Contraintes de déductibilité modulo Associativité-Commutativité

Sergiu Bursuc

Septembre, 2006

Mémoire de stage de MPRI
Encadré par Hubert Comon-Lundh

Ce travail a été effectué au LSV: Laboratoire Spécification et Vérification, Cachan, sous la direction de Hubert Comon-Lundh et avec la collaboration de Stéphanie Delaune.

1 Introduction

Les protocoles cryptographiques sont des programmes dont le but est d'établir une communication sûre entre plusieurs agents. Ils sont de plus en plus utilisés pour sécuriser les communications sur internet: des applications comme le commerce électronique, les services bancaires, les services audio-visuels, le vote électronique et beaucoup d'autres ont besoin de protocoles de sécurité pour assurer leur bon fonctionnement. Citons également comme application le porte-monnaie électronique, sur lequel nous reviendrons plus loin.

Cette importance des protocoles cryptographiques dans des applications critiques pose le problème de vérifier leur correction. En fait, même si une éxécution normale du protocole satisfait par conception les propriétés de sécurité, une attaque est possible à cause de la non-sécurité du réseau: un intrus peut intercepter tous les messages et faire éxécuter le protocole d'une manière imprévue.

On a 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 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. Le but est de concevoir des méthodes génériques s'appliquant à une classe de protocoles, en s'appuyant sur des techniques de démonstration automatique.

Une première approche de la vérification automatique consiste à utiliser 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. Le problème de sécurité d'un protocole se définit formellement dans ce modèle(parfois dit "de Dolev-Yao") et la question principale est sa décidabilité. En général, la préservation du secret d'un protocole(i.e. absence d'un attaque) est un problème indécidable. Si on borne le nombre de sessions, le problème de recherche d'une attaque devient co-NP-complet [RT03]. Même pour un nombre non-borné de sessions on obtient des résultats de décidabilité dans certains cas particuliers [CLC03, CKR$^+$03].

Mais cette approche est basée sur l'hypothèse du chiffrement parfait: 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 d'un intrus et lui permettent d'obtenir beaucoup plus d'informations que seulement la théorie classique de Dolev-Yao: des attaques impossibles dans le premier modèle 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].

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 [CKRT04], "ou" exclusif [CLS03, CKRT03], groupe abélien [MS05], théories avec un symbole d'homomorphisme [DLLT06, Del06] ... Par contre, il n'existe aucune approche générale pouvant traiter une classe pertinente de théories. Un premier pas vers une méthode systématique pour prendre en compte les propriétés algébriques a été proposée par H.Comon-Lundh et S.Delaune dans [CLD05]. Ce résultat permet de réduire un grand nombre de théories pertinentes à la théorie Associativité+Commutativité(AC) ou à la théorie vide.

La vérification des protocoles (i.e. la résolution de systèmes des contraintes) en présence de la théorie équationelle AC est donc un élément important dans cette approche. A l'heure actuelle, aucun résultat n'existe sur ce problème. Nous présentons ici un résultat de décidabilité pour un cas particulier de systèmes de contraintes bien-formés, ainsi qu'un résultat d'indécidabilité pour les systèmes de contraintes généraux (non bien-formés). Le problème reste ouvert pour le cas de systèmes de contraintes bien-formés généraux. Pour ce cas, nous montrons que si on ajoute une règle(justifiable) dans le système de déduction de l'intrus, on obtient un algoritme de décision.

Un exemple: le porte-monnaie électronique. Pour illustrer le besoin de méthodes génériques et la pertinence de l'approche décrite ici, on peut considérer un protocole de commerce électronique, développé récemment par une équipe de France Telecom R&D et modélisé par S.Delaune [Del06]. Il s'avère que la théorie équationelle nécessaire pour modèliser ce protocole ne rentre dans aucun des cas traités auparavant: il n'est pas possible d'utiliser les résultats et logiciels existants. Pour vérifier ce protocole, il faut donc soit obtenir un résultat particulier à la théorie équationelle en cause, soit appliquer un résultat générique. Heureusement, l'approche introduite dans [CLD05] est applicable ici, car cette théorie satisfait une propriété dite de variants finis [Del06]. En suivant cette voie, on pourrait donc établir non seulement un résultat pour vérifier le protocole de France Telecom, mais aussi une première application d'une méthode générique pour traiter les propriétés algèbriques.

Plan du rapport. Dans le chapitre 2 nous rappelons la modélisation du problème de recherche d'attaque comme une problème de satisfaisabilité de systèmes de contraintes bien-formés. Dans le chapitre 3 nous présentons nos résultats sur la résolution de systèmes de contraintes modulo AC - une théorie essentielle pour traiter uniformement les propriétés algébriques. Dans la section 3.2 nous prouvons l'indécidabilité de ce problème dans le cas de systèmes de

contraintes généraux. Dans la section 3.3 nous montrons que pour un cas particulier de systèmes de contraintes bien-formés la satisfaisabilité est décidable. Le cas de systèmes de contraintes bien-formés généraux est montré décidable pour un modèle de l'intrus étendu, mais réaliste, comme expliqué dans la section 3.4. Finalement, dans le chapitre 4 nous discutons le plan de recherche à suivre après avoir considéré le cas de systèmes de contraintes modulo AC.

2 Recherche d'attaque et contraintes de déductibilité

Nous rappelons ici la modélisation du problème de recherche d'attaque, à la suite de nombreux auteurs [RT03, MS03, CLS03, DLLT06]. Plus de détails peuvent être trouvés dans [Del06].

La modélisation que nous présentons ici utilise des *rôles* pour représenter les actions des différents acteurs du protocole. Un rôle est une séquence d'instructions effectuées par un agent lors d'une exécution du protocole. Chaque instruction représente une action élémentaire qui associe à un message reçu par un agent la réponse correspondante émise par ce même agent: $recv(u); send(v)$.

Les protocoles utilisent souvent la génération de nombres aléatoires (nonces) ou des "tickets", parties de messages que ne peut pas analyser l'agent qui les reçoit. C'est à cet endroit qu'un intrus peut substituer, par exemple une nonce, par un message de sa fabrication et faire accepter le message comme s'il était de la forme prévue.

