\sigma\downarrow) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$. Alors:*

- *si $u\sigma\downarrow \notin F(t)$, on a $(u\sigma)\{t \mapsto v\}\downarrow = u\sigma\downarrow\{t \mapsto v\}\downarrow$.*
- *si $u\sigma\downarrow = \chi[t] \in F(t)$, on a $(u\sigma)\{t \mapsto v\}\downarrow \in \{u\sigma\downarrow, \chi[v]\downarrow\}$*

Preuve: Si tous les facteurs de $u\sigma$ sont en forme normale, on applique la proposition 3. Sinon, $t \notin \overline{G}(\mathsf{St}(u)\sigma\downarrow) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$, on applique le lemme 17, en observant que dans ce cas $u\sigma\downarrow \notin F(t)$. □

Parfois nous aurons besoin de remplacer, à l'inverse, la constante c par le terme t. Le lemme suivant nous montre que ce remplacement conserve les pas de réécriture, puisque c n'appraît pas dans $\mathcal{R}$:

Lemme 18 *Pour tout terme v, contexte ζ et constante c n'apparaissant pas dans $\mathcal{R}$, si $v\downarrow = \zeta[c]$, $\zeta[t] \in \overline{\overline{F}}(t)$ et $t \notin \mathcal{C}_\mathcal{R}$ est en forme normale, alors $v\{c \mapsto t\}\downarrow = \zeta[t]$.*

Preuve: Puisque c n'apparaît pas dans $\mathcal{R}$, on a $c \in \mathsf{St}(v)$ et $v\downarrow = \zeta[c] \implies v\{c \mapsto t\} \to^* \zeta[t]$. Comme $\zeta[t] \in \overline{\overline{F}}(t)$ et $t \notin \mathcal{C}_\mathcal{R}$ est en forme normale, on a par la définition des systèmes purs que $\zeta[t]$ est en forme normale. On conclut donc $v\{c \mapsto t\}\downarrow = \zeta[t]$. □

Parfois, en remplaçant un gros terme t par une constante c, on perd quand même des capacités de déduction, malgré nos propriétés de compatibilité entre le remplacement et la réécriture. Le rôle de la notion suivante sera alors de fournir un moyen pour récupérer les choses perdues.

Rappelons que l'ensemble de constantes est partitionné dans $C_{priv} \uplus C_{pub}$, où C_{priv} sont les constantes privées, connues par l'intrus seulement si divulguées par les agents: il lui faut une preuve non triviale pour les avoir. L'ensemble C_{pub} représente les constantes publiques, toujours disponibles avec une preuve triviale. Nous supposons, sans grande perte de généralité, que l'intersection de C_{priv} avec les constantes qui apparaissent dans $\mathcal{R}$ est vide.

Definition 24 *Soient v un terme, t un terme en forme normale, $c_1, \ldots, c_n$ les constantes de C_{priv} qui apparaissent dans v, t et soient $c_t, c_1^p, \ldots, c_n^p$ des constantes publiques n'apparaissant pas dans $\mathcal{R}$. Soit δ_{c,c^p} le remplacement de chaque c_i par c_i^p, i.e. $\delta_{c,c^p} = \{c_1 \mapsto c_1^p\} \ldots \{c_n \mapsto c_n^p\}$. On définit*

$$v^t = v\{t \mapsto c_t\}\delta_{c,c^p}\{c_t \mapsto t\}$$

Intuitivement, v^t est obtenu en remplaçant les constantes de v par des constantes publiques nouvelles, excepté celles qui apparaissent dans une occurrence de t comme sous-terme de v. Le sens de cette transformation est de faire transparentes les parties de v qui ne correspondent pas à des occurrences de t comme sous-terme.

Exemple 34 *Par exemple,* $(a \star b + a + b)^{a\star b} = a \star b + a^p + b^p$, $(a \star b \star c)^{a\star b} = a^p \star b^p \star c^p$ *et* $((a + a \star b) \star (b + a \star b))^{a\star b} = (a^p + a \star b) \star (b^p + a \star b)$. *Notons que, si* $t \notin \mathsf{St}(v)$, *alors* v^t *ne contient que des symboles publics, i.e.* $T \vdash v^t$, *pour tout* T.

Lemme 19 *Si* v *est en forme normale, alors* v^t *est en forme normale.*

Preuve: v peut être obtenu a partir de v^t en remplaçant à l'inverse chaque c_i^p par c_i. Comme les constantes c_i^p n'apparaissent pas dans $\mathcal{R}$, si $l\sigma$ est un radical dans v^t, alors $l\sigma'$ est un radical dans v, où σ' est la substitution σ dans laquelle chaque c_i^p est remplacée par c_i. □

Le deuxième point du lemme 20 nous montre comment v^t peut être utile, quand $v{\downarrow} \in F(t)$ et $v\{t \mapsto c_t\}{\downarrow} \neq v{\downarrow}$.

Lemme 20 *Soient* u *un terme,* t *un terme en forme normale,* c_t *une constante et* σ *une substitution. Supposons que ou bien tous les facteurs de* $u\sigma$ *sont en forme normale, ou bien* $t \notin \overline{G}(\mathsf{St}(u)\sigma{\downarrow}) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$. *Soit* $v = u\sigma$. *Alors:*

1. *si* $v{\downarrow} \notin F(t)$, *alors* $v^t{\downarrow} = v{\downarrow}^t$
2. *si* $v{\downarrow} \in F(t)$, *alors* $v\{t \mapsto c_t\}{\downarrow} = v{\downarrow}$ *ou* $v^t{\downarrow} = v{\downarrow}$.

Exemple 35 *Avant la preuve, donnons un exemple. Soit* $t = a \star b$.

1. *Pour* $v = a \star b + b + J_+(b) + d$, *nous avons*

$$v^t{\downarrow} = (c_t + b^p + J_+(b^p) + d^p){\downarrow} = c_t + d^p = (t + d)^t = v{\downarrow}^t$$

2. - *Pour* $v = a \star b \star d \star J_\star(d)$, *nous avons* $v{\downarrow} = a \star b$ *et* $v\{t \mapsto c\}{\downarrow} = v{\downarrow}$
   - *Pour* $v = a\star b+b+J_+(b)$, *nous avons* $v{\downarrow} = t$. *Nous aurons besoin d'un moyen pour supplanter la perte contenue dans le fait que* $v\{t \mapsto c\}{\downarrow} = (c + b + J_+(b)){\downarrow} = c$. *Ça sera le fait que* $v^t{\downarrow} = (t + b^p + J_+(b^p)){\downarrow} = t$.

Preuve: 1. Soit $w = v\{t \mapsto c_t\}\delta_{c,c_p}$. Supposons d'abord que $v{\downarrow} \notin F(C_{priv})$. Alors $v{\downarrow} \notin F(t) \cup F(C_{priv})$ et donc, par le corollaire 3 et le fait que $c_t, c_1^p, \ldots, c_n^p$ n'apparaissent pas dans $\mathcal{R}$,

$$w{\downarrow} = v{\downarrow}\{t \mapsto c_t\}\delta_{c,c_p}{\downarrow} = v{\downarrow}\{t \mapsto c_t\}\delta_{c,c_p}$$

Puisque $w{\downarrow} \notin F(t)$, on déduit que $w{\downarrow} \neq c_t$ et, par définition de F et puisque $c_t \notin \mathcal{R}$, $w{\downarrow} \notin F(c_t)$. En appliquant à nouveau le corollaire 3, on conclut

$v^t\downarrow = w\{c_t \mapsto t\}\downarrow = w\downarrow\{c_t \mapsto t\}\downarrow = v\downarrow\{t \mapsto c_t\}\delta_{c,c_p}\{c_t \mapsto t\}\downarrow = (v\downarrow)^t\downarrow = (v\downarrow)^t$, en utilisant le lemme 19 pour le dernier pas.

Supposons maintenant que $v\downarrow \in F(C_{priv})$. Par définition de F et puisque $C_{priv} \cap \mathcal{R} = \emptyset$, on a $v\downarrow = c \in C_{priv}$. Par consequence, en appliquant le lemme 18 en conjonction avec le corollaire 3, on obtient $w\downarrow = c^p$. Donc $w\{c_t \mapsto t\}\downarrow = w\downarrow\{c_t \mapsto t\}\downarrow = c^p = v\downarrow^t$.

2. Par le corollaire 3, ou bien $v\{t \mapsto c_t\}\downarrow = v\downarrow$ (et on conclut) ou bien $v\{t \mapsto c_t\}\downarrow = \zeta[c_t]$, pour un ζ tel que $v\downarrow = \zeta[t] \in F(t)$. Notons que, puisque $c_t \notin C_{priv}$, on a $\zeta[c_t] \notin F(C_{priv})$. En applicant a nouveau le corollaire 3 on obtient: $w\downarrow = \zeta[c_t]\delta_{c,c_p} = \zeta[c_t]$.

Finalement, par le lemme 18, on obtient $v^t\downarrow = w\{c_t \mapsto t\}\downarrow = \zeta[t] = v\downarrow$, et on conclut. □

Lemme 21 *Soient v un terme et σ une substitution. Nous avons:*

$$(v\sigma)\delta_{c,c^p} = (v\delta_{c,c^p})(\sigma\delta_{c,c^p})$$

Preuve: Immédiate par récurrence et la définition du remplacement. □

Les deux lemmes qui suivent seront utiles pour montrer, dans une preuve par récurrence sur l'occurence des variables dans un système de contraintes, que $(v\sigma)^t$ est toujours déductibile de $T\sigma$, si $v\sigma$ l'est.

Lemme 22 *Soient v un terme, t un terme en forme normale et σ une substitution telle que $t \notin G(\mathsf{St}(v)\sigma) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$. Soit δ_{c,c_p} le remplacement de toute constante privée par une constante publique correspondante. Alors $(v\sigma)^t = v\delta_{c,c_p}(\sigma^t)$.*

Preuve: On a:

$$\begin{aligned}
(v\sigma)^t = (v\sigma)\{t \mapsto c_t\}\delta_{c,c^p}\{c_t \mapsto t\} = & \quad \text{par définition} \\
(v(\sigma\{t \mapsto c_t\}))\delta_{c,c^p}\{c_t \mapsto t\} = & \quad \text{par le lemme 17} \\
(v\delta_{c,c^p}(\sigma\{t \mapsto c_t\}\delta_{c,c^p}))\{c_t \mapsto t\} = & \quad \text{par le lemme 21} \\
= v\delta_{c,c^p}(\sigma\{t \mapsto c_t\}\delta_{c,c^p}\{c_t \mapsto t\}) = & \quad \text{par le lemme 17} \\
= v\delta_{c,c_p}(\sigma^t) & \quad \text{par définition}
\end{aligned}$$

□

Lemme 23 *Soient $\zeta[x_1, \ldots, x_n]$ une recette telle que $\mathsf{Fact}(\zeta) \subseteq \mathcal{X}$ et $t, v_1, \ldots, v_k$ des termes. Nous avons $\zeta[v_1, \ldots, v_k]^t = \zeta[v'_1, \ldots, v'_k]$, où:*

- *$v'_i = v_i^t$, si $v_i \notin F(t)$.*
- *$v'_i \in \{v_i, v_i^t\}$, autrement.*

Preuve: Par définition, $\zeta[v_1, \ldots, v_k]^t = \zeta[v_1, \ldots, v_k]\{t \mapsto c_t\}\delta_{c,c^p}\{c_t \mapsto t\}$.

Par le lemme 8 et la définition de F, on a $\zeta[v_1, \ldots, v_k]\{t \mapsto c_t\} = \zeta[v''_1, \ldots, v''_k]$, avec:

- $v''_i = v_i\{t \mapsto c_t\}$, si $v_i \notin F(t)$

- $v''_i \in \{v_i, \chi[c_t]\}$, autrement.

Par le lemme 8 à nouveau et puisque, pour toute constante $c \notin \mathcal{R}$, $\top(c) = c$, on a: $\zeta[v''_1, \ldots, v''_k]\delta_{c,c^p} = \zeta[v''_1\delta_{c,c^p}, \ldots, v''_k\delta_{c,c^p}]$.

Finalement, on conclut $\zeta[v''_1\delta_{c,c^p}, \ldots, v''_k\delta_{c,c^p}]\{c_t \mapsto t\} = \zeta[v'_1, \ldots, v'_k]$, avec:

- $v'_i = v_i^t$, si $v_i \notin F(t)$.
- $v'_i \in \{v_i, v_i^t\}$, autrement (on a utilisé $v_i \in F(t) \implies v_i\{t \mapsto c_t\} \in \{v_i, c_t\}$).

□

Ce lemme clé permet le remplacement d'un gros terme t dans une preuve par une constante publique. Il utilise d'une manière cruciale l'origination, la monotonie et les propriétés prouvées précédemment:

Lemme 24 *Soit $\mathcal{C} = \{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_n \stackrel{?}{\vdash} x_n\} \cup S$ un système de contraintes satisfaisable et soit σ une solution de $\mathcal{C}$. Soient $\zeta_1, \ldots, \zeta_n$ les preuves de $T_1\sigma \vdash x_1\sigma, \ldots, T_n\sigma \vdash x_n\sigma$. Soit t un terme en forme normale tel que $t \notin \overline{G}(\mathsf{St}(\mathcal{C})\sigma\downarrow) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$. On a que $\forall i \forall \zeta \in \mathsf{St}(\zeta_i) \forall v$ tels que $\zeta[T_i\sigma]\downarrow = v$:*

1. *$(T_i\sigma)\{t \mapsto c\} \vdash v\{t \mapsto c\}$.*
2. *$(T_i\sigma)\{t \mapsto c\} \vdash v^t$.*

pour une constante publique c.

Preuve:

Nous faisons une récurrence lexicographique sur la paire $(i, |\zeta|)$. Pour un ensemble des termes T et une substitution θ, nous allons noter $T \vdash \theta$, si $T \vdash x\theta$ pour chaque $x \in \mathsf{dom}(\theta)$.

Cas de base. Si $i = 1$ et ζ est une variable, alors $v \in T_1\sigma\downarrow = T_1$, car T_1 est clos, par origination. Étant donné que $t \notin \overline{G}(\mathsf{St}(T_1)\sigma\downarrow) = \overline{G}(\mathsf{St}(T_1))$, on a $T_1\sigma\{t \mapsto c\} = T_1\sigma$ et $v\{t \mapsto c\} = v$. On obtient ainsi $T_1\sigma \vdash v\{t \mapsto c\}$. De plus, puisque $t \notin \mathsf{St}(v)$, v^t ne contient que des symboles publics et donc il existe une preuve triviale de $T_1\sigma \vdash v^t$. Par conséquent, on conclut.

Pas de récurrence. Nous prouvons les deux points séparement. Notons que, par monotonie, origination et hypothèse de récurrence, nous avons: $\forall x \in \mathsf{Var}(T_i).(T_i\sigma)\{t \mapsto c\} \vdash x\sigma\{t \mapsto c\}$ et $(T_i\sigma)\{t \mapsto c\} \vdash x\sigma^t$.

1.

a. Supposons d'abord que ζ est une variable. Ainsi, il existe un $u \in T_i\sigma$ tel que $u\downarrow = v$. Soit $u = u'\sigma$, avec $u' \in T_i$. Si $v \notin F(t)$ et puisque $t \notin \overline{G}(St(u')\sigma\downarrow)$, on déduit par le corollaire 3 que $u\{t \mapsto c\}\downarrow = v\{t \mapsto c\}$ et on conclut. Autrement, si $v \in F(t)$, par le lemme 20, on a $u\{t \mapsto c\}\downarrow = u\downarrow = v$ ou $u^t\downarrow = u\downarrow = v$. Dans le premier cas, ou bien $v = \mathsf{h}(t)$ et $v\{t \mapsto c\} = v$, et on conclut, ou bien $v = t$ et $v\{t \mapsto c\} = c$, au quel cas on a une preuve triviale de $(T_i\sigma)\{t \mapsto c\} \vdash v\{t \mapsto c\}$, puisque c est une constante publique.

Dans le deuxième cas, étant donné que $t \notin \overline{G}(St(u')\sigma)$, par le lemme 22, on obtient $u^t = (u'\sigma)^t = u'\delta_{c,c^p}(\sigma^t)$. Finalement, puisque par l'hypothèse de

récurrence on a $(T_i\sigma)\{t \mapsto c\} \vdash \sigma^t$ et $u'\delta_{c,c^p}$ ne contient que des symboles publics, on déduit que $(T_i\sigma)\{t \mapsto c\} \vdash u'\delta_{c,c^p}(\sigma^t)\downarrow = u^t\downarrow = v$, et on conclut comme avant que $(T_i\sigma)\{t \mapsto c\} \vdash v\{t \mapsto c\}$.

b. Soit maintenant $\zeta = \zeta_0[\zeta_1, \ldots, \zeta_k], \mathsf{Fact}(\zeta) = \{\zeta_1, \ldots, \zeta_k\}, v_1 = \zeta_1[T_i\sigma]\downarrow, \ldots, v_k = \zeta_k[T_i\sigma]\downarrow$.

Supposons d'abord que $v \notin F(t)$. Puisque, par le lemme 6 et la proposition 1, $\zeta_0[v_1, \ldots, v_k]$ est un terme avec tous les facteurs en forme normale, par le corollaire 3, on obtient $v\{t \mapsto c\} = v\downarrow\{t \mapsto c\} = \zeta_0[v_1, \ldots, v_k]\{t \mapsto c\}\downarrow$ (*). Par le lemme 8, on obtient $\zeta_0[v_1, \ldots, v_k]\{t \mapsto c\} = \zeta_0[v'_1, \ldots, v'_k]$ (**), avec, pour tout $1 \leq j \leq k$,

- ou bien $v'_j = v_j\{t \mapsto c\}$ et alors on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v'_j$ par l'hypothèse de récurrence.
- ou bien $v'_j \in \{c, \mathsf{h}(c)\}$ et alors on a une preuve triviale de $(T_i\sigma)\{t \mapsto c\} \vdash v'_j$, puisque tous les symboles de v'_j sont publics.
- ou bien $v'_j = v_j \in F(t)$. Supposons d'abord que $v_j \in F(t) \setminus \{t\}$. Alors, par définition de F et des facteurs, nous avons $v'_j = v_j = v_j\{t \mapsto c\}$, donc on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v'_j$ par l'hypothèse de récurrence. Il nous reste le cas $v'_j = v_j = t$. Dans ce cas $v_j^t = v_j$, donc on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v'_j$ par le deuxième point de l'hypothèse de récurrence.

Ainsi, dans tous les cas, $(T_i\sigma)\{t \mapsto c\} \vdash v'_j$. Denotons par $\zeta'_1, \ldots, \zeta'_k$ toutes ces preuves de $(T_i\sigma)\{t \mapsto c\} \vdash v'_1, \ldots, v'_k$. Par (*) et (**) on obtient que $\zeta_0[\zeta'_1, \ldots, \zeta'_k]$ est une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v\{t \mapsto c\}$. Ceci conclut la première partie du lemme, quand $v \notin F(t)$.

Supposons maintenant que $v \in F(t)$ et soit $v' = \zeta_0[v_1, \ldots, v_k]$. Puisque v' a tous les facteurs en forme normale, par le lemme 20, on obtient $v'\{t \mapsto c\}\downarrow = v'\downarrow = v$ ou $v'^t\downarrow = v'\downarrow = v$. Si on est dans le premier cas, on conclut: on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v$, qui est $\zeta_0[\zeta'_1, \ldots, \zeta'_k]$, et $v \in F(t) \implies v\{t \mapsto c\} \in \{v, c\}$, ce qui implique $(T_i\sigma)\{t \mapsto c\} \vdash v\{t \mapsto c\}$.

Si on est dans le deuxième cas, par le lemme 23, on a $v'^t = \zeta_0[v'_1, \ldots, v'_k]$, avec pour tout $1 \leq i \leq k$:

- ou bien $v'_i = v_i^t$ et alors on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v'_i$ par le deuxième point de l'hypothèse de récurrence.
- ou bien $v'_i = v_i \in F(t)$. Si $v_i \in F(t)\backslash t$, alors $v_i = v_i\{t \mapsto c\}$ et donc on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v'_i$ par l'hypothèse de récurrence. Si $v_i = t$, alors $v_i^t = t$ et donc on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v'_i$ par le deuxième point de l'hypothèse de récurrence.

Par conséquent, on déduit $(T_i\sigma)\{t \mapsto c\} \vdash v'^t\downarrow$ et donc $(T_i\sigma)\{t \mapsto c\} \vdash v$. Ceci conclut la première partie du lemme: $(T_i\sigma)\{t \mapsto c\} \vdash v\{t \mapsto c\}$, puisque $v\{t \mapsto c\} \in \{v, c\}$, quand $v \in F(t)$.

2.

a. Supposons d'abord que ζ est une variable. Alors il existe un $u \in T_i\sigma$ tel que $v = u\downarrow$. Soit $u = u'\sigma$, avec $u' \in T_i$. Puisque $t \notin \overline{G}(St(u')\sigma) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$,

par le lemme 22, on obtient $u^t = (u'\sigma)^t = u'\delta_{c,c^p}(\sigma^t)$. Étant donné que, par l'hypothèse de récurrence, on a $(T_i\sigma)\{t \mapsto c\} \vdash \sigma^t$ et que $u'\delta_{c,c^p}$ ne contient que des symboles publics, on déduit que $(T_i\sigma)\{t \mapsto c\} \vdash u'\delta_{c,c^p}(\sigma^t)\downarrow = u^t\downarrow$.

Supposons d'abord que $v \notin F(t)$. Alors, puisque $t \notin \overline{G}(St(u')\sigma) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$, par le lemme 20, on a $v^t = u^t\downarrow$ et on conclut: $(T_i\sigma)\{t \mapsto c\} \vdash v^t$. Si $v \in F(t)$, par le lemme 20, on a $u\{t \mapsto c\}\downarrow = v$ ou $u^t\downarrow = v$. Dans les deux cas, on obtient $(T_i\sigma)\{t \mapsto c\} \vdash v$. Si $v = t$, alors $v^t = v$ et on obtient une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v^t$. Si $v \in F(t) \setminus \{t\}$, alors v^t ne contient que des symboles publics, donc on a une preuve triviale de $(T_i\sigma)\{t \mapsto c\} \vdash v^t$.

b. Soit maintenant $\zeta = \zeta_0[\zeta_1, \ldots, \zeta_k], Fact(\zeta) = \{\zeta_1, \ldots, \zeta_k\}, v_1 = \zeta_1[T_i\sigma]\downarrow, \ldots, v_k = \zeta_k[T_i\sigma]\downarrow$.

Soit $v' = \zeta_0[v_1, \ldots, v_k]$. Montrons d'abord qu'on a une preuve de $(T_i\sigma)\{t \mapsto c\} \vdash v'^t\downarrow$. En effet, par le lemme 23, on a $v'^t = \zeta_0[v'_1, \ldots, v'_k]$, avec $v'_i = v_i^t$ ou $v'_i \in \{v_i, v_i^t\}$, quand $v_i \in F(t)$. Par conséquent, en appliquant l'hypothèse de récurrence comme avant, on déduit $(T_i\sigma)\{t \mapsto c\} \vdash v'^t\downarrow$.

Maintenant, si $v \notin F(t)$, on conclut que $(T_i\sigma)\{t \mapsto c\} \vdash v^t$, puisque $v'^t\downarrow = v^t$, par le lemme 20. Si $v \in F(t)\backslash\{t\}$, alors v^t ne contient que des symboles publics et on a un preuve triviale de $(T_i\sigma)\{t \mapsto c\} \vdash v^t$. Si $v = t$, $v^t = v$. Par le lemme 20, on a $v'^t\downarrow = v$ ou $v'\{t \mapsto c\}\downarrow = v$. On conclut, car on a vu, lors de la preuve du premier point du lemme, que $(T_i\sigma)\{t \mapsto c\} \vdash v'^t\downarrow$ et $(T_i\sigma)\{t \mapsto c\} \vdash v'\{t \mapsto c\}\downarrow$.

Ceci conclut la preuve du lemme. □

Suit notre théoreme d'existence de "petit attaque":

Théorème 2 (Conservativité) *Soit* $\mathcal{C} = \{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_n \stackrel{?}{\vdash} x_n\} \cup S$ *un système de contraintes.* $\mathcal{C}$ *est satisfaisable si et seulement si il a une solution conservative* σ'*:* $\mathsf{St}(\mathcal{C}\sigma'\downarrow) \subseteq \overline{G}(\mathsf{St}(\mathcal{C})\sigma'\downarrow) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$.

Preuve: Soit σ une solution de $\mathcal{C}$ telle que il existe un $t \in \mathsf{St}(C\sigma\downarrow) \setminus (\overline{G}(\mathsf{St}(\mathcal{C})\sigma\downarrow) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R}))$. Par le lemme 17, on a $t \in \mathsf{St}(\sigma)$. On montre que $\sigma' = \sigma\{t \mapsto c\}$ est une solution de $\mathcal{C}$, pour une constante publique c.

D'abord, par le lemme 24, on a $\forall i.(T_i\sigma)\{t \mapsto c\} \vdash x_i\sigma\{t \mapsto c\}$.

Ensuite, par le lemme 17, on a $\forall i.(T_i\sigma)\{t \mapsto c\} = T_i(\sigma\{t \mapsto c\})$. Par conséquence, σ' est une solution pour les contraintes de deducibilité dans $\mathcal{C}$.

Soit maintenant $s_1 = s_2$ une équation de S. Puisque $s_1\sigma\downarrow = s_2\sigma\downarrow$, en utilisant le lemme 17, on obtient $s_1(\sigma\{t \mapsto c\})\downarrow = s_1\sigma\downarrow\{t \mapsto c\} = s_2\sigma\downarrow\{t \mapsto c\} = s_2(\sigma\{t \mapsto c\})\downarrow$. Par conséquent, σ' est aussi une solution de S.

De plus, $\mathsf{St}(\sigma') \subseteq \mathsf{St}(\sigma) \setminus \{t\} \cup \{c\}$: tous les mauvais termes dans $\mathsf{St}(\sigma)$ peuvent être remplacés par c, pour obtenir une solution conservative de $\mathcal{C}$. □

En d'autres termes, nous avons montré la conservativité des systèmes purs par rapport à St et $\mathcal{D} = \{h(x) \mapsto x\} \cup \{x \mapsto u \mid u \in \overline{\overline{F}}(\mathcal{C}_\mathcal{R})\}$.

6.5 Première réduction: vers des recettes pures

Definition 25 (Solutions pures, systèmes de contraintes purs) *Une solution σ d'un système de contraintes $\mathcal{C} = \{T_1 \stackrel{?}{\vdash} u_1, \ldots, T_n \stackrel{?}{\vdash} u_n\} \cup S$ est* pure *si, pour chaque $1 \leq i \leq n$, il existe une recette ζ_i telle que $\zeta_i[T_i\sigma]\downarrow = u_i\sigma\downarrow$ et $\mathsf{Fact}(\zeta_i) \subseteq \mathcal{X}$. On dit qu'un système de contraintes est pur, si on ne s'interesse qu'à ses solutions pures.*

Nous allons appeler les recettes ζ avec $\mathsf{Fact}(\zeta) \subseteq \mathcal{X}$ des recettes pures.

Dans cette section, nous réduisons la recherche des solutions d'un système de contraintes à la recherche des solutions pures. Les trois propriétés sur lesquelles la réduction repose sont la localité, la conservativité et la propriété des variants finis. La localité (théorème 1, section 6.3) et la conservativité (théorème 2, section 6.4) ont été prouvées pour les systèmes de réécriture purs dans les sections précédentes, mais la réduction presentée ici est valable pour toute théorie locale et conservative.

Quant à la propriété des variants finis, un exemple typique pour lequel elle est satisfaite est le cas des théories pour lesquelles la surréduction termine. D'autres exemples, parmi lesquels $\mathcal{R}_{EP}$, sont donnés dans [CD05, Del06]. Il y a aussi des résultats qui montrent que la propriété des variants finis est impliquée par la terminaison de restrictions de la surréduction [EMS08]. Coté négatif, ce sont par exemple les théories monoidales qui ne satisfont pas cette propriété: une équation $h(x + y) = h(x) + h(y)$ ne peut pas être orientée de manière à définir un ensemble fini de variants pour à la fois $h(x)$ et $x + y$. Dans [CR06] il y a une hypothèse nommée "réductibilité de la théorie", qui est similaire à la propriété des variants finis. Nous allons y revenir à la fin de cette partie pour une discussion, qui montrera que le relâchement des variants finis est un des travaux futurs les plus ambitieux.

Nous enoncons maintenant notre algorithme général de réduction et en prouvons la complétude.

Théorème 3 *Si E est une théorie locale, consérvative et qui a la propriété des variants finis, la satisfaisabilité des systèmes de contraintes se réduit à la satisfaisabilité des systèmes purs.*

Preuve: On suppose, sans perte de généralité, que les fonctions $\mathcal{D}$ pour la localité et la conservativité sont les mêmes (on peut faire l'union des deux ensembles de remplacements).

Soit $\mathcal{C}$ un système de contraintes et σ une solution de $\mathcal{C}$. Par consérvativité, $\mathcal{C}$ a une solution σ_1 telle que $\mathsf{St}(\mathcal{C}\sigma_1\downarrow) \subseteq \mathcal{D}(\mathsf{St}(\mathcal{C})\sigma_1\downarrow)$. Grâce à la propriété des variants finis, il existe un variant $\mathcal{C}\theta_1\downarrow$ tel que $\mathsf{St}(\mathcal{C})\sigma_1\downarrow \subseteq \mathsf{St}(\mathcal{C}\theta_1\downarrow)\sigma_2$ pour une certaine substitution σ_2 telle que $\sigma_1 = \theta_1\sigma_2$. Pour $CS_1 = \mathcal{C}\theta_1\downarrow$, il existe donc une substitution σ_2 telle que $\mathsf{St}(\mathcal{C})\sigma_1\downarrow \subseteq \mathcal{D}(\mathsf{St}(CS_1)\sigma_2)$.

Maintenant, on enlève la substitution de la portée de $\mathcal{D}$ comme suit. Si $\mathcal{D}$ est définie par les remplacements $u_i \mapsto v_i$, $1 \leq i \leq k$, on devine quels motifs u_i sont introduits par la substitution σ_2. Soit θ_2 une substitution telle que, pour tout

x, $x\theta_2$ est le renommage d'un terme dans $\mathsf{St}(u_1, \ldots, u_k)$. Pour $CS_2 = CS_1\theta_2$, il existe alors une substitution σ_3 telle que $\mathcal{D}(\mathsf{St}(CS_1)\sigma_2) \subseteq \mathcal{D}(\mathsf{St}(CS_2))\sigma_3$.

Par localité, les sous-termes intermédiaires dans les preuves de $\mathcal{C}\sigma_1{\downarrow}$ sont dans $\mathcal{D}(\mathsf{St}(\mathcal{C}\sigma_1{\downarrow})) \subseteq \mathcal{D}(\mathcal{D}(\mathsf{St}(CS_2))\sigma_3)$. Comme ci-dessus, on enlève σ_3 de la portée de $\mathcal{D}$: il existe une substitution θ_3 et un $CS_3 = CS_2\theta_3$ tels que $\mathcal{D}(\mathsf{St}(\mathcal{C}\sigma_1{\downarrow})) \subseteq \mathcal{D}(\mathcal{D}(\mathsf{St}(CS_3)))\sigma_4$, pour une certaine substitution σ_4.

Alors on choisi de manière non-déterministe $\theta = \theta_1\theta_2\theta_3$ et on rajoute à la partie équationnelle de $\mathcal{C}$ les équations $x = x\theta$ pour chaque $x \in \mathsf{Var}(\mathcal{C})$. Puis on devine quels termes dans $\mathcal{D}(\mathcal{D}(\mathsf{St}(CS_3)))$ sont déductibles et dans quel ordre. Si n est le nombre de contraintes dans $\mathcal{C}$, on construit un système $\mathcal{C}_n$ comme suit:

- Soit $\mathcal{C}_0 = \{\} \cup S$.
- Pour tout $0 \leq i < n$:
- Si $\mathcal{C}_i = \{V_{1,1} \stackrel{?}{\vdash} z_{1,1}, \ldots, V_{i,1+k_i} \stackrel{?}{\vdash} z_{i,1+k_i}\} \cup S_i$
- On devine une séquence (éventuellement vide) $v_{i+1,1}, \ldots, v_{i+1,k_{i+1}} \in \mathcal{D}(\mathcal{D}(\mathsf{St}(CS_3)))$ de termes distincts. Soit $S_{i+1} = S_i \cup \{z_{i+1,1} = v_{i+1,1}, \ldots, z_{i+1,k_{i+1}} = v_{i+1,k_{i+1}}\}$ où $z_{i+1,1}, \ldots, z_{i+1,k_{i+1}}$ sont des nouvelles variables.
- Soit $V_{i+1,1} = V_{i,1+k_i} \cup T_{i+1}$ et, pour obtenir $\mathcal{C}_{i+1}$, rajoutons à $\mathcal{C}_i$ les contraintes suivantes:

$$
\begin{array}{lcl}
V_{i+1,1} & \stackrel{?}{\vdash} & z_{i+1,1} \\
V_{i+1,1}, z_{i+1,1} & \stackrel{?}{\vdash} & z_{i+1,2} \\
 & \ldots & \\
V_{i+1,1}, z_{i+1,1}, \ldots, z_{i+1,k_{i+1}-1} & \stackrel{?}{\vdash} & z_{i+1,k_{i+1}} \\
V_{i+1,1}, z_{i+1,1}, \ldots, z_{i+1,k_{i+1}-1}, z_{i+1,k_{i+1}} & \stackrel{?}{\vdash} & x_{i+1}
\end{array}
$$

Le système resultant, $\mathcal{C}_n$, satisfait toujours la monotonie et l'origination.

On prouve maintenant que $\mathcal{C}$ a une solution si, et seulement si, un des systèmes $\mathcal{C}_n$ obtenus par la procédure non-deterministe ci-dessus a une solution pure $\sigma \uplus \theta$, où le domaine de θ est l'ensemble de variables nouvellement introduites par l'algorithme.

Correction. Supposons qu'un tel $\mathcal{C}_n$ a une solution pure σ'. Alors $\sigma = \sigma'|_{Var(\mathcal{C})}$ est une solution de $\mathcal{C}$: par récurrence sur i, $1 \leq i \leq n$, toute solution θ de $\mathcal{C}_i$ est une solution de $\{T_1 \stackrel{?}{\vdash} x_1, \ldots T_i \stackrel{?}{\vdash} x_i\}$.

Complétude. Soit σ une solution de $\mathcal{C}$. Il suffit de montrer, par récurrence, qu'il existe des choix dans la construction ci-dessus tels que, pour tout $0 \leq i < n$, σ peut être étendue à une solution pure $\mathcal{C}_{i+1}$. Comme on a vu $\sigma = \theta_1\theta_2\theta_3\sigma_4$ est

telle que pour toute contrainte $T_{i+1} \stackrel{?}{\vdash} x_{i+1} \in \mathcal{C}$, il existe une recette ζ telle que $\zeta[T_{i+1}]\sigma\downarrow = x_{i+1}\sigma$ et, pour tout $\zeta' \in \mathsf{St}(\zeta)$, il existe un $v_{\zeta'} \in \mathcal{D}(\mathcal{D}(\mathsf{St}(CS_3)))$ tel que $\zeta'[T_{i+1}]\sigma\downarrow = v_{\zeta'}\sigma_4$. Soit M l'ensemble de tous les $v_{\zeta'}$, avec $\zeta' \in \mathsf{St}(\zeta)$ et soit l'ordre $<$ sur M défini par $v_{\zeta_1} < v_{\zeta_2}$ si, et seulement si, $\zeta_1 \in \mathsf{St}(\zeta_2)$. Soit $v_{i+1,1}, \ldots, v_{i+1,k_{i+1}}, v_{i,1+k_{i+1}} = x_{i+1}\theta_1\theta_2\theta_3$ une extension totale de $<$.

Montrons par récurrence sur $<$ que, pour chaque $1 \leq j \leq 1 + k_{i+1}$, il existe une recette pure $\zeta_{i+1,j}$ telle que $\zeta_{i+1,j}[T_{i+1}\sigma, v_{i+1,1}\sigma_4, \ldots, v_{i+1,j-1}\sigma_4]\downarrow = v_{i+1,j}\sigma_4$. Si $v_{i+1,j}$ est minimal, il existe par définition de $<$, une recette $\zeta_{i+1,j}$ telle que $\zeta_{i,j}[T_i+1\sigma]\downarrow = v_{i+1,j}\sigma_4$ & $\mathsf{Fact}(\zeta_{i+1,j}) \subseteq \mathcal{X}$ et on peut conclure. Sinon, il existe une recette χ telle que $\chi[T_{i+1}\sigma]\downarrow = v_{i+1,j}\sigma_4$, $\chi = \chi_0[\chi_1, \ldots, \chi_k]$, $\mathsf{Fact}(\chi) = \{\chi_1, \ldots, \chi_k\}$ et $\chi_1[T_{i+1}\sigma]\downarrow, \ldots, \chi_k[T_{i+1}\sigma]\downarrow < v_{i+1,j}$. Nous pouvons donc choisir $\zeta_{i+1,j} = \chi_0$, car $\mathsf{Fact}(\chi_0) = \mathsf{Var}(\chi_0)$, par le lemme 6.

Par conséquent, $\sigma \cup \{z_{i+1,j} \mapsto v_{i+1,j}\sigma_4 \mid 1 \leq j \leq k_{i+1}\}$ est une solution pure de

$$\begin{array}{lcl}
T_{i+1} & \stackrel{?}{\vdash} & z_{i+1,1} \\
T_{i+1}, z_{i+1,1} & \stackrel{?}{\vdash} & z_{i+1,2} \\
 & \cdots & \\
T_{i+1}, z_{i+1,1}, \ldots, z_{i+1,k_{i+1}-1} & \stackrel{?}{\vdash} & z_{i+1,k_{i+1}} \\
T_{i+1}, z_{i+1,1}, \ldots, z_{i+1,k_{i+1}-1}, z_{i+1,k_{i+1}} & \stackrel{?}{\vdash} & x_{i+1}
\end{array}$$

et, par conséquent, de $\mathcal{C}_{i+1}$. □

Exemple 36 *Considérons le système de contraintes suivant dans* EP*, avec* $a, b, c \in C_{priv}$*:*

$$C = \left\{ \begin{array}{lcl} a \star b & \stackrel{?}{\vdash} & x \\ a \star b, exp(x, c), J_\star(b \star c), h(a) + c & \stackrel{?}{\vdash} & y \end{array} \right.$$

$\sigma = \{x \mapsto h(a \star b); y \mapsto a\}$ *est une solution de* $\mathcal{C}$*. Voyons la branche dans la transformation ci-dessus qui nous amène à un système pur, dont une extension de* σ *est une solution.*

D'abord, on calcule un variant, devinant que $x = h(z)$ *et la forme normale de* $exp(x, c)$ *est* $h(z \star c)$*. Ensuite, on devine que* θ_2, θ_3 *sont l'identité et on devine quels termes dans* $\{a, b, c, z, h(a), h(b), h(c), h(z)\} \subseteq \mathcal{D}(\mathcal{D}(\mathsf{St}_{\mathsf{EP}}(CS_3)))$ *sont déductibles et dans quel ordre. Un des systèmes ainsi obtenus est le suivant:*

$$C' = \left\{ \begin{array}{lcl@{\quad}lcl} a \star b & \stackrel{?}{\vdash} & x & x & = & h(z) \\ a \star b, exp(x, c), J_\star(b \star c), h(a) + c & \stackrel{?}{\vdash} & z_1 & z_1 & = & h(a) \\ a \star b, exp(x, c), J_\star(b \star c), h(a) + c, z_1 & \stackrel{?}{\vdash} & z_2 & z_2 & = & c \\ a \star b, exp(x, c), J_\star(b \star c), h(a) + c, z_1, z_2 & \stackrel{?}{\vdash} & y & & & \end{array} \right.$$

La substitution $\sigma' = \sigma \uplus \{z_1 \mapsto h(a); z_2 \mapsto c\}$ *est une solution pure de* C'*: les recettes utilisées sont* $exp(x_2, x_3)$*,* $x_4 + J_+(x_5)$*,* $x_1 * J_\star(x_3 \star x_6)$ *pour obtenir respectivement* $z_1\sigma', z_2\sigma', y\sigma'$*, à chaque fois* x_i *étant remplacé par le* i*-ème terme dans* T_j*.*

6.6 Seconde réduction: stabilisation des théories

Pour un système de contraintes $\mathcal{C} = \{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_n \stackrel{?}{\vdash} x_n\} \cup \mathcal{S}$, notons $\mathsf{D}(\mathcal{C}) = \{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_n \stackrel{?}{\vdash} x_n\}$ et $\mathsf{Eq}(\mathcal{C}) = \mathcal{S}$. Dans cette section, nous réduisons la recherche des solutions pures d'un système $\mathcal{C}$ à la recherche des solutions pures σ tels que chaque terme dans les contraintes de déductibilité a la théorie stable par instantiation et normalisation: $\forall t \in \mathsf{St}(\mathsf{D}(\mathcal{C})), \top(t\sigma\downarrow) = \top(t)$. Cette propriété est importante pour pouvoir fixer les termes étrangers dans chaque contrainte de deductibilité et les abstraire ensuite par des constantes. Nous appelerons *stables* les systèmes de contraintes pour lesquels on ne considère que des solutions σ qui ont la propriété ci-dessus.

Soit $\mathsf{L}(\mathcal{C}) = \overline{\overline{\overline{F}}}(\overline{G}(\mathsf{St}(\mathsf{D}(\mathcal{C}))) \cup \mathcal{C}_{\mathcal{R}})$, où $\overline{\overline{\overline{F}}}(t) = \{t, \mathsf{h}(t), \mathsf{h}(\mathsf{h}(t)), \mathsf{h}(\mathsf{h}(\mathsf{h}(t)))\}$. Nous montrons dans le lemme 29 que, si $\top(u\sigma\downarrow) \neq \top(u)$, il existe un autre terme v dans $\mathsf{L}(\mathcal{C})$, strictement inférieur (par rapport à un ordre bien choisi $\leq_{\mathcal{C}}$), tel que $u\sigma\downarrow = v\sigma\downarrow$. On rajoute alors $u = v$ à la partie équationnelle du système de contraintes et on remplace toutes les occurrences de u par v. De plus, $\leq_{\mathcal{C}}$ est défini de telle manière que le système résultant satisfait toujours l'origination. Après des remplacements successifs, on "stabilise" la théorie des termes, nous permettant de fixer les termes étrangers.

Le choix des théories de variables et des égalités entre sous-termes. On veut deviner la théorie d'un terme (après instantiation et normalisation). De tels choix sont enregistrés dans le système de contraintes en utilisant des contraintes additionelles $H(u) = \mathsf{th}$, où $\mathsf{th} \in \{1, \ldots, \ell\} \cup \mathcal{F}$. Une substitution (close) σ satisfait $H(u) = \mathsf{th}$ si $\top(u\sigma\downarrow) = \mathsf{th}$.

Pour toute variable x dans $\mathcal{C}$, on devine une théorie, en rajoutant une contrainte $H(x) = \mathsf{th}$. Par abus de notation, on pose $\top(x) = \mathsf{th}$ quand la contrainte $H(x) = \mathsf{th}$ est dans $\mathcal{C}$. Notons que la fonction $\top$ est alors modifiée en correspondance pour les termes avec des variables: pour EP par exemple, si $\top(x) = +$, alors $\top(\mathsf{h}(x)) = \bullet$.

Comme d'habitude dans les procédures de combinaison, on devine aussi toutes les égalités possibles entre les termes dans $\mathsf{L}(\mathcal{C})$. Ces choix sont enregistrés en rajoutant des égalités à la partie équationnelle du système. Après ce pas (non-déterministe), on peut ne considérer que des solutions σ telles que pour tout $u, v \in \mathsf{L}(\mathcal{C})$, si $u\sigma\downarrow = v\sigma\downarrow$, alors $u =_{\mathsf{Eq}(\mathcal{C})} v$: toutes les égalités qui sont vraies après l'instantiation sont déjà une conséquence de la partie équationnelle, $\mathsf{Eq}(\mathcal{C})$, de $\mathcal{C}$.

Exemple 37 *Pour* EP*, soit*

$$\mathcal{C} = \left\{ \begin{array}{rcl} a \bullet b, c & \stackrel{?}{\vdash} & x \\ a \bullet b, c, x + J_+(c), x \bullet J_\bullet(b) & \stackrel{?}{\vdash} & y \\ a \bullet b, c, x + J_+(c), x \bullet J_\bullet(b), y \bullet J_\bullet(a) & \stackrel{?}{\vdash} & z \end{array} \right.$$

Nous allons montrer dans la suite comment nos transformations mènent de $\mathcal{C}$ à un système de contraintes stable pour la solution σ: $x\sigma = a \bullet b + c, y\sigma = (a \bullet b + c) \bullet a, z\sigma = a$. Notons que, dans $\mathcal{C}$, les termes $x + J_+(c)$ et $y \bullet J_\bullet(a)$ ne sont pas stables par rapport à σ, car $\top(x\sigma + J_+(c)) = \bullet \neq +$ et $\top(y \bullet J_\bullet(a)) = + \neq \bullet$.

D'abord, nous devinons les égalités $x + J_+(c) = a \bullet b, y \bullet J_\bullet(a) = x$ et les théories de variables $H(x) = +, H(y) = \bullet$ et $H(z) = a$

Stabilisation des théories: lemmes préliminaires. Pour un système de contraintes $\mathcal{C}$, on définit l'ordre $\leq_\mathcal{C}$ entre les termes de $\mathsf{L}(\mathcal{C})$ comme suit. Pour $x, y \in \mathsf{Var}(\mathsf{D}(\mathcal{C}))$, $x \succ_\mathcal{C} y$ s'il existe une contrainte $T \overset{?}{\vdash} x \in \mathcal{C}$, telle que $y \in \mathsf{Var}(T)$ & $x \notin \mathsf{Var}(T)$. Ceci définit en fait un ordre $\succeq_\mathcal{C}$, grâce à l'origination et à la monotonie de $\mathcal{C}$. Cet ordre est étendu aux symboles de $\mathcal{F}$ par $x \succ_\mathcal{C} f$, pour $x \in \mathsf{Var}(\mathsf{D}(\mathcal{C})), f \in \mathcal{F}$, et $f \succ_\mathcal{C} \mathsf{h}$, pour $f \in \mathcal{F} \setminus \{\mathsf{h}\}$. Alors $\leq_\mathcal{C}$ est l'ordre MPO (multiset path ordering [DJ90]) défini à partir de la relation de priorité $\succeq_\mathcal{C}$.

Exemple 38 *Dans l'exemple 37, nous avons $x \prec_\mathcal{C} y \prec_\mathcal{C} z$ et $a \bullet b \leq_\mathcal{C} x + J_+(c), x \leq y \bullet J_\bullet(a)$.*

Notre but est de prouver la propriété suivante: pour tout $u \in \mathsf{St}(\mathcal{C})$ et toute solution σ de $\mathcal{C}$, si $\top(u\sigma\downarrow) \neq \top(u)$, alors il existe un terme $v \in \mathsf{L}(\mathcal{C})$ tel que $v <_\mathcal{C} u$ et $u\sigma\downarrow = v\sigma\downarrow$.