Pour modéliser les parties non analysables d'un message on utilise des variables, qui pourront par la suite être remplacées par des messages arbitraires: les messages sont des termes avec variables.

Un protocole est spécifié par un ensemble de rôles, où chaque rôle est une séquence d'instructions. Plus précisément:

Definition 2.1 (Rôle) Un rôle est la donnée:
- d'un ensemble de paramètres, noté $z_1, \ldots, z_p$,
- d'un ensemble de nonces, noté $\overline{n}$, et
- d'une séquence d'instructions de la forme $recv(u_i); send(u_i)$.

On le note $R(z_1, \ldots, z_p) = \nu\overline{n}.recv(u_1); send(v_1); \ldots; recv(u_k); send(v_k)$.

Les paramètres représentent des informations fixées avant l'éxecution du protocole (e.g. les noms des agents).

Exemple. Considérons le protocole suivant. On a deux acteurs. Le premier (A) initie la communication en envoyant à B le couple formé de son nom et d'un nombre aléatoire N_a, qu'il a construit. En recevant ce message, B repond par le triplet formé de son nom, N_a et un nombre aléatoire N_b. Finalement, A envoie une confirmation cfm. Ce protocole est modélisé par:

$$A(z_a, z_b) = \nu N_a. \begin{cases} ; send(< z_a, N_a >) \\ recv(< z_b, N_a, x >); send(cfm) \end{cases}$$
$$B(z_a, z_b) = \nu N_b. \begin{cases} recv(< z_a, y >); send(< z_b, y, N_b >) \\ recv(cfm); \end{cases}$$

On voit que, puisque B ne connait pas le nombre que A vas lui envoyer, il le represente par la variable y. Ainsi, A abstrait le nombre qu'il attend par x.

On voit aussi que les termes peuvent contenir des variables (x, y), des symboles publics (cfm), des paramètres (z_a, z_b) et des nonces (N_a, N_b).

Nous nous interessons au problème de la sécurité d'un protocole pour un nombre borné de sessions, c'est-à-dire en présence d'un ensemble fini d'instances de rôles.

Definition 2.2 (instance d'un rôle) Soit R un rôle ayant pour paramètres $z_1, \ldots, z_p$, pour ensemble de nonces $n_1, \ldots, n_k$ et pour séquence d'instructions S. L'instance $R(q_1, \ldots, q_p)$ du rôle R est la séquence d'instructions $S\sigma$ où σ est la substitution définie par $\{z_1 \mapsto q_1, \ldots, z_p \mapsto q_p\}$.

L'intrus. L'intrus est representé par un système de déduction, noté I, lui permettant de construire des termes nouveaux à partir de termes qu'il connait. Par exemple, le modèle de l'intrus de Dolev-Yao est représenté par les règles:

$$(Proj1)\frac{T \vdash < u, v >}{t \vdash u} \qquad (Proj2)\frac{T \vdash < u, v >}{t \vdash v} \qquad (Dec)\frac{T \vdash \{u\}_v \qquad T \vdash v}{T \vdash u}$$

$$(Comp)\frac{T \vdash u_1 \qquad \ldots \qquad T \vdash u_n}{T \vdash f(u_1, \ldots, u_n)}$$

Figure 2.1 - Modèle de Dolev-Yao: I_{DY}

D'autres exemples de systèmes I sont donnés dans [DLLT06, Del06, LLT05, MS05].

En considérant un nombre fixé d'instances de rôles, on s'intéresse aux éxecutions de protocole possibles dans ce cadre. Chaque instance de rôle étant une séquence d'instructions, tout entrelacement de ces séquences (qu'on va nommér *trace symbolique*) représente une éxecution potentielle du protocole. Une trace symbolique est *accessible* s'il existe une instanciation de cette trace pour laquelle l'intrus est capable de construire, à partir de sa connaissance initiale et des messages précédemment émis sur le réseau, les messages que les différents agents s'attendent à recevoir:

Definition 2.3 (trace symbolique) Une trace symbolique est une séquence d'instructions $instr_1; \ldots; instr_l$, où chaque $instr_i$ est de la forme $recv(u)$ ou $send(u)$.

Definition 2.4 (trace accessible) Un trace $instr_1; \ldots; instr_l$ est accessible a partir de T_0 dans le modèle de l'intrus I s'il existe un substitution σ telle que pour tout i $(1 \leq i \leq l)$ si $instr_i$ est de la forme $recv(u)$, on a:
- $u\sigma$ est déductible de $T_0 \cup \bigcup_{j=1}^{i-1}\{v\sigma \mid instr_j \; est \; de \; la \; forme \; send(v)\}$ dans I.

Exemple. Considérons l'intrus standard I_{DY} et $T_0 = \{a, k, \{k_2\}_{k_1}\}$. Soit la trace $T := recv(\{x\}_k); send(\{k_1\}_x); recv(\{y\}_{k_2}); send(\{s\}_y); recv(s)$.

Alors T est accessible a partir de T_0 si on considère $\sigma = \{x \mapsto a, y \mapsto a\}$. On a: $T_0 \vdash \{a\}_k$; $T_0, \{k_1\}_a \vdash \{a\}_{k_2}; T_0, \{k_1\}_a, \{s\}_a \vdash s$.

Pour décider si une donnée s est secrète on peut utiliser la notion d'accessibilité. En fait, si T est une trace, alors s est connu par l'intrus en éxecutant T si et seulement si la trace $T; recv(s)$ est accessible. Donc le problème de sécurité se réduit au problème de décider l'accessibilité d'une trace symbolique.

Le dernier pas de la modélisation est de transformer une trace symbolique dans un système de contraintes de déduction. On préfère les systèmes de contraintes parce qu'ils enlèvent les détails inutiles et mettent en évidence les propriétés du problème.

Soit I un système de déduction. On va noter $T \vdash_I u$ si u est déductible à partir de T dans I.

Definition 2.5 (systèmes de contraintes) Une contrainte de déduction est une expression de la forme "$T \Vdash u$" où T est un ensemble fini de termes et u un terme.

Un système C de contraintes de déduction est un ensemble fini de contraintes de déduction. Une solution de C (pour un système de déduction I) est une substitution σ telle que:

- pour tout $T \Vdash u \in C$, $T\sigma \vdash_I u\sigma$.

En utilisant les definitions 2.4 et 2.5 on peut facilement construire un système de contraintes C à partir d'une trace symbolique T de façon que T est accessible si et seulement si C est satisfaisable.

Exemple. La trace T dans l'exemple précédent se transforme en

$$C = \left\{ \begin{array}{rcl} T_0 & \Vdash & \{x\}_k \\ T_0, \{k_1\}_x & \Vdash & \{y\}_{k_2} \\ T_0, \{k_1\}_x, \{s\}_y & \Vdash & s \end{array} \right.$$

Les termes qui sont à gauche d'une contrainte représentent la conaissance de l'intrus. Le terme à droite doit être construit par l'intrus pour être envoyé sur le réseau.

Puisque les systèmes de contraintes auxquels nous nous interessons proviennent de traces symboliques, ils satisfont des propriétés particulières:

Definition 2.6 (propriété de monotonie) Un système de contraintes de déduction:

$$C = \begin{cases} T_1 & \Vdash & u_1 \\ T_2 & \Vdash & u_2 \\ & \cdots & \\ T_n & \Vdash & u_n \end{cases}$$

est dit *monotone* si $T_i \subseteq T_{i+1}$, pour tout i tel que $1 \leq i < n$

Cette propriété correspond au fait que la connaissance de l'intrus ne fait que croître au cours de l'exécution du protocole.

Definition 2.7 (propriété d'origination) Un système de contraintes de déduction $C = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ satisfait la propriété d'*origination* si $\forall i \; \forall X \in vars(T_i) \; \exists j < i \; X \in vars(u_j)$.

Cette propriété dit qu'une variable apparait a droite d'une contrainte avant d'apparaître a gauche. Elle correspond au fait que les variables représentent des termes qu'un agent attend, mais ne connait pas d'avance. L'intrus peut donc substituer les variables par des termes qu'il peut construire.

Definition 2.9 (Systèmes de contraintes bien-formés) On dit qu'un système de contraintes est *bien-formé* s'il et monotone est satisfait la propriété d'origination.

En conclusion, on a modélisé le problème de securité d'un protocole cryptographique comme un problème de résolution de systèmes de contraintes bien-formés.

3 Résolution de systèmes de contraintes modulo AC

Dans l'introduction nous avons argumenté la nécessité de résoudre des systèmes de contraintes modulo AC: la propriété de variants finis permet de réduire des nombreuses théories équationelles à la théorie AC ou à la théorie vide [CLD05]. Dans cette section, nous allons définir formellement ce problème et montrer des résultats de décidabilité et d'indécidabilité.

3.1 Définitions

Nous commençons ici par un cas particulier (non résolu à ce jour) où les messages ne sont formés qu'à partir des messages fixes en nombre fini (des constantes) et d'un symbole binaire +, associatif et commutatif. L'intrus ne peut ici effectuer qu'une seule opération: faire la somme de messages. L'espoir est de pouvoir ensuite utiliser des résultats de combinaison pour résoudre des problèmes plus généraux.

Definition 3.1 (Termes) Soit $a_1, a_2, \ldots, a_n$ un ensemble fini de constantes et + un symbole de fonction binaire, associatif et commutatif. Alors l'ensemble des termes clos construits à partir de $\{+, a_1, \ldots, a_n\}$ peut être défini comme: $T = \{\ \Sigma_i \lambda_i a_i \mid \lambda_1, \ldots, \lambda_n \in \mathbb{N}\ \}$

Si X est un ensemble fini de variables, on considère aussi l'ensemble de termes $T(X) = \{\ t + \Sigma_{x \in X} \lambda_x x \mid t \in T,\ \lambda_x \in \mathbb{N}\ \}$

Definition 3.2 (Relation de déductibilité) Soit $U \subseteq T$ et $t \in T$. On note $U \vdash t$ si t peut être obtenu par une combinaison linéaire de termes dans U, avec des coefficients dans $\mathbb{N}$: $t = \lambda_1 u_1 + \ldots + \lambda_k u_k,\ u_i \in U,\ \lambda_i \in \mathbb{N}$. On dit alors que t est déductible a partir de U.

Considérant cette notion de déductibilité on obtient, comme défini dans le chapitre précédent, des systèmes de contraintes modulo AC.

Exemple 3.3 Considérons l'ensemble de constantes $\{a, b, c\}$ et l'ensemble de variables $\{X, Y\}$. Alors:

- $a + b,\ 2a + b + c, 2b + c$ sont des termes clos, $X + a,\ X + 2Y + 2c$ sont des termes avec des variables.
- On a:
  - $a, b, c \vdash 3a + 2b + c;\ 2a, b \vdash 4a + b$
  - $a, b \nvdash a + b + c;\ 2a, b \nvdash 3a + b$

- Soit $C = \left\{ \begin{array}{rcl} 2a & \Vdash & X + a \\ a, X + c & \Vdash & Y + c \\ X + b, Y & \Vdash & 4a + b \end{array} \right.$

  Alors $\sigma = \{X \mapsto a,\ Y \mapsto 3a\}$ est une solution de C, car $\left\{ \begin{array}{rcl} 2a & \vdash & 2a \\ a, a + c & \vdash & 3a + c \\ a + b, 3a & \vdash & 4a + b \end{array} \right.$

3.2 Systèmes de contraintes non bien-formés

Dans cette patrie, nous montrons que le problème de satisfiabilité d'un système de contraintes générale (pas bien-formé) est indécidable. Ça va justifier l'importance des propriétés de monotonie et d'origination, ainsi que montrer la difficulté du problème auquel nous nous intéressons ici.

Pour montrer l'indécidabilité, on va réduire le dixième problème d'Hilbert, connu comme indécidable, au problème de décider la satisfiabilité de systèmes de contraintes.