Nous montrons d'abord que, si la théorie d'un terme change par normalisation, il est égal à l'un des ses sous-termes stricts. En fait, c'est une simple conséquence d'un lemme déjà prouvé pour la localité:

Lemme 25 *Pour tout terme u tel que $\top(u\downarrow) \neq \top(u)$, il existe un $u' \in \overline{G}(u)$ tel que, ou bien $u'\downarrow \in \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$, ou bien $top(u') \neq \mathsf{h}$ et $\exists w \in \overline{G}(\mathsf{St}_<(u'))$ tel que $u'\downarrow \in \overline{\overline{F}}(w\downarrow)$.*

Preuve: Supposons d'abord que $top(u\downarrow) \neq top(u)$. Par le lemme 15 (section 6.3), il existe alors un $w \in \overline{G}(\mathsf{St}_<(u)) \cup \mathcal{C}_\mathcal{R}$ tel que $u\downarrow \in \overline{\overline{F}}(w)\downarrow = \overline{\overline{F}}(w\downarrow)\downarrow \subseteq \overline{\overline{F}}(w\downarrow) \cup \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$, en utilisant le lemme 11 (section 6.2.2) pour l'inclusion. De plus, si $u\downarrow \notin \mathcal{C}_\mathcal{R}$, $top(u) \notin \mathsf{h}$, par définition des systèmes purs. On peut donc conclure avec $u' = u$.

Si $top(u\downarrow) = top(u)$, puisque $\top(u\downarrow) \neq \top(u)$, on a forcement $u = \mathsf{h}(u')$, $\top(u'\downarrow) \neq \top(u')$ et, si $u\downarrow \notin \overline{F}(\mathcal{C}_\mathcal{R})$, $top(u') \neq \mathsf{h}$. De plus, en utilisant $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{thsource}(\mathsf{h}) = \emptyset$, on a aussi $top(u'\downarrow) \neq top(u')$. Comme vu ci-dessus, on a alors $u'\downarrow \in \overline{\overline{F}}(\overline{G}(\mathsf{St}_<(u'))\downarrow)$ et nous pouvons conclure. □

Une solution *développée* de $\mathcal{C}$ est une substitution θ telle que, pour toute contrainte $T \overset{?}{\vdash} x \in \mathcal{C}$, il existe une recette pure ζ telle que $x\theta = \zeta[T\theta]$ (sans normaliser) et pour toute équation $s_1 = s_2 \in \mathsf{Eq}(\mathcal{C})$, $s_1\theta\downarrow = s_2\theta\downarrow$. Le but de cette notion est de nous permettre d'accéder au termes de $\mathcal{C}$ qui ont contribué aux preuves d'une solution.

Exemple 39 *Pour l'exemple 37, la solution developpée θ qui correspond à σ est*

- $x\theta = a \bullet b + c$
- $y\theta = (a \bullet b + c + J_+(c)) \bullet (a \bullet b + c) \bullet J_\bullet(b)$
- $z\theta = y\theta \bullet J_\bullet(x\theta)$

Observons comment θ explicite les termes de $\mathcal{C}$ qui sont utilisées dans la construction de σ.

Nous montrons d'abord un résultat qui nous aide dans l'analyse qui suit: on peut éliminer les recettes pures dont le symbole de la racine est h.

Lemme 26 *Pour tout système de contraintes $\mathcal{C}$ on peut calculer un ensemble de systèmes $\mathcal{C}_1, \ldots, \mathcal{C}_n$ tels que $\mathcal{C}$ a une solution pure σ ssi un $\mathcal{C}_i$, $1 \leq i \leq n$, a une solution σ' telle que, pour toute contrainte $T \stackrel{?}{\vdash} x \in \mathcal{C}_i$, il existe une recette pure ζ telle que $\zeta[T_i\sigma']\downarrow = x_i\sigma'$ et, en plus, $top(\zeta) \neq \mathsf{h}$.*

Preuve: Soit σ une solution de $\mathcal{C}$, $T \stackrel{?}{\vdash} x \in \mathcal{C}$ et ζ une recette pure telle que $\zeta[T\sigma]\downarrow = x\sigma$ et $\zeta = \mathsf{h}(\zeta')$. Notons que, puisque ζ est pure, on a $\zeta' \in \mathcal{X}$ ou bien $\top(\zeta') \in \mathsf{thsource}(\mathsf{h})$. Par la proposition 1, on a $\mathsf{Fact}(\zeta') \subseteq \mathsf{Fact}(\zeta)$. Donc ζ' est pure aussi. De plus, puisque $\mathsf{thsource}(\mathsf{h}) \cap \mathsf{thdest}(\mathsf{h}) = \emptyset$, on a $top(\zeta') \neq \mathsf{h}$.

Si, pour une nouvelle variable x', on remplace $T \stackrel{?}{\vdash} x$ par $T \stackrel{?}{\vdash} x'$ et toutes les occurrences de x dans $\mathcal{C}$ par $\mathsf{h}(x')$, on obtient un système de contraintes $\mathcal{C}'$ qui a une solution $\sigma' = \sigma \cup \{x' \mapsto \zeta'[T\sigma]\downarrow\} \setminus \{x \mapsto \zeta[T\sigma]\downarrow\}$. De plus, toute solution σ' de $\mathcal{C}'$ peut s'etendre à une solution $\sigma = \sigma' \cup \{x \mapsto \mathsf{h}(x'\sigma')\} \setminus \{x' \mapsto x'\sigma'\}$ de $\mathcal{C}$.

En choisissant de manière non-deterministe l'ensemble des tels x, on obtient la transformation $\mathcal{C} \to \mathcal{C}_1, \ldots, \mathcal{C}_n$ voulue. □

Dans la suite, on suppose donc que pour toute recette de preuve ζ, $top(\zeta) \neq \mathsf{h}$.

Pour une solution developpée θ de $\mathcal{C}$, si $\top(u\theta\downarrow) \neq \top(u\theta)$, le lemme 25 nous dit, grosso modo, que $u\theta$ est égal à un des ses sous-termes stricts. Le lemme suivant nous montre que, grâce au fait que la solution considérée est pure, ce sous-terme est une instance d'un terme de $\mathsf{L}(\mathcal{C})$ plus petit que u:

Lemme 27 *Pour tout terme $u \in \mathsf{St}(\mathsf{D}(\mathcal{C}))$ et toute solution développée θ de $\mathcal{C}$, pour tout $w \in \overline{G}(\mathsf{St}_<(u\theta))$, il existe un $v \in \overline{G}(\mathsf{St}(\mathsf{D}(\mathcal{C}))), v <_{\mathcal{C}} u$ tel que $w = v\theta$.*

Preuve: Soit $\mathcal{C} = \{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_n \stackrel{?}{\vdash} x_n\} \cup \mathcal{S}$ et $u \in \mathsf{St}(T_i, x_i)$, pour un certain $1 \leq i \leq n$.

On fait la preuve par récurrence sur i, $1 \leq i \leq n$. Si $i = 1$ et $u \in \mathsf{St}(T_i)$, u est clos et le lemme est immediat, car on peut choisir $v = w$. Si $u = x_1$, par la définition d'une solution pure, on a $u\theta = \zeta[u_1, \ldots, u_k]$, pour une recette pure ζ

et des termes $u_1, \ldots, u_k \in T_1$. Par la proposition 1, on a $\mathsf{Fact}(\zeta[u_1, \ldots, u_k]) \subseteq \mathsf{Fact}(u_1, \ldots, u_k) \cup \overline{G}(u_1, \ldots, u_k)$ et donc $\overline{G}(\mathsf{St}_<(u\theta)) \subseteq \overline{G}(\mathsf{St}(T_1))$. Puisque, pour tout $v \in \overline{G}(\mathsf{St}(T_1))$, on a $v < x_1$, on conclut.

Supposons maintenant que $i > 1$. Soit d'abord $u \in \mathsf{St}(T_i)$. On fait une récurrence sur la taille de u. Si u est une variable, par l'origination de $\mathcal{C}$, on peut conclure par l'hypothèse de récurrence. Si u est une constante, on a $\mathsf{St}_<(u\theta) = \emptyset$ et le lemme est trivialement satisfait. Autrement, soit $u = \zeta[u_1, \ldots, u_k], \mathsf{Fact}(u) = \{u_1, \ldots, u_k\}$. Notons que ζ est pure, par le lemme 6. Par la proposition 1, on obtient $\mathsf{Fact}(u\theta) \subseteq \mathsf{Fact}(u_1\theta, \ldots, u_k\theta) \cup \overline{G}(u_1\theta, \ldots, u_k\theta)$. Par conséquent, $\mathsf{St}_<(u\theta) \subseteq \overline{G}(\mathsf{St}(u_1\theta, \ldots, u_k\theta))$, en utilisant aussi le lemme 12 pour commuter St et $\overline{G}$. Par le lemme 12 encore une fois, on obtient $\overline{G}(\mathsf{St}_<(u\theta)) \subseteq \overline{G}(\overline{G}(\mathsf{St}(u_1\theta, \ldots, u_k\theta))) \subseteq \overline{G}(\mathsf{St}(u_1\theta, \ldots, u_k\theta))$. Par l'hypothèse de récurrence, on a $\forall 1 \leq j \leq k. w \in \overline{G}(\mathsf{St}_<(u_j\theta)) \implies \exists v \in \overline{G}(\mathsf{St}(\mathsf{D}(\mathcal{C}))), v <_\mathcal{C} u_j <_\mathcal{C} u : w = v\theta$. Il nous reste a considérer le cas $w \in \overline{G}(u_j\theta)$, pour un j, $1 \leq j \leq k$. Si $w = u_j\theta$, on prend $v = u_j$ et on conclut.

Sinon, soit $w \in \overline{G}(u_j\theta) \setminus \{u_j\theta\}$. Supposons d'abord que u_j n'est pas une variable. Alors $u_j = \mathsf{h}(u'_j)$, $w = u'_j\theta$ et $u'_j \in \overline{G}(\mathsf{St}_<(u))$, puisque $u_j \in St_<(u)$. On peut donc choisir $v = u'_j \in \overline{G}(\mathsf{St}_<(u))$, puisque $u'_j <_\mathcal{C} u_j <_\mathcal{C} u$. Sinon, si u_j est une variable, soit $x_{i'}$, pour un $i' < i$, la variable minimale (par rapport à $<_\mathcal{C}$) telle que $x_{i'}\theta = u_j\theta$. On a $u_j\theta = \zeta'[v_1\theta, \ldots, v_m\theta]$, pour des termes $v_1, \ldots, v_m \in T_{i'}$ est une recette pure ζ'. Si ζ' n'est pas une variable, alors $top(\zeta') \neq \mathsf{h}$ (lemme 26) et donc $\overline{G}(u_j\theta) = \{u_j\theta\}$, contredisant le choix de w. Par conséquent, ζ' est nécessairement une variable et on obtient $w \in \overline{G}(v_l\theta) \setminus \{v_l\theta\}$, pour un l, $1 \leq l \leq m$. Puisque x'_i est minimal, v_l n'est pas une variable. On a donc $v_l = \mathsf{h}(v_{l'})$ et on peut prendre $v = v_{l'} \in \overline{G}(\mathsf{St}(\mathsf{D}(\mathcal{C})))$ et $v'_l <_\mathcal{C} u$, pour conclure.

Il nous reste le cas $u = x_i$. On a alors $u\theta = \zeta[u_1\theta, \ldots, u_k\theta]$, pour une recette pure ζ et $u_1, \ldots, u_k \in T_i \subseteq \mathsf{St}(T_i)$. On peut donc conclure comme avant, car on est dans la même situation. □

En mettant les lemmes 25 et 27 ensemble, nous avons:

Lemme 28 *Pour tout terme $u \in \mathsf{St}(\mathsf{D}(\mathcal{C}))$ et toute solution développée θ de $\mathcal{C}$, si $\top(u\theta\downarrow) \neq \top(u\theta)$, il existe un $v \in \mathsf{L}(\mathcal{C}), v <_\mathcal{C} u$ tel que $u\theta\downarrow = v\theta\downarrow$.*

Preuve: Si $u\theta\downarrow = w \in \overline{\overline{F}}(\mathcal{C}_\mathcal{R})$, on conclut, car $\mathsf{Var}(u) \neq \emptyset$ et donc $w <_\mathcal{C} u$. Sinon, par le lemme 25, on a deux cas:

- ou bien $u\theta\downarrow \in \overline{\overline{F}}(w\downarrow) = \{w\downarrow, \mathsf{h}(w\downarrow), \mathsf{h}(\mathsf{h}(w\downarrow))\}$, pour un $w \in \overline{G}(\mathsf{St}_<(u\theta))$, et $top(u\theta) \neq \mathsf{h}$. Par le lemme 27, il existe un $v' \in \overline{G}(\mathsf{St}(\mathsf{D}(\mathcal{C}))), v' <_\mathcal{C} u$, tel que $w = v'\theta$. Puisque $top(u\theta) \neq \mathsf{h}$, on a $u\theta \notin \{\mathsf{h}(w), \mathsf{h}(\mathsf{h}(w))\}$. On deduit que $u \notin \{\mathsf{h}(v'), \mathsf{h}(\mathsf{h}(v'))\}$. Par définition de $<_\mathcal{C}$, ceci implique $\{v', \mathsf{h}(v'), \mathsf{h}(\mathsf{h}(v'))\} <_\mathcal{C} u$ et on peut donc choisir $v \in \{v', \mathsf{h}(v'), \mathsf{h}(\mathsf{h}(v'))\}$.

- ou bien $u = \mathsf{h}(u')$ et u' est comme dans le cas précédent. Soit alors $v' \in \overline{\overline{F}}(\overline{G}(\mathsf{St}(\mathsf{D}(C)))) \cup \mathcal{C}_\mathcal{R})$ tel que $v' <_\mathcal{C} u$ et $u'\theta\downarrow = v'\theta\downarrow$. Nous avons montrer ci-dessus qu'un tel v' existe. Alors, pour $v = \mathsf{h}(v')$, on a $v \in$

$\overline{\overline{\overline{F}}}(\overline{G}(\mathsf{St}(\mathsf{D}(C))) \cup \mathcal{C}_{\mathcal{R}}) = \mathsf{L}(\mathcal{C})$ et $u\theta\downarrow = v\theta\downarrow$. Pour conclure, notons que $v <_{\mathcal{C}} u$, puisque $u = \mathsf{h}(u')$ et $v' <_{\mathcal{C}} u'$. □

Par conséquent, quand la théorie d'un terme n'est pas stable par substitution et normalisation, il est égal à un sous-terme plus petit du système:

Lemme 29 *Soit $\mathcal{C}$ un système de contraintes. Pour tout terme $u \in \mathsf{St}(\mathsf{D}(\mathcal{C}))$ et toute solution σ de $\mathcal{C}$, si $\top(u\sigma\downarrow) \neq \top(u)$, alors il existe un terme $v \in \mathsf{L}(\mathcal{C})$ tel que $v <_{\mathcal{C}} u$ et $u\sigma\downarrow = v\sigma\downarrow$.*

Preuve: Notons que, puisque σ satisfait les contraintes sur la théorie des variables de $\mathcal{C}$, nous avons $\top(u) = \top(u\sigma)$. Soit θ la solution développée qui correspond à σ. Si $\top(u\theta) = \top(u\sigma)$, on conclut par le lemme 28, car alors $\top(u\theta) = \top(u\sigma) \neq \top(u\sigma\downarrow) = \top(u\theta\downarrow)$. Supposons maintenant que $\top(u\theta) \neq \top(u\sigma)$. Ceci est possible seulement quand $u = x$ ou $u = \mathsf{h}(x)$, pour un $x \in \mathsf{Var}(\mathsf{D}(\mathcal{C}))$. Dans le premier cas, $u\sigma$ est en forme normale est donc on contredit $\top(u\sigma) \neq \top(u\sigma\downarrow)$. Dans le deuxième cas, par la définition des systèmes de réécriture purs, $u\theta$ n'est pas en forme normale seulement si $x\theta \in \mathcal{C}_{\mathcal{R}}$ et $u\theta\downarrow \in \mathcal{C}_{\mathcal{R}}$. On peut donc choisir $v \in \mathcal{C}_{\mathcal{R}}$ et conclure le lemme. □

Exemple 40 *Pour l'exemple 37, pour le termes non-stables par rapport à σ - $x + J_+(b)$ et $y \bullet J_\bullet(a)$ - il existe $a \bullet b, x \in \mathsf{L}(\mathcal{C})$ tels que $a \bullet b <_{\mathcal{C}} x + J_+(b), a \bullet b = x\sigma + J_+(b)\downarrow$ et $x <_{\mathcal{C}} y \bullet J_\bullet(a), x\sigma = y\sigma \bullet J_\bullet(a)\downarrow$.*

Stabilisation des théories: remplacement dans les contraintes. Par le lemme 29, si la théorie u n'est pas stable par substitution et réécriture, on sait que $u\sigma\downarrow$ est égal à un terme plus petit $v \in \mathsf{L}(\mathcal{C})$. Grâce au fait que les égalités entre les termes de $\mathsf{L}(\mathcal{C})$ ont été devinées, nous avons $u =_{\mathsf{Eq}(\mathcal{C})} v$: on peut remplacer u par v, en regardant seulement les informations qu'on a déjà dans le système de contraintes. En itérant cette procédure on obtient un système dont tous les sous-termes ont des théories stables.

Procédure $\mathcal{C} \to \mathcal{C}'$. Pour un $u \in \mathsf{St}(\mathsf{D}(\mathcal{C}))$ pour lequel il existe un $v \in \mathsf{L}(\mathcal{C})$ tel que $u =_{\mathsf{Eq}(\mathcal{C})} v$, $v <_{\mathcal{C}} u$: on remplace u par v dans $\mathsf{D}(\mathcal{C})$ pour obtenir $\mathcal{C}'$.

Le lemme suivant nous permet de conclure que l'itération de cette procédure termine avec le résultat souhaité:

Lemme 30 *Dans la transformation $\mathcal{C} \to \mathcal{C}'$ décrite ci-dessus, nous avons les propriétés suivantes:*

- $|\mathsf{St}(\mathsf{D}(\mathcal{C}'))| < |\mathsf{St}(\mathsf{D}(\mathcal{C}))|$
- $\mathsf{Eq}(\mathcal{C}') = \mathsf{Eq}(\mathcal{C})$ *et, si $u_1, u_2 \in \mathsf{L}(\mathcal{C}')$ et $u_1\sigma\downarrow = u_2\sigma\downarrow$, alors $u_1 =_{\mathsf{Eq}(\mathcal{C}')} u_2$.*

Preuve: Le premier point est évident, car le remplacement de u par v fait disparaître u de $\mathsf{St}(\mathsf{D}(\mathcal{C}'))$ et, en même temps, ne peut pas faire differents deux termes qui etaient égaux.

Prouvons maintenant la deuxième partie. $\mathsf{Eq}(\mathcal{C}') = \mathsf{Eq}(\mathcal{C})$ est une propriété immédiate par construction. Soit maintenant $u_1, u_2 \in \mathsf{L}(\mathcal{C}')$ tels que $u_1\sigma\downarrow =$

$u_2\sigma\downarrow$. Si $u_1, u_2 \in \mathsf{L}(\mathcal{C})$, nous avons $u_1 =_{\mathsf{Eq}(\mathcal{C})} u_2$ et donc $u_1 =_{\mathsf{Eq}(\mathcal{C}')} u_2$. Sinon, nous avons $u_1 = v_1[v]$ et $u_2 = v_2[v]$, avec $v_1[u], v_2[u] \in \mathsf{L}(\mathcal{C})$. Donc $v_1[u] =_{\mathsf{Eq}(\mathcal{C})} v_2[u]$ et, puisque $u =_{\mathsf{Eq}(\mathcal{C})} v$, on conclut $u_1 = v_1[v] =_{\mathsf{Eq}(\mathcal{C}')} v_2[v] = u_2$. □

Par conséquence, on peut maintenant supposer que toutes les solutions préservent la théorie de tout sous-terme des contraintes de deductibilité:

Corollaire 4 *Soit $\mathcal{C}'$ le résultat des applications succéssives de la transformation ci-dessus: i.e. $\mathcal{C} \rightarrow^* \mathcal{C}'$ et $\mathcal{C}'$ est irréductible. Alors $\mathcal{C}'$ est bien défini, $Sol(\mathcal{C}) = Sol(\mathcal{C}')$ et, pour toute solution pure σ de $\mathcal{C}'$, pour tout $u \in \mathsf{St}(\mathsf{D}(\mathcal{C}'))$, $\top(u\sigma\downarrow) = \top(u)$.*

Preuve: $\mathcal{C}'$ est bien défini par le premier point du lemme 30: la transformation termine. Le deuxième point du lemme 30 est utilisé pour ne pas redeviner les égalités à chaque pas. Chaque pas de transformation préserve l'ensemble de solutions, car il remplace un terme par un autre qui est une conséquence des contraintes du système. Finalement, s'il existe une solution pure σ de $\mathcal{C}'$ et un terme $u \in \mathsf{St}(\mathsf{D}(\mathcal{C}'))$ pour lequel $\top(u\sigma\downarrow) \neq \top(u)$, par le lemme 29 on obtient une contradiction à l'irréductibilité de $\mathcal{C}'$. □

Exemple 41 *$\mathcal{C}$ de l'exemple 37 est finalement transformé dans le système $\mathcal{C}'$ ci-dessous,*

$$\mathcal{C}' = \left\{ \begin{array}{rcl} a \bullet b, c & \stackrel{?}{\vdash} & x \\ a \bullet b, c, a \bullet b, x \bullet J_\bullet(b) & \stackrel{?}{\vdash} & y \\ a \bullet b, c, a \bullet b, x \bullet J_\bullet(b), x & \stackrel{?}{\vdash} & z \\ & x + J_+(c) = a \bullet b, y \bullet J_\bullet(a) = x & \\ & H(x) = +, H(y) = \bullet, H(z) = a & \end{array} \right.$$

Chapitre 7

Résolution des systèmes purs et stables pour l'étude de cas

Les étapes précédentes de réduction sont générales. On a vu au long du chapitre 6 qu'elles s'appliquent à notre étude de cas EP. Dans ce chapitre, nous en déduisons une procédure de décision pour la satisfaisabilité des systèmes de contraintes modulo EP: on montre que les systèmes purs et stables pour EP sont de plus réductibles à des systèmes diophantiens linéaires. Cependant, une réduction naïve conduit à des équations Diophantiennes non-linéaires [BCD07a]. Une astuce spécifique aux groupes Abéliens, déjà utilisée dans [Shm04], permet de linéariser les membres gauches de contraintes. Elle est utilisée dans la deuxième étape de notre réduction, qui en a quatre:

1. On devine quel type de recette pure est utilisée pour chaque contrainte, i.e. dans quelle théorie la contrainte est a résoudre. On montre qu'il suffit de considérer trois types de recettes.

2. Dans les membres gauches de contraintes, on élimine les variables dont la théorie est la même que celle de la contrainte où elles apparaissent, en utilisant les propriétés AG de $\mathcal{R}_{\mathsf{EP}}$.

3. Ça nous permet de réduire les contraintes de déductibilité à des systèmes des équations avec des paramètres formels qui ne s'appliquent pas à des variables.

4. On résout un problème d'unification avec en plus des équations Diophantiennes, qui sont linéaires grâce à la propriété des systèmes obtenus dans l'étape 3.

7.1 Réduction à trois types de recettes.

Notons que, pour la recherche des solutions, il suffit de considérer seulement des recettes en forme normale. De plus, par le lemme 26, on peut supposer que la racine des recettes n'est pas h, le symbole d'interface pour EP.

Lemme 31 *Pour la théorie* EP*, toute recette pure, normalisée et dont la racine n'est pas* h *a l'une des formes suivantes:*

$$\begin{array}{ll} \zeta_+[x_1,\ldots,x_n] & \zeta = exp(y_0, \zeta_\star[x_1,\ldots,x_n]) \\ \zeta_\star[x_1,\ldots,x_n] & \zeta_\bullet[x_1,\ldots,x_n] \bullet h(\zeta_+[y_1,\ldots,y_m]), n \geq 1 \end{array}$$

Où $\zeta_\circ$ *est une recette dans* $\mathcal{T}(\{\circ, J_\circ, e_\circ\}, \mathcal{X})$*, pour* $\circ \in \{+, \star, \bullet\}$*.*

Preuve: Par la definition des facteurs dans l'exemple 14 (page 37), il est clair que ces recettes sont pures.

Soit maintenant ζ une recette en forme normale qui n'est dans aucun des cas ci-dessus. Dans la suite, u est un terme non-variable. Nous considérons les cas possibles pour $\top(\zeta)$:

Cas $top(\zeta) \in \{+, J_+\}$**.** Alors $\zeta = \zeta_+[u]$, avec $top(u) \notin \{+, J_+, e_+\}$. Donc $u \in \mathsf{St}(\zeta)$ et ζ n'est pas pure.

Cas $top(\zeta) \in \{\star, J_\star\}$**.** Alors $\zeta = \zeta_\star[u]$, avec $top(u) \notin \{\star, J_\star, e_\star\}$. Donc $u \in \mathsf{St}(\zeta)$ et ζ n'est pas pure.

Cas $top(\zeta) = exp$**.** Soit $\zeta = exp(\zeta_1, \zeta_2)$. Puisque ζ est en forme normale, on a $top(\zeta_1) \notin \{exp, h\}$. Donc, si ζ_1 n'est pas une variable, ζ n'est pas pure. De plus, si ζ_2 n'est pas une variable et $\zeta_2 = \zeta_\star[u], top(u) \notin \{\star, J_\star, e_\star\}$, alors $u \in \mathsf{St}(\zeta)$ et donc ζ n'est pas pure.

Cas $top(\zeta) = \bullet$**.** Alors ou bien $\zeta = \zeta_\bullet[u], top(u) \notin \{\bullet, J_\bullet, e_\bullet, h\}$ ou bien $\zeta = \zeta_\bullet[h(\zeta_+[u])], top(u) \notin \{+, J_+, e_+\}$. Alors $u \in \mathsf{St}(\zeta)$ et donc ζ n'est pas pure.

□

Dans le premier pas de notre procédure, on devine, pour chaque contrainte $T_i \stackrel{?}{\vdash} x_i \in C$, lequel des quatre types de recettes ci-dessus est utilisé. Ce choix est enregistré en rajoutant une etiquette à la contrainte, $T_i \stackrel{?}{\vdash}_f x_i$, avec $f \in \{+, \star, \bullet, exp\}$. Assez facilement, on peut éliminer le type ζ_{exp}:

Elimination de ζ_{exp}*.* Pour chaque contrainte $T_i \stackrel{?}{\vdash}_{exp} x_i$,

1. On devine $u \in T_i$ et on rajoute une équation $x_i = exp(u, y_i)$ à la partie équationnelle de $\mathcal{C}$, où y_i est une nouvelle variable.
2. On remplace $T_i \stackrel{?}{\vdash}_{exp} x_i$ par $T_i \stackrel{?}{\vdash}_\star y_i$
3. On remplace x_i par $exp(u, y_i)$ dans chaque T_j tel que $x_i \in \mathsf{Var}(T_j)$.

Lemme 32 *La transformation ci-dessus $\mathcal{C} \to \bigvee \mathcal{C}_j$ est correcte et complète.*

Preuve: Il suffit de prouver le lemme quand seulement une contrainte est transformée: le cas général résulte par itération. Notons d'abord que, si $\mathcal{C} \to \mathcal{C}'$, alors $\mathcal{C}'$ est bien un système de contraintes, qui satisfait la monotonie et l'origination.

Soit $\mathcal{C} = \{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_i \stackrel{?}{\vdash}_{exp} x_i, T_{i+1} \stackrel{?}{\vdash} x_{i+1}, \ldots, T_n \stackrel{?}{\vdash} x_n\} \cup S$ tel que la i-ème contrainte est transformée. Alors $\mathcal{C}' = \{T_1 \stackrel{?}{\vdash} x_1, \ldots, T_i \stackrel{?}{\vdash}_\star y_i, T_{i+1}\{x_i \mapsto exp(u, y_i)\} \stackrel{?}{\vdash} x_{i+1}, \ldots, T_n\{x_i \mapsto exp(u, y_i)\} \stackrel{?}{\vdash} x_n\} \cup S \cup \{x_i = exp(u, y_i)\}$, pour un $u \in T_i$.

Correction: soit $\sigma' = \sigma \cup \{y_i \mapsto t\}$ une solution pure de $\mathcal{C}'$. Pour montrer que σ est une solution pure de $\mathcal{C}$, il suffit de montrer que:

- $T_i\sigma \vdash_{exp} x_i\sigma$;
- pour tout $j > i$, $T_j\{x_i \mapsto exp(u, y_i)\}\sigma'\downarrow = T_j\sigma\downarrow$;

Notons que $x_i\sigma' = exp(u\sigma', y_i\sigma')$ et $y_i\sigma' = \zeta_\star[T_i\sigma']\downarrow$, pour une recette $\zeta_\star \in \mathcal{T}(\{\star, J_\star, e_\star\}, \mathcal{X})$. Par construction, $u\sigma' = u\sigma$ et $T_i\sigma' = T_i\sigma$. On a donc $x_i\sigma' = x_i\sigma = exp(u\sigma, y_i\sigma')$ et $y_i\sigma' = \zeta_\star[T_i\sigma]\downarrow$. Par conséquence, $x_i\sigma = exp(u\sigma, \zeta_\star[T_i\sigma])\downarrow$ et $T_i\sigma \vdash_{exp} x_i\sigma$.

Le deuxième point suit par $exp(u, y_i)\sigma'\downarrow = x_i\sigma' = x_i\sigma$ et la convergence du système de réécriture.

Complètude: soit σ une solution pure de $\mathcal{C}$. On montre qu'il existe un $\mathcal{C}'$ construit comme ci-dessus et une solution pure σ' de $\mathcal{C}'$. En effet, puisque $T_i\sigma \vdash_{exp} x_i\sigma$, il existe un $u \in T_i$ et une recette (pure) $\zeta_\star$ tels que $x_i\sigma = exp(u\sigma, \zeta_\star[T_i\sigma])\downarrow$. Choisissons ce u pour construire $\mathcal{C}'$. Il est facile de montrer, suivant les mêmes lignes qu'avant, que $\sigma' = \sigma \cup \{y_i \mapsto \zeta_\star[T_i\sigma]\downarrow\}$ est une solution pure de $\mathcal{C}'$. □

Après ce pas, on peut supposer que $\mathcal{C}$ ne contient que des contraintes de type $T_i \stackrel{?}{\vdash}_\circ v_i$, avec $\circ \in \{+, \star, \bullet\}$.

7.2 Élimination des variables de membres gauches de contraintes.

À ce stade, on pourrait considérer les facteurs dans chaque T_i comme des constantes et traduire les contraintes de déductibilité dans des systèmes Diophantiens. Pourtant, ceci nous amenerait à des systèmes Diophantiens non-linéares, comme montré dans [BCD07a]: on doit utiliser l'origination et les propriétés des groupes Abeliens pour simplifier encore le problème.

D'abord on rend apparente la dualité de $\stackrel{?}{\vdash}_\bullet$: les membres gauches des contraintes sont dupliqués: $T \stackrel{?}{\vdash}_\bullet x$ devient $[T; T] \stackrel{?}{\vdash}_\bullet x$ et, par définition, σ est une solution de $[T_1; T_2] \stackrel{?}{\vdash}_\bullet x$ s'il existe des recettes pures $\zeta_\bullet \in \mathcal{T}(\{\bullet, J_\bullet, e_\bullet\}, \mathcal{X})$

et $\zeta_+ \in \mathcal{T}(\{+, J_+, e_+\}, \mathcal{X})$ telles que $\zeta_\bullet[T_1\sigma] \bullet h(\zeta_+[T_2\sigma])\downarrow = x\sigma$. Le but de cette transformation est de calculer les facteurs par rapport à $\bullet$ dans T_1 et par rapport à $+$ dans T_2.

Soit $T_i \overset{?}{\vdash}_\circ x_i$ une contrainte de $\mathcal{C} = \{T_1 \overset{?}{\vdash} x_1, \ldots, T_n \overset{?}{\vdash} x_n\} \cup \mathcal{S}$. Dans cette étape, on élimine les variables dans $\mathsf{Fact}_\circ(T_i)$ (définition 18, page 41), si leur théorie devinée est $\circ$ et la contrainte qui les introduit est de type $\circ$. Notons que, si $H(x) = \circ$ et x n'est pas introduite par une contrainte de type $\circ$, x est eliminé dans le deuxième pas général de réduction, par le lemme 28.

Soit $\circ \in \{+, \star, \bullet\}$, $T_i \overset{?}{\vdash}_\circ x_i$ (resp. $[T_i, T_i] \overset{?}{\vdash}_\bullet x_i$) dans $\mathcal{C}$. On définit $\mathsf{fv}_i^\circ(u)$ comme suit: soit $\mathcal{X}_i^\circ$ l'ensemble des variables x_j, $j < i$ telles que $\top(x_j) = \circ$ et $T_j \overset{?}{\vdash}_\circ x_j \in \mathcal{C}$ (resp. $[T_j, T_j] \overset{?}{\vdash}_\bullet x_j \in \mathcal{C}$) et $u = \zeta_\circ[x_{i_1}, \ldots, x_{i_k}, u_1, \ldots, u_m]$ tel que $\mathsf{Fact}_\circ(u) \cap \mathcal{X}_i^\circ = \{x_{i_1}, \ldots, x_{i_k}\}$. Alors $\mathsf{fv}_i^\circ(u) \overset{\text{def}}{=} \zeta_\circ[e_\circ, \ldots, e_\circ, u_1, \ldots, u_m]\downarrow$.

Si $\circ \in \{+, \star\}$ et $T_i \overset{?}{\vdash}_\circ x_i \in \mathcal{C}$, on remplace chaque $u \in T_i$ par $\mathsf{fv}_i^\circ(u)$. Si $\circ = \bullet$ et $[T_i, T_i] \overset{?}{\vdash}_\bullet x_i \in \mathcal{C}$, on remplace tout $u = u' \bullet h(u'')$ dans la première copie de T_i par $\mathsf{fv}_i^\bullet(u') \bullet h(\mathsf{fv}_i^+(u''))$ et tout u dans la deuxième copie par $\mathsf{fv}_i^+(u)$.

Exemple 42 *Une instance de la transformation décrite ci-dessus est:*

$$a \overset{?}{\vdash} x \quad \wedge \quad a, b + x \overset{?}{\vdash}_+ y \quad \Longrightarrow \quad a \overset{?}{\vdash} x \quad \wedge \quad a, b \overset{?}{\vdash}_+ y$$

Ceci preserve les solutions: si ζ_+ est telle que $\zeta_+[a, b + x\sigma]\downarrow = y\sigma$, alors, en reconstruisant $x\sigma$ à partir de a, on peut construire une nouvelle recette ζ'_+ telle que $\zeta'_+[a, b] = y\sigma$. Reciproquement, on peut construire ζ_+ à partir de ζ'_+ en soustrayant $x\sigma$, quand ceci est nécessaire. Ces idées ont été deja appliquées pour la résolution des contraintes modulo AG *dans [Shm04].*

De la même façon, $x_1 \bullet a \bullet h(x_2 + b)$ serait remplacé par $a \bullet h(b)$, si $\top(x_1) = \bullet$ et $\top(x_2) = +$. Nous verrons dans la section suivante un exemple qui illustrera mieux le cas special de $\bullet$.

Notons que les membres gauches ne sont plus ordonnés linéairement par inclusion, mais cette propriété de monotonie n'est plus nécessaire dans la suite.

Lemme 33 *La transformation $\mathcal{C} \to \mathcal{C}'$ décrite ci-dessus est correcte, complète et*

- *si $T_i \overset{?}{\vdash}_\circ x_i \in \mathcal{C}'$ et $\circ \in \{+, \star\}$, alors pour tout $x \in \mathsf{Fact}_\circ(T_i)$, $\top(x) \neq \circ$.*
- *si $[T'_i, T''_i] \overset{?}{\vdash}_\bullet x_i \in \mathcal{C}'$, alors:*
  - *pour tout $x \in \mathsf{Fact}_+(T''_i)$, $\top(x) \neq +$.*
  - *pour tout $u = u_1 \bullet \ldots \bullet u_n \bullet h(v_1 + \ldots + v_m) \in T'_i$ tel que $\mathsf{Fact}_\bullet(u) = \{u_1, \ldots, u_n, v_1, \ldots, v_m\}$, pour tout $x \in \mathcal{X}$, $x \in \{u_1, \ldots, u_n\} \Rightarrow \top(x) \neq \bullet$ et $x \in \{v_1, \ldots, v_m\} \Rightarrow \top(x) \neq +$.*

Preuve: Les propriétés de $\mathcal{C}'$ sont immédiates par construction et par le lemme 28.

Montrons la correction et la complétude: une substitution σ est une solution (pure) de $\mathcal{C}$ si et seulement si σ est une solution (pure) de $\mathcal{C}'$.

Pour chaque $T_i \overset{?}{\vdash}_\circ x_i \in \mathcal{C}$ (resp. $[T_i', T_i''] \overset{?}{\vdash}_\bullet x_i \in \mathcal{C}$), notons par $\mathsf{fv}_i^\circ(T_i)$ (resp. $[\mathsf{fv}_i^\bullet(T_i'), \mathsf{fv}_i^+(T_i'')]$) le membre gauche obtenu en appliquant la transformation. Pour des ensembles de termes T_1, T_2, un terme t et un $\circ \in \{+, \star\}$, notons $T_1 \vdash_\circ t$ s'il existe un $\zeta_\circ \in \mathcal{T}(\{\circ, J_\circ, e_\circ\}, \mathcal{X})$ tel que $\zeta_\circ[T]\downarrow = t$, resp. $[T_1, T_2] \vdash_\bullet t$ s'ils existent $\zeta_\bullet \in \mathcal{T}(\{\bullet, J_\bullet, e_\bullet\}, \mathcal{X})$ et $\zeta_+ \in \mathcal{T}(\{+, J_+\, e_+\}, \mathcal{X})$, tels que $\zeta_\bullet[T_1] \bullet h(\zeta_+[T_2])\downarrow = t$.

Supposons que σ est une solution de $\mathcal{C}$. Si n est le nombre des contraintes dans $\mathcal{C}$, on montre par récurrence sur i, $1 \leq i \leq n$, que pour toute contrainte $T_i \overset{?}{\vdash}_\circ x_i$ (resp. $[T_i, T_i] \overset{?}{\vdash}_\bullet x_i \in \mathcal{C}$), on a $\mathsf{fv}_i^\circ(T_i) \vdash_\circ x_i\sigma$ (resp. $[\mathsf{fv}_i^\bullet(T_i), \mathsf{fv}_i^+(T_i)]\sigma \vdash_\bullet x_i\sigma$).

Quand $i = 1$, T_i est clos et $\mathsf{fv}_i^\circ(T_i)$ n'a aucun effet: on conclut immédiatement.

Pour le pas de récurrence, supposons d'abord que $\circ = +$ (le cas $\circ = \star$ est identique). Pour chaque $x \in \mathsf{Fact}_+(T_i) \cap \mathcal{X}_i^+$, par origination, monotonie et hypothèse de récurrence, il existe une recette $\zeta_+^x \in \mathcal{T}(\{+, J_+, e_+\}, \mathcal{X})$ telle que $\zeta_+^x[T_i\sigma]\downarrow = x\sigma$. Soit ζ_+ la recette telle que $\zeta_+[T_i\sigma]\downarrow = x_i\sigma$. Prenons $\zeta' = \zeta_+ + \Sigma_{u \in T_i, x \in \mathsf{Fact}_+(u) \cap \mathcal{X}_i^+} \zeta_+^x$. Il est facile de voir, par définitions, que $\zeta'[\mathsf{fv}_i^+(T_i)\sigma]\downarrow = x_i\sigma$.

Si $\circ = \bullet$, soit $\zeta_\bullet \in \mathcal{T}(\{\bullet, J_\bullet, e_\bullet\})$ et $\zeta_+ \in \mathcal{T}(\{+, J_+, e_+\}, \mathcal{X})$ les recettes telles que $\zeta_\bullet[T_i'\sigma] \bullet h(\zeta_+[T_i''\sigma])\downarrow = x_i\sigma$. La recette pour $\mathsf{fv}_i^\bullet(T_i')$ est alors $\zeta_\bullet' = \zeta_\bullet \bullet (\Pi_{u' \bullet \mathsf{h}(u'') \in T_i', x \in \mathsf{Fact}_\bullet(u') \cap \mathcal{X}_i^\bullet} \zeta_\bullet^x)$, tandis que la recette pour $\mathsf{fv}_i^+(T_i'')$ est $\zeta_+' = \zeta_+ + (\Sigma_{u' \bullet h(u'') \in T_i', x \in \mathsf{Fact}_+(u'') \cap \mathcal{X}_i^+} \zeta_+^x) + \Sigma_{u \in T_i'', x \in \mathsf{Fact}_+(u) \cap \mathcal{X}_i^+} \zeta_+^x$.

L'autre direction, quand σ est une solution de $\mathcal{C}'$, est similaire, en soustrayant les recettes pour les $\mathcal{X}_i^\circ$ pour construire une solution de $\mathcal{C}$, au lieu de les rajouter. □

7.3 Transformation des contraintes de déductibilité dans des systèmes d'équations avec des paramètres formels.

Nous illustrons maintenant comment les systèmes purs se transforment, grâce au lemme 33, dans des systèmes d'équations avec des paramètres formels, qui sont linéaires: les paramètres ne sont pas appliqués à des variables. Nous allons voir dans la section suivante comment cette linéarité permet la traduction vers les systèmes Diophantiennes linéaires.

Exemple 43 *Soit:*

$$\mathcal{C} = \left\{ \begin{array}{lll} a & \overset{?}{\vdash}_+ & x \\ a, b + x & \overset{?}{\vdash}_+ & y \end{array} \right.$$

Traduit en équations, ce système est équivalent à

$$\begin{cases} x = \lambda a \\ y = \lambda' a + \lambda'' b + \lambda'' \lambda' a \end{cases}$$

Ce système est non-linéaire, à cause du terme $\lambda''\lambda' a$. Pour eviter ça, comme vu dans la section précédente, on utilise la monotonie et l'origination pour transformer le système de contraintes dans un autre, équivalent, dont le système d'équations correspondant est linéaire:

$$\mathcal{C}' = \begin{cases} a & \overset{?}{\vdash}_+ & x \\ a, b & \overset{?}{\vdash}_+ & y \end{cases} \rightarrow \begin{cases} x = \lambda a \\ y = \lambda' a + \lambda'' b \end{cases}$$

Suit un autre exemple, qui illustre le traitement particulier de $\bullet$ dans ce pas de la procédure.

$$\mathcal{C} = \begin{cases} a & \overset{?}{\vdash}_+ & x \\ a & \overset{?}{\vdash}_\bullet & y \\ a, x+b, h(x+b), y \bullet b & \overset{?}{\vdash}_\bullet & z \end{cases}$$

Le système d'équations corréspondent est:

$$\begin{cases} x = \lambda a \\ y = a^\beta \bullet h(\beta' a) \\ z = z_1 \bullet h(z_2) \\ z_1 = a^{\gamma_1} \bullet (x+b)^{\gamma_2} \bullet y^{\gamma_4} \bullet b^{\gamma_4} \\ z_2 = \theta_1 a + \theta_2 x + \theta_2 b + \theta_3 h(x+b) + \theta_4 (y \bullet b) + \gamma_3 x + \gamma_3 b \end{cases}$$

Ce système n'est pas linéaire à cause des termes: y^{γ_4}, $\theta_2 x$ et $\gamma_3 x$. Notons que, bien que venant de la même contrainte, la non-linéarité se distribue dans deux théories, $+$ et $\bullet$. Cette observation nous amène à considérer deux copies pour le membre gauche correspondant. Le système devient:

$$\begin{cases} a & \overset{?}{\vdash}_+ & x \\ [a; a] & \overset{?}{\vdash}_\bullet & y \\ [a, x+b, h(x+b), y \bullet b; a, x+b, h(x+b), y \bullet b] & \overset{?}{\vdash}_\bullet & z \end{cases}$$

Maintenant on peut éliminer de chaque copie les éléments qui amènent la non-linéarité, tout en préservant les solutions. Comme attendu, 3 occurrences de variables ont été effacées:

$$\mathcal{C} = \begin{cases} a & \overset{?}{\vdash}_+ & x \\ [a; a] & \overset{?}{\vdash}_\bullet & y \\ [a, x+b, h(b), b; a, b, h(x+b), y \bullet b] & \overset{?}{\vdash}_\bullet & z \end{cases}$$

Le système diophantien correspondant sera ainsi linéaire:

$$\begin{cases} x = \lambda a \\ y = a^{\beta} \bullet h(\beta' a) \\ z = z_1 \bullet h(z_2) \\ z_1 = a^{\gamma_1} \bullet (x+b)^{\gamma_2} \bullet b^{\gamma_4} \\ z_2 = \theta_1 a + \theta_2 b + \theta_3 h(x+b) + \theta_4 (y \bullet b) + \gamma_3 b \end{cases}$$

Plus formellement, notre transformation des contraintes de déductibilité dans des équations est la suivante. Elle est possible par le lemme 33.

Pour $\circ \in \{+, \star\}$,

$$\sum_i t_i^1, \ldots, \sum_i t_i^n \overset{?}{\vdash}_{\circ} x \quad \Longrightarrow \quad x = \sum_{i,j} \lambda_j t_i^j$$

si, $\forall i, j. \top(t_i^j) \neq \circ$ et $\lambda_1, \ldots, \lambda_n$ sont des paramètres entiers formels, représentant le nombre des fois où le terme est selectionné dans la recette.

Pour $\overset{?}{\vdash}_{\bullet}$, on utilise ci-dessous une notation multiplicative pour $\bullet$ et une notation additive pour $+$:

$$[\ (\textstyle\prod_i t_i^1) \bullet h(\sum_i u_i^1),\ \ldots,\ (\prod_i t_i^n) \bullet h(\sum_i u_i^n)\ ;\ \sum_i v_i^1, \ldots, \sum_i v_i^m\ ] \ \overset{?}{\vdash}\ x$$
$$\Rightarrow \begin{cases} x = x_1 \bullet h(x_2) \\ x_1 = \prod_{j=1}^{n} \prod_i (t_i^j)^{\lambda_j} \\ x_2 = \sum_{j=1}^{n} \sum_i \lambda_j u_i^j + \sum_{j=1}^{m} \sum_i \mu_j v_i^j \end{cases}$$

Où $\forall i, j. \top(t_i^j) \notin \{\bullet, h\}$ & $\top(u_i^j), \top(v_i^j) \neq +$ et $\lambda_1, \ldots, \lambda_n, \mu_1, \ldots, \mu_m$ sont des variables avec des valeurs dans les entiers.

Exemple 44

$$[\ a \bullet h(2b),\ a^3 \bullet b\ ;\ 2a + 3b, 2b\ ] \ \overset{?}{\vdash}_{\bullet}\ x$$

est transformé en $x = x_1 \bullet h(x_2)$, $x_1 = a^{\lambda_1} \bullet a^{3\lambda_2} \bullet b^{\lambda_2} = a^{\lambda_1 + 3\lambda_2} \bullet b^{\lambda_2}$, $x_2 = 2\lambda_1 b + 2\mu_1 a + 3\mu_1 b + 2\mu_2 b = 2\mu_1 a + (2\lambda_1 + 3\mu_1 + 2\mu_2) b$.

Le lemme suivant est immédiat par la définition d'une solution pure:

Lemme 34 *Dans la transformation* $\mathcal{C} \implies \mathcal{C}'$, σ *est une solution pure de* $\mathcal{C}$ *si, et seulement si,* σ *est une solution de* $\mathcal{C}'$.

7.4 Résolution des équations.

On doit maintenant résoudre un système d'équations $E = E_1 \cup E_2$, où E_1 contient des équations de la forme $x = \Sigma_{\circ} \lambda_i \alpha_i t_i, H(t_i) \neq \circ$ et E_2 est un ensemble d'équations ordinaires. Après avoir appliqué à nouveau la propriété des variants finis, on obtient un problème d'unification dans une combinaison de théories disjointes.