Dixième problème d'Hilbert: Etant donné un polynôme $P(x, y, \ldots)$ avec des coefficients entiers, le dixième problème d'Hilbert consiste à décider s'il existe une substitution σ de variables avec des entiers positifs telle que $P(x\sigma, y\sigma, \ldots) = 0$.

On remarque d'abord que ce problème peut être réduit à la satisfaisabilité d'un ensemble d'équations de la forme: $x = yz$ ou $x_1 + \ldots + x_n = y_1 + \ldots + y_m$.

Exemple. Considerons le problème $x^2y + yz - xz = 0$. En introduisant les variables v_1, v_2, v_3, v_4 on obient l'ensemble d'équations:

$$\left\{ \begin{array}{l} v_1 = xy \\ v_2 = xv_1 \\ v_3 = yz \\ v_4 = xz \\ v_2 + v_3 = v_4 \end{array} \right.$$

Il nous faut donc coder dans des systèmes de contraintes le produit est la somme. Dans la suite, ont dénote par $|t|_a$ le nombre de a dans t. Par exemple: $|2a + b|_a = 2, |2a + b|_b = 1$. Une instance du dixième problème d'Hilbert va être notée $P = \{E_1, \ldots, E_n\}$.

Codage du produit. Pour une équation $x = yz$ on va construire un système de contraintes C équivalent dans le sens suivant: pour toute solution σ de C, on a $|X\sigma|_a = |Y\sigma|_a |Z\sigma|_a$. Aussi, pour toute solution (p, q, r) de $x = yz$, on a une solution σ de C telle que $|X\sigma|_a = p, |Y\sigma|_a = q, |Z\sigma|_a = r$.

Soit:

$$C = \left\{ \begin{array}{lll} a & \Vdash & X \\ a & \Vdash & Y \\ a & \Vdash & Z \\ b & \Vdash & V \\ a+b & \Vdash & Z+V \\ Y+b & \Vdash & X+V \end{array} \right.$$

V est une variable nouvelle: différente pour chaque équation à coder.

Lemme 3.10

1. Soit σ une solution de C. Alors: $|X\sigma|_a = |Y\sigma|_a |Z\sigma|_a$
2. Soit $p, q, r \in \mathbb{N}$ tels que $p = qr$. Alors $\sigma = \{X \mapsto pa, Y \mapsto qa, Z \mapsto ra, V \mapsto rb\}$ est une solution de C.

Preuve:
1. Les premières 4 contraintes assurent que $|X\sigma|_b = |Y\sigma|_b = |Z\sigma|_b = 0$ et $|V\sigma|_a = 0$. Donc, en utilisant $a+b \vdash Z\sigma+V\sigma$, on a $|Z\sigma|_a = |V\sigma|_b$. La derniere contrainte nous dit qu'il existe un $\lambda \in \mathbb{N}$ tel que: $X\sigma+V\sigma = \lambda(Y\sigma+b)$. Comme $X\sigma$ et $Y\sigma$ ne contiennent pas des b, on déduit que $\lambda = |V\sigma|_b = |Z\sigma|_a$ et enfin que $|X\sigma|_a = |Y\sigma|_a |Z\sigma|_a$.
2. La vérification est immediate. □

Codage de la somme. Soit $x_1+\ldots+x_n = y_1+\ldots+y_m$ une équation dans les entiers positifs. De même manière que pour le produit, on construit un système de contraintes equivalent:

$$C = \left\{ \begin{array}{lll} a & \Vdash & X_1, \ldots X_n, Y_1, \ldots, Y_m \\ X_1 + \ldots + X_n + c & \Vdash & Y_1 + \ldots + Y_m + c \end{array} \right.$$

Ici on a noté $\{a \Vdash u_1, \ldots, u_k\}$ au lieu de $\{a \Vdash u_1, \ldots, a \Vdash u_k\}$. Comme c est une constante n'aparaissant pas dans X_i, Y_i, l'égalité est imposée.

Codage d'un ensemble d'équations. On a vu précédemment comment coder une équation avec un système de contraintes. Dû au fait que le système représentant une équation préserve la totalité de ces solutions, on peut coder un ensemble d'équations en considérant l'union de systèmes correspondantes.

Proposition 3.11 Soit $P = \{E_1, \ldots, E_n\}$ une instance du dixième problème d'Hilbert et $C = C_1 \cup \ldots \cup C_n$ le système de contraintes où chaque C_i est construite comme expliqué auparavant. Soit $\{x_1, \ldots, x_m\}$ l'ensemble de variables dans P. Nous avons:

1. Si σ est une solution de C, alors $(x1, \ldots, x_m) = (|X_1\sigma|_a, \ldots, |X_m\sigma|_a)$ est une solution de P

2. Si $(x1, \ldots, x_m) = (p_1, \ldots, p_m)$ est une solution de P, il existe une solution σ de C telle que $|X_1\sigma|_a = p_1, \ldots, |X_m\sigma|_a = p_m$.

Preuve:

1. On doit montrer que chaque E_i est satisfaite par $(|X_1\sigma|_a, \ldots, |X_m\sigma|_a)$. Comme σ est une solution de $C = C_1 \cup \ldots \cup C_n$, on déduit que σ est une solution de C_i, pour chaque i. Par le Lemme 3.10(1) on obtient que chaque E_i est satisfaite.

2. Soit, pour chaque i, σ_i la solution de C_i donnée par le Lemme 3.10(2). Par definition des σ_i, on a que $X\sigma_i = X\sigma_j$ si $X \in dom(\sigma_i) \cap dom(\sigma_j)$. On peut donc definir $\sigma = \cup_i \sigma_i$. Comme $\sigma|_{Var(C_i)} = \sigma_i$, σ est une solution de chaque C_i, et donc de C. En plus, $|X_1\sigma|_a = p_1, \ldots, |X_m\sigma|_a = p_m$ par la construction du Lemme 3.10 (2). Donc σ est la solution cherchée. □

Corrolaire 3.12. La résolution de systèmes de contraintes généraux modulo AC est une problème indécidable.
Preuve: Etant donée une instance du dixième problème d'Hilbert P, par proposition 3.11 on peut construire un système de contraintes C tel que C a une solution si et seulement si P a une solution. D'oû on en déduit l'indécidabilité du résolution de systèmes de contraintes

Discussion Ce résultat s'applique aussi aux systèmes qui satisfont la propriété d'origination, mais pas la monotonie. La non-monotonie est utilisée pour coder le produit, ainsi que pour combiner des systèmes de contraintes. L'indécidabilité a été prouvée aussi pour des systèmes monotones, mais pour des théories plus complexes [Del06].

3.3 Systèmes de contraintes bien-formés simples

Dans cette section nous considerons un cas particulier de systèmes de contraintes modulo AC bien-formés: chaque contrainte contient au plus une variable a droite.

Definition 3.13 (Systèmes simples.) Soit $C = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$ un système de contraintes. On dit que C est *simple* si pour tout i: $u_i = \beta_i X_i + \gamma_i$, avec $\beta_i \in \mathbb{N}$, X_i - une variable et γ_i - un terme clos.

On va considérer dans la suite des systèmes de contraintes bien-formés simples. On va les représenter sous la forme $C = \{T_1 \Vdash u_1, \ldots, T_n \Vdash u_n\}$, où $C = \{T_1 \Vdash \beta_1 X_1 + \gamma_1, \ldots, T_n \Vdash \beta_n X_n + \gamma_n\}$ quand on a besoin d'expliciter les coefficients.

Exemple. Soit $C = \left\{ \begin{array}{rcl} 2a & \Vdash & X + a \\ 2a, X + c & \Vdash & Y + c \\ 2a, X + c, Y & \Vdash & 2a + c \end{array} \right.$
Alors C est un système de contraintes bien-formé simple et il a comme

solution $\{X \mapsto a,\ Y \mapsto a\}$

Pour exemplifier les difficultés qu'on peut avoir si on veut résoudre des tels systèmes, montrons qu'une simple élimination de variables nous donnes des systèmes diophantiennes généraux. Soit:

$$C = \begin{cases} a & \Vdash & X \\ a, X & \Vdash & Y \end{cases}$$

Si on essaye de traduire ce système comme un système d'équations et éliminer X, on obtient le système non-linéaire:

$$S = \begin{cases} X & = & \lambda a \\ Y & = & \lambda' a + \lambda'' \lambda a \end{cases}$$

Nous montrons dans ce paragraphe le:

Théorème 3.14 La satisfaisabilité de systèmes de contraintes bien-formés simples est décidable.

Comme nous l'avons vu plus haut, traduire les contraintes sous forme d'équations sur les entiers et tenter une élimination de variable va rapidement conduire à des systèmes Diophantiennes généraux. Il nous faut donc tirer parti des propriétés du système et en particulier de la monotonie.

L'idée générale de la preuve du théorème est la suivante:

1. On devine dans chaque T_i les termes utilisés dans la construction du $u_i\sigma$.
2. On devine une relation (de pré-ordre) d'occurence $\prec_{occ}$ entre variables, dont la relation d'équivalence associée est $=_{occ}$.
3. On résout le système C_0, restriction de C à une classe minimale E_0 de $=_{occ}$.
4. On montre que le système C, s'il a une solution, a aussi une solution qui ne s'éloigne pas trop, sur les variables de E_0, d'une solution minimale de C_0.
5. En devinant la restriction d'une solution aux variables de E_0 (dans une ensemble fini effectivement calculable par 4.), on élimine toutes les variables d'une classe minimale.

Definition 3.14 (Termes utiles) Soit C un système de contraintes et σ une solution de C. Alors, pour chaque i, la preuve de $T_i\sigma \vdash u_i\sigma$ utilise un ensemble $U_i\sigma \subseteq T_i\sigma$ d'hypothèses. On va dire que les ensembles $U_i \subseteq T_i$ sont des termes utilisés par σ.

Cette definition nous permet de deviner les termes utiles avant de chercher une solution, puisque a chaque solution correspond un multiensemble de termes utiles $U = U_1 \cup \ldots \cup U_n$. On va noter par C_U le système obtenu à partir de C en suprimmant les termes inutiles.

Les termes utiles sont donc les termes pour lequels les coefficients $\lambda \in \mathbb{N}$ sont strictement positifs dans une preuve de $T_i\sigma \vdash u_i\sigma$, pour un i. On va noter par

$$\Lambda = \begin{array}{l}(\lambda_{1,1}, \ldots, \lambda_{1,k_1}, \\ \ldots, \\ \lambda_{n,1}, \ldots, \lambda_{n,k_n})\end{array}$$

le vecteur de ces coefficients. On utilise aussi la notation Λ^σ quand on veut distinguer une solution.

En cherchant une solution de C, on cherche donc pour un vecteur (X, Λ).

Definition 3.15 (Relation d'occurrence) Soit C un système de contraintes et V l'ensemble de variables dans C. On cherche pour une solution et on suppose avoir deviné les $U_i \subseteq T_i$. On définit la relation $\prec_{occ}$ sur V de manière suivante:

$$X \prec_{occ} Y \Leftrightarrow \exists i, v : \begin{cases} Y \in Var(u_i) \\ X \in Var(v) \\ v \in U_i \end{cases}$$

Notations. Soit $\prec_{occ}$ la relation d'occurrence défini ci-dessus. On considère la clôture transitive de $\prec_{occ}$ et on note $=_{occ}$ la relation d'équivalence induite: $X =_{occ} Y$ *ssi* $\exists X_1, \ldots, X_n \; \exists Y_1, \ldots, Y_m : \; X \prec_{occ} X_1 \prec_{occ} \ldots \prec_{occ} X_n \prec_{occ} Y \prec_{occ} Y_1 \prec_{occ} \ldots \prec_{occ} Y_m \prec_{occ} X$. On denote par $[=_{occ}]$ l'ensemble de classes de variables modulo $=_{occ}$.