Les pas de résolution sont les suivants.

Pas 1: Application des variants finis. On peut calculer un ensemble fini de substitutions $\sigma_1, \ldots, \sigma_k$ tel que:

- Pour toute solution σ de E modulo EP, il existe un i, $1 \leq i \leq k$, et une substitution θ tels que $\sigma = \sigma_i\theta$ et θ est une solution de $E\sigma_i\downarrow$, modulo $\mathsf{AC}(+, \star, \bullet)$.
- Pour toute solution, modulo $\mathsf{AC}(+, \star, \bullet)$, θ de $E\sigma_i\downarrow$, $\sigma_i\theta$ est une solution de E modulo EP.

Dans la suite, on identifie E avec l'un des $E\sigma_i\downarrow$.

Pas 2: choix des égalités. Des nouvelles égalités entre les sous-termes sont peut-être introduites dans le pas précédent. Dans ce pas, on devine encore une fois les égalités entre les sous-termes de E, et on les rajoute à E_2.

Pas 3: Résolution de E_2. En s'appuyant sur [BS96] et [KN92], si les équations de E_2 (plus les contraintes sur les théories de termes) sont satisfaisables, on peut calculer un ensemble complet d'unificateurs $\theta_1, \ldots, \theta_m$. Alors E a une solution si et seulement si $E_1\theta_i$ a une solution, pour un $1 \leq i \leq m$.

Pas 4: traitement des équations avec des paramètres entiers. Maintenant on montre comment on réduit les équations formelles à des systèmes diophantiens linéaires.

Les équations de $E_1\theta_i$ sont de la forme suivante:

$$\beta_1 u_1 \circ \ldots \circ \beta_k u_k = \lambda_1\alpha_1 t_1 \circ \ldots \circ \lambda_n\alpha_n t_n, \forall i.H(t_i) \neq \circ$$

où $\beta_1, \ldots, \beta_k, \alpha_1, \ldots, \alpha_n$ sont des constantes, $\lambda_1, \ldots, \lambda_n$ sont des variables.

Notons que cette équation est satisfaite seulement si, pour tout $1 \leq j \leq k$, u_j est ou bien égal à un t_i, ou bien u_j est une variable. En effet, si u_j n'est pas une variable, puisque (suivant le pas 1) $top(u_j\sigma\downarrow) = top(u_j) \neq \circ$ et $\forall i.top(t_i\sigma\downarrow) \neq \circ$, il doit exister un t_i tel que $u_j\sigma\downarrow = t_i\sigma\downarrow$. Étant donné que les égalités ont été devinés, ceci peut se produire seulement si $t_i = u_j$. Pour tout $1 \leq i \leq n$, soit γ_i la somme des β_j tels que $t_i = u_j$.

Ensuite, si $\{u_1, \ldots, u_k\} = \{z_1, \ldots, z_m, u_{m+1}, \ldots, u_k\}$ tel que $\{u_1, \ldots, u_k\} \cap \mathcal{X} = \{z_1, \ldots, z_m\}$, on écrit

$$\begin{cases} z_1 = \mu_{1,1}t_1 \circ \ldots \circ \mu_{n,1}t_n \\ \ldots \\ z_m = \mu_{1,m}t_1 \circ \ldots \circ \mu_{n,m}t_n \end{cases}$$

Par conséquent, en projetant l'égalité sur chacun des $t_1, \ldots, t_n$, notre équation est équivalente àu système Diophantien linéaire:

$$\begin{cases} \alpha_1\lambda_1 = \beta_1\mu_1^1 + \ldots + \beta_m\mu_1^m + \gamma_1 \\ \ldots \\ \alpha_n\lambda_n = \beta_1\mu_n^1 + \ldots + \beta_m\mu_n^m + \gamma_n \end{cases}$$

En appliquant cette transformation pour toutes nos équations formelles, on obtient une conjonction de systèmes diophantiens linéaires, qu'on peut résoudre par des methodes connues (e.g. [CD94]).

En conclusion, on obtient

Théorème 4 *Le résolution des systèmes purs et stables modulo la théorie* EP *est décidable.*

Chapitre 8

Conclusions et travaux futurs

Le résultat principal de cette partie est une procédure de résolution des systèmes de contraintes pour la théorie équationnelle EP. Il résulte des théorèmes 1, 2, 3, 4 et le corollaire 4:

Théorème 5 ([BC09]) *La satisfaisabilité des systèmes de contraintes modulo* EP *est décidable.*

De plus, nous avons obtenu ce résultat en introduisant un schéma général de combinaison, qui généralise [BC09] et les résultats de combinaison existants [CR05, CR06, ACD07]. Notre résultat offre ainsi d'autres perspectives d'application, que nous esquissons maintenant.

8.1 Rechiffrement

Soit $\mathcal{F}_{\mathtt{renc}} = \{\mathtt{enc}/3, \mathtt{dec}/2, \mathtt{renc}/2, \mathtt{f}/2\}$ (avec tous les symboles publics) et considérons le système de réécriture $\mathcal{R}_{\mathtt{renc}}$ qui modélise la théorie du rechiffrement du chiffrement randomisé:

$$\begin{array}{rcl} \mathtt{dec}(\mathtt{enc}(x,y,z),x) & \to & y \\ \mathtt{renc}(\mathtt{enc}(x,y,z),z') & \to & \mathtt{enc}(x,y,\mathtt{f}(z,z')) \end{array}$$

Cette modélisation est partielle (pour avoir la propriété des variants finis), car f possède des propriétés supplémentaires.

Considérons les sous-théories suivantes $\mathcal{F}_{\mathtt{enc}} = \{\mathtt{enc}, \mathtt{renc}\}$, $\mathcal{F}_{\mathtt{dec}} = \{\mathtt{dec}\}$, $\mathcal{F}_{\mathtt{f}} = \{\mathtt{f}\}$ et $\mathcal{F}_{\perp} = \emptyset$, avec le choix des arguments

- $\mathsf{th}(\mathtt{enc}, 1) = \mathtt{dec}$, $\mathsf{th}(\mathtt{enc}, 2) = \perp$, $\mathsf{th}(\mathtt{enc}, 3) = \mathtt{f}$
- $\mathsf{th}(\mathtt{dec}, 1) = \mathtt{enc}$, $\mathsf{th}(\mathtt{dec}, 2) = \perp$

- $\mathtt{th}(\mathtt{renc}, 1) = \mathtt{enc}$, $\mathtt{th}(\mathtt{renc}, 2) = \mathtt{f}$
- $\mathtt{th}(\mathtt{f}, 1) = \mathtt{th}(\mathtt{f}, 2) = \mathtt{f}$

On n'a pas besoin de symbole d'interface pour vérifier que $\mathcal{R}_{\mathtt{renc}}$ est pur pour ces choix. Le système vérifie aussi la propriété des variants finis, car sa "boundedness" ([Del06]) est 1. Donc, par les théoremes 1 et 2, on peut appliquer le théoreme 3, pour réduire le problème à la résolution des sytèmes purs.

Pour voir quelle est la portée de ce résultat, on doit rechercher quelles sont les recettes pures pour $\mathcal{R}_{\mathtt{renc}}$:

Lemme 35 *Les recettes pures et normalisées ζ pour $\mathcal{R}_{renc}$ sont de l'un des types suivants:*

$$\begin{array}{rll} \zeta_f : & \zeta \in \mathcal{T}(\{f\}, \mathcal{X}) & \\ \zeta_{renc} : & \zeta = \mathtt{renc}(x, \zeta_f) & \\ \zeta_{enc} : & \zeta = \mathtt{enc}(\zeta_{dec}, x, \zeta_f) & \\ \zeta_{dec} : & \zeta = \mathtt{dec}(\zeta', x) & \text{pour } \zeta' \text{ de type } \zeta_{enc}, \zeta_{renc} \end{array}$$

Notre conjecture est que les systèmes purs ainsi obtenus sont facilement résolubles par une approche similaire à celle décrite dans le chapitre 7, ce qui reste un des nos travaux futurs.

8.2 Chiffrement homomorphique

Considérons la signature $\mathcal{F}_{\mathtt{hom}} = \{\mathtt{enc}/3, \mathtt{dec}/2, \langle,\rangle/2, \mathtt{proj}_1/1, \mathtt{proj}_2/1, \mathtt{proje}_1/1, \mathtt{proje}_2/1\}$ (avec tous les symboles publics) et le système de réécriture $\mathcal{R}_{\mathtt{hom}}$ qui représente les propriétés du chiffrement homomorphique:

$$\begin{array}{rcl} \mathtt{dec}(\mathtt{enc}(x, y, r), x) & \rightarrow & y \\ \mathtt{proje}_1(\mathtt{enc}(x, \langle y, z\rangle), r) & \rightarrow & \mathtt{enc}(x, y, r) \\ \mathtt{proje}_2(\mathtt{enc}(x, \langle y, z\rangle), r) & \rightarrow & \mathtt{enc}(x, z, r) \\ \mathtt{proj}_1(\langle y, z\rangle) & \rightarrow & y \\ \mathtt{proj}_2(\langle y, z\rangle) & \rightarrow & z \end{array}$$

Cette modélisation est partielle (pour avoir la propriété des variants finis), car le chiffrement homomorphique satisfait en fait l'équation $\mathtt{enc}(x, \langle y, z\rangle, r) = \langle \mathtt{enc}(x, y, r), \mathtt{enc}(x, z, r)\rangle$.

Considérons les sous-théories suivantes $\mathcal{F}_{\mathtt{enc}} = \{\mathtt{enc}, \mathtt{proje}_1, \mathtt{proje}_2\}$, $\mathcal{F}_{\mathtt{dec}} = \{\mathtt{dec}\}$, $\mathcal{F}_{\mathtt{pair}} = \{\langle,\rangle, \mathtt{proj}_1, \mathtt{proj}_2\}$, $\mathcal{F}_{\perp} = \emptyset$ et le choix des arguments

- $\mathtt{th}(\mathtt{enc}, 1) = \mathtt{dec}$, $\mathtt{th}(\mathtt{enc}, 2) = \mathtt{pair}$, $\mathtt{th}(\mathtt{enc}, 3) = \perp$
- $\mathtt{th}(\mathtt{dec}, 1) = \mathtt{enc}$, $\mathtt{th}(\mathtt{dec}, 2) = \mathtt{pair}$
- $\mathtt{th}(\mathtt{proje}_1, 1) = \mathtt{th}(\mathtt{proje}_2, 1) = \mathtt{enc}$
- $\mathtt{th}(\mathtt{proj}_1, 1) = \mathtt{th}(\mathtt{proj}_2, 1) = \mathtt{pair}$

Comme pour $\mathcal{R}_{\texttt{renc}}$, il est facile de voir que ce système est pur et satisfait la propriété des variants finis. On peut donc appliquer le théoreme 3 pour réduire le problème à la résolution des sytèmes purs.

Lemme 36 *Les recettes pures et normalisées ζ pour $\mathcal{R}_{hom}$ sont de l'un des types suivants:*

$$\begin{array}{rll}
\zeta_{enc}: & \zeta = \mathit{enc}(\zeta_{dec}, \zeta_{pair}, x) & \\
\zeta^i_{proje}: & \zeta = \mathit{proje}_i(\mathit{enc}(\zeta_{dec}, x, y)) & \textit{pour } i = 1, 2 \\
\zeta_{dec}: & \zeta = \mathit{dec}(\zeta', \zeta_{pair}) & \textit{pour } \zeta' \textit{ de type } \zeta_{enc}, \zeta_{proje_1}, \zeta_{proje_2} \\
\zeta_{pair}: & \zeta \in \mathcal{T}(\{\langle,\rangle, \mathit{proj}_1, \mathit{proj}_2\}, \mathcal{X}) &
\end{array}$$

Encore une fois, notre conjecture est que les systèmes ainsi obtenus sont plus simples.

8.3 Rechiffrement et chiffrement homomorphique

En effet, les deux théories ne sont pas disjointes. Pourtant, leur combinaison est pure: il suffirait de mettre ensemble les choix de théories pour les arguments.

8.4 Signatures en aveugle

Considérons la signature $\mathcal{F}_{\texttt{blind}} = \{\texttt{blind}, \texttt{sign}, \texttt{getmsg}, \texttt{unblind}\}$ (avec tous les symboles publics) et le système de réécriture $\mathcal{R}_{\texttt{blind}}$ suivant

$$\begin{array}{rcl}
\texttt{getmsg}(\texttt{sign}(x, y)) & \rightarrow & x \\
\texttt{unblind}(\texttt{blind}(x, y), y) & \rightarrow & x \\
\texttt{unblind}(\texttt{sign}(\texttt{blind}(x, y), z), y) & \rightarrow & \texttt{sign}(x, z)
\end{array}$$

Cette théorie modélise les signatures en aveugle, utilisées dans certains protocoles de vote. Comme remarqué dans la section 7.4.2 de [Del06], $\mathcal{R}_{\texttt{blind}}$ satisfait la propriété des variants finis. Une chose intéressante se passe si on essaye de décomposer cette théorie en sous-théories de telle manière à ce que $\mathcal{R}_{\texttt{blind}}$ soit un système pur. La seule décomposition possible est celle triviale, où tous les symboles sont dans la même théorie:

- à cause de la dernière équation, $\top(\texttt{unblind}) = \top(\texttt{sign}) = \texttt{th}(\texttt{unblind}, 1)$ et $\texttt{th}(\texttt{sign}, 1) = \top(\texttt{blind})$
- à cause de la deuxième équation, $\texttt{th}(\texttt{unblind}, 1) = \top(\texttt{blind})$ et $\texttt{th}(\texttt{blind}, 1) = \top(\texttt{unblind})$
- à cause de la première équation, $\top(\texttt{getmsg}) = \texttt{th}(\texttt{sign}, 1)$

On déduit que $\top(\texttt{blind}) = \top(\texttt{sign}) = \top(\texttt{unblind}) = \top(\texttt{getmsg})$. Cette théorie n'est donc pas décomposable, au moins pas dans le cadre de l'approche presentée ici. D'autres techniques sont donc nécéssaires pour simplifier les systèmes de contraintes modulo $\mathcal{R}_{\texttt{blind}}$. C'est un des buts de la partie qui suit.

8.5 Travaux futurs.

D'abord, comme on a vu, il nous reste à exploiter les résultats du théoreme 3 pour les théories $\mathcal{R}_{\mathtt{renc}}$, $\mathcal{R}_{\mathtt{hom}}$ et d'autres, y compris celles qui contiennent des symboles AC. Ces résultats seront partiels, car on à restreint la généralité du problème pour avoir les variants finis. Un travail futur plus ambitieux est de traiter ces théories dans toute leur généralité. Une propriété moins forte que les variants finis est peut-être suffisante. Des résultats de combinaison encore plus flexibles sont possibles. Par exemple, notre résultat nous oblige à considérer deux symboles de multiplication pour EP, pour une raison qui n'est pas de fond: $\mathsf{thdest}(\mathsf{h}) \cap \mathsf{thsource}(\mathsf{h}) = \emptyset$.

Partie III

Théories saturées et signatures en aveugle

Dans cette partie, nous allons traiter la résolution des contraintes de déductibilité d'un autre point de vue, en ayant deux raisons principales pour ce changement de perspective.

D'abord, il y a des théories élémentaires, qui ne sont pas décomposables: la seule partititon $\mathcal{F}_1 \uplus \ldots \uplus \mathcal{F}_\ell \cup \{\mathsf{h}\}$ pour laquelle la théorie est pure est la décomposition triviale, avec $\ell = 1$. C'est bien sûr le cas pour les théories avec un seul symbole, mais l'exemple des signatures en aveugle vu à la fin du chapitre précédent montre que ça peut être aussi le cas des théories plus complexes. Nous avons donc besoin d'une autre approche pour obtenir et comprendre d'un point de vue théorique les résultats de décidabilité pour ce genre de théories. Et, même pour les théories décomposables, on a vu avec les exemples de $\mathcal{R}_{\mathtt{renc}}$ et $\mathcal{R}_{\mathrm{hom}}$ que les preuves à chercher après la réduction sont dans un ensemble non-trivial, même si plus simple que celui de départ.

C'est le but de cette partie: faire une étude de la recherche de preuve pour les contraintes de déductibilité. On est dans la situation ici où les preuves ne peuvent plus être coupées pour être recomposées après, comme c'était le cas dans le chapitre précédent: on doit rechercher toute la preuve ou, du moins, avoir une représentation finie et appropriée des preuves possibles.

Une deuxième raison pour considérer une autre approche pour la résolution des contraintes de déductibilité est donné par l'intérêt des propriétés plus générales que l'existence d'une solution [Bau07, CLCZ10]. Les propriétés de traces [CLCZ10] portent sur toutes les éxécutions possibles du protocole, ce qui se traduit dans l'ensemble des solutions de systèmes de contraintes correspondants. Pour ce genre des propriétés, une procédure qui cherche l'existence d'une solution n'est plus suffisante: il nous faut une représentation finie et effective de l'ensemble des toutes les solutions. Même plus, les propriétés d'équivalence [Bau07, CD09] mettent en correspondance toute preuve dans un système de contraintes avec une preuve dans un autre: il nous faut une représentation symbolique de toutes les preuves.

C'est aussi le but de cette partie: avoir une procédure de transformation des systèmes de contraintes qui cherche toutes les preuves possibles, en suivant pas à pas les moyens de leur construction. Ça nous donne un arbre de transition qui représente toutes les possibilités pour l'intrus d'interagir avec le protocole. Les feuilles de cet arbre sont des systèmes des contraintes avec des preuves triviales: les formes resolues.

Ici nous ne considérons pas des symboles AC, qui reste un des travaux futurs.

Contributions. Nous motivons d'abord le fait que, pour la recherche des preuves, il est préférable d'éliminer d'avance le raisonement équationel. Dans le chapitre 9, nous rappellons ([Del06]) que ceci peut être fait d'une manière systématique et générale, en s'appuyant sur la surréduction et la propriété des variants finis. Nous obtenons ainsi des théories de l'intrus dont la théorie équationnelle est vide: des systèmes d'inférence. Dans le chapitre 10 nous introduisons une notion de saturation (au fait, classique) des systèmes d'inférence et montrons sa liaison avec la localité. Elle nous permet d'avoir des règles de

transformation des contraintes qui suivent fidèlement les manières possibles de construire la preuve, en nous menant vers des systèmes où les preuves sont triviales (chapitre 11). Quand le système saturé est fini, on obtient ainsi une représentation symbolique finie de l'ensemble des toutes les solutions. Quand il est infini (comme pour les signatures en aveugle), nous allons montrer une manière de regrouper les formes résolues, pour obtenir une représentation finie de solutions et déduire une procédure de décision pour la satisfaisabilité des systèmes de contraintes (chapitre 12).

Chapitre 9

Systèmes d'inférence

Soit $(\mathcal{I}_\mathcal{F}, \mathcal{R})$ une théorie de l'intrus équationnelle, où $\mathcal{R}$ est un système de réécriture convergent. Soit $T \stackrel{?}{\vdash} v$ une contrainte de déductibilité et imaginons qu'on cherche une substitution σ et une preuve π de $T\sigma \vdash v\sigma$ dans $(\mathcal{I}_\mathcal{F}, \mathcal{R})$. La manière la plus naturelle de le faire est d'effectuer une recherche de preuve en arrière, en partant de la racine. On peut ainsi suivre exactement la structure de la preuve. Pour cela, il suffit de deviner la dernière règle d'inférence utilisée dans π, disons

$$\frac{x_1 \ \ldots \ x_n}{f(x_1, \ldots, x_n)}$$

On doit alors avoir $v\sigma\downarrow = f(x_1\theta, \ldots, x_n\theta)\downarrow$, pour une substitution θ telle que $\mathsf{dom}(\theta) = \{x_1, \ldots, x_n\}$. Pour poursuivre la recherche des bouts de preuve manquants, on peut ou bien unifier v et $f(x_1, \ldots, x_n)$, modulo la théorie équationnelle, ou bien faire l'unification syntaxique de $v, f(x_1, \ldots, x_n)$ avec des membres gauches de règles du système de réécriture, comme dans [Bau07]. Dans les deux cas, les propriétés de la théorie équationnelle qui assureraient la terminaison de la procédure ne sont pas évidentes. Par exemple, la preuve de terminaison de [Bau07] repose sur des propriétés spécifiques aux systèmes sous-termes convergents et il est encore peu clair comment la généraliser à d'autres exemples et classes de théories.

Une autre approche, celle suivie dans cette partie, est de régler les questions de réécriture en amont, avant la recherche des preuves. Pour cela, on remarque que la surréduction de v et de $f(x_1, \ldots, x_n)$ peut se faire avant, indépendamment de la preuve ([Del06, CD05]). On obtient alors un système d'inférence avec des règles plus complexes, mais d'où le raisonnement équationnel est éliminé. Cela permet de réduire la compléxité de la recherche des preuves, en dissociant les propriétés dont on a besoin: pour le système de réécriture, d'une part, et pour le système d'inférence, d'une autre.

Definition 26 *Un* système d'inférence *est une théorie de l'intrus dont la théorie équationnelle est vide.*

Par surréduction de chaque règle $I(x_1), \ldots, I(x_n) \rightarrow I(f(x_1, \ldots, x_n))$ on obtient:

Exemple 45 *Un système d'inférence équivalent (nous allons voir dans quel sens ci-dessous) à la théorie* $(\mathcal{I}_{\mathsf{DY}}, \mathsf{E}_{\mathsf{DY}})$ *de l'exemple 2 est*

$$(\mathsf{E}) \quad \frac{x \quad y \quad z}{\mathtt{enc}(x, y, z)} \qquad (\mathsf{D}) \quad \frac{\mathtt{enc}(\mathtt{pub}(y), x, z) \quad \mathtt{priv}(y)}{x} \qquad (\mathsf{K}) \quad \frac{x}{\mathtt{pub}(x)}$$

$$(\mathsf{P}) \quad \frac{x \quad y}{\langle x, y \rangle} \qquad (\mathsf{Proj}_1) \quad \frac{\langle x, y \rangle}{x} \qquad (\mathsf{Proj}_2) \quad \frac{\langle x, y \rangle}{y}$$

Nous allons noter dans la suite ce système d'inférence par $\mathcal{I}_{\mathsf{DY}}$.

Notons au passage que la surréduction donne aussi les règles d'inférence suivantes:

$$(\mathsf{D}') \quad \frac{x \quad y}{\mathtt{dec}(x, y)} \qquad (\mathsf{Proj}_1') \quad \frac{x}{\mathtt{proj}_1(x)} \qquad (\mathsf{Proj}_2') \quad \frac{x}{\mathtt{proj}_2(x)}$$

La présence ou non de ces règles dans la modélisation dépend de l'implémentation des algorithmes. S'ils retournent un message d'erreur quand leurs arguments n'ont pas la forme attendue, ces règles ne doivent pas être considérées.

Pour garder les exemples simples et, surtout, puisque les règles $\mathsf{D}', \mathsf{Proj}_1', \mathsf{Proj}_2'$ ne posent aucun problème technique, nous avons decidé de ne pas les considérer. Ce sera aussi le cas pour des règles de même nature dans les autres études de cas.

La correspondance qui est montrée dans cet exemple est une instance d'une transformation plus générale, introduite dans [Del06] et qui repose sur la surréduction et la propriété des variants finis.

Nous rappelons d'abord la notion des variants, telle qu'elle est introduite dans [Del06]. Ici, on ne considère pas des symboles AC.

Definition 27 (Variants) *Soit* $\mathcal{R}$ *un système de réécriture qui a la propriété des variants finis, par rapport à la théorie vide. Par définition (définition 4), les variants d'un terme* t *sont donnés par un ensemble obtenu par surréduction:* $\mathcal{V}(t) = \{t\theta_1\downarrow, \ldots, t\theta_n\downarrow\}$.

Pour calculer les variants d'une règle d'inférence R*, elle est vue comme un terme:* $\mathcal{V}(\mathsf{R}) = \{\mathsf{R}\theta_1\downarrow, \ldots, \mathsf{R}\theta_n\downarrow\}$. *Pour une théorie de l'intrus* $(\mathcal{I}, \mathcal{R})$, $(\mathcal{V}(\mathcal{I}), \emptyset)$ *est le système d'inférence obtenu en rajoutant à* $\mathcal{I}$ *les variants de chaque règle d'inférence et en oubliant la théorie équationnelle.*

Les variants d'un système de contraintes $\mathcal{C}$ *sont, de même, calculés en considérant* $\mathcal{C}$ *comme étant un (grand) terme:* $\mathcal{V}(\mathcal{C}) = \{\mathcal{C}\theta_1\downarrow, \ldots, \mathcal{C}\theta_n\downarrow\}$.

Exemple 46 *Le système d'inférence obtenu par surréduction pour la théorie Dolev-Yao est donné dans l'exemple 45. Les variants de D, de l'exemple 5*

(page 21), sont D et D':

$$D' := \left\{ \begin{array}{llll} & a \overset{?}{\vdash} \langle x_0^1, x_0^2 \rangle & \wedge & \\ & a \overset{?}{\vdash} x_1 & \wedge & \\ \mathtt{enc}(x_0^1, \langle b, x_1 \rangle, r),\ \mathtt{priv}(a), & a \overset{?}{\vdash} x_2 & \wedge & x_2 = b \end{array} \right.$$

Pour une preuve π dans $(\mathcal{I}, \mathcal{R})$, nous allons dénoter par $\mathcal{V}(\pi)$ l'ensemble des preuves dans $(\mathcal{V}(\mathcal{I}), \emptyset)$, où chaque règle d'inférence est remplacée par l'un de ses variants.

Théorème 6 ([Del06]) *Soit $\mathcal{C} = \{T_1 \overset{?}{\vdash} x_1 \cup \ldots T_n \overset{?}{\vdash} x_n\} \cup \mathcal{S}$ un système de contraintes et $(\mathcal{I}, \mathcal{R})$ une théorie de l'intrus telle que $\mathcal{R}$ a la propriété des variants finis par rapport à la théorie vide. $\mathcal{C}$ a une solution σ, modulo $(\mathcal{I}, \mathcal{R})$, si, et seulement si, il existe un $\mathcal{C}' = \mathcal{C}\theta\downarrow \in \mathcal{V}(\mathcal{C})$ et une solution σ' de $\mathcal{C}'$, modulo $(\mathcal{V}(\mathcal{I}), \emptyset)$, telle que $\mathcal{C}\sigma\downarrow = \mathcal{C}'\sigma'$.*

De plus, les preuves sont les mêmes: pour chaque preuve π de $T_i\sigma\downarrow \vdash x_i\sigma$, il existe une preuve $\pi' \in \mathcal{V}(\pi)$ de $T_i\theta\downarrow\sigma' \vdash x_i\theta\sigma'$ et réciproquement.

Preuve: Ce sont le théoreme 7.14 et la proposition 7.15 de [Del06]. Le résultat plus précis concernant les mêmes solutions et les mêmes (modulo les variants) preuves n'est pas dans les ennoncés de [Del06], mais peut être dérivé de leurs preuves. □

Systèmes de contraintes. Notons que les systèmes $\mathcal{C}' = \{T_1 \overset{?}{\vdash} u_1, \ldots, T_n \overset{?}{\vdash} u_n\} \cup \mathcal{S}$ à résoudre dans $(\mathcal{V}(\mathcal{I}), \emptyset)$ ne contiennent plus seulement des variables dans les membres droits. Ils satisfont la monotonie et la propriété d'origination suivante: $\forall 1 \leq i \leq n.\ \mathsf{Var}(T_i) \subseteq \mathsf{Var}(\{u_1, \ldots, u_{i-1}\})$. On peut aussi résoudre $\mathcal{S}$ (dans la théorie vide) et appliquer les unificateurs les plus généraux à $\{T_1 \overset{?}{\vdash} u_1, \ldots, T_n \overset{?}{\vdash} u_n\}$. Les sytèmes de contraintes considérés dans la suite sont donc des ensembles de contraintes de déductibilité qui satisfont la monotonie et l'origination, sans équations.

9.1 Exemples

Ici nous montrons quels sont les systèmes d'inférence obtenus par l'application des variants finis aux exemples considérés dans la suite: les théories sous-termes convergentes et les théories du chapitre 8. Pour ce calcul, il faut surréduire (avec des substitutions en forme normale) toutes les règles d'inférence $I(x_1), \ldots, I(x_n) \rightarrow I(f(x_1, \ldots, x_n))$ de la théorie de l'intrus de départ. Au fait, puisque nous ne considérons pas de symboles AC, la surréduction *basique* (i.e. la surréduction ne se fait pas dans la partie substitution d'un des pas précédents) est suffisante, cf [Del06], section 7.3.2.

9.1.1 Théories sous-termes convergentes

Un système de réécriture est sous-terme convergent s'il est donné par un ensemble fini de règles $\mathcal{R} = \{l_1 \to r_1, \ldots, l_n \to r_n\}$ tel que, pour tout i, r_i est un sous-terme strict de l_i et tel que la relation de réécriture associée $\to_{\mathcal{R}}$ est convergente. Il est alors facile de voir que toute séquence de surréduction basique issue de $f(x_1, \ldots, x_n)$ est de longueur au plus 1 et le calcul des variants nous donne le système d'inférence suivant:

$$\mathcal{I}_{\mathtt{sc}} = \begin{cases} \dfrac{l_1 \quad \cdots \quad l_n}{r} & \text{si } f(l_1, \ldots, l_n) \to r \in \mathcal{R} \text{ pour un } f \in \mathcal{F}_{\mathtt{pub}}, \\ & \quad \text{et } l_1, \ldots, l_n \text{ sont en forme normale} \\ \dfrac{x_1 \quad \cdots \quad x_n}{f(x_1, \ldots, x_n)} & \text{si } f \in \mathcal{F}_{\mathtt{pub}} \end{cases}$$

Considérons maintenant les trois théories de la fin de la partie précédente.

9.1.2 Rechiffrement

Pour $\mathcal{R}_{\mathtt{renc}}$, chaque séquence de surréduction est de longueur au plus 1 (seulement $\mathtt{renc}(x, y)$ et $\mathtt{dec}(x, y)$ ont des séquences de longueur 1) et on obtient le système d'inférence $\mathcal{I}_{\mathtt{renc}}$:

$$\frac{x \quad y \quad z}{\mathtt{enc}(x, y, z)} \qquad\qquad \frac{\mathtt{enc}(x, y, z) \quad x}{y}$$

$$\frac{\mathtt{enc}(x, y, z) \quad z'}{\mathtt{enc}(x, y, f(z, z'))}\,(\mathsf{R})$$

9.1.3 Chiffrement homomorphique

Pour $\mathcal{R}_{\mathtt{hom}}$, chaque séquence de surréduction est de longueur au plus 1 (seulement $\mathtt{proje}_1, \mathtt{proje}_2, \mathtt{proj}_1, \mathtt{proj}_2$ ont des séquences de longueur 1) et on obtient le système d'inférence $\mathcal{I}_{\mathtt{hom}}$: $\mathcal{I}_{\mathsf{DY}}$ plus les deux règles $(\mathsf{R_t})$

$$\frac{\mathtt{enc}(x, \langle y, z\rangle, r)}{\mathtt{enc}(x, y, r)} \qquad \frac{\mathtt{enc}(x, \langle y, z\rangle, r)}{\mathtt{enc}(x, z, r)}$$

9.1.4 Signatures en aveugle

Pour $\mathcal{R}_{\mathtt{blind}}$, chaque séquence de surréduction est de longueur au plus 1 (seulement `getmsg` et `unblind` ont des séquences de longueur 1) et on obtient le système d'inférence $\mathcal{I}_{\mathtt{blind}}$:

$$(\mathsf{S}) \quad \frac{x \quad y}{\mathtt{sign}(x, y)} \qquad (\mathsf{C}) \quad \frac{\mathtt{sign}(x, y)}{x}$$

$$(\mathsf{B})\quad \frac{x \quad y}{\mathtt{blind}(x,y)} \qquad (\mathsf{UB})\quad \frac{\mathtt{blind}(x,y) \quad y}{x} \qquad (\mathsf{UB}_1)\quad \frac{\mathtt{sign}(\mathtt{blind}(x,y),z) \quad y}{\mathtt{sign}(x,z)}$$

9.2 Une classification des règles d'inférence

Pour faire une analyse plus fine des preuves dans la suite, nous faisons ici une classification des règles selon le rapport de la conclusion avec les prémisses. Elle généralise la classification traditionnelle composition/décomposition. Les termes qui apparaisent dans les règles sont comparés par rapport à l'ordre sous-terme. Dans cette partie, les sous-termes sont syntaxiques. Les sous-termes de t sont notés par $st(t)$. L'ordre sous-terme $\lhd$ est défini par $u \lhd v \Leftrightarrow u \in st(v)$.

Une règle R est une

- *Composition*, si sa conclusion est le seul terme maximal dans R
- *Décomposition*, si tous les termes maximaux dans R sont des prémisses
- *Versatile*, si la conclusion et quelques prémisses sont maximaux dans R

Exemple 47 *Toute règle de la forme $I(x_1), \ldots, I(x_n) \rightarrow I(f(x_1, \ldots, x_n))$ est une composition, e.g. (P), (E), pour Dolev-Yao, et (S), (B) pour $\mathcal{I}_{blind}$.*

(D), pour Dolev-Yao, et (UB), pour $\mathcal{I}_{blind}$, sont des décompositions.

(UB_1), dans $\mathcal{I}_{blind}$, (R), dans $\mathcal{I}_{renc}$, et (R_t), dans $\mathcal{I}_{hom}$, sont des exemples de règles versatiles.

Chapitre 10

Théories saturées

Le problème de recherche de preuve pour les contraintes de déductibilité est évidemment lié à la recherche des preuves dans des théories logiques: Entscheidungsproblem. D. McAllester [McA93] a montré que Entscheidungsproblem peut se résoudre en temps polynomial si la théorie est locale et a proposé un algorithme qui vérifie la propriété de localité. D. Basin et H. Ganzinger [BG01] ont montré que la localité est équivalente à la saturation de la théorie par résolution ordonnée. Notre point d'intérêt dans cette partie peut être vu comme le relévement de ces résultats à la recherche des preuves dans des ensembles de termes non-clos. Les théories qu'on considère sont en effet saturées par résolution ordonnée, avec une notion de redondance similaire, mais légèrement différente de celle de [BG01]. Additionnellement, nous aurons des restrictions sur la forme des règles (section 10.2.2), pour guider la recherche non-close des preuves. Cependant, ces restrictions syntaxiques ne sont pas fondamentales et un des travaux futurs consiste à montrer que la localité seule est suffisante pour la résolution des contraintes de déductibilité.

Dans ce chapitre nous introduisons la saturation, l'illustrons sur nos études de cas (section 10.1) et montrons comment elle est liée à la recherche de preuves et à la localité (section 10.2).

10.1 Saturation

Pour une preuve π, on va noter par

- `step`(π) l'ensemble des termes qui etiquettent π
- `leaves`(π) le multi-ensemble des termes qui etiquettent les feuilles de π
- `last`(π) le dernier pas d'inférence de π
- `premises`(π) les preuves des prémisses de `last`(π)
- `conc`(π) la conclusion de π

Plus formellement,

$$\text{si} \quad \pi = \frac{\pi_1 \ \cdots \ \pi_n}{u} \quad \text{alors} \quad \begin{cases} \mathtt{last}(\pi) = \dfrac{\mathtt{conc}(\pi_1) \ \cdots \mathtt{conc}(\pi_n)}{u} \\ \mathtt{premises}(\pi) = \{\pi_1, \ldots, \pi_n\} \\ \mathtt{conc}(\pi) = u \end{cases}$$

Exemple 48 *Considérons la preuve π dans $\mathcal{I}_{\mathsf{DY}}$:*

$$\dfrac{\mathit{enc}(\mathit{pub}(k), a, r) \qquad \dfrac{\langle \mathit{priv}(k), a\rangle}{\mathit{priv}(k)}\,(\mathsf{Proj}_1)}{a}\,(\mathsf{D})$$

On a

- $\mathit{premises}(\pi) = \{\mathit{enc}(\mathit{pub}(k), a, r),\ \dfrac{\langle \mathit{priv}(k), a\rangle}{\mathit{priv}(k)}\}$
- $\mathit{conc}(\pi) = a$
- $\mathit{last}(\pi) = \dfrac{\mathit{enc}(\mathit{pub}(k), a, r)\ \mathit{priv}(k)}{a}$
- $\mathit{leaves}(\pi) = \{\mathit{enc}(\mathit{pub}(k), a, r), \langle \mathit{priv}(k), a\rangle\}$

Dans la suite, nous introduisons notre notion de saturation. Intuitivement, s'il existe une preuve telle que l'un des ses pas intermédiaires est trop grand (on va nommer une telle preuve *mauvaise* et le grand pas un *mauvais motif*), alors il existe une preuve plus simple de la même chose.

Si $\mathsf{R} = \dfrac{s_1 \ \cdots \ s_n}{s_0}$ est une règle d'inférence, on dénote par $\mathtt{Max}(\mathsf{R})$ le multi-ensemble des termes maximaux s_i, par rapport à l'ordre sous-terme $\lhd$.

Definition 28 (mauvais(e) motif / preuve) *Une* mauvaise preuve *est une preuve π de la forme:*

$$\dfrac{u_1 \ \cdots \ u_n \qquad \dfrac{v_1 \ \cdots \ v_m}{u_{n+1}}\,\mathsf{R}_1 \qquad u_{n+2} \ \cdots \ u_{n+k}}{v}\,\mathsf{R}_2$$

telle que $\mathsf{R}_1 = \dfrac{s_1 \ \cdots \ s_m}{s}$, $\mathsf{R}_2 = \dfrac{t_1 \ \cdots \ t_{n+k}}{t}$, $s \in \mathit{Max}(\mathsf{R}_1)$ *et* $t_{n+1} \in \mathit{Max}(\mathsf{R}_2)$.
Un mauvais motif *dans une preuve π est une sous-preuve de π de la forme:*

$$\dfrac{\pi_2^1 \ \cdots \ \pi_2^{i-1} \qquad \dfrac{\pi_1^1 \ \cdots \ \pi_1^m}{\mathit{conc}(\pi_2^i)}\,\mathsf{R}_1 \qquad \pi_2^{i+1} \ \cdots \ \pi_2^n}{v}\,\mathsf{R}_2$$

telle que la preuve suivante est mauvaise

$$\dfrac{\mathit{conc}(\pi_2^1)\ \cdots\ \mathit{conc}(\pi_2^{i-1}) \quad \dfrac{\mathit{conc}(\pi_1^1)\ \cdots\ \mathit{conc}(\pi_1^m)}{\mathit{conc}(\pi_2^i)} \quad \mathit{conc}(\pi_2^{i+1})\ \cdots\ \mathit{conc}(\pi_2^n)}{v}$$

Notons que, suivant notre classification des règles d'inférence, un mauvais motif est l'application d'une règle de composition ou versatile suivie de l'application d'une règle de décomposition ou versatile telle que (l'instance de) la conclusion de la première règle est (une instance d') une prémisse maximale de la deuxième règle.

Si $\pi = \mathsf{R}\theta$ est une instance d'une règle d'inférence R et $\mathtt{Max}(\mathsf{R}) = \{s_1, \ldots, s_k\}$, alors $\mu(\pi)$ est par définition le multi-ensemble $\{s_1\theta, \ldots, s_k\theta\}$. Si π est une preuve, $\mu(\pi)$ est par définition le multi-ensemble des $\mu(\pi')$, pour tous les pas d'inférence π' de π. Formellement, si $\mathtt{premises}(\pi) = \{\pi_1, \ldots, \pi_n\}$, on a:

$$\mu(\pi) = \mu(\pi_1) \uplus \cdots \uplus \mu(\pi_n) \uplus \mu(\mathtt{last}(\pi)).$$

Les multi-ensembles sont ordonnés en utilisant l'extension multi-ensemble de l'ordre pour leur éléments. Si $\succeq$ est un ordre, on dénote par $\succeq_m$ son extension multi-ensemble.

Definition 29 (système d'inférence saturé) *Un système d'inférence est* saturé *s'il existe une extension totale et bien-fondée $\prec$ de l'ordre sous-terme $\lhd$ telle que, pour toute mauvaise preuve π, il existe une preuve π' de $\mathit{leaves}(\pi) \vdash \mathit{conc}(\pi)$ avec $\mathit{leaves}(\pi') \subseteq \mathit{leaves}(\pi)$ (l'inclusion multi-ensemble) et $\mu(\pi')$ $(\prec_m)_m$ $\mu(\pi)$.*

Exemple 49 *Le système d'inférence Dolev-Yao de l'exemple 45 est saturé. En effet, les seules mauvaises preuves sont*

$$\dfrac{\dfrac{\mathit{pub}(u_1)\ \ u_2\ \ u_3}{\mathit{enc}(\mathit{pub}(u_1), u_2, u_3)} \quad \mathit{priv}(u_1)}{u_2} \qquad \dfrac{\dfrac{u_1 \quad u_2}{\langle u_1, u_2\rangle}}{u_i}\ \ i = 1, 2$$

Évidemment, pour chacune de ces preuves, il existe une preuve plus petite, triviale, π' de $\mathit{leaves}(\pi) \vdash \mathit{conc}(\pi)$ telle que $\mathit{leaves}(\pi') \subseteq \mathit{leaves}(\pi)$ et $\mu(\pi')$ $(\prec_m)_m$ $\mu(\pi)$ pour toute extension totale et bien-fondée de l'ordre sous-terme.

Si un système n'est pas saturé, on peut le compléter dans un système saturé équivalent, par résolution ordonnée. Dans les deux exemples qui suivent, le système saturé ainsi obtenu est fini:

10.1.1 Un exemple de saturation

Nous montrons d'abord un exemple (artificiel) illustrant comment on peut saturer un système d'inférence, qui n'est pas saturé au départ.

Considérons le système contenant la seule règle versatile

$$\frac{\mathtt{f}(\mathtt{g}(x),\mathtt{g}(y)) \quad \mathtt{g}(x) \quad y}{\mathtt{h}(x,y)}$$

et des règles de composition pour les symboles $\mathtt{f},\mathtt{g},\mathtt{h}$. Ce système n'est pas saturé, car il n'y a que des mauvaises preuves (minimales) de e.g. $\mathtt{g}(a),\mathtt{g}(b),b \vdash \mathtt{h}(a,b)$.

On complète le système de la manière suivante. D'abord, on superpose la règle versatile avec la règle

$$\frac{x_1 \quad x_2}{\mathtt{f}(x_1,x_2)}$$

On obtient une nouvelle règle versatile

$$\frac{\mathtt{g}(x) \quad \mathtt{g}(y) \quad y}{\mathtt{h}(x,y)}$$

qu'on rajoute au système d'inférence. On a maintenant une "bonne" preuve de $\mathtt{g}(a),\mathtt{g}(b),b \vdash \mathtt{h}(a,b)$. Pourtant, le système n'est pas encore saturé, puisqu'il n'y a que des preuves (minimales) mauvaises de e.g. $\mathtt{g}(a),b \vdash \mathtt{h}(a,b)$. Un autre pas de résolution ordonnée, superposant

$$\frac{y}{\mathtt{g}(y)}$$

avec la nouvelle règle versatile, rajoute la règle

$$\frac{\mathtt{g}(x) \quad y}{\mathtt{h}(x,y)}$$

Le système résultant, ci-dessous, est saturé, car toutes les mauvaises preuves sont maintenant contournables (par des preuves plus petites). On ne va pas faire la preuve, mais notons juste que dans cet exemple n'importe quelle extension totale et bien-fondée de l'ordre sous-terme marche, car on enlève toujours un terme dans la mesure μ d'une mauvaise preuve π, quand on lui associe une "bonne" preuve π'.

$$\frac{x \quad y}{\mathtt{f}(x,y)} \qquad \frac{x \quad y}{\mathtt{h}(x,y)} \qquad \frac{x}{\mathtt{g}(x)}$$

$$\frac{\mathtt{f}(\mathtt{g}(x),\mathtt{g}(y)) \quad \mathtt{g}(x) \quad y}{\mathtt{h}(x,y)} \qquad \frac{\mathtt{g}(x) \quad \mathtt{g}(y) \quad y}{\mathtt{h}(x,y)} \qquad \frac{\mathtt{g}(x) \quad y}{\mathtt{h}(x,y)}$$

Considérons maintenant les études de cas, dont les systèmes d'inférence sont presentés dans de la section 9.1.

10.1.2 Systèmes sous-terme convergents

Le système d'inférence $\mathcal{I}_{\mathtt{sc}}$ associé à un système de réécriture sous-terme convergent n'est pas toujours saturé, comme le montre l'exemple suivant:

Exemple 50 *Soit la signature* $\{\mathtt{f}, \mathtt{g}\}$*, avec tous les symboles publics, et le système de réécriture* $\{\mathtt{g}(\mathtt{f}(\mathtt{f}(x,y),\mathtt{g}(y))) \to y\}$

Le système d'inférence correspondant est formé des règles de composition pour $\mathtt{f}, \mathtt{g}$ *et*

$$\frac{\mathtt{f}(\mathtt{f}(x,y),\mathtt{g}(y))}{y}$$

Dans ce système, la preuve minimale (pour toute extension de l'ordre sous-terme) de $\mathtt{f}(a,b), \mathtt{g}(b) \vdash b$ *est mauvaise.*

Cependant, on peut appliquer une simple complétion pour saturer, en un nombre fini de pas, le système.

On applique itérativement la procédure suivante à $\mathcal{I}_{\mathtt{sc}}$:

1. on élimine de $\mathcal{I}_{\mathtt{sc}}$ les règles d'inférence dont la conclusion apparaît parmi les prémisses
2. on remplace les règles d'inférence $I(s_1) \ldots, I(x), \ldots, I(s_n) \to I(t)$ par $I(s_1), \ldots, \ldots, I(s_n) \to I(t)$ (i.e. on élimine $I(x)$ de l'ensemble des prémisses), quand x est une variable qui n'apparaît ni dans t ni comme un sous-terme strict d'un des s_i.
3. pour chaque symbole public $\mathtt{g}$ et chaque règle d'inférence

$$\frac{l_1 \ \cdots \ l_k \ \ \mathtt{g}(u_1, \ldots, u_m) \ \ \cdots l_n}{r}$$

on rajoute la règle

$$\frac{l_1 \ \cdots \ l_k \ \ u_1 \ \cdots \ \ u_m \ \ \cdots l_n}{r}$$

Jusqu'à ce que un point fixe soit atteint. Notons par $\overline{\mathcal{I}}_{\mathtt{sc}}$ le système d'inférence ainsi obtenu.

Puisque la taille (le nombre des symboles de fonctions) des nouvelles règles est strictement inférieure à la taille d'une règle d'inférence dans l'ensemble de départ, le point fixe est obtenu en un nombre borné de pas.

Exemple 51 *Pour l'exemple précédent, la seule règle*

$$\frac{\mathtt{f}(x,y) \ \ \mathtt{g}(y)}{y}$$

est rajoutée.

Lemme 37 *La transformation $\mathcal{I}_{sc} \rightarrow \overline{I}_{sc}$ décrite ci-dessus préserve la même relation de déductibilité, i.e. $T \vdash_{\mathcal{I}_{sc}} v \Leftrightarrow T \vdash_{\overline{\mathcal{I}}_{sc}} v$, pour tout T, v.*

Preuve: En effet, chaque nouvelle règle d'inférence est une conséquence des autres règles d'inférence et toute règle qui est enlevée (ou remplacée) peut aussi être prouvée en utilisant les autres règles. □Notons que le lemme précédent est vrai pour toute théorie de l'intrus.