Si C_1, C_2 sont deux classes de $[=_{occ}]$, on note $C_1 \prec_{occ} C_2$ si pour un $X \in C_1$ et un $Y \in C_2$ on a $X \prec_{occ} Y$. On voit aisement que $\prec_{occ}$ est une relation d'ordre sur $[=_{occ}]$.

On va considérer l'ordre produit sur les vecteurs d'entiers: $(u_1, \ldots, u_k) \leq (v_1, \ldots, v_k)$ *ssi* $\forall i : u_i \leq v_i$. Soit u, v deux termes clos. On va noter $u \leq v$ si $|u|_c \leq |v|_c$, pour toute constante c. Donc $u \leq v$ si $(|v|_{c_1}, \ldots, |v|_{c_k}) \leq (|v|_{c_1}, \ldots, |v|_{c_k})$, pour l'ensemble de constantes $\{c_1, \ldots, c_k\}$. On considère aussi l'extension de cet ordre aux substitutions, vues comme vecteurs de termes.

Pour un terme clos t, on va noter par $max_c(t)$ le maximum de ces coefficients: $max_c(t) = max(\{|t|_{c_1}, \ldots, |t|_{c_k}\})$.

Systèmes linéaires. Ici nous rappelons la definition de systèmes d'équations diophantiennes linéaires ainsi qu'un résultat sur la forme de solutions de ces systèmes [Sch86].

Definition 3.16 (systèmes linéaires, systèmes homogènes) Un système d'équations diophantiennes linéaire est un système de la forme:

$$S = \begin{cases} a_{11}x_1 + \ldots + a_{1q}x_q & = & b_1 \\ \ldots & & \\ a_{i1}x_1 + \ldots + a_{iq}x_q & = & b_i \\ \ldots & & \\ a_{p1}x_1 + \ldots + a_{pq}x_q & = & b_p \end{cases}$$

où $\forall i, j\ a_{ij} \in \mathbb{Z}, \forall i\ b_i \in \mathbb{Z}$.

Une solution de S est une affectation $\sigma(x_1, \ldots, x_p) \rightarrow \mathbb{N}^p$ qui satisfait ces équations.

Le système homogène associé à S est:

$$S_h = \begin{cases} a_{11}x_1 + \ldots + a_{1q}x_q &= 0 \\ \ldots & \\ a_{i1}x_1 + \ldots + a_{iq}x_q &= 0 \\ \ldots & \\ a_{p1}x_1 + \ldots + a_{pq}x_q &= 0 \end{cases}$$

Solutions de systèmes linéaires. On va utiliser les résultats suivants sur l'ensemble des solutions d'un sytème linéaire:

1. L'ensemble de solutions minimales (pour $\leq$) non nulles de S (resp. S_h) est fini.

2. Si $M_1, \ldots, M_n$ sont des solutions minimales non nulles de S_h et X_0 est une solution minimale de S, alors pour tous $c_1, \ldots, c_n \in \mathbb{N}$: $X = X_0 + \Sigma_i c_i M_i$ est une solution de S. Réciproquement: pour toute solution X de S, il existent $n,\ X_0,\ M_1, \ldots, M_n,\ c_1, \ldots, c_n$ comme ci-dessus tels que: $X = X_0 + \Sigma_i c_i M_i$.

Dans la suite, nous montrons que si $|[=_{occ}]| = 1$, la résolution de systèmes de contraintes modulo AC se réduit à la résolution de systèmes lineaires d'équations. Ce résultat est valable pas seulement pour les systèmes bien-formés.

Lemme 3.17 Soit C_U le système de contraintes obtenu a partir d'un système C en devinant U, avec $|[=_{occ}]| = 1$. Soit $U' \subseteq U$ le sous-ensemble de termes non-clos de U. Alors il existe un vecteur A, tel que $\Lambda^\sigma|_{U'} \leq A$, pour toute σ - solution de C_U.

Preuve. Si $Var(C_U) = \{X\}$ et $X \not\prec_{occ} X$, on peut résoudre C_U simplement: si X apparaît dans une contrainte à gauche, on peut borner sa valeur; si X n'apparaît qu'à droite, le système est déjà linéaire. On suppose donc sans perte de généralité que tous sousensemble de $Var(C_U)$ est relié par un cycle.

Considerons n'importe quel cycle $X_1 \prec_{occ} X_2 \prec_{occ} \ldots \prec_{occ} X_n \prec_{occ} X_1$. On a donc:

$$\begin{array}{rcl} \lambda_1 X_1\sigma + t_1 &=& \beta_1 X_2\sigma + \gamma_1 \\ \lambda_2 X_2\sigma + t_2 &=& \beta_2 X_3\sigma + \gamma_2 \\ & \ldots & \\ \lambda_n X_n\sigma + t_n &=& \beta_n X_1\sigma + \gamma_n \end{array},$$

pour certains termes t_i. On va borner λ_1 et par symétrie on obtiendra une borne pour chaque λ_i.

Puisque tous les coefficients sont positifs, on obtient les inégalités:

$$\begin{array}{rcl} \lambda_1 X_1\sigma &\leq& \beta_1 X_2\sigma + \gamma_1 \\ \lambda_2 X_2\sigma &\leq& \beta_2 X_3\sigma + \gamma_2 \\ & \ldots & \\ \lambda_n X_n\sigma &\leq& \beta_n X_1\sigma + \gamma_n \end{array}$$

Si on multiple la deuxième inégalité par β_1/λ_2, la troisème par $\beta_1\beta_2/\lambda_2\lambda_3$, ..., la n-ième par $\beta_1 \ldots \beta_{n-1}/\lambda_2 \ldots \lambda_n$ on obtient:

$$\begin{array}{rcl} \lambda_1 X_1\sigma & \leq & \beta_1 X_2\sigma + \gamma_1 \\ \beta_1 X_2\sigma & \leq & \beta_1\beta_2/\lambda_2 X_3\sigma + \beta_1/\lambda_2\gamma_2 \\ \beta_1\beta_2/\lambda_2 X_3\sigma & \leq & \beta_1\beta_2\beta_3/\lambda_2\lambda_3 X_4\sigma + \beta_1\beta_2/\lambda_2\lambda_3\gamma_3 \\ & \ldots & \\ \beta_1 \ldots \beta_{n-1}/\lambda_2 \ldots \lambda_{n-1} X_n\sigma & \leq & \beta_1 \ldots \beta_n/\lambda_2 \ldots \lambda_n X_1\sigma + \beta_1 \ldots \beta_{n-1}/\lambda_2 \ldots \lambda_n\gamma_n \end{array}$$