Quand la théorie est sous-terme convergente, une autre propriété du système obtenu est qu'il n'a pas de règles versatiles: toutes les règles d'inférence sont ou bien des compositions $I(x_1), \ldots, I(x_n) \rightarrow I(f(x_1, \ldots, x_n))$ ou bien des décompositions $I(u_1), \ldots, I(u_n) \rightarrow I(r)$. En effet, dans la transformation ci-dessus, on garde l'invariant que la conclusion d'une règle d'inférence, qui n'est pas une composition, est un sous-terme de l'une des prémisses.

Lemme 38 *$\overline{\mathcal{I}}_{sc}$ est saturé.*

Preuve: Puisque les prémisses maximales d'une décomposition ne sont pas des variables (deuxième transformation ci-dessus), les seuls mauvais motifs sont de la forme:

$$\frac{u_1 \ \cdots \ u_k \ \dfrac{x_1\sigma \ \cdots x_m\sigma}{\mathsf{g}(x_1, \ldots, x_m)\sigma} \ u_{k+2} \ \cdots \ u_n}{r}$$

où $\mathsf{g} \in \mathcal{F}_{\mathtt{pub}}$, $u_{k+1} = \mathsf{g}(x_1\sigma, \ldots, x_m\sigma)$. Grâce à notre complétion (troisième point), il existe alors une preuve plus simple:

$$\frac{u_1 \ \cdots \ u_k \ x_1\sigma \ \cdots \ x_m\sigma \ u_{k+2} \ \cdots \ u_n}{r}$$

Ceci nous permet de conclure, car la mesure de cette preuve est plus petite pour toute extension totale et bien-fondée de l'ordre sous-terme. □

Dans les trois autres études de cas, le système saturé s'avère infini.

10.1.3 Rechiffrement

Le système $\mathcal{I}_{\mathtt{renc}}$ n'est pas saturé. En effet, notons que les termes maximaux dans la règle (R) sont $\mathtt{enc}(x, y, z)$ et $\mathtt{enc}(x, y, \mathtt{f}(z, z'))$. Considérons une preuve de $\mathtt{enc}(x, y, r), r', r'' \vdash \mathtt{enc}(x, y, \mathtt{f}(\mathtt{f}(r, r'), r''))$:

$$\frac{\dfrac{\mathtt{enc}(x, y, r) \quad r'}{\mathtt{enc}(x, y, \mathtt{f}(r, r'))} \quad r''}{\mathtt{enc}(x, y, \mathtt{f}(\mathtt{f}(r, r'), r''))}$$

C'est une mauvaise preuve, car elle superpose la conclusion (maximale) de la première instance de (R) avec l'hypothèse maximale de la deuxième. Pourtant, il n'existe pas de preuve plus simple de $\mathtt{enc}(x, y, r), r', r'' \vdash \mathtt{enc}(x, y, \mathtt{f}(\mathtt{f}(r, r'), r''))$.

Cependant, on peut saturer le système, en rajoutant (par résolution ordonnée) un ensemble infini de règles. Soit $t_n(z_1, \ldots, z_n)$ le terme défini par

- $t_1(z_1) = z_1$
- $t_{n+1}(z_1, \ldots, z_{n+1}) = f(t_n(z_1, \ldots, z_n), z_{n+1})$

On rajoute au système les règles d'inférence (pour tout $n \geq 2$)

$$\frac{\mathtt{enc}(x, y, z_1) \quad z_2 \ \ldots \ z_n}{\mathtt{enc}(x, y, t_n(z_1, \ldots, z_n))} \, (\mathsf{R}_n)$$

Ces règles sont des raccourcis pour éviter toutes les mauvaises preuves.

Lemme 39 *Le système (récursif) ainsi obtenu est saturé.*

Preuve: À part les mauvaises preuves provenant de la partie $\mathcal{I}_{\mathsf{DY}}$ du système, et qui peuvent être évitées par des preuves triviales, toutes les mauvaises preuves sont des superpositions π de deux règles (R_n) et (R_m):

$$\frac{\dfrac{\mathtt{enc}(x, y, r_1) \quad r_2 \ \ldots \ r_n}{\mathtt{enc}(x, y, t_n(r_1, \ldots, r_n))} \, (\mathsf{R}_n) \qquad r_{n+1} \ \ldots \ r_{n+m}}{t_{n+m}(r_1, \ldots, r_{n+m})} \, (\mathsf{R}_n)$$

Elles peuvent être remplacées par le résultat de leur résolution $\pi' = (\mathsf{R}_{n+m})$:

$$\frac{\mathtt{enc}(x, y, r_1) \quad r_2 \ \ldots \ r_{n+m}}{\mathtt{enc}(x, y, t_{n+m}(r_1, \ldots, r_{n+m}))} \, (\mathsf{R}_{n+m})$$

On a

$$\mu(\pi') = \{\{\mathtt{enc}(x, y, r_1), \mathtt{enc}(x, y, t_{n+m}(r_1, \ldots, r_{n+m}))\}\}$$

$$\mu(\pi) = \left\{ \begin{array}{l} \{\mathtt{enc}(x, y, r_1), \mathtt{enc}(x, y, t_n(r_1, \ldots, r_n))\}, \\ \{\mathtt{enc}(x, y, t_n(r_1, \ldots, r_n)), \mathtt{enc}(x, y, t_{n+m}(r_1, \ldots, r_{n+m}))\} \end{array} \right\}$$

Pour assurer que $\mu(\pi')$ $(\prec_m)_m$ $\mu(\pi)$, pour chaque telle paire π, π', il suffit de choisir une extension totale et bien-fondée $\prec$ de l'ordre sous-terme telle que $\mathtt{enc}(x, y, u_1) \prec \mathtt{enc}(x, y, t_n(u_1, \ldots, u_n))$, pour tout $n > 1$ et termes $u_1, \ldots, u_n$. □

10.1.4 Chiffrement homomorphique

À cause des enchaînements de la règle R_t, $\mathcal{I}_{\mathtt{hom}}$ n'est pas saturé. Pourtant, si on définit $P(x)$ comme étant le plus petit ensemble tel que

- $x \in P(x)$
- $t \in P(x) \Rightarrow \forall u.\ \langle t, u \rangle \in P(x) \wedge \langle u, t \rangle \in P(x)$

alors l'ensemble de règles ci-dessous, avec les règles de $\mathcal{I}_{\mathsf{DY}}$, est saturé.

$$\frac{\mathtt{enc}(x, t, r)}{\mathtt{enc}(x, y, r)} \, (\mathsf{R}_\mathsf{t}) \quad \text{pour tout } t \in P(y)$$

Lemme 40 *Le système d'inférence ainsi obtenu est saturé.*

Preuve: En plus des mauvaises preuves dans $\mathcal{I}_{\mathsf{DY}}$ (contournables par des preuves triviales), on a trois types de mauvaises preuves

$$\frac{\dfrac{\mathtt{enc}(t,u,r)}{\mathtt{enc}(t,v,r)}\,(\mathsf{R_t})}{\mathtt{enc}(t,w,r)}\,(\mathsf{R_t}) \qquad \frac{\dfrac{t\ \ u\ \ r}{\mathtt{enc}(t,u,r)}\,(\mathsf{E})}{\mathtt{enc}(t,v,r)}\,(\mathsf{R_t}) \qquad \frac{\dfrac{\mathtt{enc}(\mathtt{pub}(t),u,r)}{\mathtt{enc}(\mathtt{pub}(t),v,r)}\,(\mathsf{R_t}) \qquad \mathtt{priv}(t)}{v}\,(\mathsf{D})$$

avec $u \in P(v)$ et $v \in P(w)$.

Soit π une mauvaise preuve du premier type. Alors, par définition de P, on a $u \in P(w)$ et donc la preuve π' de $\mathtt{leaves}(\pi) \vdash \mathtt{conc}(\pi)$:

$$\frac{\mathtt{enc}(t,u,r)}{\mathtt{enc}(t,w,r)}\,(\mathsf{R_t})$$

Évidemment, $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. On a $\mu(\pi') = \{\{\mathtt{enc}(t,u,r), \mathtt{enc}(t,w,r)\}\}$ et $\mu(\pi) = \{\{\mathtt{enc}(t,u,r), \mathtt{enc}(t,v,r)\}, \{\mathtt{enc}(t,v,r), \mathtt{enc}(t,w,r)\}\}$. Pour avoir $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$, on doit choisir une extension $\prec$ de l'ordre sous-terme telle que $\mathtt{enc}(t,w,r) \prec \mathtt{enc}(t,v,r)$, pour tous t, r, w, v tels que $v \in P(w)$.

Soit maintenant π une mauvaise preuve de deuxième type. Puisque $u \in P(v)$, on a $u \vdash v$, en utilisant seulement Proj_1 et Proj_2. Soit π_{uv} cette preuve de $u \vdash v$ et soit π' la preuve suivante de $\mathtt{leaves}(\pi) \vdash \mathtt{conc}(\pi)$

$$\frac{t\ \ \pi_{uv}\ \ r}{\mathtt{enc}(t,v,r)}\,(\mathsf{E})$$

Évidemment, on a $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. On a aussi

$$\mu(\pi') = \{\{u\}, \{u_1\}, \ldots, \{u_n\}, \{\mathtt{enc}(t,v,r)\}\}$$
$$\mu(\pi) = \{\{\mathtt{enc}(t,u,r)\}, \{\mathtt{enc}(t,u,r), \mathtt{enc}(t,v,r)\}\}$$

où pour tout $1 \leq i \leq n$, $u_i \in st(u)$. Par conséquent, pour toute extension $\prec$ de l'ordre sous-terme, on a $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$.

Finalement, soit π une mauvaise preuve de troisième type. Puisque $u \in P(v)$, on a $u \vdash v$ en utilisant seulement Proj_1 et Proj_2. Soit π_{uv} cette preuve. On considère alors la preuve π' de $\mathtt{leaves}(\pi) \vdash \mathtt{conc}(\pi)$ qui consiste en l'application de (D), aux prémisses $\mathtt{enc}(\mathtt{pub}(t),u,r)$ et $\mathtt{priv}(t)$, suivie de la preuve π_{uv}:

$$\frac{\mathtt{enc}(\mathtt{pub}(t),u,r) \quad \mathtt{priv}(t)}{\pi_{uv}}\,(\mathsf{D})$$

On a $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$ et

- $\mu(\pi') = \{\{\mathtt{enc}(\mathtt{pub}(t),u,r), \mathtt{priv}(t)\}, \{u\}, \{u_1\}, \ldots \{u_n\}\}$,

- $\mu(\pi) = \{\{\mathtt{enc}(\mathtt{pub}(t), u, r), \mathtt{enc}(\mathtt{pub}(t), v, r)\}, \{\mathtt{enc}(\mathtt{pub}(t), v, r), \mathtt{priv}(t)\}\}$.

Puisque $u_1, \ldots, u_n \in st(u)$, pour avoir $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$ il suffit de choisir une extension $\prec$ de l'ordre sous-terme telle que $\mathtt{priv}(t) \prec \mathtt{pub}(t)$, pour tout t. Notons que cette restriction de l'ordre n'est pas en contradiction avec celle requise pour contourner les mauvaises preuves dans les 2 cas traités précédemment.

On conclut qu'il existe un ordre $\prec$, extension totale et bien-fondée de l'ordre sous-terme, qui rend le système saturé. □

10.1.5 Signatures en aveugle.

À cause de la règle (UB_1), le système n'est pas saturé. Par exemple, la preuve π suivante est mauvaise:

$$\dfrac{\dfrac{\mathtt{sign}(\mathtt{blind}(\mathtt{blind}(x, x_1), x_2), y) \quad x_2}{\mathtt{sign}(\mathtt{blind}(x, x_1), y)} \quad x_1}{\mathtt{sign}(x, y)}$$

Pourtant, pour toute extension bien-fondée $\prec$ de l'ordre sous-terme $\lhd$, il n'existe pas de preuve π' de $\mathtt{leaves}(\pi) \vdash \mathtt{sign}(x, y)$ telle que $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$ et $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$.

Cependant, par résolution ordonnée, on peut rajouter au système toutes les raccourcis qui correspondent à des mauvaises preuves. Soit $b_n(x, x_1, \ldots, x_n)$ défini par:

- $b_1(x, x_1) = \mathtt{blind}(x, x_1)$
- $b_{n+1}(x, x_1, \ldots, x_{n+1}) = \mathtt{blind}(b_n(x, x_1, \ldots, x_n), x_{n+1})$

On rajoute à notre sytème les règles suivantes (pour tout $n \geq 1$),

$$\dfrac{\mathtt{sign}(b_n(x, x_1, \ldots, x_n), y) \quad x_1 \quad \ldots \quad x_n}{\mathtt{sign}(x, y)}\ (\mathsf{UB_n})$$

Lemme 41 *Le système d'inférence ainsi obtenu est saturé.*

Preuve: On a les types suivants des mauvaises preuves:

$$\dfrac{\dfrac{u \quad v}{\mathtt{sign}(u, v)}\ (\mathsf{S})}{u}\ (\mathsf{C}) \qquad \dfrac{\dfrac{b_n(u, v_1 \ldots, v_n) \quad s}{\mathtt{sign}(b_n(u, v_1, \ldots, v_n), s)}\ (\mathsf{S}) \quad v_1 \ \ldots \ v_n}{\mathtt{sign}(u, s)}\ (\mathsf{UB}_n)$$

$$\dfrac{\dfrac{u \quad v}{\mathtt{blind}(u, v)}\ (\mathsf{B}) \quad v}{u}\ (\mathsf{UB}) \qquad \dfrac{\mathtt{sign}(b_n(u, v_1, \ldots, v_n), s) \quad v_1 \ \ldots \ v_n}{\dfrac{\mathtt{sign}(u, s)}{u}\ (\mathsf{C})}\ (\mathsf{UB}_n)$$

$$\dfrac{\mathtt{sign}(b_{n+m}(v, t_1, \ldots, t_n, v_1, \ldots, v_m), s) \quad v_1 \ \ldots \ v_m}{\dfrac{\mathtt{sign}(b_n(v, t_1, \ldots, t_n), s)}{\mathtt{sign}(v, s)}\ (\mathsf{UB}_n) \quad t_1 \ \ldots \ t_n}\ (\mathsf{UB}_m)$$

Dans les deux premiers cas ($\mathsf{S}-\mathsf{C}$ et $\mathsf{B}-\mathsf{UB}$), on a comme dans le cas $\mathcal{I}_{\mathsf{DY}}$ des preuves correspondantes triviales, qui sont plus petites par rapport à toute extension bien-fondée de l'ordre sous-terme.

Supposons maintenant que la mauvaise preuve π est du type $\mathsf{S}-\mathsf{UB}_\mathsf{n}$. Soit π' la preuve qui consiste dans quelques applications de UB suivies d'une application de S. On a évidemment $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. De plus, nous avons que

$$\mu(\pi') = \{\{b_n(u, v_1, \ldots, v_n)\}, \{b_{n-1}(u, v_1, \ldots, v_{n-1})\}, \ldots, \{b_1(u, v_1)\}, \{\mathtt{sign}(u, s)\}\}$$
$$\mu(\pi) = \{\{\mathtt{sign}(b_n(u, v_1, \ldots, v_n), s)\}, \{\mathtt{sign}(b_n(u, v_1, \ldots, v_n), s), \mathtt{sign}(u, s)\}\}$$

et donc $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$ pour toute extension bien-fondée $\prec$ de l'ordre sous-terme.

Quand la mauvaise preuve est du type $\mathsf{UB}_\mathsf{n}-\mathsf{C}$, la situation est similaire.

Finalement, supposons que la mauvaise preuve π est du type $\mathsf{UB}_\mathsf{m}-\mathsf{UB}_\mathsf{n}$. On considère alors la preuve π'

$$\frac{\mathtt{sign}(b_{n+m}(v, t_1, \ldots, t_n, v_1, \ldots, v_m), s)\ \ t_1\ \ldots\ t_n\ \ v_1\ \ldots\ v_m}{\mathtt{sign}(v, s)}\ (\mathsf{UB}_{n+m})$$

Encore une fois, l'inclusion $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$ est évidente. De plus, nous avons que

$$\mu(\pi') = \{\{\mathtt{sign}(b_{n+m}(v, t_1, \ldots, t_n, v_1, \ldots, v_m), s), \mathtt{sign}(v, s)\}\}$$
$$\mu(\pi) = \left\{ \begin{array}{c} \{\mathtt{sign}(b_{n+m}(v, t_1, \ldots, t_n, v_1, \ldots, v_m), s), \mathtt{sign}(b_n(v, t_1, \ldots, t_n), s)\}, \\ \{\mathtt{sign}(b_n(v, t_1, \ldots, t_n), s), \mathtt{sign}(v, s)\} \end{array} \right\}$$

Par conséquent, pour assurer $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$, il est suffisant de considérer un extension totale et bien-fondée $\prec$ de l'ordre sous-terme telle que pour tout $n, v, s, t_1, \ldots, t_n$, on a $\mathtt{sign}(v, s) \prec \mathtt{sign}(b_n(v, t_1, \ldots, t_n), s)$. □

10.2 Preuves normales et localité

Dans cette section nous montrons que, grâce à la saturation, il suffit de considérer des preuves qui ont une certaine structure (section 10.2.1) et un certain contenu (section 10.2.2): chaque preuve normale est locale, i.e. ses termes intermédiaires sont dans les sous-termes des hypothèses ou de la conclusion.

10.2.1 Preuves normales et preuves simples

Nous allons nommer *normale* une preuve sans mauvais motif. Une conséquence immédiate de la saturation est l'existence des preuves normales parmi les preuves avec un ensemble minimal d'hypothèses:

Lemme 42 *Pour toute preuve π de $T \vdash u$ il existe une preuve normale π' telle que $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$*

Preuve: Soit $\prec$ l'extension totale et bien-fondée de l'ordre sous-terme $\lhd$ temoignant du fait que le système d'inférence est saturé. Nous allons faire une récurrence sur $\mu(\pi)$ pour prouver qu'il existe une preuve normale π' de $T \vdash u$ avec $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$ et $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$.

Si π est normale (en particulier, un feuille, avec $\mu(\pi)$ minimal), on conlut. Sinon, soit π_1 un mauvais motif de π. Nous avons:

$$\pi_1 \;=\; \left\{ \dfrac{\pi_2^1 \;\cdots\; \pi_2^{i-1} \quad \dfrac{\pi_1^1 \;\cdots\; \pi_1^m}{\pi_2^i}\,\mathsf{R}_1 \quad \pi_2^{i+1} \;\cdots\; \pi_2^n}{v}\,\mathsf{R}_2 \right.$$

tel que $\widehat{\pi_1}$ définie ci-dessous est une mauvaise preuve.

$$\dfrac{\mathtt{conc}(\pi_2^1)\cdots\mathtt{conc}(\pi_2^{i-1}) \quad \dfrac{\mathtt{conc}(\pi_1^1)\cdots\mathtt{conc}(\pi_1^m)}{\mathtt{conc}(\pi_2^i)}\,\mathsf{R}_1 \quad \mathtt{conc}(\pi_2^{i+1})\cdots\mathtt{conc}(\pi_2^n)}{v}\,\mathsf{R}_2$$

Par saturation, on sait qu'il existe une preuve $\widehat{\pi_2}$ telle que $\mathtt{conc}(\widehat{\pi_2}) = v$ et $\mathtt{leaves}(\widehat{\pi_2}) \subseteq \mathtt{leaves}(\widehat{\pi_1})$ (inclusion multi-ensemble) et $\mu(\widehat{\pi_2})(\prec_m)_m\mu(\widehat{\pi_1})$. Soit π_2 la preuve $\widehat{\pi_2}$ dans laquelle chaque hypothèse $\mathtt{conc}(\pi_i^j)$ est remplacée avec la preuve correspondante π_i^j.
Soit $M = \biguplus_{1\le j\le m} \mu(\pi_1^j) \uplus \biguplus_{1\le j<i} \mu(\pi_2^j) \uplus \biguplus_{i<j\le n} \mu(\pi_2^j)$. Nous avons:

$$\mu(\pi_2) = M \uplus \mu(\widehat{\pi_2})\ (\prec_m)_m\ M \uplus \mu(\widehat{\pi_1}) = \mu(\pi_1).$$

Notons que l'inégalité est aussi une conséquence de $\mathtt{leaves}(\widehat{\pi_2}) \subseteq \mathtt{leaves}(\widehat{\pi_1})$, l'inclusion multi-ensemble étant importante pour ne pas dupliquer les noeuds et faire croître μ. Soit π' la preuve obtenue en remplaçant π_1 avec π_2 dans π. Nous avons $\mu(\pi')\ (\prec_m)_m\ \mu(\pi)$ et $\mathtt{leaves}(\pi') \subseteq \mathtt{leaves}(\pi)$. On applique l'hypothèse de récurrence pour conclure. □

Definition 30 (preuve simple) *Soit $H_1 \subseteq H_2 \subseteq \cdots \subseteq H_n$ une séquence d'ensembles de termes. Une preuve π de $H_i \vdash u$ est* gauche-minimale *si pour tout $j < i$ tel que $H_j \vdash u$, π est une preuve de $H_j \vdash u$. Une preuve est* simple *si elle est normale et gauche-minimale.*

Exemple 52 *Considérons le système d'inférence Dolev-Yao de l'exemple 45. Soit $H_1 = \{\mathtt{enc}(\mathtt{pub}(k), a, r),\ \mathtt{priv}(k),\ a\}$ et $H_2 = H_1 \cup \{\langle a, b\rangle\}$. On a $H_2 \vdash a$. En effet, les preuves π_1, π_2 et π_3 ci-dessous sont des temoins de ce fait:*

$$\dfrac{\langle a, b\rangle}{a} \qquad \dfrac{\mathtt{enc}(\mathtt{pub}(k), a, r) \quad \mathtt{priv}(k)}{a} \qquad \dfrac{\dfrac{a \quad a}{\langle a, a\rangle}}{a}$$

La preuve π_2 est simple, tandis que π_1 et π_3 ne le sont pas. Notons que la preuve de $H_2 \vdash a$ réduite à une feuille est simple aussi.

Lemme 43 *Soit $H_1 \subseteq H_2 \subseteq \cdots \subseteq H_n$ un séquence croissante d'ensembles de termes et $i \in \{1, \ldots, n\}$. Si $H_i \vdash u$, alors il existe une preuve simple de $H_i \vdash u$.*

Preuve: Soit π une preuve gauche-minimale de $H_i \vdash u$. Une telle preuve existe, car il suffit de considérer le j minimal tel que $H_j \vdash u$. Par le lemme 42, il existe une preuve normale π' de $H_i \vdash u$ telle que $\texttt{leaves}(\pi') \subseteq \texttt{leaves}(\pi)$. Il est facile de voir que π' est aussi gauche-minimale. La preuve π' est donc simple. □

10.2.2 Localité

Dans la suite, nous supposons que:

1. pour toute règle de composition, la conclusion est $\texttt{f}(x_1, \ldots, x_n)$ où $x_1, \ldots, x_n$ sont des variables.
2. pour toute règle versatile :
   (a) chaque sous-terme strict de la conclusion est sous-terme d'une des prémisses.
   (b) chaque prémisse qui n'est pas maximale dans la règle est un sous-terme strict d'une autre prémisse dans la même règle.

Ces conditions sont utilisées pour restreindre l'espace de la recherche des preuves pour les contraintes de déductibilité. Seulement la condition 2b est nécessaire pour le lemme 44 qui suit. Les deux autres sont utilisées pour prouver la complétude et la terminaison de nos règles de transformation des systèmes de contraintes.

Toutes ces conditions sont satisfaites par les études de cas $\mathcal{I}_{\texttt{DY}}$, $\mathcal{I}_{\texttt{sc}}$, $\mathcal{I}_{\texttt{blind}}$, $\mathcal{I}_{\texttt{hom}}$ et l'exemple de la section 10.1.1. Par contre, le système d'inférence $\mathcal{I}_{\texttt{renc}}$ ne satisfait pas la condition 2a.

Notons aussi que la première condition est satisfaite par toute théorie de l'intrus (équationelle) de départ, avant l'application des variants finis. Dans tous nos exemples, le calcul des variants finis la préserve, mais il serait intéressant de voir quelles propriétés du système de réécriture sont en jeu ici.

Le lemme suivant est un résultat de localité assez precis. Il nous donne non seulement une borne sur l'ensemble des termes utiles dans les preuves, mais aussi des informations spécifiques suivant le type de la dernière règle dans la preuve.

Lemme 44 (Localité) *Soit π une preuve normale de $H \vdash u$. On est alors dans un des cas suivants:*

- $\texttt{last}(\pi)$ *est une composition et* $\texttt{step}(\pi) \subseteq st(H \cup \{u\})$*;*
- π *est une feuille ou* $\texttt{last}(\pi)$ *est une décomposition et* $\texttt{step}(\pi) \subseteq st(H)$*;*
- $\texttt{last}(\pi)$ *est une règle versatile et* $\texttt{step}(\pi') \subseteq st(H)$ *pour toute sous-preuve stricte π' de π.*

Preuve: Soit π une preuve normale de $H \vdash u$. On fait une récurrence sur la taille de π.

Cas de base: π est une feuille. Le résultat est immédiat.

Étape de récurrence. Si π se termine par une instance d'une règle de composition, le résultat suit facilement par l'hypothèse de récurrence. Sinon, π se termine par une instance d'une règle versatile ou d'une décomposition. Nous avons

$$\pi = \left\{ \begin{array}{c} \dfrac{\pi_1 \quad \cdots \quad \pi_n}{u}\ \mathsf{R} \text{ avec } \mathsf{R} \end{array} \right. = \dfrac{v_1 \quad \cdots \quad v_n}{v_0}$$

Soit $\mathtt{conc}(\pi_i) = u_i$ et $I = \{i \in \{0, 1, \ldots, n\} \mid v_i \in \mathtt{Max}(\mathsf{R})\}$. Pour tout $i \in \{1, \ldots, n\}$, soit R_i la dernière règle d'inférence dans π_i. Par définition d'une preuve normale, π_i ne se termine pas par une instance d'une règle de composition ou d'une règle versatile pour tout $i \in I \setminus \{0\}$. Par conséquent, en appliquant l'hypothèse de récurrence, nous avons $\mathtt{step}(\pi_i) \subseteq st(H)$ pour tout $i \in I \setminus \{0\}$. Soit maintenant $J = \{1, \ldots, n\} \setminus I$ et $j \in J$. On a $v_j \in st(v_i)$ pour un $i \in I$. En fait, on a $i \neq 0$. Ceci suit par définition, quand R est une décomposition, et grâce à notre condition additionnelle 2b, quand R est versatile. Donc, pour tout $j \in J$, il existe un $i \in I \setminus \{0\}$, tel que $u_j \in st(u_i)$. De plus, par hypothèse de récurrence, $\mathtt{step}(\pi_j) \subseteq st(H \cup \{u_j\})$. On obtient $\mathtt{step}(\pi_i) \subseteq st(H)$ pour tout $i \in \{1, \ldots, n\}$. Ceci nous permet de conclure. □

L'exemple suivant montre que la condition 2b est nécéssaire pour le lemme 44.

Exemple 53 *Considérons le système d'inférence (saturé) formé par les deux règles ci-desssous. La première est une composition et la deuxième est versatile, avec $\mathtt{Max}(\mathsf{R}_2) = \{\mathsf{h}(x), \mathsf{f}(\mathsf{g}(x))\}$.*

$$\frac{x}{\mathsf{g}(x)}\ \mathsf{R}_1 \qquad \frac{\mathsf{h}(x) \quad \mathsf{g}(x)}{\mathsf{f}(\mathsf{g}(x))}\ \mathsf{R}_2$$

Soit $H = \{a, \mathsf{h}(a)\}$. On a $H \vdash \mathsf{f}(\mathsf{g}(a))$. La preuve normale temoignant de ce fait se termine par une instance de la règle versatile R_2. Pourtant, nous avons que $\mathsf{g}(a) \notin st(H)$. Ce système ne satisfait pas notre condition 2b. En effet, la prémisse $\mathsf{g}(x)$ n'est pas maximale dans la règle, elle doit donc être sous-terme strict d'une autre prémisse. Ce n'est pas le cas.

Chapitre 11

Résolution des systèmes de contraintes

Dans ce chapitre nous considérons un système d'inférence saturé et nous allons montrer que chaque système de contraintes peut se transformer en un ensemble de systèmes de contraintes en forme résolue. Ces derniers sont une représentation de toutes ses solutions (section 11.2). Si, de plus, le système saturé est fini, cette transformation est effective: la procédure de transformation termine et la représentation symbolique des solutions obtenue est finie (section 11.3). Si le système est infini, le branchement de notre transformation l'est aussi. Nous allons voir dans le chapitre 12 comment contourner ce problème. Pour l'exemple des signatures en aveugle, nous obtiendrons ainsi une procédure de décision pour la satisfaisabilité des systèmes de contraintes.

11.1 Systèmes de contraintes

Selon une observation faite après l'application des variants finis, les systèmes de contraintes à résoudre sont maintenant des ensembles de contraintes de déductibilité. Pourtant, dans le processus de recherche de preuve, on doit introduire des nouvelles variables (liées) et deviner des nouvelles égalités entre sous-termes. Ces deux points motivent la généralisation de la classe des systèmes que nous allons considérer:

Definition 31 (Système de contraintes) *Un* système de contraintes D *est une formule de la forme* $\exists \tilde{z}.[C \mid \mathsf{E}]$*, avec:*

- $\tilde{z}$ *une séquence de variables;*
- $\mathsf{E}(D) = \mathsf{E}$ *est un ensemble d'équations en forme résolue, identifié à une substitution* θ_{E}*;*

- *C est une conjonction de contraintes de déductibilité $H_1 \stackrel{?}{\vdash} u_1 \wedge \ldots \wedge H_n \stackrel{?}{\vdash} u_n$ avec $\mathsf{Var}(C) \cap \mathsf{dom}(\theta_{\mathsf{E}}) = \emptyset$, $H_1, \ldots, H_n$ des ensembles finis de termes et $u_1, \ldots, u_n$ des termes. De plus, la* monotonie *et l'*origination *sont satisfaites:*
  - Monotonie*: $\emptyset \neq H_1 \subseteq H_2 \subseteq \ldots \subseteq H_n$;*
  - Origination*: $\mathsf{Var}(H_i) \subseteq \mathsf{Var}(\{u_j \mid H_j \subsetneq H_i\})$ pour tout $1 \leq i \leq n$.*

On notera $fvar(D) = \mathsf{Var}(D) \setminus \tilde{z}$, les variables libres de D, et $\mathsf{LH}(D) = \{H_1, \ldots, H_n\}$. Parfois la conjonction C sera représenté comme un ensemble.

Definition 32 (Solution) *Étant donné un système d'inférence, une* solution *d'un système de contraintes $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ est une substitution close σ, avec $\mathsf{dom}(\sigma) = fvar(D)$, telle qu'il existe une substitution close τ, avec $\mathsf{dom}(\tau) = \tilde{z}$, telle que:*

- *$H(\sigma \cup \tau) \vdash u(\sigma \cup \tau)$, pour chaque $H \stackrel{?}{\vdash} u \in C$, et*
- *$u(\sigma \cup \tau) = v(\sigma \cup \tau)$, pour tout $u = v \in \mathsf{E}$.*

On dénote par $Sol(D)$ l'ensemble des solutions de D.

Exemple 54 *Considérons le système d'inférence $\mathcal{I}_{\mathsf{DY}}$ et le système de contraintes suivant*

$$D := \begin{cases} H_1 = a \stackrel{?}{\vdash} x_0 \quad \wedge \quad a \stackrel{?}{\vdash} x_1 \\ H_2 = \mathtt{enc}(x_0, \langle b, x_1 \rangle, r),\ \mathtt{priv}(a),\ a \stackrel{?}{\vdash} b \end{cases}$$

$H_1 \subseteq H_2$ et les variables x_0, x_1 apparaissent d'abord à droite. La substitution $\sigma = \{x_0 \mapsto \mathtt{pub}(a), x_1 \mapsto \langle a, a \rangle\}$ est une solution de D.

Ici, on n'a ni variable liée, ni équation. On verra plus tard comment elles s'introduisent dans les systèmes.

Notation. Soit $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ un système de contraintes. Pour toute variable $x \in \mathsf{Var}(D)$, on note H_x le plus petit ensemble $H \in \mathsf{LH}(D)$ pour lequel il existe une contrainte $H \stackrel{?}{\vdash} u \in D$ avec $x \in \mathsf{Var}(u) \setminus \mathsf{Var}(H)$. Autrement dit, H_x est la partie gauche de la contrainte de déductibilité qui introduit la variable x pour la première fois. Par origination et monotonie, H_x est bien défini pour chaque $x \in \mathsf{Var}(C)$. Par convention, $H_x = \emptyset$ quand x n'apparaît pas dans C.

11.2 Transformation des contraintes

Nous montrons ici que l'on peut résoudre les contraintes de déductibilité en préservant toutes les solutions du système, comme dans [CLCZ10]. L'idée de base des règles de transformation est simple. On devine la dernière règle de

la preuve et on fait une recherche de preuve en arrière, en unifiant le membre droit de la contrainte à résoudre avec la conclusion cette règle. Si $\mathsf{R} = I(u_1), \ldots, I(u_n) \to I(u)$ est dévinée comme étant la dernière règle dans la preuve de $H\sigma \vdash v\sigma$, on fait simplement la transformation suivante:

$$\exists \tilde{z}.[C \wedge H \overset{?}{\vdash} v \mid \mathsf{E}] \rightsquigarrow \exists \tilde{z'}.[C\theta \wedge H\theta \overset{?}{\vdash} u_1\theta \wedge \ldots \wedge H\theta \overset{?}{\vdash} u_n\theta \mid \mathsf{E'}]$$

où $\tilde{z'} = \tilde{z} \cup \mathsf{Var}(\mathsf{R})$, $\theta = \mathsf{mgu}(u, v)$ et $\mathsf{E'} = \mathsf{E} \cup \theta$.

Cependant, ceci ne termine pas, même pour un système d'inférence très simple et des ensembles de termes clos. Considérons par exemple la seule règle (Proj_1) de $\mathcal{I}_{\mathsf{DY}}$. On obtient:

$$\begin{array}{rcl} H \overset{?}{\vdash} v & \rightsquigarrow & \exists x_1, x_2.[H \overset{?}{\vdash} \langle v, x_2\rangle \mid x_1 = v] \\ & \rightsquigarrow & \exists x_1, x_2, y_1, y_2.[H \overset{?}{\vdash} \langle\langle v, x_2\rangle, y_2\rangle \mid x_1 = v \wedge y_1 = \langle v, x_2\rangle] \rightsquigarrow \ldots \end{array}$$

De même pour (P) :

$$H \overset{?}{\vdash} x \rightsquigarrow \exists x_1, x_2.[H \overset{?}{\vdash} x_1 \wedge H \overset{?}{\vdash} x_2 \mid x = \langle x_1, x_2\rangle] \rightsquigarrow \ldots$$

Tout d'abord, nous ne souhaitons pas enumérer explicitement toutes les solutions, mais seulement calculer des *formes résolues*, qui sont une représentation concise de toutes les solutions. Spécifiquement, $H \overset{?}{\vdash} v$ sera résolue quand v est une variable. Ceci écarte le deuxième exemple présenté ci-dessus.

En présence de règles de décomposition ou de règles versatiles, nous pouvons encore avoir le comportement décrit dans le premier exemple ci-dessus. C'est là que nous utilisons la localité: on contrôle l'application de telles règles, en demandant que les prémisses maximales soient des sous-termes de H.

Ceci n'est pourtant pas complet, puisque le lemme 44 montre seulement que, dans le cas d'une règle versatile ou d'une décomposition, les prémisses sont sous-termes des hypothèses au niveau *clos*. Autrement dit, si on a deviné que la dernière règle, dans la preuve d'une instance (avec la substitution σ) de $H \overset{?}{\vdash} v$, est une décomposition, on sait seulement que les prémisses du dernier pas de la preuve sont dans $st(H\sigma)$. Nous utilisons alors la propriété suivante des sous-termes syntaxiques: $st(H\sigma) = st(H)\sigma \cup st(\sigma)$.

Si les prémisses sont dans $st(H)\sigma$, tout va bien: nous pouvons deviner les sous-termes de H qui forment les prémisses. Sinon, la suite n'est pas évidente, car σ est inconnue. C'est là où nous utilisons des stratégies additionnelles, basées sur la monotonie et l'origination du système de contraintes.

Une autre difficulté vient de l'introduction des variables. Si nous introduisons des variables et des équations sans cesse, les membres gauches (et leur sous-termes) peuvent croître sans limite. Le choix des prémisses dans les sous-termes de H ne se fait alors pas nécessairement dans un ensemble borné.

(Axiom) $\exists\tilde{z}.[C \wedge H \overset{?}{\vdash} u \mid \sigma] \rightsquigarrow \exists\tilde{z}.[C\theta \mid \sigma \cup \theta]$
où $\theta = \mathsf{mgu}(u, v)$, $v \in H$ et $u \notin \mathcal{X}$

(Triv) $\exists\tilde{z}.[C \wedge H \overset{?}{\vdash} x \wedge H' \overset{?}{\vdash} x \mid \sigma] \rightsquigarrow \exists\tilde{z}.[C \wedge H \overset{?}{\vdash} x \mid \sigma]$
quand $H \subseteq H'$ et $x \in \mathcal{X}$

(Comp) $\exists\tilde{z}.[C \wedge H \overset{?}{\vdash} \mathtt{f}(u_1, \ldots, u_n) \mid \sigma] \rightsquigarrow \exists\tilde{z}.[C \wedge H \overset{?}{\vdash} u_1 \wedge \ldots \wedge H \overset{?}{\vdash} u_n \mid \sigma]$
si $\mathtt{f}$ est un symbole public

(Dec) $\exists\tilde{z}.[C \wedge H \overset{?}{\vdash} v \mid \sigma] \rightsquigarrow \exists\tilde{z}\cup\tilde{x}.[C\theta \wedge H\theta \overset{?}{\vdash} w_1\theta \wedge \ldots H\theta \overset{?}{\vdash} w_n\theta \mid \sigma \cup \theta]$
$\wedge\, H'\theta \overset{?}{\vdash} v_1\theta \wedge \ldots \wedge H'\theta \overset{?}{\vdash} v_m\theta$

où:

- $\mathsf{R} = \dfrac{v_1 \ \ldots \ v_m \ w_1 \ \ldots \ w_n}{w}$ est une règle de décomposition ou versatile telle que $\mathtt{Max}(\mathsf{R}) \subseteq \{w_1, \ldots, w_n\}$ et $\tilde{x} = \mathsf{Var}(\mathsf{R})$;
- $\theta = \mathsf{mgu}(\langle w, w_1, \ldots, w_n\rangle, \langle v, u_1, \ldots, u_n\rangle)$, $u_1, \ldots, u_n \in st(H) \setminus \mathcal{X}$, et $v \notin \mathcal{X}$;
- H' est un membre gauche d'une contrainte de déductibilité tel que $H' \subsetneq H$.

Figure 11.1: Transformation des constraintes de déductibilité

11.2.1 Règles de transformation

Ces règles sont présentées dans la figure 11.1. Elles sont appliquées d'une manière non-déterministe. Quand des nouvelles variables sont introduites (dans la règle Dec), elles sont supposées fraîches, par renommage.

La règle Axiom s'applique quand la preuve de la contrainte est réduite à une feuille. La règle Triv est due au fait que toute preuve pour la contrainte $H \overset{?}{\vdash} x$ est une preuve pour la contrainte $H' \overset{?}{\vdash} x$, quand $H \subseteq H'$. La règle Comp est utilisée quand la dernière règle de la preuve est une composition.

Donnons quelques explications pour la règle Dec. Nous avons deviné ici une règle de décomposition ou versatile. Les prémisses $w_1, \ldots, w_n$ sont celles dont les instances correspondent à des termes dans $st(H)\sigma$: nous pouvons deviner les termes correspondants dans $st(H)$, c'est-à-dire $u_1, \ldots, u_n$. Les autres prémisses (qui sont alors des sous-termes de la partie substitution) sont contraintes à être prouvées avec des hypothèses strictement plus petites. Nous allons montrer que ceci est toujours possible, établissant ainsi la complétude de notre système des règles.

Exemple 55 *Prenons le système de contraintes D donné dans l'exemple 54 pour le système d'inférence $\mathcal{I}_{\mathsf{DY}}$. En considérant la règle de décomposition*

(Proj_1) *et en appliquant* Dec *à la troisième contrainte, nous obtenons:*

$\exists x', y'.[a \overset{?}{\vdash} x_0, a \overset{?}{\vdash} x_1, \mathbf{enc}(x_0, \langle b, x_1\rangle, r), \mathbf{priv}(a), a \overset{?}{\vdash} \langle b, x_1\rangle \mid \{x' \mapsto b, y' \mapsto x_1\}]$.
Maintenant, en considérant (D) *et en appliquant à nouveau* Dec *à la troisième contrainte on obtient:*

$$D' = \left\{ \begin{array}{ll} \exists x, y, z, x', y'.[& a \overset{?}{\vdash} x_0\theta, \quad a \overset{?}{\vdash} x_1\theta \\ & H_2\theta \overset{?}{\vdash} \mathbf{enc}(x_0\theta, \langle b, x_1\theta\rangle, r) \\ & H_2\theta \overset{?}{\vdash} \mathbf{priv}(a) \mid \theta \cup \{x' \mapsto b, y' \mapsto x_1\}] \end{array} \right.$$

où $\theta = \mathsf{mgu}(\langle x, \mathbf{enc}(\mathbf{pub}(y), x, z), \mathbf{priv}(y)\rangle, \langle\langle b, x_1\rangle, \mathbf{enc}(x_0, \langle b, x_1\rangle, r), \mathbf{priv}(a)\rangle)$
$= \{x \mapsto \langle b, x_1\rangle,\ y \mapsto a,\ z \mapsto r,\ x_0 \mapsto \mathbf{pub}(a)\}$.

11.2.2 Correction

Nous montrons d'abord que nos règles transforment un système de contraintes en un système de contraintes, sans introduire de fausses solutions:

Lemme 45 (Correction) *Si D est un système de contraintes tel que $D \rightsquigarrow D'$, alors D' est un système de contraintes et $Sol(D') \subseteq Sol(D)$.*

Preuve: Soit $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ un système de contraintes et soit D' tel que $D \rightsquigarrow D'$. Montrons que D' est un système de contraintes. On a $D' = \exists \tilde{z'}.[C' \mid \mathsf{E}']$, où:

- $\tilde{z'}$ est une séquence de variables;
- E' est un ensemble d'équations en forme résolue;
- C' est une conjonction des contraintes de déductibilité et, puisque la substitution calculée est appliquée aux contraintes de C, on a $\mathsf{Var}(C') \cap \mathsf{dom}(\mathsf{E}') = \emptyset$. Il est aussi facile de remarquer que la monotonie est preservée. Pour prouver l'origination, on considère chacune des règles.

Notons d'abord que l'origination est preservée par l'application d'une substitution: si C satisfait l'origination, alors $C\theta$ satisfait l'origination pour toute substitution θ.

Règle Axiom. Dans ce cas, on a que $C' = C\theta \setminus \{H\theta \overset{?}{\vdash} u\theta\}$, pour une contrainte $H \overset{?}{\vdash} u \in C$ telle que $u\theta \in H\theta$. Puisque $C\theta$ satisfait l'origination et, puisque $u\theta \in H\theta$, aucune variable ne peut pas être introduite dans $u\theta$, on conclut que C' satisfait l'origination.

Règle Triv. Dans ce cas, on a $C' = C \setminus \{H' \overset{?}{\vdash} x\}$, où $H \overset{?}{\vdash} x \in C$ pour $H \subseteq H'$. La contrainte que nous avons éliminée n'introduit pas aucune variable et donc on conclut facilement.

Règle Comp. Dans ce cas, on a:

$$C' = C \setminus \{H \overset{?}{\vdash} \mathtt{f}(t_1, \ldots, t_n)\} \cup \{H \overset{?}{\vdash} t_1, \ldots, H \overset{?}{\vdash} t_n\}.$$

Il est facile de voir que l'origination est toujours satisfaite par C'.

Règle Dec. Dans ce cas, on a:

$$C' = (C\theta \setminus \{H\theta \overset{?}{\vdash} v\theta\}) \cup \{H\theta \overset{?}{\vdash} u_i\theta \mid 1 \leq i \leq n\} \cup \{H'\theta \overset{?}{\vdash} v_j\theta \mid 1 \leq j \leq m\}$$

où $\theta = \mathsf{mgu}(\langle w, w_1, \ldots, w_n\rangle, \langle v, u_1, \ldots, u_n\rangle)$, $u_1, \ldots, u_n \in st(H) \setminus \mathcal{X}$. De plus, les variables qui apparaissent dans les termes $w, w_1, \ldots, w_n$ sont fraîches et on a $\mathsf{Var}(w) \subseteq \mathsf{Var}(\{w_1, \ldots, w_n\})$.

Par notre première observation, $C\theta$ satisfait l'origination. Les contraintes nouvellement introduites ne posent pas de problèmes pour l'origination. Cependant, pour enlever sans dommage $H\theta \overset{?}{\vdash} v\theta$, on doit vérifier que le terme $v\theta$, avec $v\theta = w\theta$, n'introduit aucune variable pour la première fois. Soit $x \in \mathsf{Var}(v\theta)$. Puisque $\mathsf{Var}(v) \subseteq \mathsf{Var}(\{v_1, \ldots, v_m, w_1, \ldots, w_n\})$, nous avons $\mathsf{Var}(v\theta) \subseteq \mathsf{Var}(\{v_1\theta, \ldots, v_m\theta, w_1\theta, \ldots, w_n\theta\})$. Si $x \in \mathsf{Var}(v_i\theta)$, on peut conclure facilement, grâce à la présence de $H'\theta \overset{?}{\vdash} v_j\theta$. Sinon, on a $x \in \mathsf{Var}(w_i\theta) = \mathsf{Var}(u_i\theta)$ et on déduit que $x \in \mathsf{Var}(H\theta)$, puisque $u_i \in st(H)$. Ceci nous permet de conclure que $\mathcal{C}'$ satisfait la propriété d'origination.

En considérant chaque règle de transformation, il est facile de montrer que, si $D \rightsquigarrow D'$ et $\sigma \in Sol(D')$, alors $\sigma \in Sol(D)$. Ceci permet de conclure la preuve. □

11.2.3 Complétude

Nous allons montrer que, en utilisant les règles de transformation, la résolution des systèmes de contraintes peut être réduite à la résolution de systèmes élémentaires, que nous allons nommer des formes résolues. Les formes résolues sont sans structure: la procédure de transformation peut être vue comme le défrichement pas à pas du champ possible pour les preuves, jusqu'à ce que n'importe quelle preuve (qui n'essaie pas de creuser ses hypothèses) marche sans problème.

Definition 33 (Forme résolue) *Un système de contraintes* $D = \exists \tilde{z}.[H_1 \overset{?}{\vdash} x_1 \wedge \ldots \wedge H_n \overset{?}{\vdash} x_n \mid \mathsf{E}]$ *est en* forme résolue *quand* $x_1, \ldots, x_n$ *sont des variables distinctes.*

Lemme 46 *Une forme résolue* $D = \exists \tilde{z}.[H_1 \overset{?}{\vdash} x_1 \wedge \ldots \wedge H_n \overset{?}{\vdash} x_n \mid \mathsf{E}]$ *a toujours une solution.*

Preuve: Soit $\pi_1, \ldots, \pi_n$ n'importe quelles preuves à partir (respectivement) des hypothèses $H_1, \ldots, H_n$, qui ne contiennent que des règles de composition. Par

monotonie et origination, elles definissent une substitution σ, avec $\mathsf{dom}(\sigma) = \{x_1, \ldots, x_n\}$, telle que $\pi_1, \ldots, \pi_n$ sont des preuves de $H_1\sigma \vdash x_1\sigma, \ldots, H_n\sigma \vdash x_n\sigma$. Si $\theta = \mathsf{mgu}(\mathsf{E})$, $\sigma \cup \theta\sigma$ est alors une solution de $H_1 \stackrel{?}{\vdash} x_1 \wedge \ldots \wedge H_n \stackrel{?}{\vdash} x_n \wedge \mathsf{E}$, qui, projétée sur $fvar(D)$, nous donne une solution de D. □

Pour montrer qu'on atteint les formes résolues, nous allons montrer que, pour un système D et une solution donnée σ, une certaine mesure décroît en appliquant les règles de transformation. Pour que cette mesure soit bien définie, il faut pouvoir étendre σ aux variables liées de D d'une manière unique. C'est le but de l'invariant suivant.

Definition 34 (système uniquement determiné) *Un système de contraintes* $D = \exists\tilde{z}.[\![C \mid \mathsf{E}]\!]$ *est* uniquement determiné *si, pour toute substitution close* σ *telle que* $\mathsf{dom}(\sigma) = fvar(D)$*, il existe des termes clos* $u_1, \ldots, u_\ell$ *tels que, si* $\mathsf{mgu}(\mathsf{E}\sigma) \neq \bot$*, alors* $\mathsf{mgu}(\mathsf{E}\sigma) = \{z_1 = u_1, \ldots, z_\ell = u_\ell\}$*, où* $\tilde{z} = \{z_1, \ldots, z_\ell\}$*. Dans ce cas, on définit* $\overline{\sigma}$ *par* $\sigma \cup \mathsf{mgu}(\mathsf{E}\sigma)$*. Notons que* $\overline{\sigma}$ *est une solution du système de contraintes* $[\![C \mid \mathsf{E}]\!]$*. Nous appellerons* $\overline{\sigma}$ *l'*extension *de* σ *par rapport à* D*.*

Exemple 56 *Soit* D' *le système de contraintes donné dans l'exemple 55. On a* $fvar(D) = \{x_0, x_1\}$*. Dès que des valeurs sont assignées aux variables* x_0, x_1*, il existe une unique substitution* τ *qui satisfait les équations de* $\mathsf{E}(D')$*.*

Notons que les systèmes de contraintes de départ (avant l'application des règles de transformation) sont trivialement uniquement determinés, car leur ensemble d'équations est vide.

Le lemme suivant repose sur le fait que, si un ensemble d'équations E a une solution unique, tout ensemble E', tel que $\mathsf{E} \subseteq \mathsf{E}', \mathsf{Var}(E) = \mathsf{Var}(E')$ et $\mathsf{mgu}(E') \neq \bot$, a une solution unique.

Lemme 47 *Soit* D *un système de contraintes uniquement determiné et* D' *tel que* $D \rightsquigarrow D'$*. Le système* D' *est alors uniquement determiné.*

Preuve: Soit $D = \exists\tilde{z}.[\![C \mid \mathsf{E}]\!]$ et $D' = \exists\tilde{z'}.[\![C' \mid \mathsf{E}']\!]$ deux systèmes de contraintes tels que $D \rightsquigarrow D'$. Nous montrons le résultat en effectuant une étude de cas sur la règle de transformation utilisée.

- Règles Triv et Comp. Dans ce cas, on a $\tilde{z'} = \tilde{z}$ et $\mathsf{E}' = \mathsf{E}$: on conclut immédiatement.

- Règle Axiom. Dans ce cas, $\tilde{z'} = \tilde{z}$ et $\mathsf{E}' = \mathsf{E} \cup \theta$, avec θ un ensemble d'équations tel que $\mathsf{Var}(\theta) \subseteq \mathsf{Var}(C)$. Soit ρ une substitution close avec $\mathsf{dom}(\rho) = fvar(D') = fvar(D)$ et supposons que $\mathsf{mgu}(\mathsf{E}'\rho) \neq \bot$. Puisque $\mathsf{E}' = \mathsf{E} \cup \theta$, on a alors $\mathsf{mgu}(\mathsf{E}\rho) \neq \bot$. En utilisant le fait que D est uniquement determiné, on déduit que $\tau = \mathsf{mgu}(\mathsf{E}\rho)$ est une substitution close. On a donc $\mathsf{mgu}(\mathsf{E}'\rho) = \mathsf{mgu}(\mathsf{E}\rho) = \tau$, ce qui nous permet de conclure.

- Règle Dec. Dans ce cas, on a $\tilde{z'} = \tilde{z} \cup \tilde{x}$ et $\mathsf{E}' = \mathsf{E} \cup \theta$, avec $\theta = \mathsf{mgu}(u, v)$ pour des termes u, v tels que $\mathsf{Var}(u) \subseteq \mathsf{Var}(C)$ et $\mathsf{Var}(v) = \tilde{x}$. Soit ρ une substitution close avec $\mathsf{dom}(\rho) = fvar(D') = fvar(D)$ et supposons que $\mathsf{mgu}(\mathsf{E}'\rho) \neq \bot$. Comme dans le cas précédent, on déduit que $\tau = \mathsf{mgu}(\mathsf{E}\rho)$ est une substitution close. D'autre part, on a

$$\mathsf{mgu}(\mathsf{E}'\rho) = \mathsf{mgu}(\mathsf{E}\rho \cup \{u\rho = v\}) = \tau \cup \mathsf{mgu}(u(\rho \cup \tau) = v).$$

  Il est facile de voir que cette substitution est close et que son domaine est $\tilde{z} \cup \tilde{x}$. Ceci est du au fait que $\tilde{x} = \mathsf{Var}(v)$ et $u(\rho \cup \tau)$ est un terme clos. □

Dans la suite, nous considérons donc seulement des systèmes uniquement determinés, en le rappellant seulement quand nous en avons besoin.

Nous allons maintenant montrer le résultat suivant: à toute solution σ d'un système D, nos règles de transformation associent un chemin vers une forme resolue, dans le système de transitions sous-jacent. Ce chemin correspond aux parties non-triviales des preuves pour σ, qui sont contraintes par le système, tandis que la forme résolue correspond aux parties malléables de preuves.

Soit $H_1 \subseteq H_2 \subseteq \ldots \subseteq H_n$ une séquence d'ensembles de termes. Soit π une preuve de $H_i \vdash u$, pour un i, $1 \leq i \leq n$. Nous associons à π l'ensemble minimal $\mathtt{Hyp}(\pi) \in \{H_1, \ldots, H_n\}$ qui contient les feuilles de π. Bien sûr, $\mathtt{Hyp}(\pi) \subseteq H_i$. Étant donné un système de contraintes $D = \exists\tilde{z}.[C \mid \mathsf{E}]$ uniquement determiné et une solution σ de D, une preuve *simple* par rapport à D est par définition une preuve simple par rapport à la séquence d'ensembles $\mathsf{LH}(D)\overline{\sigma}$.

Dans un premier temps, nous allons montrer que les sous-termes qui apparaissent dans les preuves sont ou bien des sous-termes des hypothèses, ou bien leur preuve simple termine par une règle de composition ou une règle versatile.

Lemme 48 *Soit D un système de contraintes de la forme $[C \mid \mathsf{E}]$, i.e. sans variables liées. Soit $H \in \mathsf{LH}(D)$ tel que pour tout $y \in \mathsf{Var}(H)$ il existe une contrainte $H_y \overset{?}{\vdash} y \in D$. Soit σ une solution de D et v un terme tel que $H\sigma \vdash v$. Soit $u \in st(v)$. On a:*

1. *ou bien $u \in (st(H) \setminus \mathcal{X})\sigma$;*
2. *ou bien $H\sigma \vdash u$ et toute preuve simple π de $H\sigma \vdash u$ termine par une règle de composition ou une règle versatile.*

Preuve: Soit Π l'ensemble des preuves simples de $H\sigma \vdash v$. Nous allons prouver le résultat par récurrence sur la paire (H, d) où d est la taille d'une preuve minimale (en taille) dans Π. Nous distinguons deux cas suivant la dernière règle d'inférence de preuves dans Π.

- *Il existe une preuve $\pi \in \Pi$ qui se termine avec une règle de décomposition ou un axiome.* Dans ce cas, grâce au lemme 44 (et trivialement, dans le cas d'un axiome), on a $v \in st(H\sigma)$, et donc $u \in st(H\sigma)$. Ou bien

$u \in (st(H) \setminus \mathcal{X})\sigma$ et on conclut facilement (ceci se produit toujours quand H est clos, notre cas de base). Sinon, $u \in st(y\sigma)$ pour un $y \in \mathsf{Var}(H)$. Par origination et notre hypothèse sur D, on a $H_y \overset{?}{\vdash} y \in D$, avec $H_y \subsetneq H$. Puisque σ est une solution de D, $H_y\sigma \vdash y\sigma$. Par conséquent, on peut appliquer l'hypothèse de récurrence (pour H_y, $y\sigma$ et u) pour conclure.

- *Chaque preuve $\pi \in \Pi$ se termine par une règle de composition ou une règle versatile.* Choisissons une preuve $\pi \in \Pi$ minimale en taille parmi les preuves dans Π. Soit $\pi_1, \ldots, \pi_n$ les sous-preuves immédiates de π et $u_1, \ldots, u_n$ les conclusion de $\pi_1, \ldots, \pi_n$. Si $u = v$, on conclut. Sinon, *puisque les sous-termes stricts de la conclusion sont des sous-termes des prémisses* (notre condition 2a sur les systèmes d'inférence), on a $u \in st(u_j)$, pour un j tel que $1 \leq j \leq n$. On a alors $H\sigma \vdash u_j$ et $u \in st(u_j)$. Par conséquent, on peut appliquer l'hypothèse de récurrence, puisque π_j est une preuve simple de $H\sigma \vdash u_j$ et la taille de π_j est plus petite que la taille de π. Ce qui nous permet de conclure le lemme. □

Maintenant, nous définissons une mesure de complexité pour les preuves qui temoignent du fait que σ est une solution de D. Nous montrons, dans le lemme 49, qu'il existe toujours une règle qui donne une complexité strictement inférieure, jusqu'à ce qu'une forme résolue soit atteinte.

Mesure PS. Soit $D = \exists\tilde{z}.[C \mid \mathsf{E}]$ un système de contraintes uniquement determiné, σ une solution de D et $\overline{\sigma}$ l'extension de σ par rapport à D.

- Si $H \overset{?}{\vdash} u \in C$ alors $\mathsf{PS}(H \overset{?}{\vdash} u, \sigma)$ est par définition la taille (i.e. le nombre de noeuds) minimale d'une preuve simple de $H\overline{\sigma} \vdash u\overline{\sigma}$.

- Si $\mathsf{LH}(D) = \{H_1, \ldots, H_n\}$, avec $H_1 \subsetneq \ldots \subsetneq H_n$, alors le *niveau* $\mathsf{lev}(H \overset{?}{\vdash} u, D)$ d'une contrainte de déductibilité $H \overset{?}{\vdash} u \in C$ est l'indice i tel que $H = H_i$.

La mesure PS est étendue aux systèmes de contraintes en définissant, pour toute solution σ de D, $\mathsf{PS}(D, \sigma)$ comme étant le multi-ensemble des paires $(\mathsf{lev}(H \overset{?}{\vdash} u, D), \mathsf{PS}(H \overset{?}{\vdash} u, \sigma))$, pour toutes les contraintes de déductibilité $H \overset{?}{\vdash} u \in D$. Les multi-ensembles $\mathsf{PS}(D, \sigma)$ sont comparés en utilisant l'extension multi-ensemble de la composition lexicographique des ordres.

Notons que, par les règles de transformation, le nombre de niveaux différents dans un système de contraintes peut décroître, mais jamais croître.

Si D n'est pas en forme résolue, il doit exister une contrainte $H \overset{?}{\vdash} u \in D$ telle que u n'est pas une variable. Nous considérons une telle contrainte, avec un membre gauche minimal. En fonction de la dernière règle d'une preuve simple de $H\sigma \vdash u\sigma$ minimale en taille, on peut alors appliquer une règle de transformation, qui nous donne un PS plus petit:

Lemme 49 *Si D est un système de contraintes qui est uniquement determiné et $\sigma \in Sol(D)$, alors ou bien D est en forme résolue ou bien il existe un D' tel que $D \rightsquigarrow D'$, $\sigma \in Sol(D')$ et $\mathsf{PS}(D,\sigma) > \mathsf{PS}(D',\sigma)$.*

Preuve: Soit $D = \exists \tilde{z}.[C \mid \mathsf{E}]$ un système de contraintes qui n'est pas en forme résolue. Supposons d'abord que chaque contrainte de déductibilité de D est de la forme $H \stackrel{?}{\vdash} x$, où x est une variable. Notons que, pour avoir une forme résolue, toutes ces variables doivent être distinctes. Si D n'est pas en forme résolue, on peut donc appliquer Triv, en obtenant un système D' tel que $\mathsf{PS}(D,\sigma) > \mathsf{PS}(D',\sigma)$.

Supposons maintenant qu'il existe une contrainte de déductibilité dont le membre droit n'est pas une variable. Considérons une telle contrainte, disons $H \stackrel{?}{\vdash} s$, telle que s n'est pas une variable et dont le membre gauche H est minimal par rapport à l'inclusion. Puisque σ est une solution de D, on sait qu'il existe une substitution close τ avec $\mathsf{dom}(\tau) = \tilde{z}$ et telle que $\sigma \cup \tau$ satisfait chaque contrainte dans C et E. Puisque D est uniquement determiné, on sait qu'une telle substitution τ est unique. Soit $\overline{\sigma} = \sigma \cup \tau$.

Soit π une preuve simple de $H\overline{\sigma} \vdash s\overline{\sigma}$ qui est minimale en taille (notons qu'une telle preuve existe, grâce au lemme 43). Nous distinguons trois cas, en fonction de la dernière règle de π.

- Si π est réduite à une feuille, on peut appliquer Axiom. On obtient un système D' tel que $\mathsf{PS}(D,\sigma) > \mathsf{PS}(D',\sigma)$. En effet, une contrainte de déductibilité a été supprimée.

- Si π termine par une instance d'une règle de composition, on peut appliquer Comp. On obtient un système de contraintes D'. On a évidemment σ solution de D' et $\mathsf{PS}(D,\sigma) > \mathsf{PS}(D',\sigma)$.

- Si π termine par une instance d'une règle de décomposition ou d'une règle versatile, disons
$$\mathsf{R} = \frac{v_1 \quad \ldots \quad v_m}{w}$$
alors on montre qu'on peut appliquer Dec.

  Soit θ la substitution close telle que $\mathsf{dom}(\theta) = \mathsf{Var}(\mathsf{R})$ et $\mathsf{R}\theta$ correspond à l'instance utilisée dans le dernier pas de la preuve π. On a $w\theta = s\overline{\sigma}$ et $v_1\theta, \ldots, v_m\theta$ sont les prémisses du dernier pas de la preuve. Soit $\pi_1, \ldots, \pi_m$ les sous-preuves immédiates de π. Puisque π est simple et donc normale, par le lemme 44, on a $v_i\theta \in st(H\overline{\sigma})$, pour tout i, $1 \leq i \leq m$. Pour chaque $1 \leq i \leq m$, on a:

  1. ou bien $v_i\theta \in (st(H) \setminus \mathcal{X})\overline{\sigma}$
  2. ou bien $v_i\theta \in st(x\overline{\sigma})$, pour un $x \in \mathsf{Var}(H)$. Par le choix de H et la propriété d'origination, on a $H_x\overline{\sigma} \vdash x\overline{\sigma}$, avec $H_x \subsetneq H$. Grâce au lemme 48 (appliqué à $[C \mid \mathsf{E}]$, $H_x\overline{\sigma}$, $x\overline{\sigma}$ et $v_i\theta$), on déduit que π_i est

une preuve simple de $H_x\overline{\sigma} \vdash v_i\theta$ et π_i termine par une instance d'une règle versatile ou d'une règle de composition. Par conséquent, les prémisses qui ne sont pas dans $(st(H) \setminus \mathcal{X})\overline{\sigma}$ peuvent être prouvées avec un membre gauche strictement plus petit, $H_x \subsetneq H$.

De plus, si v_i est maximal parmi $\{v_1, \ldots, v_m, w\}$, alors $v_i\theta \in (st(H) \setminus \mathcal{X})\overline{\sigma}$. En effet, dans un tel cas, puisque la preuve π est normale (i.e. sans mauvais motif), la preuve π_i ne peut pas se terminer avec une instance d'une règle versatile ou de composition. Ainsi le deuxième cas ci-dessus n'est pas possible.

Ainsi, nous avons que $D \rightsquigarrow D'$ avec la règle Dec et σ est une solution de D'. De plus, $\mathsf{PS}(D, \sigma) > \mathsf{PS}(D', \sigma)$ et on peut conclure. □

Pour conclure la complétude, il nous reste à énoncer le lemme suivant, qui suit immédiatement par récurrence sur $\mathsf{PS}(D, \sigma)$, en appliquant le lemme 49 à chaque pas. Le lemme 47 nous permet d'assurer que le système résultant est uniquement déterminé et d'appliquer l'hypothèse de récurrence.

Lemme 50 (complétude) *Si D est un système de contraintes qui est uniquement determiné et $\sigma \in Sol(D)$, alors il existe un système de contraintes en forme résolue D' tel que $D \rightsquigarrow^* D'$ et $\sigma \in Sol(D')$.*

11.2.4 Terminaison

Si le système d'inférence saturé est infini, notre procédure n'est évidemment pas effective, puisqu'à chaque règle d'inférence correspond une règle de transformation. Nous verrons dans le chapitre suivant comment on peut traiter ce genre d'exemple. Nous supposons pour l'instant que le système d'inférence est fini et nous cherchons à montrer que dans ce cas il existe une stratégie pour laquelle nos règles terminent.

Comme en témoigne l'exemple ci-dessous, même avec nos restrictions sur l'application de Dec et pour un système d'inférence fini, nos règles ne terminent pas:

Exemple 57 *Considérons une théorie avec la règle de décomposition:*

$$\frac{f(f(x,y), f(x',y')) \quad f(x',y')}{f(x,y)}$$

et une règle de composition pour f. En appliquant nos règles de transformation, on a la séquence infinie suivante ($T \overset{?}{\vdash} u, v$ est un raccourci pour $T \overset{?}{\vdash} u \wedge T \overset{?}{\vdash} v$):

$$\mathcal{C} = \left\{ \begin{array}{rcl} [a & \stackrel{?}{\vdash} & x_0, y_0 \\ a, f(x_0, y_0) & \stackrel{?}{\vdash} & f(x, y) \\ & - & \\ & \emptyset &] \end{array} \right.$$

$$\rightsquigarrow_{\mathsf{Dec}} \quad \exists x_1, y_1. \left\{ \begin{array}{rcl} [a & \stackrel{?}{\vdash} & f(x, y), f(x_1, y_1) \\ a & \stackrel{?}{\vdash} & f(x_1, y_1) \\ a, f(f(x, y), f(x_1, y_1)) & \stackrel{?}{\vdash} & f(f(x, y), f(x_1, y_1)) \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(x_1, y_1)] \end{array} \right.$$

$$\rightsquigarrow^*_{\mathsf{Comp},\mathsf{Triv}} \quad \exists x_1, y_1. \left\{ \begin{array}{rcl} [a & \stackrel{?}{\vdash} & x, y, x_1, y_1 \\ a, f(f(x, y), f(x_1, y_1)) & \stackrel{?}{\vdash} & f(x, y) \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(x_1, y_1)] \end{array} \right.$$

$$\rightsquigarrow_{\mathsf{Dec}} \quad \exists x_1, y_1, x_2, y_2. \left\{ \begin{array}{rcl} [a & \stackrel{?}{\vdash} & x, y, f(x, y), f(x_2, y_2) \\ a & \stackrel{?}{\vdash} & f(x_2, y_2) \\ a, f(f(x, y), f(f(x, y), f(x_2, y_2))) & \stackrel{?}{\vdash} & f(f(x, y), f(x_2, y_2)) \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(f(x, y), f(x_2, y_2)) \\ x_1 & \mapsto & f(x, y) \\ y_1 & \mapsto & f(x_2, y_2)] \end{array} \right.$$

$$\rightsquigarrow^*_{\mathsf{Comp},\mathsf{Triv}} \quad \exists x_1, y_1, x_2, y_2. \left\{ \begin{array}{rcl} [a & \stackrel{?}{\vdash} & x, y, x_2, y_2 \\ a, f(f(x, y), f(f(x, y), f(x_2, y_2))) & \stackrel{?}{\vdash} & f(x, y) \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(f(x, y), f(x_2, y_2)) \\ x_1 & \mapsto & f(x, y) \\ y_1 & \mapsto & f(x_2, y_2)] \end{array} \right.$$

$$\rightsquigarrow_{\mathsf{Dec}} \quad \ldots$$

11.3 Stratégie pour la terminaison

L'exemple 57 nous montre qu'il nous faut des stratégies supplémentaires pour obtenir la terminaison. Pour cela, dans certaines applications de Dec, nous allons mettre de coté certaines contraintes et certaines équations. Les règles de transformation ne pourront pas s'appliquer sur ces contraintes, tant que le reste (la partie dite active) ne sera pas sous forme résolue. Une fois la partie active du système en forme résolue, nous ouvrirons le frigidaire afin de continuer la transformation. Pour obtenir la terminaison nous allons montrer qu'une forme résolue sur la partie active est atteinte en un nombre fini de pas et que le nombre total d'ouvertures du frigidaire est borné par des paramètres du système de départ.

11.3.1 La stratégie

Pour une variable x et un système de contraintes D, le *niveau* $\mathsf{lev}(x, D)$ de x est le niveau de H_x dans D, si $x \in \mathsf{Var}(D)$, et 0, sinon. Les contraintes de déductibilité de D sont séparées dans une *partie active* $\mathsf{Act}(D)$ et une *partie gelée* $\mathsf{Fr}(D)$.

Definition 35 (Système de contraintes étendu) *Un* système de contraintes étendu D *est une formule* $\exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$*, où:*

- $\tilde{z}$ *est une séquence de variables;*
- $\mathsf{E}(D) \stackrel{\mathsf{def}}{=} \mathsf{E}$ *est un ensemble d'équations (pas nécessairement en forme résolue), avec* $\mathsf{mgu}(\mathsf{E}) \neq \bot$*;*
- $\mathsf{Act}(D) \stackrel{\mathsf{def}}{=} \mathsf{A}$*, la* partie active *de* D*, et* $\mathsf{Fr}(D) \stackrel{\mathsf{def}}{=} \mathsf{F}$*, la* partie gelée *de* D*, sont des ensembles de contraintes de déductibilité;* A *et* $(\mathsf{A} \cup \mathsf{F})\mathsf{mgu}(\mathsf{E})$ *sont des systèmes de contraintes.*

Soit $\theta = \mathsf{mgu}(\mathsf{E})$*. Une solution de* $\exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$ *est par définition une solution de* $\exists\tilde{z}.[(\mathsf{A} \cup \mathsf{F})\theta \mid \theta]$*. Le système* D *est en forme résolue quand* $\mathsf{Fr}(D) = \emptyset$ *et* $\mathsf{Act}(D)\theta$ *est en forme résolue.*

Les règles de transformation sont traduites dans des règles de transformation pour les systèmes étendus. Dans le système initial, rien n'est gelé. De plus, toutes les règles s'appliquent sur la partie active et toutes les règles (à part Dec) modifient seulement la partie active.

- Quand la règle Dec est appliquée à une contrainte $H \stackrel{?}{\vdash} v$ telle que, pour une variable $x \in \mathsf{Var}(v)$, on a $\mathsf{lev}(x, \mathsf{Act}(D)) = \mathsf{lev}(H, \mathsf{Act}(D))$, elle contribue aussi seulement à la partie active.
- Autrement, quand pour tout $x \in \mathsf{Var}(v)$, $\mathsf{lev}(x, \mathsf{Act}(D)) < \mathsf{lev}(H, \mathsf{Act}(D))$, alors seulement les contraintes dont les membres droits sont des sous-termes de la contrainte sont gardées dans la partie active, le reste étant stocké dans la partie gelée.

Quand $\mathsf{Act}(D)$ est en forme résolue (et seulement à ce moment là), on ouvre le frigidaire et on verse son contenu dans la partie active, en effectuant tous les remplacements nécéssaires. Ces règles sont décrites dans la figure 11.2. Nous dénoterons cette rélation de transition par $\mapsto$. Parfois, nous utiliserons $\mapsto_{\mathsf{A/F}}$ à la place de $\mapsto_{\mathsf{A}} \cup \mapsto_{\mathsf{F}}$.

(Active) $\quad \exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}] \mapsto_{\mathsf{A}} \exists\tilde{z} \cup \tilde{x}.[\mathsf{A}' \mid \mathsf{F} \mid \mathsf{E} \cup \theta]$

si $\mathsf{A} \rightsquigarrow \exists\tilde{x}.[\mathsf{A}' \mid \theta]$ en utilisant Axiom, Triv, Comp; ou Dec sur $H \overset{?}{\vdash} v$ et il existe $x \in \mathsf{Var}(v)$ tel que $\mathsf{lev}(x, \mathsf{A}) = \mathsf{lev}(H, \mathsf{A})$. On assume que $\mathsf{mgu}(\mathsf{E} \cup \theta) \neq \bot$.

(Freeze) $\quad \exists\tilde{z}.[\mathsf{A} \wedge H \overset{?}{\vdash} v \mid \mathsf{F} \mid \mathsf{E}] \mapsto_{\mathsf{F}} \exists\tilde{z} \cup \tilde{x}.[\mathsf{A} \wedge H \overset{?}{\vdash} u_1 \wedge \ldots H \overset{?}{\vdash} u_n \mid \mathsf{F} \wedge H' \overset{?}{\vdash} v_1 \wedge \ldots \wedge H' \overset{?}{\vdash} v_m \mid \mathsf{E} \cup \theta]$

où:

- $\mathsf{R} = \dfrac{v_1 \ \ldots \ v_m \ \ w_1 \ \ldots \ w_n}{w}$ est une règle de décomposition ou versatile telle que $\mathtt{Max}(\mathsf{R}) \subseteq \{w_1, \ldots, w_n\}$ et $\tilde{x} = \mathsf{Var}(\mathsf{R})$;
- $\theta = \mathsf{mgu}(\langle w, w_1, \ldots, w_n\rangle, \langle v, u_1, \ldots, u_n\rangle)$, $u_1, \ldots, u_n \in st(H) \setminus \mathcal{X}$, et $v \notin \mathcal{X}$;
- H' est un membre gauche d'une contrainte de déductibilité dans A tel que $H' \subsetneq H$;
- $\mathsf{mgu}(\mathsf{E} \cup \theta) \neq \bot$ et $\mathsf{lev}(x, \mathsf{A} \wedge H \overset{?}{\vdash} v) < \mathsf{lev}(H, A \wedge H \overset{?}{\vdash} v)$ pour tout $x \in \mathsf{Var}(v)$.

(Open) $\quad \exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}] \mapsto_{\mathsf{O}} \exists\tilde{z}.[(\mathsf{A} \cup \mathsf{F})\theta \mid \emptyset \mid \theta]$ quand A est en forme résolue et $\theta = \mathsf{mgu}(\mathsf{E})$

Figure 11.2: Transformation des systèmes étendus

Dernière restriction de la stratégie: quand on est dans une boucle sur la partie active, en utilisant seulement les règles Active et Freeze, i.e. $D \mapsto^* D_1 \mapsto^*_{\mathsf{A/F}} D_2$ et $\mathsf{Act}(D_1) = \mathsf{Act}(D_2)$, on enlève toutes les branches qui commencent avec un tel prefixe.

Nous allons montrer que cette stratégie:

- est *complète*: pour tout D, pour tout $\sigma \in Sol(D)$, il existe une séquence $D \mapsto^* D'$ autorisée par la stratégie telle que $\sigma \in Sol(D')$ et D' est en forme résolue (section 11.3.2).
- *termine*: il n'existe pas de séquence infinie de transformation (section 11.3.3).

Montrons d'abord comment cette stratégie fait terminer la transformation de l'exemple précédent.

Exemple 58 *La première application de* Dec *est active, car les variables x, y sont introduites dans la contrainte en question. Ainsi, rien ne change dans la transformation. Pour la deuxième application de* Dec*, la stratégie nous oblige à appliquer* Freeze *et ainsi mettre $a \overset{?}{\vdash} f(x_2, y_2)$ au frigidaire et ne pas appliquer la substitution $x_1 \mapsto f(x, y), y_1 \mapsto f(x_2, y_2)$ sur la partie active:*

$$\mathcal{C} = \left\{ \begin{array}{rcl} [a & \overset{?}{\vdash} & x_0, y_0 \\ a, f(x_0, y_0) & \overset{?}{\vdash} & f(x, y) \\ & - & \\ & \emptyset & \\ & - & \\ & \emptyset &] \end{array} \right.$$

$$\mapsto^*_{\mathsf{A(Dec,Comp,Triv)}} \quad \exists x_1, y_1. \left\{ \begin{array}{rcl} [a & \overset{?}{\vdash} & x, y, x_1, y_1 \\ a, f(f(x, y), f(x_1, y_1)) & \overset{?}{\vdash} & f(x, y) \\ & - & \\ & \emptyset & \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(x_1, y_1)] \end{array} \right.$$

$$\mapsto_{\mathsf{F(Dec)}} \quad \exists x_1, y_1, x_2, y_2. \left\{ \begin{array}{rcl} [a & \overset{?}{\vdash} & x, y, x_1, y_1 \\ a, f(f(x, y), f(x_1, y_1)) & \overset{?}{\vdash} & f(x_1, y_1) \\ & - & \\ a & \overset{?}{\vdash} & f(x_2, y_2) \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(x_1, y_1) \\ x_1 & \mapsto & f(x, y) \\ y_1 & \mapsto & f(x_2, y_2)] \end{array} \right.$$

Puisque les boucles sur la partie active sont interdites, les seules transformations possibles restent les chemins vers le système suivant, où la partie active

est en forme résolue.

$$\exists x_1, y_1, x_2, y_2.\left\{\begin{array}{lcl} [a & \overset{?}{\vdash} & x, y, x_1, y_1 \\ & - & \\ a & \overset{?}{\vdash} & f(x_2, y_2) \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(x_1, y_1) \\ x_1 & \mapsto & f(x, y) \\ y_1 & \mapsto & f(x_2, y_2)] \end{array}\right.$$

De plus, quand on ouvre le frigidaire ($\mapsto_{\mathsf{O}}$), le système obtenu a perdu un niveau:

$$\left\{\begin{array}{lcl} [a & \overset{?}{\vdash} & x, y, f(x, y), f(x_2, y_2) \\ a & \overset{?}{\vdash} & f(x_2, y_2) \\ & - & \\ & \emptyset & \\ & - & \\ x_0 & \mapsto & f(x, y) \\ y_0 & \mapsto & f(f(x, y), f(x_2, y_2)) \\ x_1 & \mapsto & f(x, y) \\ y_1 & \mapsto & f(x_2, y_2)] \end{array}\right.$$

Il est facile de voir que toutes les séquences de transformation à partir de ce système sont finies.

Dans un premier temps, nous allons montrer la correction de nos règles de transformation.

Lemme 51 (Correction) *Si D est un système de contraintes étendu tel que $D \mapsto D'$, alors D' est un système de contraintes étendu et $Sol(D') \subseteq Sol(D)$.*

Preuve: En utilisant la définition des règles de transformation et le lemme 45, il est facile de voir que $(\mathsf{Act}(D') \cup \mathsf{Fr}(D'))\theta'$, ou $\theta' = \mathsf{mgu}(\mathsf{E}(D'))$ est un système de contraintes. Le plus important à montrer est que $\mathsf{Act}(D')$ est un système de contraintes. On distingue trois cas:

Regle Active. Le résultat est immédiat, car $\mathsf{Act}(D) \rightsquigarrow \exists \tilde{x}.[\mathsf{Act}(D') \mid \theta]$, pour certaines variables $\tilde{x}$, et une substitution θ. Le lemme 45 nous permet de conclure.

Regle Freeze. Dans ce cas, $\mathsf{Act}(D') = \mathsf{Act}(D) \setminus \{H \overset{?}{\vdash} v\} \cup \{H \overset{?}{\vdash} u_1, \ldots, H \overset{?}{\vdash} u_n\}$ et on a $H_x \subsetneq H$ pour tout $x \in \mathsf{Var}(v)$. L'origination est donc satisfaite pour le système $\mathsf{Act}(D')$.

Regle Open. Dans ce cas, on a $\mathsf{Act}(D') = (\mathsf{Act}(D) \cup \mathsf{Fr}(D))\theta$, où $\theta = \mathsf{mgu}(\mathsf{E})$. Par définition, $\mathsf{Act}(D')$ est un système de contraintes, puisque D est un système de contraintes étendu.

Le fait que $Sol(D') \subseteq Sol(D)$ est une conséquence immédiate du lemme 45.
□

11.3.2 Terminaison de la stratégie

Nous allons montrer la terminaison en deux temps. Dans un premier temps, nous prouvons que la relation $\mapsto_{\mathsf{A/F}}$ termine: il n'existe pas de séquence infinie de transformation sans ouvrir le frigidaire. Dans un deuxième temps, nous montrons que le nombre total d'ouvertures du frigidaire est, lui aussi, borné.

Terminaison sans ouverture du frigidaire.

Maintenant, on clarifie le rôle du frigidaire, en montrant que le niveau des variables dans la partie active ne croît pas. De plus, si des nouvelles variables sont introduites dans la partie active, leur niveau est strictement inférieur au niveau d'une ancienne variable, dont le niveau a décru strictement.

Lemme 52 *Si $D \mapsto_{\mathsf{A/F}} D'$, alors:*

1. $\mathsf{lev}(x, \mathsf{Act}(D')) \leq \mathsf{lev}(x, \mathsf{Act}(D))$*, pour tout* $x \in \mathsf{Var}(\mathsf{Act}(D))$.
2. *Soit* $M = \mathsf{Var}(\mathsf{Act}(D')) \setminus \mathsf{Var}(\mathsf{Act}(D))$. *Si* $M \neq \emptyset$*, alors il existe un* $x \in \mathsf{Var}(\mathsf{Act}(D))$ *tel que* $\mathsf{lev}(z, \mathsf{Act}(D')) < \mathsf{lev}(x, \mathsf{Act}(D))$*, pour tout* $z \in M \cup \{x\}$.

Preuve:
Nous faisons une étude de cas, en fonction de la règle de transformation utilisée dans la transition $D \mapsto_{\mathsf{A/F}} D'$.

Règle Triv ou Comp (Active): Dans ce cas, $\mathsf{Var}(\mathsf{Act}(D')) \setminus \mathsf{Var}(\mathsf{Act}(D)) = \emptyset$ et il est facile de voir que $\mathsf{lev}(x, \mathsf{Act}(D')) \leq \mathsf{lev}(x, \mathsf{Act}(D))$ pour tout $x \in \mathsf{Var}(\mathsf{Act}(D))$.

Règle Axiom (Active): Dans ce cas, $M = \mathsf{Var}(\mathsf{Act}(D')) \setminus \mathsf{Var}(\mathsf{Act}(D)) = \emptyset$, puisque la règle Axiom n'introduit pas de nouvelles variables. En fait, on a $\mathsf{Act}(D') = (\mathsf{Act}(D) \setminus \{H \stackrel{?}{\vdash} u\})\theta$ où $\theta = \mathsf{mgu}(u, v)$ avec $v \in H$. L'application d'une substitution, telle que θ, peut seulement faire décroître le niveau des variables existantes. On peut conclure, car l'élimination d'une contrainte ne peut pas faire croître le niveau d'une variable.

Règle Dec (Freeze): Dans ce cas, $M = \mathsf{Var}(\mathsf{Act}(D')) \setminus \mathsf{Var}(\mathsf{Act}(D)) = \emptyset$. On doit donc montrer seulement le premier point. Pour conclure, il est en fait facile d'observer que $\mathsf{lev}(x, \mathsf{Act}(D')) = \mathsf{lev}(x, \mathsf{Act}(D))$ pour tout $x \in \mathsf{Var}(\mathsf{Act}(D))$.

Règle Dec (Active): soit $H \stackrel{?}{\vdash} v$ la contrainte sur laquelle Dec est appliquée. Soit $\mathsf{R} = \dfrac{v_1 \ \ldots \ v_m \ \ w_1 \ \ldots \ w_n}{w}$ la règle d'inférence appliquée. Par définition d'une décomposition active, il existe $x_0 \in \mathsf{Var}(v)$ telle que $\mathsf{lev}(x_0, \mathsf{Act}(D)) = \mathsf{lev}(H, \mathsf{Act}(D))$. Soit θ le mgu calculé par la transformation.

- On a $\mathsf{lev}(x, \mathsf{Act}(D')) \leq \mathsf{lev}(x, \mathsf{Act}(D))$, pour tout $x \in \mathsf{Var}(\mathsf{Act}(D))$. En effet, l'application d'une substitution peut seulement faire decroître le niveau d'une variable qui est déjadans $\mathsf{Act}(D)$. Si la variable disparaît, son niveau devient 0.

- Soit $z \in M \cup \{x_0\}$. On montre que $\mathsf{lev}(z, \mathsf{Act}(D')) < \mathsf{lev}(H, \mathsf{Act}(D)) = \mathsf{lev}(x_0, \mathsf{Act}(D))$. Si $z \in \mathsf{dom}(\theta)$, alors $\mathsf{lev}(z, \mathsf{Act}(D')) = 0$ et on conclut. Admettons que ce n'est pas le cas. Si $z = x_0$, puisque $v\theta = w\theta$, il existe une variable $y \in \mathsf{Var}(w)$ telle que $z \in \mathsf{Var}(y\theta)$. Grâce a notre condition 2b sur les règles versatiles, on a:

  $$\mathsf{Var}(w) \subseteq \mathsf{Var}(\{v_1, \ldots, v_m, w_1, \ldots, w_n\}) \subseteq \mathsf{Var}(\{w_1, \ldots, w_n\}).$$

  Par conséquent, il existe i tel que $y \in \mathsf{Var}(w_i)$. De même, pour $z \in M \subseteq \mathsf{Var}(\mathsf{R})$, il existe i tel que $z \in \mathsf{Var}(w_i)$. Prenons une variable $w' \in \{y, z\}$.

  On sait que $w_i\theta = u_i\theta$ pour un $u_i \in st(H)$. Par origination, on a $H_x \subsetneq H$ pour tout $x \in \mathsf{Var}(u_i)$. Alors, ou bien il existe $u' \in st(u_i)$ tel que $w'\theta = u'\theta$, ou bien il existe une variable $x \in \mathsf{Var}(u_i)$ telle que $w'\theta \in st(x\theta)$. Dans les deux cas, on déduit que, pour chaque $z_0 \in \mathsf{Var}(w'\theta)$, il existe un $y_0 \in \mathsf{Var}(H)$ tel que $z_0 \in \mathsf{Var}(y_0\theta)$. On conclut par origination que

  $$\mathsf{lev}(z_0, \mathsf{Act}(D')) \leq \mathsf{lev}(H, \mathsf{Act}(D))$$

  pour chaque $z_0 \in \mathsf{Var}(w'\theta)$ et $w' \in \{y, z\}$. Comme $x_0 \in \mathsf{Var}(y\theta)$ et $z\theta = z$, si $z \in M$, on conclut que $\mathsf{lev}(z, \mathsf{Act}(D')) \leq \mathsf{lev}(H, \mathsf{Act}(D))$, pour chaque $z \in M \cup \{x_0\}$. □

Nous montrons maintenant un premier lemme de terminaison: il n'existe pas de séquence infinie de transformation sans ouverture du frigidaire. La raison est que, ou bien on n'introduit pas de variables, et alors il doit exister une boucle (ceci est interdit par la stratégie), ou bien, grâce au lemme 52, il existe un niveau ℓ tel que le niveau d'une variable de niveau ℓ décroît strictement, tandis que le niveau des variables nouvelles est strictement inférieur a ℓ.

Lemme 53 *Pour tout système de contraintes D, toute séquence $\mapsto^*_{\mathsf{A/F}}$, issue de D, qui respecte la stratégie (i.e. pas de boucles sur la partie active) termine.*

Preuve: Étant donné un système de contraintes D, considérons le multi-ensemble de niveaux, dans la partie active, de variables de $\mathsf{Act}(D)$, i.e.

$$M(D) = \{\mathsf{lev}(x, \mathsf{Act}(D)) \mid x \in \mathsf{Var}(\mathsf{Act}(D))\}.$$

Ces multi-ensembles sont ordonnés en utilisant l'extension multi-ensemble de l'ordre sur les entiers.

Grâce au lemme 52, nous savons que $D \mapsto_{\mathsf{A/F}} D'$ implique:

- ou bien $\mathsf{Var}(\mathsf{Act}(D')) \subseteq \mathsf{Var}(\mathsf{Act}(D))$ et $M(D') \leq M(D)$;

- ou bien $M(D') < M(D)$.

Par conséquent, dans toute séquence de transformation, il ne peut exister qu'un nombre fini de règles qui introduisent des variables. Ainsi, dans toute séquence de transformation infinie, il doit exister une (sous-)séquence infinie dont l'ensemble des variables dans la partie active est constant.

Montrons que, dès que l'ensemble des variables dans la partie active est constant, on ne peut rajouter aucun nouveau sous-terme dans la partie active du système de contraintes. En effet, dans ce cas, seule une règle de type Active Dec peut rajouter des nouveaux sous-termes. Cependant, pour chaque nouveau sous-terme maximal $v_i\theta$ introduit par une règle Active Dec, il existe un terme correspondant $u_i \in st(\mathsf{Act}(D))$, tel que $u_i\theta = v_i\theta$. Puisque l'ensemble des variables est fixé dans la partie active, on a $u_i\theta = u_i$, et donc $v_i\theta = u_i \in st(\mathsf{Act}(D))$. Puisque tout $w_j\theta$, avec w_j non-maximal, introduit par Dec est sous-terme d'un $v_i\theta$ maximal (grâce à la condition 2b), on conclut que les sous-termes des constraintes dans la partie active sont fixés. Par conséquent, toute séquence infinie doit boucler sur la partie active, ce qui est interdit par la stratégie. □

Le nombre d'ouvertures du frigidaire est borné.

Pour déduire la terminaison de la strategie, il est suffisant maintenant de montrer que le nombre d'applications de la règle $\mapsto_{\mathsf{O}}$ est borné. C'est le but de cette section. L'idée est que les variables x de niveau maximal apparaissent dans une contrainte $H \overset{?}{\vdash} x$ à l'ouverture du frigidaire. De plus, dans le frigidaire, toutes les variables ont un niveau strictement inférieur. Il s'ensuit que $H \overset{?}{\vdash} x$ ne va plus participer à aucune transformation. Les transformations futures s'effectueront seulement à des niveaux inférieurs. Plus précisement, nous montrons que, si ce n'est pas le cas, alors le nombre des niveaux differents dans le système décroit, comme illustré dans l'exemple 58.

Dans un premier temps, nous montrons un lemme préliminaire d'unification, qui va nous permettre d'analyser comment les variables de niveau maximal se distribuent dans la contrainte après l'ouverture du frigidaire:

Lemme 54 *Soit* $\mathsf{E} = \mathsf{E}_\mathsf{A} \cup \mathsf{E}_\mathsf{F}$ *un ensemble d'équations et* $L = L_1 \uplus L_2$ *un ensemble de variables qui satisfait les conditions suivantes:*

1. E_A *est un ensemble d'équations tel que* $\mathsf{mgu}(\mathsf{E}_\mathsf{A}) \neq \bot$. *On pose* $\sigma_\mathsf{A} = \mathsf{mgu}(\mathsf{E}_\mathsf{A})$.
2. $L \cap \mathsf{Var}(\mathsf{E}_\mathsf{F}) = \emptyset$
3. $\forall x \in L.\ \forall y \in \mathsf{dom}(\sigma_\mathsf{A}).\ x \in \mathsf{Var}(y\sigma_\mathsf{A}) \implies y \in L_2$
4. $\forall x \in L_1.\ x \notin \mathsf{dom}(\sigma_\mathsf{A})$

Soit $\theta = \mathsf{mgu}(\mathsf{E})$. *Nous avons:*

- $\forall x \in L_1. \forall y \in \mathsf{dom}(\theta).\ x \in \mathsf{Var}(y\theta) \implies y \in L_2$

- $\forall x \in L_1.\ x \notin \mathsf{dom}(\theta)$

Preuve: Soit $\sigma_{\mathsf{A}} = \sigma' \cup \sigma_0$, avec $\mathsf{dom}(\sigma') = \mathsf{dom}(\sigma_{\mathsf{A}}) \cap L_2$. On a donc $\mathsf{dom}(\sigma_0) = \mathsf{dom}(\sigma_{\mathsf{A}}) \setminus L_2$ et, grâce au point 4, on a $\mathsf{dom}(\sigma_0) \cap L_1 = \emptyset$. On en déduit que $\mathsf{dom}(\sigma_0) \cap L = \emptyset$. Grâce au point 3, on obtient $\mathsf{Var}(\mathsf{img}(\sigma_0)) \cap L = \emptyset$. Ainsi on conclut que $\mathsf{Var}(\mathsf{E_F} \cup \sigma_0) \cap L = \emptyset$.

Par conséquent, on déduit que $\theta = \mathsf{mgu}(\mathsf{E_F} \cup \sigma_0 \cup \sigma') = \sigma' \mathsf{mgu}(\mathsf{E_F} \cup \sigma_0) \cup \mathsf{mgu}(\mathsf{E_F} \cup \sigma_0)$. On obtient $\mathsf{dom}(\theta) \cap L_1 = \emptyset$ et , si $x \in \mathsf{Var}(y\theta) \cap L_1$ pour un $y \in \mathsf{dom}(\theta)$, alors $y \in L_2$. Cela nous permet de conclure. □

La définition suivante caractérise un niveau dans un système de contraintes au-dessus duquel on ne peut plus avoir de transformations:

Definition 36 (niveau actif non-résolu) *Soit D un système de contraintes étendu. Son* niveau actif non-résolu, *noté par* $\mathsf{unsolvedActlev}(D)$, *est le niveau minimal $\ell \in \{0, 1, \ldots, |\mathsf{LH}(\mathsf{Act}(D))|\}$ tel que:*

pour toute contrainte $H \overset{?}{\vdash} v \in \mathsf{Act}(D)$ de niveau (strictement) supérieur à ℓ, v est une variable et $\mathsf{lev}(v, \mathsf{Act}(D)) = \mathsf{lev}(H \overset{?}{\vdash} v, \mathsf{Act}(D))$.

L'ensemble de contraintes de niveau strictement supérieur à $\mathsf{unsolvedActlev}(D)$ contient exactement une contrainte $H_x \overset{?}{\vdash} x$, pour chaque des variable x telles que $\mathsf{lev}(x, \mathsf{Act}(D)) > \ell$. Puisqu'aucune règle (à part Triv) ne s'applique à des contraintes de la forme $H \overset{?}{\vdash} x$, il n'existe pas de règle de transformation qui affecte les contraintes dont le niveau est plus grand que $\mathsf{unsolvedActlev}(D)$. Par conséquent, le niveau actif non-résolu d'un système de contraintes ne peut pas croître le long d'une séquence de transformations.

Pour un système de contraintes étendu D, on dénote par $\mathsf{sActlev}(x, D)$ la taille (i.e. le nombre des elements) du membre gauche de la contrainte qui introduit la variable x dans $\mathsf{Act}(D)$. Par convention, $\mathsf{sActlev}(x, D) = 0$ quand $x \notin \mathsf{Act}(D)$. Enfin, on note $\mathsf{sunsolvedActlev}(D)$ la taille du membre gauche des contraintes de niveau $\mathsf{unsolvedActlev}(D)$.