En sommant toutes ces inégalités et en faisant les simplifications on obtient:

$\lambda_1 X_1 <$
$\beta_1 \ldots \beta_n/\lambda_2 \ldots \lambda_n X_1\sigma + \gamma_1 + \beta_1/\lambda_2\gamma_2 + \ldots + \beta_1 \ldots \beta_{n-1}/\lambda_2 \ldots \lambda_n\gamma_n <$
$\beta_1 \ldots \beta_n X_1\sigma + \gamma_1 + \beta_1\gamma_2 + \ldots + \beta_1 \ldots \beta_{n-1}\gamma_n$

Finalement on a $\lambda_1 < \beta_1 \ldots \beta_n + max_c(\gamma_1) + \beta_1 max_c(\gamma_2) \ldots + \beta_1 \ldots \beta_{n-1} max_c(\gamma_n)$ et donc on a borné λ_1 par une constante. Puisque on a une seule classe d'occurrence, toute apparition d'une variable est reliée par un cycle aux autres et on peut borner son contexte λ comme on l'a fait pour λ_1. □

Corrolaire 3.18 Si C_U est un système comme dans le lemme précédent, il existe un ensemble fini $E = \{C_1, \ldots, C_n\}$ de systèmes *linéaires* tel que σ est une solution de C_U si et seulement si $\exists C_i \in E$ tel que σ est une solution de C_i.
Preuve. Soit V l'ensemble de vecteurs $\{v \mid v \leq A\}$, où A est le vecteur donné par le lemme précédent. Soit $\Lambda' = \Lambda|_{U'}$, où U' est l'ensemble de termes non-clos de U. Alors, pour chaque vecteur $v \in V$ on obtient un système C_v en substituant Λ' par v. Soit $E = \{C_v \mid v \in V\}$.

D'abord, on remarque que E est un ensemble de systèmes linéaires. En fait, si on a un terme non-clos $t_j \in U_i$ dans C_v, pour certains i, j, v, alors on a remplacé λ_{ij} dans C_U par v_{ij} pour obtenir C_v. Donc tous les termes avec des variables ont un coefficient constant dans C_v, pour tout $C_v \in E$.

Soit (X, Λ) une solution d'un $C_v \in E$. Soit $\Lambda_1 = \Lambda \cup v$. Alors il est evident que (X, Λ_1) est une solution de C_U. Soit maintenant (X, Λ) un solution de C_U. Par le lemme 3.17, on a que $\Lambda|_{U'} = v \leq A$. Donc $(X, \Lambda|_{U \setminus U'})$ est une solution de C_v. □

Corrolaire 3.19 Si C_U est un système comme dans le lemme précédent, l'ensemble de solutions de C_U peut être représenté par $S = \cup_{C_k \in E}\{\theta + \Sigma_i c_i w_i \mid \theta \in sm(C_k),\ w_i \in sm(C_k^h),\ c_i \in \mathbb{N}\}$, où E est l'ensemble du corrolaire 3.18 et $sm(C)$ représente l'ensemble de solutions minimales de C.
Preuve. Résulte immediatement du corrolaire 3.18 et du resultat sur la forme de solutions de systèmes linéaires, rappelé dans cette section.□

Système homogène. On regarde ici les systèmes de contraintes avec $|[=_{occ}]| = 1$ comme une union finie de systèmes de contraintes linéaires. Voyons comment se définit le système homogène associé a un système de contraintes

linéaires. Après avoir borné les λ_i correspondant à des termes non-clos de U_i, on obtient un système de la forme:

$$C = \left\{ \begin{array}{rcl} \lambda_{11}u_{11} + \ldots + \lambda_{1k_1}u_{1k_1} + t_{11} + \ldots + t_{1l_1} & = & \beta_1 X_1 + \gamma_1 \\ & \ldots & \\ \lambda_{n1}u_{n1} + \ldots + \lambda_{nk_n}u_{nk_n} + t_{n1} + \ldots + t_{nl_n} & = & \beta_n X_n + \gamma_n \end{array} \right. ,$$

avec $\forall i, j:$ u_{ij} - clos, t_{ij} - non-clos.

Pour un terme t, soit t^v le terme obtenu en enlevant la partie close de t: $t = t^v + \gamma$, γ - clos, $|t^v|_c = 0$ pour tous c. Par exemple: $(x+2y+a+2b)^v = x+2y$.

Alors la definition du système homogène associé à un système linéaire est dans notre cas:

$$C_h = \left\{ \begin{array}{rcl} \lambda_{11}u_{11} + \ldots + \lambda_{1k_1}u_{1k_1} + t^v_{11} + \ldots + t^v_{1l_1} & = & \beta_1 X_1 \\ & \ldots & \\ \lambda_{n1}u_{n1} + \ldots + \lambda_{nk_n}u_{nk_n} + t^v_{n1} + \ldots + t^v_{nl_n} & = & \beta_n X_n \end{array} \right. ,$$

Cette definition est en fait une reformulation de la definition classique d'un système homogène, donnée dans le rappel précédent sur les systèmes linéaires.

Grâce à la propriété d'origination, pour chaque variable X il existe une contrainte, qu'on va noter $T_{i_X} \Vdash \beta_{i_X} X + \gamma_{i_X}$, telle que $\forall i < i_X :\ X \notin Var(T_i \cup u_i)$ et $X \notin Var(T_{i_X})$. On va dire que $T_{i_X} \Vdash \beta_{i_X} X + \gamma_{i_X}$ introduit X.