Exemple 59 *Considérons le système de contraintes suivant:*

$$D = \left\{ \begin{array}{rcl} a & \overset{?}{\vdash} & x \\ a, x & \overset{?}{\vdash} & \mathit{enc}(x, y) \\ a, x, b & \overset{?}{\vdash} & y \\ a, x, b, y & \overset{?}{\vdash} & z \end{array} \right.$$

dans lequel toutes les contraintes sont actives. On a $\mathsf{unsolvedActlev}(D) = 3$ *(le niveau de la troisème contrainte),* $\mathsf{sunsolvedActlev}(D) = 3$ *(la taille de la troisième contrainte),* $\mathsf{sActlev}(x, D) = 1$, $\mathsf{sActlev}(y, D) = 2$ *et* $\mathsf{sActlev}(z, D) = 4$.

Avant la terminaison, nous montrons que la taille du membre gauche de la contrainte qui introduit une variable ne peut pas croître par transformation.

Lemme 55 *Soient D, D' deux systèmes de contraintes étendus tels que $D \mapsto_{\mathsf{A/F}} D'$. Pour tout $y \in \mathsf{Var}(D)$, on a $\mathsf{sActlev}(y, D') \leq \mathsf{sActlev}(y, D)$.*

Preuve: Soit $y \in \mathsf{Var}(D)$. Si $y \notin \mathsf{Var}(\mathsf{Act}(D))$, alors $y \notin \mathsf{Var}(\mathsf{Act}(D'))$ et donc $\mathsf{sActlev}(y, D') = \mathsf{sActlev}(y, D) = 0$. Supposons que $y \in \mathsf{Var}(\mathsf{Act}(D))$ et soit $H \overset{?}{\vdash} v$ la contrainte dans $\mathsf{Act}(D)$ qui introduit y. On a $y \in \mathsf{Var}(v)$ et $\mathsf{sActlev}(y, D) = |H|$. On distingue alors deux cas, suivant si la transformation $D \mapsto_{\mathsf{A/F}} D'$ est appliquée à la contrainte $H \overset{?}{\vdash} v$ ou non.

- *La règle de transformation n'est pas appliquée à $H \overset{?}{\vdash} v$.* Dans ce cas, il se peut que $y \notin \mathsf{Var}(\mathsf{Act}(D'))$. C'est le cas quand une règle Active est appliquée et $y \in \mathsf{dom}(\theta)$, où θ est la substitution impliquée dans l'application de la règle. Dans ce cas, le resultat est trivial. Sinon, on a $\mathsf{sActlev}(y, D') \leq \mathsf{sActlev}(y, D)$.

- *La règle de transformation est appliquée à $H \overset{?}{\vdash} v$.* Notons que, dans ce cas, la règle ne peut pas, par définition, être de type Freeze. Encore une fois donc, ou bien on a $y \notin \mathsf{Var}(\mathsf{Act}(D'))$ ou bien $\mathsf{sActlev}(y, D') \leq \mathsf{sActlev}(y, D)$. □

Nous sommes prêts maintenant à prouver notre deuxième lemme de terminaison. Dans la suite, nous notons $\mathsf{MLH}(D)$ l'ensemble maximal dans $\mathsf{LH}(D)$.

Lemme 56 *Soit D un sytème de contraintes étendu tel que $\mathsf{E}(D) = \mathsf{Fr}(D) = \emptyset$. Toute séquence de transformation issue de D utilise au plus $2|\mathsf{LH}(D)|+|\mathsf{MLH}(D)|$ la règle* Open.

Preuve: Considérons une séquence de transformation comme suit:

$$D_1 \mapsto^*_{\mathsf{A/F}} S_1 \mapsto_{\mathsf{O}} D_2 \mapsto^*_{\mathsf{A/F}} S_2 \mapsto_{\mathsf{O}} \cdots$$

Notons que les systèmes de contraintes D_i ont une partie gelée vide, tandis que les S_i ont une partie active résolue.

Dans le reste de la preuve, nous montrons que:

- $\mathsf{unsolvedActlev}(D_{i+1}) < \mathsf{unsolvedActlev}(D_i)$, ou
- $|\mathsf{MLH}(D_{i+1})| < |\mathsf{MLH}(D_i)|$, ou
- $|\mathsf{LH}(D_{i+1})| < |\mathsf{LH}(D_i)|$.

Puisque ces trois mesures ne peuvent pas croître, ceci nous permettra de conclure.

Nous supposons que $|\mathsf{MLH}(D_{i+1})| = |\mathsf{MLH}(D_i)|$ et $|\mathsf{LH}(D_{i+1})| = |\mathsf{LH}(D_i)|$, et nous allons montrer que $\mathsf{unsolvedActlev}(D_{i+1}) < \mathsf{unsolvedActlev}(D_i)$. Soit $\ell = \mathsf{unsolvedActlev}(D_i)$ et $k_\ell = \mathsf{sunsolvedActlev}(D_i)$. Nous allons prouver, plus tard, par récurrence sur la longueur de la séquence $D_i \mapsto^*_{\mathsf{A/F}} D' \mapsto_{\mathsf{A/F}} S$ les affirmations suivantes:

Aff. 1 *Nous avons* $\mathsf{sunsolvedActlev}(S) \leq k_\ell$.

i.e. le niveau actif non-résolu ne grossit pas.

Aff. 2 *Pour tout* $y \in \mathsf{Var}(\mathsf{Fr}(S))$, *on a* $\mathsf{sActlev}(y, S) < k_\ell$.

i.e. le niveau des variables dans le frigidaire est strictement plus petit que le niveau actif non-résolu.

Aff. 3 *Supposons que* $\mathsf{mgu}(\mathsf{E}(S)) \neq \bot$ *et soit* $\sigma_S = \mathsf{mgu}(\mathsf{E}(S))$. *Pour tout* $x \in \mathsf{Var}(\mathsf{Act}(S))$, *avec* $\mathsf{sActlev}(x, S) \geq k_\ell$, *on a:*

1. $x \notin \mathsf{dom}(\sigma_S)$ *et*
2. $\forall y \in \mathsf{Var}(\mathsf{Act}(S) \cup \mathsf{Fr}(S)) \cap \mathsf{dom}(\sigma_S).\ x \notin \mathsf{Var}(y\sigma_S)$

i.e. les variables de niveau maximal ne seront pas affectées par l'ouverture du frigidaire.

Aff. 4 *Soit* $\mathsf{LH}(\mathsf{Fr}(S))$ *les membres gauches des contraintes dans* $\mathsf{Fr}(S)$. *Pour tout* $H' \in \mathsf{LH}(\mathsf{Fr}(S))$, *nous avons* $|H'| < k_\ell$.

i.e. toutes les contraintes dans le frigidaire ont un ensemble des hypothèses plus petit que le niveau actif non-résolu.

Montrons d'abord que ces affirmations suffisent pour conclure le lemme, quand elles sont vraies pour $S = S_i$. Supposons que $\mathsf{mgu}(\mathsf{E}(S_i)) \neq \bot$ et soit $\sigma_{i+1} = \mathsf{mgu}(\mathsf{E}(S_i))$. Nous avons:

$$D_{i+1} = \exists \tilde{z}.[(\mathsf{Act}(S_i) \cup \mathsf{Fr}(S_i))\sigma_{i+1} \mid \emptyset \mid \sigma_{i+1}].$$

Soit $H \overset{?}{\vdash} u \in (\mathsf{Act}(S_i) \cup \mathsf{Fr}(S_i))\sigma_{i+1}$ une contrainte de $\mathsf{Act}(D_{i+1})$ telle que $\mathsf{lev}(H, \mathsf{Act}(D_{i+1})) \geq \ell$. D'abord, puisque $|\mathsf{MLH}(D_{i+1})| = |\mathsf{MLH}(D_i)|$ et $|\mathsf{LH}(D_{i+1})| = |\mathsf{LH}(D_i)|$, on a $|H| \geq k_\ell$. En conséquence, par l'affirmation 4, on a $H \overset{?}{\vdash} u \notin \mathsf{Fr}(S_i)\sigma_{i+1}$. Ainsi on a $H \overset{?}{\vdash} u \in \mathsf{Act}(S_i)\sigma_{i+1}$. Puisque $\mathsf{Act}(S_i)$ est en forme résolue, il existe une contrainte $H' \overset{?}{\vdash} x \in \mathsf{Act}(S_i)$, avec $(H' \overset{?}{\vdash} x)\sigma_{i+1} = H \overset{?}{\vdash} u$ et $|H'| = |H| \geq k_\ell$ (on a encore une fois utilisé le fait que $\mathsf{MLH}(D_{i+1}) = \mathsf{MLH}(D_i)$). On en déduit donc que $\mathsf{sActlev}(x, S_i) \geq k_\ell$ et, par l'affirmation 3 (point 1), on a $x \notin \mathsf{dom}(\sigma_{i+1})$. On en déduit que $u = x$. Par le point 2 de l'affirmation 3, on a

$$\forall y \in \mathsf{Var}(\mathsf{Act}(S_i) \cup \mathsf{Fr}(S_i)) \cap \mathsf{dom}(\sigma_{i+1}).\ x \notin \mathsf{Var}(y\sigma_{i+1}).$$

De plus, par l'affirmation 2, nous avons $x \notin \mathsf{Var}(\mathsf{Fr}(S_i))$. On obtient donc que cette occurrence de x est sa seule occurrence dans le membre droit des contraintes de déductibilité de D_{i+1}. Cela nous permet de conclure que le niveau actif non-résolu de D_{i+1} est strictement inférieur à ℓ et on peut terminer la preuve du lemme. □

Il nous reste donc a prouver les quatre affirmations énoncées dans la preuve du lemme 56.

Preuves des affirmations. Nous faisons la preuve par récurrence sur la longueur de la séquence de transformation $D_i \mapsto^*_{\mathsf{A/F}} D' \mapsto_{\mathsf{A/F}} S$. Nous allons dénoter par $H \overset{?}{\vdash} v$ la contrainte dans $\mathsf{Act}(D')$ sur laquelle la dernière règle de transformation est appliquée et par θ le mgu qui a été calculé dans cette application. Notons que θ peut être appliquée à la partie active ou simplement mémorisée, dans le cas d'une règle Freeze.

Aff. 1. Nous avons $\mathsf{sunsolvedActlev}(S) \leq k_\ell$.
Preuve: Le cas de base est évident. En effet, dans ce cas on a $S = D_i$ et $\mathsf{sunsolvedActlev}(S) = k_\ell$. Supposons par récurrence que $\mathsf{sunsolvedActlev}(D') \leq k_\ell$. Pour toute contrainte $H_u \overset{?}{\vdash} u \in \mathsf{Act}(D')$ avec $|H_u|$ strictement plus grand que k_ℓ, on a $\mathsf{lev}(H_u \overset{?}{\vdash} u, \mathsf{Act}(D')) > \mathsf{unsolvedActlev}(D')$. On en déduit donc que u est une variable, $H \subsetneq H_u$ et u ne peut pas apparaître dans le domaine ou dans l'image de θ.

On en déduit que $H_u\theta \overset{?}{\vdash} u$ (dans le cas d'une règle Active) ou $H_u \overset{?}{\vdash} u$ (dans le cas d'une règle Freeze) est la seule contrainte dans $\mathsf{Act}(S)$ qui contient la variable u parmi les contraintes de $\mathsf{Act}(S)$. Ainsi on a $\mathsf{sunsolvedActlev}(S) \leq k_\ell$. □

Aff. 2. Pour tout $y \in \mathsf{Var}(\mathsf{Fr}(S))$, on a $\mathsf{sActlev}(y, S) < k_\ell$.
Preuve: Le cas de base est immédiat. En effet, on a alors $S = D_i$ et $\mathsf{Fr}(S) = \emptyset$. Sinon, soit un $y \in \mathsf{Var}(\mathsf{Fr}(S))$. On distingue deux cas:

- Si $y \in \mathsf{Var}(\mathsf{Fr}(D'))$, on a par hypothèse de récurrence que $\mathsf{sActlev}(y, D') < k_\ell$. Par le lemme 55, on conclut $\mathsf{sActlev}(y, S) \leq \mathsf{sActlev}(y, D') < k_\ell$.

- Sinon, une règle Freeze a été appliquée à $H \overset{?}{\vdash} v$ et la variable y a été introduite dans la partie gelée du système. Si y est une variable fraîche, alors $y \notin \mathsf{Act}(S)$ et $\mathsf{sActlev}(y, S) = 0$: on conclut. Sinon, on a $y \in \mathsf{Var}(H)$. Par l'affirmation 1 appliquée à D', on a $\mathsf{sunsolvedActlev}(D') \leq k_\ell$. Puisque une règle est appliquée à $H \overset{?}{\vdash} v$, on a $|H| \leq \mathsf{sunsolvedActlev}(D')$. Puisque $y \in \mathsf{Var}(H)$, par origination, on a $\mathsf{sActlev}(y, D') < |H|$ et donc $\mathsf{sActlev}(y, D') < k_\ell$. Par conséquent, par le lemme 55, on conclut que $\mathsf{sActlev}(y, S) < k_\ell$. □

Aff. 3. Supposons que $\mathsf{mgu}(\mathsf{E}(S)) \neq \bot$ et soit $\sigma_S = \mathsf{mgu}(\mathsf{E}(S))$. Pour tout $x \in \mathsf{Var}(\mathsf{Act}(S))$, avec $\mathsf{sActlev}(x, S) \geq k_\ell$, on a:

1. $x \notin \mathsf{dom}(\sigma_S)$ et
2. $\forall y \in \mathsf{Var}(\mathsf{Act}(S) \cup \mathsf{Fr}(S)) \cap \mathsf{dom}(\sigma_S).\ x \notin \mathsf{Var}(y\sigma_S)$

Preuve: Pour faciliter la preuve, nous séparons l'ensemble d'équations $\mathsf{E}(S)$ en deux parties:

- celles qui viennent de l'application d'une règle Active ou de $\mathsf{E}(D_i)$, et
- celles qui viennent de l'application d'une règle Freeze.

Soit $\mathsf{E}(S) = \mathsf{E}_\mathsf{A}(S) \cup \mathsf{E}_\mathsf{F}(S)$, avec $\mathsf{E}(D_i) \subseteq \mathsf{E}_\mathsf{A}(S)$. On dénote par σ_A la substitution $\mathsf{mgu}(\mathsf{E}_\mathsf{A}(S))$ et par σ_i la substitution $\mathsf{mgu}(\mathsf{E}(D_i))$. Soit

- $L_1 = \{x \in \mathsf{Var}(\mathsf{Act}(S)) \mid \mathsf{sActlev}(x, S) \geq k_\ell\}$, et
- $L_2 = \mathsf{dom}(\sigma_i) \cup \{y \mid \exists S', S''.\ D_i \mapsto^*_{\mathsf{A/F}} S' \mapsto_\mathsf{A} S'' \mapsto^*_{\mathsf{A/F}} S.$
  $\mathsf{sActlev}(y, S')) \geq k_\ell$ et $\mathsf{sActlev}(y, S'') = 0\}$.

Intuitivement, les variables de L_1 sont celles qui sont introduites par une contrainte dont la taille est plus grande que k_ℓ, tandis que celles de L_2 sont celles qui ont déjà disparu par des règles de transformation et, si elles etaient présentes dans D_i, leur niveau etait maximal.

Montrons que $\mathsf{E}(S) = \mathsf{E}_\mathsf{A}(S) \cup \mathsf{E}_\mathsf{F}(S)$ et $L = L_1 \uplus L_2$ satisfont les conditions requises pour appliquer le lemme 54.

1. $\mathsf{E}_\mathsf{A}(S)$ est un ensemble d'équations et $\mathsf{mgu}(\mathsf{E}_\mathsf{A}(S)) \neq \bot$ puisque $\mathsf{mgu}(\mathsf{E}(S)) \neq \bot$.
2. $L \cap \mathsf{Var}(\mathsf{E}_\mathsf{F}(S)) = \emptyset$. Par contradiction, supposons qu'il existe une variable $y \in L \cap \mathsf{Var}(\mathsf{E}_\mathsf{F}(S))$.

   - Si $y \in \mathsf{Var}(\mathsf{E}_\mathsf{F}(S))$, ça veut dire qu'une règle Freeze a été appliquée à une contrainte $H' \overset{?}{\vdash} v'$ dans $\mathsf{Act}(S')$ sur un système S' tel que $D_i \mapsto^*_{\mathsf{A/F}} S' \mapsto^*_{\mathsf{A/F}} S$, avec $y \in \mathsf{Var}(H', v')$. Dans ce cas, par définition d'une règle Freeze, on a $0 \neq \mathsf{lev}(y, \mathsf{Act}(S')) < \mathsf{lev}(H', \mathsf{Act}(S')) \leq \mathsf{unsolvedActlev}(S')$, et donc on obtient $0 \neq \mathsf{sActlev}(y, S') < \mathsf{sunsolvedActlev}(S') \leq k_\ell$ en nous appuyant sur l'affirmation 1 pour la dernière inégalité.
   - Si $y \in L$, on a $\mathsf{sActlev}(y, S'') \geq k_\ell$ ou $\mathsf{sActlev}(y, S'') = 0$ pour chaque S'' le long de la dérivation $D_i \mapsto^*_{\mathsf{A/F}} S$. Ceci suit par la définition de L_1, L_2 et par le lemme 55.

   Par conséquent, on a, d'une part, $0 \neq \mathsf{sActlev}(y, S') < k_\ell$ et, d'autre part, $\mathsf{sActlev}(y, S') \geq k_\ell$ ou $\mathsf{sActlev}(y, S') = 0$. Par contradiction, on conclut $L \cap \mathsf{Var}(\mathsf{E}_\mathsf{F}(S)) = \emptyset$.

3. Soit $x \in L$ et considérons l'ensemble Y_x definit comme suit:

$$Y_x = \{y \in \mathsf{dom}(\sigma_\mathsf{A}) \mid y \notin L_2 \text{ et } x \in \mathsf{Var}(y\sigma_\mathsf{A})\}$$

Supposons par contradiction que Y_x n'est pas vide. Soit $y \in Y_x$ tel que $y\sigma_\mathsf{A}$ est minimal dans $\{z\sigma \mid z \in Y_x\}$ par rapport à l'ordre sous-terme. Puisque $y \in \mathsf{dom}(\sigma_\mathsf{A})$ et $y \notin L_2$, on a:

- il existe deux systèmes de contraintes S', S'', avec $D_i \mapsto^*_{\mathsf{A/F}} S' \mapsto_\mathsf{A} S'' \mapsto^*_{\mathsf{A/F}} S$ tels que $y \in \mathsf{Var}(\mathsf{Act}(S'))$ et $y \notin \mathsf{Var}(\mathsf{Act}(S''))$; et
- $0 \neq \mathsf{sActlev}(y, S') < k_\ell$. considérons la contrainte $H' \overset{?}{\vdash} v' \in \mathsf{Act}(S')$ temoignant de ce fait, i.e. $y \in \mathsf{Var}(v')$, $\mathsf{lev}(H', \mathsf{Act}(S')) = \mathsf{lev}(y, \mathsf{Act}(S'))$ et $|H'| < k_\ell$.

Soit θ le mgu calculé dans la transition $S' \mapsto_\mathsf{A} S''$ et $H \overset{?}{\vdash} v$ la contrainte sur laquelle la règle est appliquée.

Notons que $x \in \mathsf{Var}(y\theta)$. En effet, si ce n'est pas le cas, il doit exister une variable $y' \in \mathsf{Var}(y\theta)$ telle que $y' \in \mathsf{Var}(\mathsf{Act}(S''))$ et $x \in \mathsf{Var}(y'\sigma_\mathsf{A})$. Pour contredire la minimalité de y, on doit montrer que $y' \in Y_x$. Pour ce faire, on va d'abord montrer que $\mathsf{sActlev}(y', S'') < k_\ell$. On distingue deux cas:

(a) Si $\mathsf{sActlev}(y', S') < k_\ell$, on conclut facilement par le lemme 55.

(b) Autrement on a $\mathsf{sActlev}(y', S') \geq k_\ell$. Puisque $y' \in \mathsf{img}(\theta)$, par l'affirmation 1, on en déduit que $\mathsf{sActlev}(y', S') = k_\ell$. On a donc $|H| = k_\ell$. Puisque $|H'| < k_\ell$, on en déduit que $H' \subsetneq H$. Par conséquent, on a $H'\theta \overset{?}{\vdash} v'\theta \in \mathsf{Act}(S')$ et donc $\mathsf{sActlev}(y', S'') < k_\ell$.

En conséquence, dans les deux cas, on a $\mathsf{sActlev}(y', S'') < k_\ell$ et, grâce au lemme 55, on sait que le niveau actif de la variable y' ne croît pas le long de la dérivation. Ainsi, par définition de l'ensemble L_2, on obtient $y' \notin L_2$ (en observant aussi que $y' \notin \mathsf{dom}(\sigma_i)$). Par suite, on contredit la minimalité de y et on conclut que $x \in \mathsf{Var}(y\theta)$.

Maintenant, on distingue deux cas:

(a) ou bien $H \overset{?}{\vdash} v = H' \overset{?}{\vdash} v'$. Autrement dit, la contrainte sur laquelle la règle de transformation est appliquée pour passer de S' à S'' est la contrainte qui temoigne du fait que $\mathsf{sActlev}(y, S') < k_\ell$. Puisque $x \in \mathsf{Var}(y\theta)$, on sait que x apparaît dans $H \overset{?}{\vdash} v$, et donc, puiqu'une règle Active est appliqueée, $0 \neq \mathsf{sActlev}(x, S') \leq |H| = |H'| < k_\ell$.

(b) Autrement, on a $H \overset{?}{\vdash} v \neq H' \overset{?}{\vdash} v'$. Dans ce cas, $H'\theta \overset{?}{\vdash} v'\theta \in \mathsf{Act}(S'')$ et $|H'\theta| < k_\ell$. Puisque $x \in \mathsf{Var}(y\theta)$ et $y \in \mathsf{Var}(v')$, on en déduit facilement que $0 \neq \mathsf{sActlev}(x, S'') < k_\ell$.

Par le lemme 55, dans le deux cas on en déduit que $\mathsf{sActlev}(x, S) < k_\ell$ et on obtient que $x \notin L_1$. Puisque $x \in \mathsf{Var}(y\theta)$ et donc x apparaît dans les contraintes de S', il est aussi clair que $x \notin \mathsf{dom}(\sigma_i)$. Par le lemme lemme 55 et $0 < \mathsf{sActlev}(x, S') < k_\ell \vee 0 < \mathsf{sActlev}(x, S'') < k_\ell$, on en déduit que $x \notin L_2$. Ainsi $x \notin L$ et on a une contradiction.

4. $L_1 \cap \mathsf{dom}(\sigma_\mathsf{A}) = \emptyset$. En effet, on a $L_1 \subseteq \mathsf{Var}(\mathsf{Act}(S))$ et en même temps $\mathsf{dom}(\sigma_\mathsf{A}) \cap \mathsf{Var}(\mathsf{Act}(S)) = \emptyset$ (les équations "actives" ont été appliquées à la partie active).

En conclusion, on peut appliquer le lemme 54. Puisque $\sigma_S = \mathsf{mgu}(\mathsf{E}(S))$, le lemme 54 nous dit que pour tout $x \in \mathsf{Var}(\mathsf{Act}(S))$ avec $\mathsf{sActlev}(x, S) \geq k_\ell$,

1. $x \notin \mathsf{dom}(\sigma_S)$

2. si $x \in \mathsf{Var}(y\sigma_S)$, avec $y \in \mathsf{dom}(\sigma_S)$, alors $y \in L_2$. Pour conclure l'affirmation, il suffit maintenant de montrer que $L_2 \cap \mathsf{Var}(\mathsf{Act}(S) \cup \mathsf{Fr}(S)) = \emptyset$.

   - Le fait que $L_2 \cap \mathsf{Var}(\mathsf{Act}(S)) = \emptyset$ suit immediatement de la définition de L_2.
   - Soit $y \in L_2$. Si $y \in \mathsf{dom}(\sigma_i)$, on conclut $y \notin \mathsf{Var}(\mathsf{Fr}(S))$ facilement. Sinon, soit S', S'' les systèmes de contraintes temoignant du fait que $y \in L_2$. On a $\mathsf{sActlev}(y, S') \geq k_\ell$ et, par l'affirmation 2, on en déduit que $y \notin \mathsf{Var}(\mathsf{Fr}(S'))$. Puisque $S' \mapsto_\mathsf{A} S''$ (par l'application d'une règle Active), on déduit que $y \notin \mathsf{Var}(\mathsf{Fr}(S''))$. Puisque $\mathsf{sActlev}(y, S'') = 0$ (i.e. y n'apparaît plus dans la partie active), il n'y a plus de moyen de mettre y dans la partie gelée. Par conséquent, pour toute séquence $S'' \mapsto^* S'''$, on a $y \notin \mathsf{Var}(\mathsf{Fr}(S'''))$. On en déduit donc que $y \notin \mathsf{Var}(\mathsf{Fr}(S))$.

   On conclut que $\forall y \in \mathsf{Var}(\mathsf{Act}(S) \cup \mathsf{Fr}(S)) \cap \mathsf{dom}(\sigma_S).\ x \notin \mathsf{Var}(y\sigma_S)$. □

Aff. 4. Pour tout $H' \in \mathsf{LH}(\mathsf{Fr}(S))$, on a $|H'| < k_\ell$.
Preuve: Encore une fois, le cas de base est évident, car $S = D_i$ et donc $\mathsf{Fr}(S) = \emptyset$. Sinon, soit $H' \in \mathsf{LH}(\mathsf{Fr}(S))$. Si $H' \in \mathsf{LH}(\mathsf{Fr}(D'))$, on conclut directement en appliquant l'hypothèse de récurrence que $|H'| < k_\ell$. Sinon, $H' \in \mathsf{LH}(\mathsf{Fr}(S)) \setminus \mathsf{LH}(\mathsf{Fr}(D'))$, une règle Freeze a été appliquée à $H \overset{?}{\vdash} v$. Par définition de Freeze, on a $H' \subsetneq H$ et donc $|H'| < |H|$. Puisque $|H| \leq \mathsf{sunsolvedActlev}(D')$ et, par l'affirmation 1, $\mathsf{sunsolvedActlev}(D') \leq k_\ell$, on conclut $|H'| < k_\ell$. □

Des lemmes 53 et 56, on déduit le corollaire suivant:

Corollaire 5 (Terminaison) *Notre stratégie est fortement terminante: il n'existe pas de séquence de transformation infinie de systèmes (étendus) de contraintes de déductibilité.*

11.3.3 Complétude de la stratégie.

Nous allons montrer ici que, si un système de contraintes a une solution, on peut toujours atteindre une forme résolue suivant notre stratégie. En particulier, nous allons montrer que pour toute solution σ (avec un ensemble de preuves pour les contraintes de déductibilité), il existe une règle de transformation qui s'applique et qui diminiue la mesure PS (cf page 127) de la partie active. Cela permet d'assurer que couper les boucles sur la partie active n'enlève pas des branches nécéssaires pour capturer toutes les solutions.

Pour un sytème de contraintes étendu $D = \exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$, nous posons $\mathsf{Open}(D) = \exists\tilde{z}.[(\mathsf{A}\cup\mathsf{F})\mathsf{mgu}(\mathsf{E}) \mid \mathsf{mgu}(\mathsf{E})]$ et nous allons dire que D est uniquement déterminé si le système $\mathsf{Open}(D)$ est uniquement déterminé. Notons que, par définitions, nous avons $Sol(D) = Sol(\mathsf{Open}(D))$ et que la propriété d'unique détermination est préservée par $\mapsto$. En effet, si $D \mapsto D'$ et D est uniquement déterminé,

- ou bien $\mathsf{Open}(D) \rightsquigarrow^* \mathsf{Open}(D')$, et alors D' est uniquement déterminé par le lemme 47.
- ou bien $D \mapsto_{\mathsf{Triv}} D'$, et alors $\mathsf{E}(D) = \mathsf{E}(D')$: l'unique détermination est préservée car elle est une propriété de l'ensemble d'équations d'un système.

Le corollaire suivant nous garanti que la mesure PS décroît sur la partie active.

Corollaire 6 (du lemme 49) *Soit D un système (étendu) de contraintes uniquement déterminé tel que $\mathsf{Act}(D)$ n'est pas en forme résolue et $\sigma \in Sol(D)$. Il existe un système D' tel que $D \mapsto_{\mathsf{A/F}} D'$ et $\sigma \in Sol(D')$.*

De plus, $\mathsf{PS}(\mathsf{A}, \overline{\sigma}|_{\mathsf{Var(A)}}) > \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{\mathsf{Var(A')}})$, où $\mathsf{A} = \mathsf{Act}(D)$, $\mathsf{A}' = \mathsf{Act}(D')$ et $\overline{\sigma}$ est l'extension de σ par rapport à D'.

Preuve: Soit $D = \exists\tilde{z}.[\mathsf{A} \mid \mathsf{F} \mid \mathsf{E}]$ un système de contraintes étendu uniquement déterminé et $\sigma \in Sol(D)$. Soit σ' l'extension de σ par rapport à D. Notons que $\sigma'|_{\mathsf{Var(A)}}$ est une solution de A et A n'est pas en forme résolue. Par le lemme 49, on obtient $[\mathsf{A} \mid \emptyset] \rightsquigarrow \exists\tilde{z'}.[\mathsf{A}' \mid \theta]$, de telle manière que $\sigma'|_{\mathsf{Var(A)}} \in Sol(\exists\tilde{z'}.[\mathsf{A}' \mid \theta])$ et $\mathsf{PS}(\mathsf{A}, \sigma'|_{\mathsf{Var(A)}}) > \mathsf{PS}(\exists\tilde{z'}.[\mathsf{A}' \mid \theta], \sigma'|_{\mathsf{Var(A)}})$. Par définition des règles de transformation des systèmes étendus, on déduit que:

- ou bien on peut appliquer une règle Active, pour obtenir $D \mapsto_{\mathsf{A}} D' = \exists\tilde{z} \cup \tilde{z'}.[\mathsf{A}' \mid \mathsf{F} \mid \mathsf{E} \cup \theta]$. On a $\sigma \in Sol(D')$, D' est uniquement déterminé et $\overline{\sigma} = \sigma' \cup \sigma_0$, où $\mathsf{dom}(\sigma_0) = \tilde{z'}$.

  De plus, on a $\mathsf{PS}(\exists\tilde{z'}.[\mathsf{A}' \mid \theta], \sigma'|_{\mathsf{Var(A)}}) = \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{\mathsf{Var(A')}})$. En conséquence, on a

  $$\mathsf{PS}(\mathsf{A}, \sigma'|_{\mathsf{Var(A)}}) > \mathsf{PS}(\exists\tilde{z'}.[\mathsf{A}' \mid \theta], \sigma'|_{\mathsf{Var(A)}}) = \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{\mathsf{Var(A')}}).$$

  Puisque $\overline{\sigma}|_{\mathsf{Var(A)}} = \sigma'|_{\mathsf{Var(A)}}$, ceci nous permet de conclure que $\mathsf{PS}(\mathsf{A}, \overline{\sigma}|_{\mathsf{Var(A)}}) > \mathsf{PS}(\mathsf{A}', \overline{\sigma}|_{\mathsf{Var(A')}})$, où $\mathsf{A} = \mathsf{Act}(D)$ et $\mathsf{A}' = \mathsf{Act}(D')$.

- ou bien on peut appliquer une règle Freeze, pour obtenir $D \mapsto_{\mathsf{F}} D' = \exists \tilde{z} \cup \tilde{z'}.[\mathsf{A}_a \mid \mathsf{F} \cup \mathsf{A}_f \mid \mathsf{E} \cup \theta]$, où $\mathsf{A}' = (\mathsf{A}_a \cup \mathsf{A}_f)\theta$. Encore une fois, il est facile de voir que $\sigma \in Sol(D')$. De plus, on a $\mathsf{Act}(D') = \mathsf{A}_a$ et $\mathsf{A}_a\overline{\sigma} \subseteq \mathsf{A}'\overline{\sigma}$. On obtient en conclusion que

$$\mathsf{PS}(\mathsf{A}, \overline{\sigma}|_{\mathsf{Var(A)}}) > \mathsf{PS}(\exists \tilde{z'}.[\mathsf{A}' \mid \theta], \overline{\sigma}|_{\mathsf{Var(A)}}) \geq \mathsf{PS}(\mathsf{A}_a, \overline{\sigma}|_{\mathsf{Var}(\mathsf{A}_a)})$$

□

En conséquence, s'il y a une boucle sur la partie active, on peut enlever la branche, en préservant la complétude:

Lemme 57 *Notre stratégie est complète.*

Preuve: Soit D_0 un système de contraintes étendu et $\sigma \in Sol(D_0)$. On doit montrer qu'il existe un système D' en forme résolue tel que $\sigma \in Sol(D')$ et $D_0 \mapsto^* D'$ par une séquence autorisée par la stratégie. Grâce au corollaire 6, on sait qu'il existe une séquence

$$D_0 \mapsto^*_{\mathsf{A/F}} S_1 \mapsto_{\mathsf{O}} D_1 \mapsto^*_{\mathsf{A/F}} S_2 \mapsto_{\mathsf{O}} D_2 \mapsto^*_{\mathsf{A/F}} \cdots$$

telle que σ est une solution de chaque système dans la séquence et la mesure PS sur la partie active decroît le long de chaque séquence $D_i \mapsto^*_{\mathsf{A/F}} S_{i+1}$.

De plus, grâce au corollaire 5, on sait que cette séquence est finie. Soit D' son dernier système. Pour conclure, il nous reste à montrer que D' est en forme résolue.

Supposons par contradiction que D' n'est pas en forme résolue. On distingue deux cas:

- Si $\mathsf{Act}(D')$ est en forme résolue, Open peut être appliquée amenant à une contradiction.
- Si $\mathsf{Act}(D')$ n'est pas en forme résolue, par le corollaire 6, on en déduit qu'il existe un système D'' tel que $D' \mapsto_{\mathsf{A/F}} D''$, $\sigma \in Sol(D'')$ et $\mathsf{PS}(\mathsf{Act}(D'), \overline{\sigma}|_{\mathsf{Var(Act}(D'))}) > \mathsf{PS}(\mathsf{Act}(D''), \overline{\sigma}|_{\mathsf{Var(Act}(D''))})$, où $\overline{\sigma}$ est l'extension de σ par rapport à D''.

  Puisque la transition $D' \mapsto_{\mathsf{A/F}} D''$ est interdite par la stratégie, on en déduit qu'il existe une sous-séquence $D_j \mapsto^*_{\mathsf{A/F}} D'$ telle que $\mathsf{Act}(D_j) = \mathsf{Act}(D'')$ (une boucle est enlevée). Ceci implique que la mesure PS ne decroît pas dans cette séquence: contradiction. □

11.4 Résultat principal et discussion

Nous avons obtenu:

Théorème 7 ([BDC09]) *Soit $\mathcal{I}$ un système d'inférence saturé et fini, qui satisfait en plus les restrictions syntaxiques de la page 116. Étant donné un système de contraintes D, on peut calculer un ensemble fini de formes résolue $D_1, \ldots, D_n$ tel que $Sol(D) = Sol(D_1) \cup \ldots \cup Sol(D_n)$.*

Preuve: On étend D avec un frigidaire et on calcule un ensemble fini de formes résolues étendues, suivant notre stratégie. En oubliant les frigidaires, on obtient par définition les formes résolues voulues $D_1, \ldots, D_n$. □

Des théories pour lesquelles ce théoreme s'applique sont $\mathcal{I}_{\mathtt{DY}}$ (Dolev-Yao) et $\mathcal{I}_{\mathtt{sc}}$ (systèmes sous-terme convergents). Rappelons cependant que ces systèmes ne contiennent pas de règles versatiles. Notre résultat est ainsi une généralisation stricte de [CLCZ10], [CK07] et, si on se restreint aux propriétés de traces, de [Bau07]. L'idée d'utiliser la localité pour contrôler un processus systématique de recherche de preuve est déjà utilisée dans e.g. [CLCZ10]. Ici nous montrons que la localité est la *seule* propriété essentielle pour effectuer cela, quand le système d'inférence est fini. De même, notre résultat explique [CK07], qui effectue déjà la saturation d'un système sous-terme convergent particulier pour dériver la complétude d'un ensemble de règles de simplification. Une autre différence avec [Bau07] et [CK07] est méthodologique: la surréduction de la théorie de l'intrus fait partie du processus de la recherche de preuves dans [Bau07], elle fait partie de la saturation dans [CK07] et s'applique avant la saturation dans notre procédure.

Si le système d'inférence saturé est infini, comme c'est le cas pour $\mathcal{I}_{\mathtt{blind}}$ ou $\mathcal{I}_{\mathtt{hom}}$, nos règles de transformation restent complètes (lemme 50), mais ne sont plus effectives. Pour en déduire quand même une procédure de décision, nous allons dans le chapitre suivant grouper un nombre infini des règles de transformation dans une seule règle, qui va les représenter toutes symboliquement.

Chapitre 12

Signatures en aveugle

Dans ce chapitre nous nous intéressons aux systèmes d'inférence saturés et infinis. En fait, nous allons considérer seulement le système $\mathcal{I}_{\texttt{blind}}$. Notre but ici est de montrer comment un système saturé infini peut être traité à partir des résultats précédents.

Pour couper le branchement infini, nous allons introduire un prédicat dont la sémantique va englober un nombre infini de règles de transformation. Une nouvelle notion de forme résolue et des nouvelles règles de transformation sont alors nécéssaires pour gérer le nouveau prédicat. Pour faciliter les preuves de complétude et de terminaison de ces règles, nous allons d'abord remarquer que l'introduction des variables existentielles est, dans ce cas, contournable et que les égalités entre les sous-termes qui apparaissent dans les contraintes peuvent être dévinées au début.

12.1 Simplifications

12.1.1 Simplification de la règle Dec

Nous remarquons d'abord que, toute en préservant la complétude de notre système générique de transformation, on peut remplacer la règle Dec par la règle

$$(\mathsf{Dec}') \quad \exists \tilde{z}.[C \wedge H \overset{?}{\vdash} v \mid \sigma] \rightsquigarrow \exists \tilde{z} \cup \tilde{x}.[C\theta \wedge H\theta \overset{?}{\vdash} w_1\theta \wedge \ldots H\theta \overset{?}{\vdash} w_n\theta \wedge H\theta \overset{?}{\vdash} v_1\theta \wedge \ldots \wedge H\theta \overset{?}{\vdash} v_m\theta \mid \sigma \cup \theta]$$

où:

- $\mathsf{R} = \dfrac{v_1 \ \ldots \ v_m \ \ w_1 \ \ldots \ w_n}{w}$ est une règle de décomposition ou versatile telle que $\mathtt{Max}(\mathsf{R}) = \{w_1, \ldots, w_n\}$ et $\tilde{x} = \mathsf{Var}(\mathsf{R})$;
- $\theta = \mathsf{mgu}(\langle w, w_1, \ldots, w_n\rangle, \langle v, u_1, \ldots, u_n\rangle)$, $u_1, \ldots, u_n \in st(H) \setminus \mathcal{X}$, et $v \notin \mathcal{X}$;

Il y a deux différences entre Dec et Dec':

- aucune prémisse n'est contrainte à avoir une preuve plus petite dans Dec'. Cette condition était utilisée pour la terminaison de règles de transformation. Cette propriété sera prouvée autrement dans cette section. Pour la complétude, nous observons que la mesure PS d'une contrainte $H\theta \overset{?}{\vdash} w_i\theta$ est au moins aussi grande que celle de la contrainte $H'\theta \overset{?}{\vdash} w_i\theta$, si $H' \subseteq H$. Ainsi, si la mesure PS decroît pour les contraintes avec $H' \subsetneq H$, introduites par Dec, elle decroît aussi pour les contraintes avec H, introduites par la règle Dec'.
- $\mathtt{Max}(\mathsf{R}) = \{w_1, \ldots, w_n\}$ dans la règle Dec', tandis que $\mathtt{Max}(\mathsf{R}) \subseteq \{w_1, \ldots, w_n\}$ dans la règle Dec. Puisque pour aucune contrainte l'ensemble d'hypothèses n'est forcé à être plus petit dans la règle Dec', la nouvelle condition affecte seulement les équations (il y en a moins) du système obtenu en appliquant la règle, et non pas ses contraintes de déductibilité, qui seules determinent la mesure PS.

Ces deux points nous assurent que la complétude est préservée en remplaçant Dec par Dec'.

Lemme 58 *Considérons la relation de transformation $\rightsquigarrow'$ obtenue en remplaçant, pour chaque règle de décomposition ou versatile, ses règles Dec par une seule règle Dec'. Soient D, D' deux systèmes de contraintes tels que $D \rightsquigarrow D'$ et σ une solution de D, D'. Il existe un système de contraintes D'' tel que $D \rightsquigarrow' D''$ et $\mathsf{PS}(D'', \sigma) \leq \mathsf{PS}(D', \sigma)$.*

Preuve: Immédiate par les deux observations faites ci-dessus. □

Dans la suite, on va noter par $\rightsquigarrow$ la relation de transformation où les règles Dec sont remplacées par les règles Dec'.

Nous observons maintenant qu'on peut simplifier plus certains règles de transformation, dans le cas de $\mathcal{I}_{\mathtt{blind}}$. Par exemple, considérons la règle Dec' qui correspond à la règle de décomposition UB:

$$T \overset{?}{\vdash} u \rightsquigarrow \exists x, y.[T\theta \overset{?}{\vdash} \mathtt{blind}(x, y)\theta \wedge T\theta \overset{?}{\vdash} y\theta \mid \theta]$$

où $\theta = \mathsf{mgu}(\langle \mathtt{blind}(x, y), x\rangle, \langle \mathtt{blind}(u', v), u\rangle)$ et $\mathtt{blind}(u', v) \in st(T)$

En simplifiant les équations, on obtient alors la règle:

$$(\mathsf{R}_2)\quad \begin{array}{l} T \overset{?}{\vdash} u \rightsquigarrow [T\theta \overset{?}{\vdash} \mathtt{blind}(u'\theta, v\theta) \wedge T\theta \overset{?}{\vdash} v\theta \mid \theta] \\ \quad \text{où } \theta = \mathsf{mgu}(u', u) \text{ et } \mathtt{blind}(u', v) \in st(T) \end{array}$$

Notons que cette règle n'introduit pas de nouvelles variables.

Une simplification similaire peut être faite pour l'autre règle de décomposition, qui correspond à la règle d'inférence (C):

$$(\mathsf{R}_3)\quad \begin{array}{l} T \overset{?}{\vdash} u \rightsquigarrow [T\theta \overset{?}{\vdash} \mathtt{sign}(u'\theta, v\theta) \mid \theta] \\ \quad \text{où } \mathtt{sign}(u', v) \in st(T) \text{ et } \theta = \mathsf{mgu}(u, u') \end{array}$$

12.1.2 Introduction d'un nouveau prédicat

Le système $\mathcal{I}_{\texttt{blind}}$ est infini à cause des règles versatiles $\mathsf{UB_n}$:

$$\frac{\texttt{sign}(b_n(x, x_1, \ldots, x_n), y) \quad x_1 \quad \ldots \quad x_n}{\texttt{sign}(x, y)}$$

On cherche ici à décrire les règles de transformation qui correspondent à ces règles d'inférence d'une manière finie et effective: pour une contrainte $T \overset{?}{\vdash} \texttt{sign}(u, v)$ on veut un représentation symbolique pour l'ensemble des prémisses possibles quand la dernière règle dans la preuve est une des $\mathsf{UB_n}, n \geq 0$. Notons d'abord que, pour chaque règle $\mathsf{UB_n}$, on a une règle de transformation Dec' correspondante:

$$T \overset{?}{\vdash} \texttt{sign}(v, w) \rightsquigarrow \exists x, x_1, \ldots, x_n, y.[T\theta \overset{?}{\vdash} \texttt{sign}(u_1, u_2)\theta \wedge T\theta \overset{?}{\vdash} x_1\theta \wedge \ldots \wedge T\theta \overset{?}{\vdash} x_n\theta \mid \theta]$$

pour $\theta = \mathsf{mgu}(\langle \texttt{sign}(b_n(x, x_1, \ldots, x_n), y), \texttt{sign}(x, y)\rangle, \langle \texttt{sign}(u_1, u_2), \texttt{sign}(v, w)\rangle)$ et $\texttt{sign}(u_1, u_2) \in st(T)$

Pour obtenir un ensemble fini de règles de transformation, on introduit un prédicat $'' \in \mathcal{B}d''$ et sa version contrainte $'' \overset{?}{\in} \mathcal{B}d''$ dont la sémantique englobe les propriétés requises pour $x_1, \ldots, x_n$. Pour un ensemble de termes T et deux termes u, v, on a $u \in \mathcal{B}d(T, v)$, s'il existe un entier n et des termes $v_1, \ldots, v_n$ tels que $u = b_n(v, v_1, \ldots, v_n)$ et $T \vdash v_1, \ldots, T \vdash v_n$. Une substitution σ satisfait $u \overset{?}{\in} \mathcal{B}d(T, v)$ si $u\sigma \in \mathcal{B}d(T\sigma, v\sigma)$. Avec la simplification des équations, ce prédicat nous permet d'écrire la règle de transformation suivante:

$$(\mathsf{R_4}) \quad T \overset{?}{\vdash} \texttt{sign}(v, w) \rightsquigarrow T\theta \overset{?}{\vdash} \texttt{sign}(u_1, u_2)\theta \wedge u_1\theta \overset{?}{\in} \mathcal{B}d(T\theta, v\theta), \quad \text{où } \texttt{sign}(u_1, u_2) \in st(T) \text{ et } \theta = \mathsf{mgu}(u_2, w)$$

qui représente, et remplace, toutes les règles Dec' correspondant à $\mathsf{UB_n}, n \geq 0$.

12.1.3 Deviner les égalités en avance

Nous observons maintenant qu'aucune des règles $\mathsf{R_2}, \mathsf{R_3}, \mathsf{R_4}$ n'introduit de nouveaux termes dans le système de contraintes: les égalités qui sont dévinées lors de la recherche des preuves peuvent ainsi être dévinées à l'avance.

Definition 37 (solution non-confondante) *Soit T un ensemble de termes. Une substitution σ est* non-confondante *pour T si pour tout $t_1, t_2 \in T$ tels que $t_1 \neq t_2$, on a $t_1\sigma \neq t_2\sigma$.*

Soit $\mathcal{D}$ un ensemble de contraintes. Une substitution σ est une solution non-confondante *de $\mathcal{D}$ si $\sigma \in Sol(\mathcal{D})$ et σ est non-confondante pour $st(\mathcal{D})$. On notera par $Sol_{\mathsf{NC}}(\mathcal{D})$ l'ensemble de solutions non-confondantes de $\mathcal{D}$.*

Lemme 59 *Soit $\mathcal{C}$ un système de contraintes et $\sigma \in Sol(\mathcal{C})$. Il existe une relation d'équivalence $\approx$ sur $st(\mathcal{C})$ et $\sigma' \in Sol_{\mathsf{NC}}(\mathcal{C}\sigma_{\approx})$ telle que $\sigma = \sigma_{\approx}\theta$, où $\sigma_{\approx} = \mathsf{mgu}(\bigwedge_{s \approx t} s = t)$.*

Preuve: Soit σ une solution de $\mathcal{C}$ et $\approx$ la relation d'équivalence sur $st(\mathcal{C})$ définie par $t \approx u$ si et seulement si $t\sigma = u\sigma$. La substitution σ est un unificateur pour les classes d'équivalence modulo $\approx$ et il existe un unificateur le plus général $\sigma_{\approx}$. De plus, par la définition d'un mgu, il existe une substitution θ telle que $\sigma = \sigma_{\approx}\theta$, ce qui implique que θ est une solution de $\mathcal{C}\sigma_{\approx}$.

Pour la deuxième partie du lemme, on utilise l'observation suivante: si ϕ est un mgu de u, v, qui n'introduit pas de nouvelles variables, alors, pour chaque variable $x \in \mathsf{Var}(u, v)$, $st(x\phi) \subseteq st(\{u, v\})\phi$. Alors tout $t \in st(\mathcal{C}\sigma_{\approx})$ est ou bien dans $st(\mathcal{C})\sigma_{\approx}$ ou bien il existe une variable x telle que $t \in st(x\sigma_{\approx})$. Cependant, dans le dernier cas, comme on vient de remarquer, on a aussi $t \in st(\mathcal{C})\sigma_{\approx}$.

Par conséquent, pour tout $t, u \in st(\mathcal{C}\sigma_{\approx})$, il existent $t_0, u_0 \in st(\mathcal{C})$ tels que $t = t_0\sigma_{\approx}$ et $u = u_0\sigma_{\approx}$. Alors $t\theta = u\theta$ implique $t_0\sigma = u_0\sigma$, donc $t_0 \approx u_0$, et en conséquence $t_0\sigma_{\approx} = u_0\sigma_{\approx}$, nous donnant $t = u$. □

Par conséquent, nous allons considérer dans la suite seulement des solutions non-confondantes d'un système. Dans ce cas, les égalités à deviner par les règles de transformation sont purement syntaxiques, sans le besoin d'une instance. Cette observation nous donne l'ensemble des règles $\mathsf{Axiom}, \mathsf{Triv}, \mathsf{R}_2 - \mathsf{R}_4$ de la figure 12.1 pour la transformation des contraintes dans $\mathcal{I}_{\mathtt{blind}}$. Nous expliquerons dans la section 12.3 la présence des règles $\mathsf{R}_5 - \mathsf{R}_7$.

12.2 Systèmes de contraintes, formes résolues et bonne-formation

Les égalités sont devinées et le mgu est appliqué au système. On revient ainsi aux systèmes de contraintes sans équations et sans variables liées. Cependant, puisqu'un nouveau type de contraintes $u \stackrel{?}{\in} \mathcal{B}d(T, v)$ est introduit par les règles de transformation, la notion de système de contraintes doit à nouveau être étendue. Nous allons voir que la définition des formes résolues doit être également étendue, afin de répresenter certains ensembles de solutions. Enfin, pour assurer que les formes résolues ont toujours une solution, un invariant de bonne-formation est nécéssaire.

12.2.1 Systèmes de contraintes

On étend l'origination et la monotonie et on rajoute une troisième condition sur l'occurence des variables. Cette dernière a pour but de faciliter la preuve de complétude de nos règles de transformation. D'autre part, selon nos observations précédentes, nous n'avons pas besoin des variables liées et des équations.

Une *contrainte élémentaire* C est ou bien une *contrainte de déductibilité* de la forme $T \stackrel{?}{\vdash} u$ ou bien *une contrainte d'appartenance* de la forme $v \stackrel{?}{\in} \mathcal{B}d(T, u)$,

$$
\begin{array}{lrcl}
\mathsf{Axiom}: & \mathcal{C} \wedge T \overset{?}{\vdash} u & \rightsquigarrow & \mathcal{C} \\
 & & & \text{si } u \in T \setminus \mathcal{X} \\
\mathsf{Triv}: & \mathcal{C} \wedge T \overset{?}{\vdash} x \wedge T' \overset{?}{\vdash} x & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} x \\
 & & & \text{si } T \subseteq T' \text{ et } x \in \mathcal{X} \\
\mathsf{R}_1: & \mathcal{C} \wedge T \overset{?}{\vdash} \mathsf{f}(t_1, \ldots, t_n) & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} t_1, \ldots, T \overset{?}{\vdash} t_n \\
 & & & \text{pour } \mathsf{f} \in \{\mathtt{blind}, \mathtt{sign}\} \\
\mathsf{R}_2: & \mathcal{C} \wedge T \overset{?}{\vdash} v & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} \mathtt{blind}(v, u) \ \wedge \ T \overset{?}{\vdash} u \\
 & & & \text{si } \mathtt{blind}(v, u) \in st(T) \text{ et } v \notin \mathcal{X} \\
\mathsf{R}_3: & \mathcal{C} \wedge T \overset{?}{\vdash} v & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} \mathtt{sign}(v, u) \\
 & & & \text{si } \mathtt{sign}(v, u) \in st(T) \text{ et } v \notin \mathcal{X} \\
\mathsf{R}_4: & \mathcal{C} \wedge T \overset{?}{\vdash} \mathtt{sign}(v, u) & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} \mathtt{sign}(w, u) \ \wedge w \overset{?}{\in} \mathcal{B}d(T, v) \\
 & & & \text{si } \mathtt{sign}(w, u) \in st(T) \\
\mathsf{R}_5: & \mathcal{C} \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T', v) & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} x \wedge T \overset{?}{\vdash} v \wedge x \overset{?}{\in} \mathcal{B}d(T, v) \\
 & & & \text{si } T \subsetneq T' \\
\mathsf{R}_6: & \mathcal{C} \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T', v) & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} x \wedge T \overset{?}{\vdash} w \wedge x \overset{?}{\in} \mathcal{B}d(T, w) \wedge w \overset{?}{\in} \mathcal{B}d(T', v) \\
 & & & \text{si } T \subsetneq T' \text{ et } w \in st(T) \\
\mathsf{R}_7: & \mathcal{C} \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T, v) \wedge x \overset{?}{\in} \mathcal{B}d(T, v') & \rightsquigarrow & \mathcal{C} \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T, v) \wedge v \overset{?}{\in} \mathcal{B}d(T, v') \\
 & & & \text{si } T_x = T
\end{array}
$$

Figure 12.1: Règles de transformation pour les signatures en aveugle

où T est un ensemble fini de termes et u, v sont des termes. L'ensemble T est l'ensemble de termes associé à C.

Étant donné un ensemble $\mathcal{D}$ de contraintes élémentaires et $x \in \mathsf{Var}(\mathcal{D})$, on notera par T_x l'ensemble minimal (par rapport à l'inclusion) de termes tel que:

- $T_x \overset{?}{\vdash} u \in \mathcal{D}$, avec $x \in \mathsf{Var}(u)$, ou
- $v \overset{?}{\in} \mathcal{B}d(T_x, u) \in \mathcal{D}$, avec $x \in \mathsf{Var}(u)$.

Definition 38 (Système de contraintes) *Un* système de contraintes $\mathcal{C}$ *est ou bien* $\bot$ *ou bien un ensemble de contraintes élémentaires. On demande que les contraintes dans* $\mathcal{C}$ *puissent s'ordonner en* $C_1, \ldots, C_\ell$ *de façon à ce que les conditions suivantes soient satisfaites:*

1. Monotonie*:* $\emptyset \neq T_1 \subseteq T_2 \ldots \subseteq T_\ell$*;*
2. Origination*: pour chaque* $1 \leq i \leq \ell$*, on a* $\mathsf{Var}(T_i \cup \{v_i\}) \subseteq \mathsf{Var}(\{u_1, \ldots, u_{i-1}\})$*;*

où C_i *est de la forme* $T_i \overset{?}{\vdash} u_i$ *ou* $v_i \in \mathcal{B}d(T_i, u_i)$*, pour chaque* $1 \leq i \leq \ell$*. Additionnellement, on suppose que pour chaque variable* $x \in \mathsf{Var}(\mathcal{C})$*, on a:*

- *il existe* $T_x \overset{?}{\vdash} u \in \mathcal{C}$ *avec* $x \in \mathsf{Var}(u)$*, ou*
- *il existe* $v \in \mathcal{B}d(T_x, u) \in \mathcal{C}$ *avec* $x \in \mathsf{Var}(u)$ *et* $T_y \subsetneq T_x$*, pour tout* $y \in \mathsf{Var}(v)$*.*

Exemple 60 *La séquence de contraintes suivante satisfait la monotonie et l'origination. Pourtant,* $\mathcal{C}$ *n'est pas un système de contraintes puisque la contrainte d'appartenance* $C_2 = \mathtt{blind}(x, a) \overset{?}{\in} \mathcal{B}d(\{a\}, y)$ *est la seule qui introduit la variable* y *et on n'a pas* $T_x \subsetneq \{a\}$*, car* $T_x = \{a\}$*.*

$$\mathcal{C} = \left\{ \begin{array}{rcl} a & \overset{?}{\vdash} & x \\ \mathtt{blind}(x, a) & \overset{?}{\in} & \mathcal{B}d(\{a\}, y) \end{array} \right.$$

Lemme 60 *Si* $\mathcal{C} \rightsquigarrow \mathcal{C}'$ *en utilisant une règle* Dec' *qui correspond à une règle d'inférence* $\mathsf{UB_n}$ *et* σ *est une solution de* $\mathcal{C}$ *et de* $\mathcal{C}'$*, alors il existe un* $\mathcal{C}''$ *tel que* $\mathcal{C} \rightsquigarrow \mathcal{C}''$ *en utilisant la règle* $\mathsf{R_4}$ *et tel que* $\sigma \in Sol(\mathcal{C}'')$*. De plus, l'ensemble de preuves qui temoignent que* σ *est une solution de* $\mathcal{C}''$ *est le même que celui pour temoigner que* $\sigma \in Sol(\mathcal{C}')$*.*

Preuve: Immédiate par la sémantique de $\overset{?}{\in} \mathcal{B}d$. □

$$
\begin{array}{lrcll}
\mathsf{S}_1: & \mathcal{C} \wedge u \stackrel{?}{\in} \mathcal{B}d(T,u) & \rightarrow & \mathcal{C} & \\
\mathsf{S}_2: & \mathcal{C} \wedge \mathtt{blind}(u,v) \stackrel{?}{\in} \mathcal{B}d(T,w) & \rightarrow & \mathcal{C} \wedge T \stackrel{?}{\vdash} v \wedge u \stackrel{?}{\in} \mathcal{B}d(T,w) & \text{si } \mathtt{blind}(u,v) \neq w \\
\mathsf{S}_3: & \mathcal{C} \wedge \mathsf{f}(t_1,\ldots,t_n) \stackrel{?}{\in} \mathcal{B}d(T,v) & \rightarrow & \bot & \text{si } \mathsf{f} \neq \mathtt{blind} \text{ et } \mathsf{f}(t_1,\ldots,t_n) \neq v \\
\mathsf{S}_4: & \mathcal{C} \wedge x_1 \stackrel{?}{\in} \mathcal{B}d(T_1, v_1[x_2]) \wedge \ldots & & & \\
& \ldots \wedge x_n \stackrel{?}{\in} \mathcal{B}d(T_n, v_n[x_1]) & \rightarrow & \bot & \text{si } \exists i.(v_i \neq \epsilon \ \vee |\{x_1,\ldots,x_n\}| \geq 1)
\end{array}
$$

Figure 12.2: Règles de simplification

Simplification des contraintes d'appartenance

La sémantique de $'' \stackrel{?}{\in} \mathcal{B}d''$ permet quelques règles de simplification, notées $\mathcal{C} \rightarrow \mathcal{C}'$, qui préservent les solutions non-confondantes, $Sol_{\mathsf{NC}}(\mathcal{C}) = Sol_{\mathsf{NC}}(\mathcal{C}')$. Ces règles sont décrites dans la figure 12.2. Les règles S_1 et S_2 sont un développement de la sémantique de $\stackrel{?}{\in} \mathcal{B}d$. Les règles S_3 et S_4 sont dues au fait que les solutions considérées sont non-confondantes. Le but de la règle S_4 est aussi d'enlever les cycles dans une relation d'occurence définie plus tard.

Notons d'abord que ces règles terminent. En effet, la mesure

$$\mu(\mathcal{C}) = \sum_{(v \stackrel{?}{\in} \mathcal{B}d(T,u)) \in \mathcal{C}} |v|$$

decroît strictement à chaque application. Nous allons considérer dans la suite seulement des systèmes en forme simplifiée (aucune règle de simplification ne s'applique), dénotée par $\mathcal{C}{\downarrow_S}$. Notons que, dans un système simplifié, toutes les contraintes d'appartenance sont de la forme $x \stackrel{?}{\in} \mathcal{B}d(T,u)$.

Le lemme suivant montre que les systèmes simplifiés sont en effet des systèmes de contraintes et que les solutions sont préservées par simplification.

Lemme 61 *Soit $\mathcal{D}$ un ensemble de contraintes élémentaires. Alors,*

1. *si $\mathcal{D}$ est un système de contraintes, $\mathcal{D}{\downarrow_S}$ est un système de contraintes;*
2. *$Sol(\mathcal{D}{\downarrow_S}) \subseteq Sol(\mathcal{D})$ et $Sol_{\mathsf{NC}}(\mathcal{D}) \subseteq Sol_{\mathsf{NC}}(\mathcal{D}{\downarrow_S})$.*

Preuve: Montrons les deux points séparément.

1. Il suffit de montrer que chaque règle transforme un système de contraintes en un système de contraintes. Supposons que $\mathcal{D} \rightarrow \mathcal{D}'$. La monotonie est évidemment préservée. L'origination reste vraie car, quand on enlève une contrainte sans obtenir un système trivial (e.g. S_1), il est clair qu'elle n'introduit aucune variable nouvelle. Il nous reste à montrer que la dernière condition sur les variables dans la définition 38 est satisfaite. Supposons

que $\mathcal{D}' \neq \bot$ et soit $x \in \mathsf{Var}(\mathcal{D}') = \mathsf{Var}(\mathcal{D})$. S'il existe une contrainte $T_x \overset{?}{\vdash} u \in \mathcal{D}$ avec $x \in \mathsf{Var}(u)$, alors cette contrainte de déductibilité est présente aussi dans $\mathcal{D}'$ et on peut facilement conclure. Sinon, il existe une contrainte d'appartenance $v \overset{?}{\in} \mathcal{B}d(T_x, u) \in \mathcal{D}$ avec $x \in \mathsf{Var}(u)$ et $T_y \subsetneq T_x$, pour chaque $y \in \mathsf{Var}(v)$. Le seul cas où on ne peut pas conclure immédiatement est celui de la règle $\mathsf{S_2}$. Cependant, puisque $\mathsf{Var}(u) \subseteq \mathsf{Var}(\mathtt{blind}(u, v))$, on conclut facilement.

2. Soient $\mathcal{D}, \mathcal{D}'$ deux ensembles de contraintes tels que $\mathcal{D} \to \mathcal{D}'$. Pour établir les deux résultats, il suffit de montrer que $Sol(\mathcal{D}') \subseteq Sol(\mathcal{D})$ et $Sol_{\mathsf{NC}}(\mathcal{D}) \subseteq Sol_{\mathsf{NC}}(\mathcal{D}')$. Considérons chacune des règles de simplification. Dans le cas de $\mathsf{S_1}$ les deux inclusions suivent immédiatement par la sémantique de $'' \overset{?}{\in} \mathcal{B}d''$. Pour les autres règles, il est facile de voir que $Sol(\mathcal{D}') \subseteq Sol(\mathcal{D})$. Pour l'autre inclusion $Sol_{\mathsf{NC}}(\mathcal{D}) \subseteq Sol_{\mathsf{NC}}(\mathcal{D}')$, on se repose sur le fait que la solution σ considérée est non-confondante. Dans le cas de $\mathsf{S_2}$ (le cas de $\mathsf{S_3}$ est similaire), puisque $\mathtt{blind}(u, v) \neq w$, on a aussi que $\mathtt{blind}(u\sigma, v\sigma) \neq w\sigma$. Puisque $\mathtt{blind}(u\sigma, v\sigma) \in \mathcal{B}d(T\sigma, w\sigma)$, on doit nécessairement avoir $u\sigma \in \mathcal{B}d(T\sigma, w\sigma)$ et $T\sigma \vdash v\sigma$. Ceci nous permet de déduire $\sigma \in Sol(\mathcal{D}')$. Puisque les règles de simplification n'introduisent pas de nouveaux sous-termes, on conclut que $\sigma \in Sol_{\mathsf{NC}}(\mathcal{D}')$. Dans le cas de $\mathsf{S_4}$, en reposant sur le fait que σ est non-confondante, on obtient une contradiction, d'où $Sol_{\mathsf{NC}}(\mathcal{D}) = \emptyset$. □

12.2.2 Formes résolues et systèmes bien-formés

Comme dans le chapitre précédent, nous considérons une notion de forme résolue. Celle-ci doit cependant être étendue (cf exemple), afin de prendre en compte le nouveau prédicat.

Exemple 61 *Considérons le système suivant:*

$$\begin{array}{rcl} a & \overset{?}{\vdash} & x \\ a, \mathtt{sign}(x, b) & \overset{?}{\vdash} & \mathtt{sign}(a, b) \end{array}$$

L'ensemble des solutions de ce système est $\bigcup_{n\geq 0}\{x \mapsto b_n(a, \ldots, a)\}$. *Sans avoir recours aux contraintes d'appartenance, cet ensemble ne peut pas être décrit par un ensemble fini de formes résolues.*

La forme résolue de ce système sera, d'après la définition suivante et nos règles de transformation,

$$\begin{array}{rcl} a & \overset{?}{\vdash} & x \\ x & \overset{?}{\in} & \mathcal{B}d(\{a\}, a) \end{array}$$

L'idée dans la définition des formes résolues pour les signatures en aveugle est que, pour chaque variable x, ou bien on arrive à une description de x sans

contrainte d'appartenance (c'est le cas pour toutes les variables dans les formes résolues génériques), ou bien, comme dans l'exemple ci-dessus, on a une seule contrainte d'appartenance pour x dont l'ensemble des termes associé est le même que pour la (seule) contrainte de déductibilité qui introduit x.

Definition 39 (Forme résolue) *Un système de contraintes $\mathcal{C} = \{C_1, \ldots, C_\ell\}$ est en* forme résolue *si chaque C_i est ou bien une contrainte de déductibilité de la forme $T_i \stackrel{?}{\vdash} x_i$ ou bien une contrainte d'appartenance de la forme $x_i \stackrel{?}{\in} \mathcal{B}d(T_i, u_i)$, où x_i est une variable. De plus, on demande que pour chaque $x \in \mathsf{Var}(\mathcal{C})$:*

1. *il existe une unique contrainte de déductibilité $T \stackrel{?}{\vdash} x$; et*
2. *il existe au plus une contrainte d'appartenance $x \stackrel{?}{\in} \mathcal{B}d(T', u)$ et on a nécessairement $T' = T$.*

Exemple 62 *Considérons les systèmes de contraintes suivants:*

$$\mathcal{C}_1 = \left\{ \begin{array}{lll} a & \stackrel{?}{\vdash} & y \\ y & \stackrel{?}{\in} & \mathcal{B}d(\{a\}, a) \\ y & \stackrel{?}{\in} & \mathcal{B}d(\{a, b\}, a) \end{array} \right. \qquad \mathcal{C}_2 = \left\{ \begin{array}{lll} a & \stackrel{?}{\vdash} & \texttt{blind}(x, y) \\ y & \stackrel{?}{\in} & \mathcal{B}d(\{a\}, x) \end{array} \right.$$

$$\mathcal{C}_3 = \left\{ \begin{array}{lll} a & \stackrel{?}{\vdash} & x \\ a, x & \stackrel{?}{\vdash} & y \\ y & \stackrel{?}{\in} & \mathcal{B}d(\{a, x\}, x) \end{array} \right. \qquad \mathcal{C}_4 = \left\{ \begin{array}{lll} a & \stackrel{?}{\vdash} & x \\ x & \stackrel{?}{\in} & \mathcal{B}d(\{a\}, b) \end{array} \right.$$

Seulement $\mathcal{C}_3$ et $\mathcal{C}_4$ sont en forme résolue.

Notons que, même si $\mathcal{C}_4$ de l'exemple 62 est en forme résolue, il n'a pas de solution. $\mathcal{C}_3$, oui - une infinité -, car il est *bien-formé*.

La notion de bonne-formation, introduite ci-dessous, assure que les contraintes d'appartenance d'un système suivent une certaine structure, qui va aider à la construction d'une solution d'une forme résolue. Elle est basée sur un ordre d'occurence entre les variables:

Definition 40 (Ordre d'occurence) *Soit $\mathcal{C}$ un système de contraintes en forme simplifiée. On définit $\leq_{\mathcal{C}}$ sur $\mathsf{Var}(\mathcal{C})$ comme étant la plus petite relation close par transitivité et réflexivité telle que*

$$x \stackrel{?}{\in} \mathcal{B}d(T, u) \in \mathcal{C} \text{ et } y \in \mathsf{Var}(u) \Longrightarrow y \leq_{\mathcal{C}} x$$

Notons que, grâce à S_4, si $x \leq_{\mathcal{C}} y$ et $y \leq_{\mathcal{C}} x$, alors $x = y$ et donc $\leq_{\mathcal{C}}$ est un ordre partiel. On l'étend à un pré-ordre partiel sur des ensembles de variables comme suit:

$$V_1 \preceq_{\mathcal{C}} V_2 \text{ si et seulement si } \forall x \in V_1\, \exists y \in V_2 \text{ tel que } x \leq_{\mathcal{C}} y.$$

Pour un unsemble de variables V, on va dénoter par $\mathcal{C}_{\preceq V}^{\preceq}$ l'ensemble des contraintes dans $\mathcal{C}$ qui ne contiennent que des variables plut petites (par $\preceq_{\mathcal{C}}$) que celles dans V, i.e.

$$\mathcal{C}_{\preceq V}^{\preceq} = \{C \in \mathcal{C} \mid \mathsf{Var}(C) \preceq_{\mathcal{C}} V\}$$

Pour un ensemble de termes T, on ecrira $\mathcal{C}_T$ pour $\mathcal{C}_{\preceq \mathsf{Var}(T)}^{\preceq}$.

Exemple 63 *Considérons le système de contraintes suivant:*

$$\mathcal{C} := \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a & \stackrel{?}{\vdash} & y \\ a, \mathtt{blind}(x,a) & \stackrel{?}{\vdash} & z \\ y & \stackrel{?}{\in} & \mathcal{B}d(\{a, \mathtt{blind}(x,a)\}, z) \end{array} \right.$$

On $a\ z \preceq_{\mathcal{C}} y$, $\mathcal{C}_{\{y,x\}}^{\preceq} = \mathcal{C}$ et $\mathcal{C}_{\{z,x\}}^{\preceq} = \{a \stackrel{?}{\vdash} x,\ a, \mathtt{blind}(x,a) \stackrel{?}{\vdash} z\}$.

Pour deux ensembles de contraintes M, M', on notera par $M \models M'$ si toute solution de M est une solution de M'.

Definition 41 (Système bien-formé) *Un système simplifié $\mathcal{C}$ est* bien-formé *si pour chaque contrainte d'appartenance $y \stackrel{?}{\in} \mathcal{B}d(T_i, u_i)$ dans $\mathcal{C}$ on a:*

1. *ou bien $T_y \subsetneq T_i$;*
2. *ou bien $T_y = T_i$ et $\mathcal{C}_{\preceq V}^{\preceq} \models (T_i \stackrel{?}{\vdash} u_i)$, où $V = \mathsf{Var}(T_i \cup \{u_i\})$.*

Exemple 64 *Le système $\mathcal{C}$ de l'exemple 63 et les systèmes $\mathcal{C}_1, \mathcal{C}_3$ de l'exemple 62 sont bien-formés:*

- *$\mathcal{C}$, puisque $T_y = \{a\} \subsetneq \{a, blind(x,a)\}$.*
- *$\mathcal{C}_1$, puisque $a \vdash a$ et $\{a\} \subsetneq \{a,b\}$.*
- *$\mathcal{C}_3$, puisque $a, x \vdash x$.*

Les systèmes $\mathcal{C}_2$ et $\mathcal{C}_4$ de l'exemple 62 ne sont pas bien formés, car $a \nvdash x$ et $a \nvdash b$.

Le lemme suivant motive le fait que, dans la suite, nous aurons des règles de transformation qui mènent vers une forme résolue, en gardant à chaque pas l'invariant de bonne formation.

Lemme 62 *Si $\mathcal{C}$ est un sytème des contraintes bien-formé et en forme résolue, il a toujours une solution.*

Preuve: On considère un ordre parmi les variables de $\mathcal{C}$ défini comme suit:

$$x \leq y \text{ si et sulement si } \begin{cases} \text{ou bien } T_x \subsetneq T_y, \\ \text{ou bien } T_x = T_y \text{ et } x \leq_{\mathcal{C}} y \end{cases}$$

Notons que $x \leq_{\mathcal{C}} y$ implique $T_x \subseteq T_y$. En effet $x \leq_{\mathcal{C}} y$ implique qu'il existe $n \geq 1$ et n contraintes d'appartenance dans $\mathcal{C}$ de la forme $x_i \stackrel{?}{\in} \mathcal{B}d(T_i, u_i)$ $(1 \leq i \leq n)$, avec $y = x_1$, $x_{i+1} \in \mathsf{Var}(u_i)$, pour $1 \leq i < n$, et $x \in \mathsf{Var}(u_n)$. Puisque le système $\mathcal{C}$ est en forme résolue, on a $T_{x_i} = T_i$ et, en utilisant en plus l'origination, $T_{i+1} \subseteq T_i$, pour chaque i. On obtient donc $T_x \subseteq T_y$.

On en déduit donc que $x \leq_{\mathcal{C}} y$ implique $x \leq y$.

Soit $x_1, \ldots, x_n$ les variables de $\mathcal{C}$ renommées de manière à ce que $x_i \leq x_j$ implique $i \leq j$. Considèrons que les contraintes dans $\mathcal{C}$ sont ordonnées suivant la séquence $x_1, \ldots, x_n$, i.e.

$$\mathcal{C} := \begin{cases} T_1 \stackrel{?}{\vdash} x_1 \ \wedge \ [x_1 \stackrel{?}{\in} \mathcal{B}d(T_1, u_1)] \\ \qquad\qquad \ldots \\ T_n \stackrel{?}{\vdash} x_n \ \wedge \ [x_n \stackrel{?}{\in} \mathcal{B}d(T_n, u_n)] \end{cases}$$

La notation $[x \stackrel{?}{\in} \mathcal{B}d(T, u)]$ est utilisée pour indiquer que cette partie est optionelle. Par définition de $\leq$, il est clair que $T_1 \subseteq \ldots \subseteq T_n$. Montrons que, pour chaque i, $\mathsf{Var}(T_i, u_i) \subseteq \{x_1, \ldots, x_{i-1}\}$. En effet, pour chaque $y \in \mathsf{Var}(T_i)$, on a par origination que $T_y \subsetneq T_i$ et donc $y < x_i$, i.e $y \in \{x_1, \ldots, x_{i-1}\}$. De même, pour chaque $y \in \mathsf{Var}(u_i)$, on a $y <_{\mathcal{C}} x_i$ et donc $y < x_i$, i.e. $y \in \{x_1, \ldots, x_{i-1}\}$.

Soit maintenant σ la substitution définie, par:

- $x_i\sigma = u_i\sigma$, si $x_i \stackrel{?}{\in} \mathcal{B}d(T_i, u_i)$ est dans $\mathcal{C}$.
- $x_i\sigma \in T_i\sigma$, sinon.

pour tout $1 \leq i \leq n$. Grâce à nos observations précédentes, σ est une substitution bien-définie. Il nous reste à montrer que $\sigma \in Sol(\mathcal{C})$. On le fait par récurrence sur $1 \leq i \leq n$. *Cas de base:* $i = 1$. S'il n'existe pas de contrainte d'appartenance pour x_1, on a $T_1\sigma \vdash x_1\sigma$ par construction. Sinon, la contrainte d'appartenance est satisfaite par construction et il nous reste à montrer que $T_1 \vdash u_1$, car T_1 et u_1 sont clos. On a $\mathcal{C}^{\preceq}_{\mathsf{Var}(T_1 \cup \{u_1\})} = \emptyset$ et, puisque $\mathcal{C}$ est bien-formé, on en déduit que $\emptyset \models (T_1 \stackrel{?}{\vdash} u_1)$, et donc que $T_1 \vdash u_1$.

Pas de récurrence: on doit montrer que les contraintes $T_i \stackrel{?}{\vdash} x_i$ et $x_i \stackrel{?}{\in} \mathcal{B}d(T_i, u_i)$ (si elle existe) sont satisfaites par σ. Comme dans le cas de base, la difficulté est de montrer que $T_i\sigma \vdash u_i\sigma$, quand $x_i \stackrel{?}{\in} \mathcal{B}d(T_i, u_i)$ est dans $\mathcal{C}$. On a $\mathsf{Var}(T_i \cup \{u_i\}) \subseteq \{x_1, \ldots, x_{i-1}\}$. Soit $V_i = \mathsf{Var}(T_i \cup \{u_i\})$. En reposant sur l'hypothèse de récurrence, on sait que σ satisfait toutes les contraintes dans $\mathcal{C}^{\preceq}_{V_i}$. Puisque $\mathcal{C}$ est bien-formé, on a $C^{\preceq}_{V_i} \models (T_i \stackrel{?}{\vdash} u_i)$ et on conclut que $T_i\sigma \vdash u_i\sigma$. □

12.3 Transformation des contraintes et invariants

Les règles de transformation $\mathsf{Axiom}, \mathsf{Triv}, \mathsf{R}_1 - \mathsf{R}_4$ de la figure 12.1 ne suffisent pas pour mettre un système en forme résolue. En effet, elles ne s'appliquent pas par exemple sur le système $\mathcal{C}$ de l'exemple 63 et le système $\mathcal{C}_1$ de l'exemple 62. Pourtant, ces sytèmes ne sont pas en forme résolue. Il nous faut donc des règles additionnelles ($\mathsf{R}_5, \mathsf{R}_6$ et R_7) pour traiter les contraintes d'appartenance et arriver à une forme résolue. Les systèmes sont (implicitement) simplifiés après l'application de chaque règle.

Donnons quelques explications pour ces règles. Notons que, par définition, une solution σ d'une contrainte d'appartenance $x \overset{?}{\in} \mathcal{B}d(T', v)$ vient avec des témoins $v_1, \ldots, v_n$ tels que $x\sigma = b_n(v\sigma, v_1, \ldots, v_n)$ et $T'\sigma \vdash v_1, \ldots, T'\sigma \vdash v_n$. La règle R_5 va s'appliquer quand on peut trouver des preuves de $v_1, \ldots, v_n$ avec un ensemble d'hypothèses plus petit $T'\sigma \subsetneq T\sigma$. La règle R_6 va s'appliquer lorsque seulement quelques-uns des $v_1, \ldots, v_n$ peuvent être prouvés avec moins d'hypothèses. Si on a deux contraintes d'appartenance $x \overset{?}{\in} \mathcal{B}d(T, v)$ et $x \overset{?}{\in} \mathcal{B}d(T, v')$, une des séquences témoin est forcement un prefixe de l'autre. La règle R_7 va garder seulement une copie de cette partie commune.

Nous considérons dans la suite l'ensemble des règles $\mathsf{Triv}, \mathsf{Axiom}, \mathsf{R}_1 - \mathsf{R}_7$, modulo les règles de simplification.

12.3.1 Correction

Lemme 63 (Correction) *Soit $\mathcal{C}$ un système de contraintes (simplifié) tel que $\mathcal{C} \rightsquigarrow \mathcal{C}'$. Alors $\mathcal{C}'$ est un système de contraintes tel que $st(\mathcal{C}') \subseteq st(\mathcal{C})$ et $Sol(\mathcal{C}') \subseteq Sol(\mathcal{C})$.*

Preuve: En reposant sur le lemme 61, on peut supposer qu'aucune simplification ne s'effectue lors de la transition $\mathcal{C} \rightsquigarrow \mathcal{C}'$.

Montrons d'abord que $\mathcal{C}'$ est en effet un système de contraintes. Il est facile de voir que la monotonie et l'origination sont préservées. Vérifions que la troisième condition de la définition 38 est aussi satisfaite, après chaque règle de transformation.

Les règles $\mathsf{R}_1, \mathsf{Triv}$ ne posent aucun problème.

Les règles $\mathsf{R}_2, \mathsf{R}_3$ *et* Axiom affectent seulement des contraintes qui n'introduisent pas de nouvelles variables. Elles ne posent donc aucun problème.

Si la règle R_4 est appliquée, alors, même si la contrainte d'appartenance introduite, i.e. $w \overset{?}{\in} \mathcal{B}d(T, v)$, introduit une variable pour la première fois, on a $T_y \subsetneq T$ pour tout $y \in \mathsf{Var}(w)$, par origination de $\mathcal{C}$ et puisque $w \in st(T)$. Ceci nous permet de conclure.

Dans le cas de R_5 *(resp.* R_6*)*, la contrainte d'appartenance additionnelle n'introduit pas de nouvelles variables, grâce à la présence de la contrainte de déductibilité $T \overset{?}{\vdash} v$ (resp. $T \overset{?}{\vdash} w$). Dans le cas de R_6, supposons que la contrainte d'appartenance

$w \overset{?}{\in} \mathcal{B}d(T', v)$ introduit une variable. On a alors $T_y \subsetneq T \subsetneq T'$ pour chaque $y \in \mathsf{Var}(w)$, car $w \in st(T)$. On peut ainsi conclure.

Dans le cas de $\mathsf{R_7}$, la contrainte d'appartenance additionelle ne nuit pas à notre condition, puisque v' ne peut pas introduire de variable. En effet, dans le cas contraire, $\mathcal{C}$ ne satisfait pas lui-même la condition.

Ceci nous permet de conclure que nos règles transforment un système de contraintes dans un système de contraintes.

L'inclusion $st(\mathcal{C}') \subseteq st(\mathcal{C})$ est immédiate pour toutes les règles.

Montrons maintenant que $Sol(\mathcal{C}') \subseteq Sol(\mathcal{C})$. Soit $\sigma \in Sol(\mathcal{C}')$ et considérons le cas de chaque règle de transformation. En fait, il suffit de considérer les règles qui transforment des contraintes d'appartenance, car dans le cas des règles $\mathsf{Axiom}, \mathsf{Triv}, \mathsf{R_1}, \mathsf{R_2}, \mathsf{R_3}, \mathsf{R_4}$ on peut conclure par le lemme 45.

Cas $\mathsf{R_5}$*:* Les arbres de preuve qui témoignent que $\sigma \in Sol(\mathcal{C}')$ peuvent être utilisées pour montrer que $\sigma \in Sol(\mathcal{C})$. L'arbre de preuve qui témoigne du fait que $T \overset{?}{\vdash} v$ n'est même pas utile pour cela.

Cas $\mathsf{R_6}$*:* On doit grouper les séquences de preuves témoignant $x \overset{?}{\in} \mathcal{B}d(T, w)$ et $w \overset{?}{\in} \mathcal{B}d(T', v)$ pour voir que $x\sigma \in \mathcal{B}d(T'\sigma, v\sigma)$.

Cas $\mathsf{R_7}$*:* On doit grouper les séquences de preuves témoignant de faits $x \overset{?}{\in} \mathcal{B}d(T, v)$ et $v \overset{?}{\in} \mathcal{B}d(T, v')$ pour voir que $x\sigma \in \mathcal{B}d(T\sigma, v'\sigma)$. □

12.3.2 Complétude

Mesure PS. Pour prouver la complétude de notre système de règles pour $\mathcal{I}_{\texttt{blind}}$, par rapport à la nouvelle notion de forme résolue, on étend la mesure de complexité PS pour les systèmes de contraintes avec des contraintes d'appartenance. Soit $\mathcal{C}$ un système de contraintes et σ une solution de $\mathcal{C}$. Comme avant, pour toute contrainte de déductibilité $T \overset{?}{\vdash} u \in \mathcal{C}$, $\mathsf{PS}(T \overset{?}{\vdash} u, \sigma)$ est la taille d'une preuve simple et minimale de $T\sigma \vdash u\sigma$. Pour une contrainte d'appartenance $x \overset{?}{\in} \mathcal{B}d(T, u)$, il existe par définition $u_1, \ldots, u_n$ tels que $x\sigma = b_n(u, u_1, \ldots, u_n)$ et $T\sigma \vdash u_1, \ldots, T\sigma \vdash u_n$. On définit alors $\mathsf{PS}(x \overset{?}{\in} \mathcal{B}d(T, u), \sigma)$ comme étant la somme des tailles de preuves simples et minimales de $T\sigma \vdash u_1, \ldots, T\sigma \vdash u_n$, pour un n minimal. La mesure PS est étendue aux systèmes de contraintes en définissant, pour toute solution σ de $\mathcal{C}$, $\mathsf{PS}(\mathcal{C}, \sigma)$ comme étant le multi-ensemble de paires $(\mathsf{lev}(C, \mathcal{C}), \mathsf{PS}(C, \sigma))$, pour toutes les contraintes élémentaires $C \in \mathcal{C}$. Les multi-ensembles $\mathsf{PS}(\mathcal{C}, \sigma)$ sont comparés en utilisant l'extension multi-ensemble de la composition lexicographique des ordres. Notons que, si le système ne contient pas de contraintes d'appartenance, la définition de PS coincide avec la définition donnée dans la partie générique.

L'idée de la procédure est la même: on va considérer la première contrainte non-résolue et montrer qu'on peut appliquer une règle de transformation et diminuer la mesure PS. Lorsque cette contrainte est une contrainte de

déductibilité, les arguments sont les mêmes que dans notre procédure générique. Pour une contrainte d'appartenance, on va montrer qu'une des règles $\mathsf{R}_5, \mathsf{R}_6, \mathsf{R}_7$ s'applique et que la mesure diminue.

Étant donné un ensemble de termes T, on dénotera par $\mathcal{C}(T)$ les contraintes de $\mathcal{C}$ dont T est l'ensemble de termes associé, i.e.

$$\mathcal{C}(T) = \{C \in \mathcal{C} \mid C = T \stackrel{?}{\vdash} u \text{ ou } C = v \stackrel{?}{\in} \mathcal{B}d(T, u) \text{ pour certains } u, v\}$$

Definition 42 (contrainte minimale non-résolue) *Soit $\mathcal{C}$ un système de contraintes (simplifié) qui n'est pas en forme résolue. Soit $T_{\min}$ l'ensemble minimal (par rapport à l'inclusion) de termes tel que $\mathcal{C}^- = \bigcup_{T_i \subseteq T_{\min}} \mathcal{C}(T_i)$ n'est pas résolu.*

Parmi $\mathcal{C}(T_{\min})$, soit S un ensemble maximal de contraintes tel que $\bigcup_{T_i \subsetneq T_{\min}} \mathcal{C}(T_i) \cup S$ est en forme résolue. Alors $\mathsf{M}_{\mathcal{C}} = \mathcal{C}(T_{\min}) \setminus S$ sont les contraintes non-résolues minimales de $\mathcal{C}$.

Pour une contrainte $C \in \mathsf{M}_{\mathcal{C}}$, on pose $S(\mathcal{C}, C) = \bigcup_{T_i \subsetneq T_{\min}} \mathcal{C}(T_i) \cup S \cup C$.

Notons que $\mathsf{M}_{\mathcal{C}}$ n'est pas toujours unique:

Exemple 65 *Considérons le système de contraintes suivant, où $T = \{a, b\}$.*

$$\mathcal{C} := \left\{ \begin{array}{c} T \stackrel{?}{\vdash} x \\ x \stackrel{?}{\in} \mathcal{B}d(T, \mathtt{blind}(a, b)) \\ x \stackrel{?}{\in} \mathcal{B}d(T, a) \end{array} \right.$$

Il n'est pas résolu. On a $\mathcal{C} = \mathcal{C}(T)$ et $T_{\min} = T$. Cependant, pour $\mathsf{M}_{\mathcal{C}}$, on a deux possibilités: $\mathsf{M}_{\mathcal{C}} = \{x \stackrel{?}{\in} \mathcal{B}d(T, \mathtt{blind}(a, b))\}$ ou $\mathsf{M}_{\mathcal{C}} = \{x \stackrel{?}{\in} \mathcal{B}d(T, a)\}$.

Pour montrer qu'un système $\mathcal{C}$, qui n'est pas en forme résolue, peut être simplifié, nous distinguons deux cas: ou bien parmi les contraintes minimales non-résolues il existe une contrainte de déductibilité (lemme 64) ou bien toutes ces contraintes sont d'appartenance (lemme 65). Les différents cas sont résumés dans le tableau 12.3.

Lemme 64 (complétude - contrainte de déductibilité) *Soit $\mathcal{C}$ un système de contraintes non-résolu tel que $\mathsf{M}_{\mathcal{C}}$ contient une contrainte de déductibilité. Soit $\sigma \in Sol_{\mathsf{NC}}(\mathcal{C})$. Il existe un système de contraintes $\mathcal{C}'$ tel que $\mathcal{C} \rightsquigarrow \mathcal{C}'$, $\sigma \in Sol_{\mathsf{NC}}(\mathcal{C}')$ et $\mathsf{PS}(\mathcal{C}', \sigma) < \mathsf{PS}(\mathcal{C}, \sigma)$.*

Preuve: Soit $T \stackrel{?}{\vdash} v$ une contrainte de déductibilité dans $\mathsf{M}_{\mathcal{C}}$. Alors, par définitions, on peut extraire de $S(\mathcal{C}, T \stackrel{?}{\vdash} v)$ un système de contraintes $\mathcal{D}$ qui ne contient que des contraintes de déductibilité, n'est pas en forme résolue et

$\mathcal{C}$ contient parmi d'autres	$\mathsf{M}_{\mathcal{C}}$ contient C	La dernière règle dans la preuve P, témoin de C	règle
	$T \overset{?}{\vdash} \mathsf{f}(u_1, u_2)$	$(\mathsf{S}), (\mathsf{B})$	R_1
	$T \overset{?}{\vdash} u$	(UB)	R_2
	$T \overset{?}{\vdash} u$	(C)	R_3
	$T \overset{?}{\vdash} \mathtt{sign}(u_1, u_2)$	$(\mathsf{UB_n})$	R_4
$T \overset{?}{\vdash} x$	$x \overset{?}{\in} \mathcal{B}d(T', v)$ avec $T \subsetneq T'$	T' peut être affaibli à T dans toutes les preuves auxiliaires	R_5
$T \overset{?}{\vdash} x$	$x \overset{?}{\in} \mathcal{B}d(T', v)$ avec $T \subsetneq T'$	T' peut être affaibli à T dans certains des preuves auxiliaires	R_6
$T \overset{?}{\vdash} x$ $x \overset{?}{\in} \mathcal{B}d(T, v')$	$x \overset{?}{\in} \mathcal{B}d(T, v)$		R_7

Figure 12.3: Résumé: preuve de complétude

dont la contrainte minimale non-résolue est $T \overset{?}{\vdash} v$. Par le lemme 49, il existe un système de contraintes $\mathcal{D}'$ tel que $\mathcal{D} \rightsquigarrow \mathcal{D}'$, en utilisant $\mathsf{Axiom}, \mathsf{Triv}, \mathsf{R}_1, \mathsf{R}_2, \mathsf{R}_3$ ou $\mathsf{Dec}'(\mathsf{UB_n})$ et tel que $\sigma \in \mathit{Sol}(\mathcal{D}')$, $\mathsf{PS}(\mathcal{D}', \sigma) < \mathsf{PS}(\mathcal{D}, \sigma)$. Si la règle utilisée est differente de $\mathsf{Dec}'(\mathsf{UB_n})$, alors, par définitions, on peut étendre cette transition à $\mathcal{C}$: $\mathcal{C} \rightsquigarrow \mathcal{C}'$ avec $\sigma \in \mathit{Sol}(\mathcal{C}')$ et $\mathsf{PS}(\mathcal{C}', \sigma) < \mathsf{PS}(\mathcal{C}, \sigma)$. Si la règle utilisée est $\mathsf{Dec}'(\mathsf{UB_n})$, alors, par le lemme 60, il existe un $\mathcal{D}''$ tel que $\mathcal{D} \rightsquigarrow \mathcal{D}''$, en utilisant R_4, $\sigma \in \mathit{Sol}(\mathcal{D}'')$ et $\mathsf{PS}(\mathcal{D}'', \sigma) = \mathsf{PS}(\mathcal{D}', \sigma)$. Encore une fois, on peut étendre cette transition à $\mathcal{C}$, pour obtenir le $\mathcal{C}'$ demandé. □

Lemme 65 (complétude - contrainte d'appartenance) *Soit $\mathcal{C}$ un système de contraintes non-résolu tel que $\mathsf{M}_{\mathcal{C}}$ ne contient que des contraintes d'appartenance. Soit $\sigma \in \mathit{Sol}_{\mathsf{NC}}(\mathcal{C})$. Il existe un système de contraintes $\mathcal{C}'$ tel que $\mathcal{C} \rightsquigarrow \mathcal{C}'$, $\sigma \in \mathit{Sol}_{\mathsf{NC}}(\mathcal{C}')$ et $\mathsf{PS}(\mathcal{C}', \sigma) < \mathsf{PS}(\mathcal{C}, \sigma)$.*

Preuve: Notons que, puisque $\mathcal{C}$ est simplifié, toutes les contraintes d'appartenance portent sur des variables. Soit $V_{\mathsf{M}} = \{x \mid x \overset{?}{\in} \mathcal{B}d(T', v) \in \mathsf{M}_{\mathcal{C}}\}$ et choisissons une contrainte $x \overset{?}{\in} \mathcal{B}d(T', v)$ dans $\mathsf{M}_{\mathcal{C}}$ telle que x est maximal dans V_{M} pour l'ordre d'occurence $\leq_{\mathcal{C}}$. Notons que, grâce à l'origination et à la maximalité de x, il existe une contrainte $T \overset{?}{\vdash} x$ qui apparaît dans $\mathcal{C}$, avec $T \subseteq T'$. En effet, supposons que x est introduite dans une contraite $y \overset{?}{\in} \mathcal{B}d(T'', u)$, avec $x \in \mathsf{Var}(u)$ et $T'' \subseteq T'$. Par la maximalité de x, nous avons $y \overset{?}{\in} \mathcal{B}d(T'', u) \notin \mathsf{M}_{\mathcal{C}}$. Par

définition des formes résolues, il existe $T \subseteq T''$ tel que $T \overset{?}{\vdash} x \in \mathcal{C}$.

On distingue plusieurs cas, suivant si $T \subsetneq T'$ ou $T = T'$.

- Supposons que $T \subsetneq T'$. On va montrer que dans ce cas on peut appliquer $\mathsf{R_5}$ ou $\mathsf{R_6}$. Par définition d'une solution, on a $x\sigma = \mathtt{blind}(\ldots \mathtt{blind}(v\sigma, v_1), \ldots, v_k)$ avec $T'\sigma \vdash v_1, \ldots, T'\sigma \vdash v_k$. Notons que $k > 0$ puisque σ est non-confondante et $\mathcal{C}$ est simplifié. Suivant si toutes les preuves de $T'\sigma \vdash v_i$ peuvent être affaiblies ou pas, on applique $\mathsf{R_5}$ ou $\mathsf{R_6}$.

  - Supposons que $T\sigma \vdash v_i$, pour tout $1 \leq i \leq k$. Soit $\mathcal{C}'$ le système de contraintes obtenu en appliquant la règle $\mathsf{R_5}$ à $\mathcal{C}$. On a $T\sigma \vdash x\sigma$ et donc $T\sigma \vdash v\sigma$. Une preuve témoignant de ce fait peut être obtenue en appliquant k fois la règle d'inférence UB à la conclusion de la preuve de $T \vdash x\sigma$ et en utilisant les preuves qui témoignent de $T\sigma \vdash v_i$, pour chaque $1 \leq i \leq k$. Par conséquent, on obtient $\sigma \in Sol(\mathcal{C}')$ et, puisque aucun sous-terme n'est introduit, on a $\sigma \in Sol_{\mathsf{NC}}(\mathcal{C}')$.

    De plus, on a $\mathsf{PS}(\mathcal{C}', \sigma) < \mathsf{PS}(\mathcal{C}, \sigma)$: on a remplacé une paire (T', n) par deux paires $(T, n_1), (T, n_2)$ avec $T \subsetneq T'$.

  - Sinon, soit i_0 tel que $T\sigma \vdash v_j$ pour chaque $j > i_0$ et $T\sigma \nvdash v_{i_0}$. Notons que, si on n'est pas dans le premier cas, un tel i_0 existe et $i_0 \geq 1$. Notons aussi que, si $i_0 = k$, toute preuve (donc toute preuve simple) de $T\sigma \vdash x\sigma$ finit avec un axiome ou une règle de décomposition. Par le lemme 48, on en déduit qu'il existe un $t \in st(T) \setminus \mathcal{X}$ tel que $t\sigma = x\sigma$. Puisque σ est non-confondante, ce cas est impossible et on obtient $1 \leq i_0 < k$.

    On a $T\sigma \vdash \mathtt{blind}(\ldots \mathtt{blind}(v\sigma, v_1), \ldots, v_{i_0})$, en considérant la preuve de $T\sigma \vdash x\sigma$, les preuves de $T\sigma \vdash v_k, \ldots, T\sigma \vdash v_{i_0+1}$ et en appliquant $k - i_0$ fois la règle d'inférence UB. Soit P une preuve simple de $T\sigma \vdash \mathtt{blind}(\ldots \mathtt{blind}(v\sigma, v_1), \ldots, v_{i_0})$. Puisque $T\sigma \nvdash v_{i_0}$, en inspectant les règles d'inférence, P ne peut pas se terminer avec une règle de composition ou un règle versatile. En conséquence, par le lemme 48, on en déduit qu'il existe un $w \in (st(T) \setminus \mathcal{X})$ tel que $w\sigma = \mathtt{blind}(\ldots \mathtt{blind}(v\sigma, v_1), \ldots, v_{i_0})$, impliquant que $w\sigma \in \mathcal{B}d(T'\sigma, v\sigma)$. On peut donc appliquer la règle $\mathsf{R_6}$ pour obtenir un système de contraintes $\mathcal{C}'$. Comme on vient de voir, σ satisfait chaque contrainte rajoutée à $\mathcal{C}'$, donc $\sigma \in Sol(\mathcal{C}')$ et $\sigma \in Sol_{\mathsf{NC}}(\mathcal{C}')$.

    Enfin, on a $\mathsf{PS}(\mathcal{C}', \sigma) < \mathsf{PS}(\mathcal{C}, \sigma)$: on remplace une paire (T', n) par trois paires $(T, n_1), (T, n_2)$ et (T', n_3), avec $T \subsetneq T'$ et $n_3 < n$. Cette dernière inégalité est due au fait que $i_0 < k$.

- Cas $T = T'$. Dans ce cas, puisque $x \overset{?}{\in} \mathcal{B}d(T', v) \in \mathsf{M}_{\mathcal{C}}$, il doit exister une autre contrainte d'appartenance $x \overset{?}{\in} \mathcal{B}d(T', v') \in \mathcal{C}$. De plus, on a $T_x = T$.

  En effet, autrement on aurait $T \overset{?}{\vdash} x \in \mathsf{M}_{\mathcal{C}}$, contredisant les hypothèses du lemme. Ainsi on a

– $x\sigma = \texttt{blind}(\ldots\texttt{blind}(v\sigma, v_1), \ldots, v_k)$ avec $T\sigma \vdash v_i$, pour $1 \leq i \leq k$; et

– $x\sigma = \texttt{blind}(\ldots\texttt{blind}(v'\sigma, v'_1), \ldots, v'_p)$ avec $T\sigma \vdash v'_i$, pour $1 \leq i \leq p$.

Clairement, on a ou bien $v\sigma \in st(v'\sigma)$ et $v'\sigma \in \mathcal{B}d(T\sigma, v\sigma)$, ou bien, symétriquement, $v'\sigma \in st(v\sigma)$ et $v\sigma \in \mathcal{B}d(T\sigma, v'\sigma)$. Supposons, sans perte de généralité, que nous sommes dans le deuxième cas. On peut appliquer la règle R_7 pour obtenir un système de contraintes $\mathcal{C}'$. Selon nos observations précédentes, il est clair que $\sigma \in Sol_{\mathsf{NC}}(\mathcal{C}')$.

Il nous reste à montrer que la séquence de preuves témoignant du fait que $v\sigma \in \mathcal{B}d(T\sigma, v'\sigma)$ est strictement plus petite que la séquence témoignant du fait que $x\sigma \in \mathcal{B}d(T\sigma, v\sigma)$. Ceci provient du fait que $v'\sigma$ est un sous-terme strict de $x\sigma$. En effet, on a $v'\sigma \in st(x\sigma)$ et $w\sigma \neq x\sigma$, car σ est non-confondante et le système est simplifié. □

Corollaire 7 *Soit $\mathcal{C}$ un système de contraintes et $\sigma \in Sol_{\mathsf{NC}}(\mathcal{C})$. Il existe un système de contraintes $\mathcal{C}'$ en forme résolue tel que $\mathcal{C} \rightsquigarrow^* \mathcal{C}'$ et $\sigma \in Sol_{\mathsf{NC}}(\mathcal{C}')$.*

Preuve: C'est une simple récurrence sur $\mathsf{PS}(\mathcal{C}, \sigma)$, en appliquant, selon le cas, le lemme 64 ou le lemme 65 et en utilisant le fait que les règles de simplification ne font pas croître la mesure PS. □

12.3.3 Bonne-formation

Comme montré dans la section 12.2.2, une forme résolue n'a pas nécessairement une solution. Conformément au lemme 62, il nous faut un invariant de bonne-formation pour utiliser le corollaire 7 dans une procédure de décision. Le but de cette section est de montrer cet invariant: si $\mathcal{C}$ est un système bien-formé et $\mathcal{C} \rightsquigarrow \mathcal{C}'$, alors $\mathcal{C}'$ est un système bien-formé.

On montre d'abord que la relation $\leq_{\mathcal{C}}$ peut être seulement raffinée par les règles de transformation.

Exemple 66 *Considérons le système $\mathcal{C}$ de l'exemple 63. On a $y \geq_{\mathcal{C}} z$ et $\mathcal{C} \rightsquigarrow \mathcal{C}'$ (par R_6):*

$$\mathcal{C}' = \left\{ \begin{array}{rcl} a & \stackrel{?}{\vdash} & x \\ a & \stackrel{?}{\vdash} & y \\ a, \texttt{blind}(x, a) & \stackrel{?}{\vdash} & z \\ y \stackrel{?}{\in} \mathcal{B}d(\{a\}, \texttt{blind}(x, a)) & \wedge & x \stackrel{?}{\in} \mathcal{B}d(\{a, \texttt{blind}(x, a)\}, z) \end{array} \right.$$

où $y >_{\mathcal{C}'} x >_{\mathcal{C}'} z$.

Lemme 66 (propriété de $\leq_{\mathcal{C}}$) *Soit $\mathcal{C}$ un système de contraintes simplifié et $\mathcal{C}'$ un système de contraintes tel que $\mathcal{C} \rightsquigarrow \mathcal{C}'$ et $\mathcal{C}' \neq \bot$. Alors $\leq_{\mathcal{C}} \subseteq \leq_{\mathcal{C}'}$.*

Preuve: Considérons chaque règle de transformation. Le cas des règles Axiom, Triv, R_1, R_2 et R_3 est trivial, car elles n'affectent pas les contraintes d'appartenance. La règle R_4 introduit une nouvelle contrainte d'appartenance et donc il est facile de voir que $\leq_{\mathcal{C}} \subseteq \leq_{\mathcal{C}'}$. Il nous restent les trois cas suivants:

- R_5: $\mathcal{C}_0 \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T', v) \rightsquigarrow \mathcal{C}_0 \wedge T \overset{?}{\vdash} x \wedge T \overset{?}{\vdash} v \wedge x \overset{?}{\in} \mathcal{B}d(T, v)$ avec $T \subsetneq T'$. Dans ce cas, l'ordre n'est pas affecté: $\leq_{\mathcal{C}} = \leq_{\mathcal{C}'}$.

- R_6: $\mathcal{C}_0 \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T', v) \rightsquigarrow \mathcal{C}_0 \wedge T \overset{?}{\vdash} x \wedge T \overset{?}{\vdash} w \wedge x \overset{?}{\in} \mathcal{B}d(T, w) \wedge w \overset{?}{\in} \mathcal{B}d(T', v)$ avec $T \subsetneq T'$ et $w \in st(T)$. On doit montrer que $y \leq_{\mathcal{C}'} x$, pour chaque $y \in \mathsf{Var}(v)$. Les règles de simplification appliquées à $w \overset{?}{\in} \mathcal{B}d(T', v)$ mènent à une contrainte d'appartenance de la forme $z \overset{?}{\in} \mathcal{B}d(T', v)$, avec $z \in \mathsf{Var}(w)$, ou à un contrainte vide seulement si $v \in st(w)$. Dans les deux cas, on conclut facilement.

- R_7: $\mathcal{C}_0 \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T, v) \wedge x \overset{?}{\in} \mathcal{B}d(T, v') \rightsquigarrow \mathcal{C}_0 \wedge T \overset{?}{\vdash} x \wedge x \overset{?}{\in} \mathcal{B}d(T, v) \wedge v \overset{?}{\in} \mathcal{B}d(T, v')$ avec $T_x = T$. On doit montrer que $y \leq_{\mathcal{C}'} x$, pour chaque $y \in \mathsf{Var}(v')$. Les règles de simplification appliquées à $v \overset{?}{\in} \mathcal{B}d(T, v')$ mènent à une contrainte d'appartenance de la forme $z \overset{?}{\in} \mathcal{B}d(T, v')$, avec $z \in \mathsf{Var}(v)$, ou à une contrainte vide seulement si $v' \in st(v)$. Dans les deux cas, on conclut facilement. □

Un résultat essentiel avant la preuve de l'invariant est la compatibilité entre la projection d'un système sur un ensemble de variables V et les règles de transformation.

Exemple 67 *Considérons les systèmes $\mathcal{C}$ et $\mathcal{C}'$ de l'exemple 66. On a $\mathcal{C}_{y,x} \rightsquigarrow {\mathcal{C}'}_{y,x}$, tandis que $\mathcal{C}_{x,z} \subseteq {\mathcal{C}'}_{x,z}$. Dans les deux cas, on en déduit que ${\mathcal{C}'}_V \models \mathcal{C}_V$.*

Lemme 67 *Soit $\mathcal{D}$ un ensemble de contraintes simplifié et $\mathcal{D}'$ tel que $\mathcal{D} \rightsquigarrow \mathcal{D}'$. Si V est un ensemble de variables, alors $\mathcal{D}'_V \models \mathcal{D}_V$.*

Preuve: Chaque règle de transformation appliquée dans $\mathcal{D} \rightsquigarrow \mathcal{D}'$ est de la forme $C_1, \ldots, C_n \rightsquigarrow C'_1, \ldots, C'_m$. On sépare la preuve dans deux cas: ou bien $\{C_1, \ldots, C_n\} \nsubseteq \mathcal{D}_V$ ou bien $\{C_1, \ldots, C_n\} \subseteq \mathcal{D}_V$.

Supposons d'abord que $\{C_1, \ldots, C_n\} \nsubseteq \mathcal{D}_V$. On montre que $\mathcal{D}_V \subseteq \mathcal{D}'_V$. En effet, soit $C \in \mathcal{D}_V$. Si $C \notin \{C_1, \ldots, C_n\}$, on obtient $C \in \mathcal{D}'_V$ par le lemme 66. Supposons maintenant que $C \in \{C_1, \ldots, C_n\}$. Puisque $\{C_1, \ldots, C_n\} \nsubseteq \mathcal{D}_V$, ceci est le cas seulement si C est la contrainte $T \overset{?}{\vdash} x$ dans les règles R_5 et R_6. Donc $C \in \mathcal{D}'$ et, par le lemme 66, $C \in \mathcal{D}'_V$. On conclut que $\mathcal{D}_V \subseteq \mathcal{D}'_V$ et donc $\mathcal{D}'_V \models \mathcal{D}_V$.

Supposons maintenant que $\{C_1, \ldots, C_n\} \subseteq \mathcal{D}_V$. On montre que $\mathcal{D}_V \rightsquigarrow \mathcal{DV}$, pour un ensemble de contraintes $\mathcal{DV}$ tel que $\mathcal{DV} \subseteq \mathcal{D}'_V$. En effet, on a $\mathcal{D}_V = \{C_1, \ldots, C_n\} \cup M \rightsquigarrow \{C'_1, \ldots, C'_m\} \cup M$. Ceci nous donne $\mathcal{DV}$. Montrons que

chaque contrainte dans $\mathcal{DV} = \{C'_1, \ldots, C'_m\} \cup M$ est un élément de $\mathcal{D}'_V$. D'abord, pour les contraintes $C \in M \subseteq \mathcal{D}_V$, on peut répeter l'argument d'avant, basé sur le lemme 66. Ensuite, pour tout $1 \leq i \leq m$, on a $\mathsf{Var}(C'_i) \subseteq \mathsf{Var}(C_1, \ldots, C_n)$ et, puisque $\{C_1, \ldots, C_n\} \subseteq \mathcal{D}_V$, on en déduit que $\mathsf{Var}(C'_i) \prec_{\mathcal{D}} V$. Par le lemme 66, on obtient $\mathsf{Var}(C'_i) \prec_{\mathcal{D}'} V$ et donc $C'_i \in \mathcal{D}'_V$.

On obtient donc $\mathcal{D}_V \rightsquigarrow \mathcal{DV} \subseteq \mathcal{D}'_V$. On en conclut que $\mathcal{D}'_V \models \mathcal{DV} \models \mathcal{D}_V$, en utilisant la correction des règles de transformation pour obtenir $\mathcal{DV} \models \mathcal{D}_V$. □

On est maintenant pret à prouver notre invariant.

Exemple 68 *Dans la transformation suivante, le premier système est bien-formé grâce à la première condition de la définition, tandis que le deuxième est bien-formé grâce à la deuxième.*

$$\mathcal{C}' = \left\{ \begin{array}{lll} a & \overset{?}{\vdash} & x \\ x & \overset{?}{\in} & \mathcal{B}d(\{a, b\}, y) \end{array} \right.$$

mène, par $\mathsf{R_5}$*, à la forme résolue bien-formée:*

$$\mathcal{C}'' = \left\{ \begin{array}{lll} a & \overset{?}{\vdash} & x \\ a & \overset{?}{\vdash} & y \\ x & \overset{?}{\in} & \mathcal{B}d(\{a\}, y) \end{array} \right.$$

Lemme 68 *Si $\mathcal{C}$ est un système de contraintes bien-formé et $\mathcal{C} \rightsquigarrow \mathcal{C}'$, alors $\mathcal{C}'$ est bien-formé.*

Preuve: On doit montrer que chaque contrainte d'appartenance $M = x \overset{?}{\in} \mathcal{B}d(T, u)$ dans $\mathcal{C}'$ est telle que:

- ou bien $T_x \subsetneq T$
- ou bien $T_x = T$ et $\mathcal{D}^{\preceq}_V \models (T \overset{?}{\vdash} u)$, où $V = \mathsf{Var}(T \cup \{u\})$.

Considérons d'abord une contrainte d'appartenance $M = x \overset{?}{\in} \mathcal{B}d(T, u)$ qui est aussi dans $\mathcal{C}$. Si $T_x \subsetneq T$ dans $\mathcal{C}$, alors $T_x \subsetneq T$ reste vraie dans $\mathcal{C}'$ et on peut conclure. Sinon, on a $T_x = T$ et $\mathcal{C}^{\preceq}_V \models (T \overset{?}{\vdash} u)$. Par le lemme 67, on en déduit que $\mathcal{C}'^{\preceq}_V \models \mathcal{C}^{\preceq}_V$. On adonc $\mathcal{C}'^{\preceq}_V \models (T \overset{?}{\vdash} u)$.

Par conséquent, dans la suite on considère seulement des contraintes d'appartenance dans $\mathcal{C}' \setminus \mathcal{C}$. Prenons chaque cas à part.

- Dans le cas des règles $\mathsf{Axiom}, \mathsf{Triv}, \mathsf{R_1}$, $\mathsf{R_2}$ et $\mathsf{R_3}$, on n'a pas de nouvelles contraintes d'appartenance et on peut conclure.
- règle $\mathsf{R_4}$. Dans ce cas, si la nouvelle contrainte d'appartenance ne disparaît pas par simplification, on a $\mathcal{C}' = (\mathcal{C} \setminus \{T \overset{?}{\vdash} \mathtt{sign}(v, u)\}) \cup \{T \overset{?}{\vdash}$

$\mathtt{sign}(w,u),\ y \overset{?}{\in} \mathcal{B}d(T,v), T \overset{?}{\vdash} w_1, \ldots, T \overset{?}{\vdash} w_n\}$ avec $\mathtt{sign}(w,u) \in st(T)$, $y \in \mathsf{Var}(w)$ et $w_1, \ldots, w_n \in st(w)$. Puisque $w \in st(T)$, on a par origination $T_y \subsetneq T$ et on peut conclure.

- règle $\mathsf{R_5}$. Dans ce cas, on a $\mathcal{C}' = (\mathcal{C} \setminus \{x \overset{?}{\in} \mathcal{B}d(T',v)\}) \cup \{T \overset{?}{\vdash} v,\ x \overset{?}{\in} \mathcal{B}d(T,v)\}$, avec $T \subsetneq T'$. Clairement, on a $\mathcal{C}'^{\preceq}_{\mathsf{Var}(T,v)} \models (T \overset{?}{\vdash} v)$.

- règle $\mathsf{R_6}$. Dans ce cas, on a $\mathcal{C}' = (\mathcal{C} \setminus \{x \overset{?}{\in} \mathcal{B}d(T,v')\}) \cup \{T \overset{?}{\vdash} w,\ x \overset{?}{\in} \mathcal{B}d(T,w),\ y \overset{?}{\in} \mathcal{B}d(T',v), T' \overset{?}{\vdash} w_1, \ldots, T' \overset{?}{\vdash} w_n\}$, avec $w \in st(T)$, $y \in \mathsf{Var}(v)$ et $w_1, \ldots, w_n \in st(w)$. Puisque, d'une part, $T \overset{?}{\vdash} w \in \mathcal{C}'$ et, d'autre part, $T_y \subsetneq T$ (par origination), on peut conclure.

- règle $\mathsf{R_7}$. Dans ce cas, si la nouvelle contrainte d'appartenance ne disparaît pas, on a $\mathcal{C}' = (\mathcal{C} \setminus \{x \overset{?}{\in} \mathcal{B}d(T,v')\}) \cup \{y \in \mathcal{B}d(T,v'), T \overset{?}{\vdash} v_1, \ldots, T \overset{?}{\vdash} v_n\}$, avec $y \in \mathsf{Var}(v)$ et $v_1, \ldots, v_n \in st(v)$. En utilisant $T_x = T$ et la bonne-formation de $x \overset{?}{\in} \mathcal{B}d(T,v')$ dans $\mathcal{C}$, on en déduit que $\mathcal{C}^{\preceq}_{\overline{V'}} \models (T \overset{?}{\vdash} v')$, où $V' = \mathsf{Var}(T \cup \{v'\})$. En conséquence, grâce au lemme 67, on a $\mathcal{C}'^{\preceq}_{V'} \models (T \overset{?}{\vdash} v')$ et on peut conclure. □

12.4 Résultat principal et discussion

Le dernier élément qui nous manque pour avoir une procédure de décision est un argument de terminaison des règles. Cependant, elles ne terminent pas:

Exemple 69 *Considérons le système de contraintes suivant:*

$$T = c, x, y, \mathtt{sign}(\mathtt{blind}(x, \mathtt{sign}(y,k)),k),\ \mathtt{sign}(\mathtt{blind}(y, \mathtt{sign}(x,k)),k) \quad \begin{array}{rcl} c & \overset{?}{\vdash} & x \\ c & \overset{?}{\vdash} & y \\ & \overset{?}{\vdash} & \mathtt{sign}(x,k) \end{array}$$

où c, k sont des constantes et, pour toute solution possible, k ne peut pas être déduite. La seule manière pour obtenir une solution (non-confondante) est d'utiliser $(\mathsf{UB_2})$ *pour déduire $\mathtt{sign}(x,k)\sigma$ dans la troisième contrainte. Ceci nous donne (par* $\mathsf{R_4}$*):*

$$\begin{array}{rcl} c & \overset{?}{\vdash} & x \\ c & \overset{?}{\vdash} & y \\ T & \overset{?}{\vdash} & \mathtt{sign}(\mathtt{blind}(x, \mathtt{sign}(y,k)),k) \\ \mathtt{blind}(x, \mathtt{sign}(y,k)) & \overset{?}{\in} & \mathcal{B}d(T,x) \end{array}$$

Par Axiom *et* S_1*, nous revenons au système de départ, dans lequel les variables* x *et* y *ont été interchangées.*

La non-terminaison pourrait être évitée avec un contrôle minutieux de l'application des règles. Cependant, ceci n'est pas nécéssaire, grâce à notre résultat de complétude plus précis. En effet, on montre non-seulement qu'une forme résolue peut être atteinte, mais que sur le chemin qui nous mène vers elle la mesure PS decroît à chaque pas. Maintenant, si on a une derivation infinie, comme on n'introduit pas de nouveaux sous-termes, on doit avoir une boucle. Mais alors on a une partie du chemin sur laquelle la mesure PS n'a pas decru: toutes les formes résolues peuvent être atteintes en ne considérant que des chemins qui ne bouclent pas, donc que des chemins finis.

Procédure de décision pour la satisfaisabilité. Soit $\mathcal{C}_0$ un système de contraintes.

1. On devine les égalités entre les sous-termes de $\mathcal{C}_0$; soit $\mathcal{C}_1, \ldots, \mathcal{C}_k$ les systèmes de contraintes ainsi obtenus.

2. on applique les règles de transformation à chaque $\mathcal{C}_i$ jusqu'à obtenir une forme résolue ou bien un système de contraintes qui a été déjà considéré par la procédure; pour chaque $\mathcal{C}_i$, soit $\mathcal{C}_i^1, \ldots \mathcal{C}_i^{p_i}$ les formes résolues atteignables à partir de $\mathcal{C}_i$.

3. Si une forme résolue est atteinte, $\mathcal{C}_0$ est satisfaisable, sinon, non.

Théorème 8 *Soit* $\mathcal{C}_0$ *un système de contraintes. La procédure décrite ci-dessus est correcte et complète.*

Preuve: C'est une conséquence directe des lemmes 59, 62, 63, 68 et du corollaire 7. □

Même plus, nous calculons un ensemble fini de formes résolues qui représente symboliquement toutes les solutions de $\mathcal{C}$. Comme expliqué dans [CLCZ10], c'est une propriété qui permet de décider d'autres propriétés de trace. On étend ainsi un résultat de [BC08], qui montre déjà que le problème du secret dans un nombre borné de sessions est décidable pour les signatures en aveugle. La nouveauté de notre approche est aussi méthodologique, en nous rattachant à la notion classique de localité et à l'algorithme général que nous avons présenté dans le chapitre précédent. Ceci se fait à travers l'introduction d'un nouveau prédicat pour couper le branchement infini, qui est inhérent pour certaines théories saturées.

Nous suggérons maintenant que cette construction s'adapte facilement à d'autres théories saturées infinies.

Chiffrement homomorphique. Nous enrichissons la syntaxe de contraintes avec un prédicat $u \overset{?}{\in} P(v)$. Les solutions de tels contraintes sont des substitutions σ telles que ou bien $u\sigma = v\sigma$ ou bien il existe u_1, u_2 tels que $u\sigma = \langle u_1, u_2 \rangle$ et σ est une solution de $u_1 \overset{?}{\in} P(v)$ ou de $u_2 \overset{?}{\in} P(v)$.

$$
\begin{array}{rcl}
f(u_1,\ldots,u_n) \overset{?}{\in} P(u) & \rightarrow & f(u_1,\ldots,u_n) = u \\
 & & \text{si } f \neq \langle\cdot,\cdot\rangle \\
u_1 \overset{?}{\in} P(v_1[u_2]) \wedge \ldots \wedge u_n \overset{?}{\in} P(v_n[u_1]) & \rightarrow & u_1 = v_1[u_2] \wedge \ldots \wedge u_n = v_n[u_1] \\
\langle u, v\rangle \overset{?}{\in} P(w) & \rightarrow & \langle u, v\rangle = w \vee u \overset{?}{\in} P(w) \vee v \overset{?}{\in} P(w) \\
H \overset{?}{\vdash} x \wedge x \overset{?}{\in} P(u) & \rightsquigarrow & H \overset{?}{\vdash} x \wedge x \overset{?}{\in} P(u) \wedge H \overset{?}{\vdash} u \\
H \overset{?}{\vdash} \mathtt{enc}(v, u, r) & \rightsquigarrow & H \overset{?}{\vdash} \mathtt{enc}(v', w, r') \wedge w \overset{?}{\in} P(u) \wedge v = v' \wedge r = r' \\
 & & \text{si } \mathtt{enc}(v', w, r') \in st(H)
\end{array}
$$

Figure 12.4: Règles de transformation

Des règles de simplification et de transformation spécifiques gèrent le nouveau prédicat, dans la figure 12.4.

La relation $x \overset{?}{\in} P(u)$ définit encore une fois un ordre entre les variables: $y \leq_D x$ si $y \in \mathsf{Var}(u)$ et $x \overset{?}{\in} P(u)$ est dans D. Cet ordre nous permet de construire une solution pour une forme résolue bien-formée:

Definition 43 *Une* forme résolue D *est un système de contraintes de la forme*

$$
\begin{array}{c}
x_1 = t_1 \wedge \ldots \wedge x_n = t_n \wedge H_1 \overset{?}{\vdash} y_1 \wedge \ldots \wedge H_m \overset{?}{\vdash} y_m \\
\wedge z_1 \overset{?}{\in} P(u_1) \wedge \ldots \wedge z_k \overset{?}{\in} P(u_k)
\end{array}
$$

avec

- $x_1, \ldots, x_n$ *des variables qui apparaissent une seule fois dans le système,*
- $y_1, \ldots, y_m$ *des variables distinctes,*
- $\{z_1, \ldots, z_k\} \subseteq \{y_1, \ldots, y_m\}$ *et, si* $z_i = y_j$*, alors* $D_{H_{y_j} \cup \{u_i\}} \models H_{y_j} \overset{?}{\vdash} u_i$

Ainsi, le cas des signatures en aveugle n'est pas atypique et il pose les bases pour un travail futur important: sa généralisation.

Chapitre 13

Conclusion et travaux futurs

Pour les théories locales, nos règles de transformation associent à un système de contraintes $\mathcal{C}$ un ensemble de formes résolues, qui représente symboliquement toutes les solutions de $\mathcal{C}$.

Un premier travail futur consisterait à lever les restrictions syntaxiques posées sur la théorie dans la section 10.2.2. Ceci permettrait le traitement de la théorie du rechiffrement dans le cadre presenté ici.

D'autres travaux futurs, plus ambitieux, sont esquissées dans la suite.

13.1 Théories saturées infinies

La question principale ouverte par le chapitre 12 est la généralisation du prédicat qui coupe le branchement infini des théories saturées. Ceci nous permettrait un traitement systématique des théories de l'intrus, dans un aller-retour entre la théorie de départ, le système saturé infini (implicite) et le système saturé fini (explicite) grâce au nouveau prédicat.

Est-il possible de mettre en place une construction générale pour couper le branchement infini des théories saturées, qui aurait les signatures en aveugle et le chiffrement homomorphique comme des cas particuliers d'application ?

13.2 Propriétés d'équivalence

Les propriétés d'équivalence [Bau07, CD09] demandent le même ensemble de preuves pour deux systèmes de contraintes $\mathcal{C}, \mathcal{C}'$. C'est un des problèmes ouverts les plus stimulants pour l'analyse formelle des protocoles de sécurité. Si on considère les arbres $\mathcal{C} \rightsquigarrow \mathcal{C}_1, \ldots, \mathcal{C}_n$ et $\mathcal{C}' \rightsquigarrow \mathcal{C}_1, \ldots, \mathcal{C}_m$ construits par nos règles de transformation, peut-on traduire l'équivalence de $\mathcal{C}$ et $\mathcal{C}'$ comme une relation de bisimulation entre ces deux arbres ?

13.3 Symboles AC

Il y a maintes travaux qui considèrent déjà des symboles AC, mais la plupart dans le cadre de théories particulières. De plus, c'est l'existence d'une solution qui est presque toujours decidée, sans avoir une représentation de toutes les solutions. En l'occurence, aucun résultat existe ne permet de décider l'équivalence de systèmes de contraintes en présence de symboles AC. Même si on peut voir une certaine similarité avec les systèmes saturés infinis, l'étude de la recherche de preuve pour les systèmes d'inférence avec des symboles AC est pour l'instant le travail futur le plus ambitieux.

Chapitre 14

Conclusion et perspectives

Les deux parties de cette thèse sont fortement liées. On a vu d'abord qu'il y a un passage logique entre la première et la deuxième partie, pour traiter des théories non-décomposables et étudier des propriétés plus générales que l'existence d'une solution.

Cependant, des liens plus profonds unissent les deux parties: la localité, les variants finis, les sous-termes sémantiques dont le pendant dans la deuxième partie est la saturation ... Pour cela, nous pensons qu'elles sont des instances d'un résultat encore plus général (notons comment le chiffrement homomorphique surgit dans les deux parties). Quelques pas vers sa recherche et des perspectives pour la résolution de systèmes de contraintes sont esquissés dans les conclusions de chacune de parties.

Nos résultats s'encadrent aussi dans des perspectives plus générales.

Combinaison. Les principes de la combinaison sont universels: identifier les éléments de base d'un système et les lois qui régissent leur interaction.

Correction calculatoire. Maintes travaux, e.g. [CLC08, CKKW06, CW05], ratachent la sécurité logique à une notion de sécurité plus forte, où les actions des agents sont modélisées par des calculs de machines de Turing. Si deux modèles logiques sont corrects, est'ce-que leur combinaison l'est?

Protocoles. De plus en plus, les protocoles sont éxécutés ensemble, en interdependance. Par exemple, google permet de lire le mail, effectuer des achats, publier des contenus, etc, à partir d'un seul compte. Quel est l'effet de la combinaison de protocoles sur leur sécurité? Peut-on utiliser sans dommage le même mot de passe pour deux services?

Propriétés. Si un protocole satisfait la propriété A et un autre satisfait la propritété B, peut-on construire un protocole qui satisfait $A \wedge B$? Plus généralement, sur quelles conditions la conjonction $A_1 \wedge \ldots \wedge A_1$ est réalisable? Par exemple, c'est une question pertinente pour les protocoles de vote électronique, où l'existence de protocoles pour assurer à la fois l'anonymat,

l'absence de coercion et la vérifiabilité de son vote par un votant n'est pas évidente.

Aspects. Les systèmes informatiques sont concurents (réseaux de petri) en temps (automates temporisées) [Emo], utilisent des listes d'entiers [YKB02], melangent horloges et compteurs [BFS09]. Peut-on vérifier de manière hiérarchique la combinaison de ces aspects, ou est-on obligés à faire les raisonements directement dans le modèle complexe qui les unifie?

Localité et formes résolues. Les principes de la localité sont universels: toute solution efficace d'un problème passe par des éléments qui lui sont intrinseques, même si souvent elle lui rajoute des constructions théoriques.

Correction calculatoire. Plus riche le modèle formel, plus de chances d'avoir la correction calculatoire. Y a-t-il alors une contradiction entre la correction calculatoire et la localité? Si oui, comment concilier les deux, pour avoir à la fois des fortes garanties et des preuves automatiques de sécurité?

Protocoles. Le calcul des formes résolues nous pointe non seulement l'existence d'une attaque, mais aussi les actions de l'intrus qui y mènent, sa cause. Serait-il possible de retourner en arrière sur cette trace et corriger (semi-) automatiquement le protocole?

Propriétés. Y a-t-il une hiérarchie des propriétés? Étant donné une propriété, les executions qui la satisfont sont'elles "locales"? Est'ce qu'une propriété peut être simplifiée, reduite à des propriétés résolues?

Aspects. Pour voir le rôle de la localité dans d'autres aspects de la vérification, citons par exemple [SS06, JSS07, IJSS08]. Ces travaux et le notre sont liés, par un fil qui reste à découvrir.

Bibliography

[ACD07] Mathilde Arnaud, Véronique Cortier, and Stéphanie Delaune. Combining algorithms for deciding knowledge in security protocols. In Franck Wolter, editor, *Proceedings of the 6th International Symposium on Frontiers of Combining Systems (FroCoS'07)*, volume 4720 of *Lecture Notes in Artificial Intelligence*, pages 103–117, Liverpool, UK, September 2007. Springer.

[AL00] Roberto M. Amadio and Denis Lugiez. On the reachability problem in cryptographic protocols. In Catuscia Palamidessi, editor, *CONCUR*, volume 1877 of *Lecture Notes in Computer Science*, pages 380–394. Springer, 2000.

[ANR07] Siva Anantharaman, Paliath Narendran, and Michaël Rusinowitch. Intruders with caps. In *Proc. 18th Int. Conf. on Rewriting Techniques and Applications (RTA)*, pages 20–35, 2007.

[Bau07] Mathieu Baudet. *Sécurité des protocoles cryptographiques : aspects logiques et calculatoires*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, January 2007.

[BC08] Vincent Bernat and Hubert Comon-Lundh. Normal proofs in intruder theories. In *Revised Selected Papers of the 11th Asian Computing Science Conference (ASIAN'06)*, volume 4435 of *Lecture Notes in Computer Science*, pages 151–166. Springer, 2008.

[BC09] Sergiu Bursuc and Hubert Comon-Lundh. Protocol security and algebraic properties: decision results for a bounded number of sessions. In Ralf Treinen, editor, *Proceedings of the 20th International Conference on Rewriting Techniques and Applications (RTA'09)*, volume 5595 of *Lecture Notes in Computer Science*, pages 133–147, Brasília, Brazil, June-July 2009. Springer.

[BCD07a] Sergiu Bursuc, Hubert Comon-Lundh, and Stéphanie Delaune. Associative-commutative deducibility constraints. In Wolfgang Thomas and Pascal Weil, editors, *Proceedings of the 24th Annual Symposium on Theoretical Aspects of Computer Science*

(STACS'07), volume 4393 of *Lecture Notes in Computer Science*, pages 634–645, Aachen, Germany, February 2007. Springer.

[BCD07b] Sergiu Bursuc, Hubert Comon-Lundh, and Stéphanie Delaune. Deducibility constraints, equational theory and electronic money. In Hubert Comon-Lundh, Claude Kirchner, and Hélène Kirchner, editors, *Rewriting, Computation and Proof — Essays Dedicated to Jean-Pierre Jouannaud on the Occasion of his 60th Birthday*, volume 4600 of *Lecture Notes in Computer Science*, pages 196–212, Cachan, France, 2007. Springer.

[BDC09] Sergiu Bursuc, Stéphanie Delaune, and Hubert Comon-Lundh. Deducibility constraints. In Anupam Datta, editor, *Proceedings of the 13th Asian Computing Science Conference (ASIAN'09)*, volume 5913 of *Lecture Notes in Computer Science*, pages 24–38, Seoul, Korea, December 2009. Springer.

[Ber07a] Gerard Berry. A la chasse aux bugs, la vérification des programmes et circuits. Chaire d'Innovation technologique - Liliane Bettencourt, College de France, 2007. `http://www.college-de-france.fr/default/EN/all/inn_tec2007/index.htm`.

[Ber07b] Gerard Berry. Pourquoi et comment le monde devient numérique. Chaire d'Innovation technologique - Liliane Bettencourt, College de France, 2007. `http://www.college-de-france.fr/default/EN/all/inn_tec2007/index.htm`.

[BFS09] Florent Bouchy, Alain Finkel, and Arnaud Sangnier. Reachability in timed counter systems. In Peter Habermehl and Tomáš Vojnar, editors, *Joint Proceedings of the 8th, 9th and 10th International Workshops on Verification of Infinite State Systems (INFINITY'06,'07,'08)*, volume 239 of *Electronic Notes in Theoretical Computer Science*, pages 167–178. Elsevier Science Publishers, July 2009.

[BG01] David Basin and Harald Ganzinger. Automated complexity analysis based on ordered resolution. *Journal of the Association of Computing Machinery*, 48(1):70–109, January 2001.

[BGW01] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: the insecurity of 802.11. In *MOBICOM*, pages 180–189, 2001.

[Bor01] Michele Boreale. Symbolic trace analysis of cryptographic protocols. In Fernando Orejas, Paul G. Spirakis, and Jan van Leeuwen, editors, *ICALP*, volume 2076 of *Lecture Notes in Computer Science*, pages 667–681. Springer, 2001.

[BS96] Franz Baader and Klaus U. Schulz. Unification in the union of disjoint equational theories: Combining decision procedures. *Journal of Symbolic Computation*, 21(2):211–243, 1996.

[CD94] Evelyne Contejean and Hervé Devie. An efficient incremental algorithm for solving systems of linear diophantine equations. *Information and Computation*, 113(1):143–172, 1994.

[CD05] Hubert Comon-Lundh and Stéphanie Delaune. The finite variant property: How to get rid of some algebraic properties. In Jürgen Giesl, editor, *Proceedings of the 16th International Conference on Rewriting Techniques and Applications (RTA'05)*, volume 3467 of *Lecture Notes in Computer Science*, pages 294–307, Nara, Japan, April 2005. Springer.

[CD09] Véronique Cortier and Stéphanie Delaune. A method for proving observational equivalence. In *Proceedings of the 22nd IEEE Computer Security Foundations Symposium (CSF'09)*, pages 266–276, Port Jefferson, NY, USA, July 2009. IEEE Computer Society Press.

[CDL06] Véronique Cortier, Stéphanie Delaune, and Pascal Lafourcade. A survey of algebraic properties used in cryptographic protocols. *Journal of Computer Security*, 14(1):1–43, 2006.

[CJ97] John Clark and Jeremy Jacob. A survey of authentication protocol. literature: Version 1.0. `http://www.cs.york.ac.uk/~jac/papers/drareview.ps.gz`, 1997.

[CK07] Yannick Chevalier and Mounira Kourjieh. Key substitution in the symbolic analysis of cryptographic protocols. In Vikraman Arvind and Sanjiva Prasad, editors, *FSTTCS*, volume 4855 of *Lecture Notes in Computer Science*, pages 121–132. Springer, 2007.

[CKKW06] Véronique Cortier, Steve Kremer, Ralf Küsters, and Bogdan Warinschi. Computationally sound symbolic secrecy in the presence of hash functions. In *Foundations of Software Technology and Theoretical Computer Science (FSTTCS'06)*, volume 4337 of *Lecture Notes in Computer Science*, pages 176–187, 2006.

[CKR+03] Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, Mathieu Turuani, and Laurent Vigneron. Extending the Dolev-Yao intruder for analyzing an unbounded number of sessions. In *Computer Science Logic*, volume 2803 of *Lecture Notes in Computer Science*, pages 128–141. Springer, 2003.

[CKRT03a] Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. Deciding the security of protocols with Diffie-Hellman exponentiation and products in exponents. In J. Radhakrishnan and P.K. Pandya, editors, *Proc. FST/TCS, Mumbai*, volume 2914 of *Lecture Notes in Computer Science*, pages 124–135, 2003.

[CKRT03b] Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. An NP decision procedure for protocol insecurity with xor. In Kolaitis [Kol03], pages 261–270.

[CKRT05] Yannick Chevalier, Ralf Küsters, Michaël Rusinowitch, and Mathieu Turuani. Deciding the security of protocols with commuting public key encryption. *Electr. Notes Theor. Comput. Sci.*, 125(1):55–66, 2005.

[CLC03] Hubert Comon-Lundh and Véronique Cortier. New decidability results for fragments of first-order logic and application to cryptographic protocols. In *14th Int. Conf. Rewriting Techniques and Applications (RTA'2003), Valencia, Spain*, volume 2706 of *Lecture Notes in Computer Science*, pages 148–164. Springer, Jun. 2003.

[CLC08] Hubert Comon-Lundh and Véronique Cortier. Computational soundness of observational equivalence. In *ACM Conference on Computer and Communications Security*, pages 109–118, 2008.

[CLCZ10] Hubert Comon-Lundh, Véronique Cortier, and Eugen Zalinescu. Deciding security properties for cryptographic protocols. application to key cycles. *ACM Trans. Comput. Log.*, 11(2), 2010.

[CLS02] Hubert Comon-Lundh and Vitaly Shmatikov. Is it possible to decide whether a cryptographic protocol is secure or not ? *Journal of Telecommunications and Information Technology*, 4, 2002.

[CLS03] Hubert Comon-Lundh and Vitaly Shmatikov. Intruder deductions, constraint solving and insecurity decision in preence of exclusive or. In Kolaitis [Kol03].

[CLT03] Hubert Comon-Lundh and Ralf Treinen. Easy intruder deductions. In *Verification: Theory and Practice, Essays Dedicated to Zohar Manna on the Occasion of His 64th Birthday*, volume 2772 of *Lecture Notes in Computer Science*, pages 225–242. Springer-Verlag, 2003.

[CM96] Evelyne Contejean and Claude Marché. Cime: Completion modulo E. In *Proc. Rewriting Techniques and Applications*, volume 1103 of *Lecture Notes in Computer Science*, pages 416–419, 1996.

[CR05] Yannick Chevalier and Michaël Rusinowitch. Combining Intruder Theories. In Luís Caires, Giuseppe F. Italiano, Luís Monteiro, Catuscia Palamidessi, and Moti Yung, editors, *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings*, volume 3580 of *Lecture Notes in Computer Science*, pages 639–651. Springer, 2005. Available at `http://dx.doi.org/10.1007/11523468_52`.

[CR06] Yannick Chevalier and Michaël Rusinowitch. Hierarchical combination of intruder theories. In *Proc. Rewriting Techniques and Application*, volume 4098 of *Lecture Notes in Computer Science*, pages 108–122, 2006.

[CW05] V. Cortier and B. Warinschi. Computationally sound, automated proofs for security protocols. In *Proc. 14th European Symposium on Programming (ESOP'05)*, volume 3444 of *Lecture Notes in Computer Science*, pages 157–171, 2005.

[DEK82] Danny Dolev, Shimon Even, and Richard M. Karp. On the security of ping-pong protocols. In *CRYPTO*, pages 177–186, 1982.

[Del06] Stéphanie Delaune. *Vérification des protocoles cryptographiques et propriétés algébriques*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2006.

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 243–309. North-Holland, 1990.

[DLLT06] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis in presence of a homomorphism operator and *exclusive or*. In Michele Buglesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming (ICALP'06) — Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 132–143, Venice, Italy, July 2006. Springer.

[DLLT08] Stéphanie Delaune, Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Symbolic protocol analysis for monoidal equational theories. *Information and Computation*, 206(2-4):312–351, February-April 2008.

[DY83] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–207, 1983.

[Emo] Emoticon : Equivalences between models with time and concurrency. Projet de l'institut Farman de l'ENS Cachan avec la participation de LSV (Laboratoire Spécification et Vérification) et LURPA (Laboratoire Universitaire de Recherche en Production Automatisée).

[EMS08] Santiago Escobar, José Meseguer, and Ralf Sasse. Effectively checking the finite variant property. In *Proc. Rewriting Techniques and Applications*, pages 79–93, 2008.

[Hui99] Antti Huima. Efficient infinite-state analysis of security protocols. FLOC Workshop on Formal Methods in Security Protocols, 1999.

[IJSS08] Carsten Ihlemann, Swen Jacobs, and Viorica Sofronie-Stokkermans. On local reasoning in verification. In C. R. Ramakrishnan and Jakob Rehof, editors, *TACAS*, volume 4963 of *Lecture Notes in Computer Science*, pages 265–281. Springer, 2008.

[JK86] Jean-Pierre Jouannaud and Hélène Kirchner. Completion of a set of rules modulo a set of equations. *SIAM J. Comput.*, 15(4):1155–1194, 1986.

[JSS07] Swen Jacobs and Viorica Sofronie-Stokkermans. Applications of hierarchical reasoning in the verification of complex systems. *Electr. Notes Theor. Comput. Sci.*, 174(8):39–54, 2007.

[KN92] Deepak Kapur and Paliath Narendran. Double-exponential complexity of computing a complete set of AC-unifiers. In *Logic in Computer Science*, pages 11–21. IEEE Computer Society, 1992.

[Kol03] P. Kolaitis, editor. *Eighteenth Annual IEEE Symposium on Logic in Computer Science*, Ottawa, Canada, June 2003. IEEE Computer Society.

[Laf06] Pascal Lafourcade. *Vérification des protocoles cryptographiques en présence de théories équationnelles*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, September 2006. 209 pages.

[LLT07] Pascal Lafourcade, Denis Lugiez, and Ralf Treinen. Intruder deduction for the equational theory of abelian groups with distributive encryption. *Information and Computation*, 205(4):581–623, 2007.

[McA93] David McAllester. Automatic recognition of tractability in inference relations. *Journal of the ACM*, 40(2), 1993.

[MDL+99] Durgin Lincoln Mitchell, N. A. Durgin, P. D. Lincoln, J. C. Mitchell, and A. Scedrov. Undecidability of bounded security protocols, 1999.

[MS01] Jonathan K. Millen and Vitaly Shmatikov. Constraint solving for bounded-process cryptographic protocol analysis. In *Proc. 8th ACM Conference on Computer and Communications Security*, 2001.

[MS05] Jonathan K. Millen and Vitaly Shmatikov. Symbolic protocol analysis with an abelian group operator or diffie-hellman exponentiation. *Journal of Computer Security*, 13(3):515–564, 2005.

[Nar96] Paliath Narendran. Solving linear equations over polynomial semirings. In *Proc. LICS 1996*, Lecture Notes in Computer Science, pages 466–472, 1996.

[RT01] Michaël Rusinowitch and Mathieu Turuani. Protocol insecurity with finite number of sessions is NP-complete. In *Proc.14th IEEE Computer Security Foundations Workshop*, Cape Breton, Nova Scotia, June 2001.

[RT03] Michaël Rusinowitch and Mathieu Turuani. Protocol insecurity with a finite number of sessions, composed keys is NP-complete. *Theor. Comput. Sci.*, 1-3(299):451–475, 2003.

[Shm04] Vitaly Shmatikov. Decidable analysis of cryptographic protocols with products and modular exponentiation. In *Proc. European Symposium on Programming (ESOP'04)*, volume 2986 of *Lecture Notes in Computer Science*. Springer-Verlag, 2004.

[Sim94] Gustavus J. Simmons. Cryptoanalysis and protocol failures. *Commun. ACM*, 37(11):56–65, 1994.

[SS06] Viorica Sofronie-Stokkermans. Interpolation in local theory extensions. In Ulrich Furbach and Natarajan Shankar, editors, *IJCAR*, volume 4130 of *Lecture Notes in Computer Science*, pages 235–250. Springer, 2006.

[YKB02] Tuba Yavuz-Kahveci and Tevfik Bultan. Automated verification of concurrent linked lists with counters. In Manuel V. Hermenegildo and Germán Puebla, editors, *SAS*, volume 2477 of *Lecture Notes in Computer Science*, pages 69–84. Springer, 2002.