Lemme 3.20 Soit $C = \{T_1 \Vdash \beta_1 X_1 + \gamma_1, \ldots, T_n \Vdash \beta_n X_n + \gamma_n\}$ un système de contraintes bien-formé simple et U son ensemble de termes utiles, avec $|[=_{occ}]| = 1$. Alors, par le corrolaire 3.18, C_U est une union finie de systèmes linéaires et soit C_h n'importe quel de systèmes homogènes associés à C_U. Soit $X_1, \ldots, X_m$ l'ordre d'apparition des variables dans C et $\beta_{X_j} = \beta_{i_{X_1}} \ldots \beta_{i_{X_j}}$, pour $1 \leq j \leq m$. Si σ est une solution de C_h, alors pour tout $X \in Var(C)$: $\beta_X X\sigma = \Sigma_k \mu_k t_k$, avec $t_k \in T_{i_X}$ et t_k clos.

Preuve. On va prouver cette propriété par induction sur l'ordre d'apparition des variables. On va noter par $T^h_i \Vdash \beta_i X_i$ les contraintes de C_h

Si $X = X_1$, alors T_{i_X} ne contient que des termes clos, à cause de l'origination. Donc $T^h_{i_X} = T_{i_X}$. Comme σ est une solution de C_h, on a $\beta_{i_X} X\sigma = \lambda_1 t_1 + \ldots + \lambda_k t_k$, avec $t_l \in T^h_{i_X} = T_{i_X}$, pour $1 \leq l \leq k$. D'où le résultat, puisque $\beta_X = \beta_{i_X}$, pour $X = X_1$.

Si $X = X_p$, avec $p > 1$, on a encore une fois $\beta_{i_X} X\sigma = \lambda_1 t_1\sigma + \ldots + \lambda_k t_k\sigma(*)$, avec $t_l \in T^h_{i_X}$, pour $1 \leq l \leq k$. Pour un l, si t_l est clos, alors $\lambda_l t_l$ est de la forme souhaitée et on va le garder dans la somme. Si t_l contient des variables, alors $\lambda_l = 1$ et il est de la forme $t_l = \alpha_1 Y_1 + \ldots + \alpha_r Y_r$. Soit Y la plus grande variable (dans l'ordre d'apparition) dans $T^h_{i_X}$ (Y est plus petite strictement que X). En multipliant (*) par β_Y on obtient: $\beta_Y \beta_{i_X} X\sigma = \lambda_1 \beta_Y t_1\sigma + \ldots + \lambda_k \beta_Y t_k\sigma(**)$ et $\beta_Y t_l\sigma = \alpha_1 \beta_Y Y_1\sigma + \ldots + \alpha_r \beta_Y Y_r\sigma(***)$, si t_l n'est pas clos.

Par la maximalité de Y et par la definition de β_Y on a $\beta_{i_{Y_j}} \mid \beta_Y$, pour chaque $Y_j \in Var(T^h_{i_X})$. Comme on a aussi $Y_j < X$, on peut utiliser l'hypothèse

d'induction pour Y_j pour déduire que $\beta_{i_{Y_j}} Y_j\sigma$ est une combinaison linéaire de termes clos de $T_{i_{Y_j}}$ et donc $\beta_Y Y_j\sigma = \Sigma_t \mu_t t$, avec $t \in T_{i_{Y_j}} \subseteq T_{i_X}$(par monotonie) et t clos. Donc on peut remplacer dans (***) chaque $\beta_Y Y_j\sigma$ par une combinaison linéaire de termes de T_{i_X}. (**) devient ensuite:

$$\beta_Y\beta_{i_X} X\sigma = \lambda_1(\sum_{t\in T_{i_X}} \mu_t^1 t) + \ldots + \lambda_k(\sum_{t\in T_{i_X}} \mu_t^k t),$$

où les $t \in T_{i_X}$ sont clos pour chaque somme. Comme, par definition, $\beta_Y\beta_{i_X} \mid \beta_X$, on obtient finalement $\beta_X X\sigma = \Sigma_t \mu_t t$, avec $t \in T_{i_X}$ et t clos. □

Lemme 3.21 Soit $C = \{T_1 \Vdash \beta_1 X_1 + \gamma_1, \ldots, T_n \Vdash \beta_n X_n + \gamma_n\}$ un système de contraintes bien-formé simple, E_0 une classe minimale de $[=_{occ}]$ et $C_0 = \{U_i \Vdash u_i \mid T_i \Vdash u_i \in C,\ Var(u_i) \in E_0\}$. Alors il existe $w(C_0)$ - un vecteur ne dépendant que de C_0, tel que pour toute σ - solution minimale de C, il existe θ - une solution minimale de C_0 telle que: $X\sigma \leq X\theta + w$, pour tout $X \in E_0$.
Preuve. Soit $Var(C_0) = \{X_1, \ldots, X_k\}$ et $\beta = \beta_{i_{x_1}} \ldots \beta_{i_{x_k}}$.

Premièrement, observons que C_0 satisfait les hypothèses du lemme 3.17 (sa unique classe d'équivalence est E_0).

Soit alors σ une solution minimale de C. La projection de σ sur les variables de C_0 est une solution de C_0. Par le corrolaire 3.19, il existe une solution minimale de C_0 (θ), des entiers $c_1, \ldots, c_l$ et des solutions minimales de C_0^h $w_1, \ldots, w_l$ t.q: $X\sigma = X\theta + \Sigma_j c_j w_j^X$, pour tout $X \in Var(C_0)$. Nous allons prouver que $c_j \leq \beta$, pour tout $1 \leq j \leq l$.

Effectivement, supposons que pour un j: $c_j > \beta$. Nous montrons comment construire une solution σ' plus petite que σ, en contredisant sa minimalité. Considerons $X\sigma' = X\sigma - \beta w_j^X$, pour tout $X \in Var(C_0)$, et $X\sigma' = X\sigma$, si $X \in Var(C)\backslash Var(C_0)$. Nous allons vérifier que chaque contrainte de C est satisfaite par σ'. Il est facile de voir que σ' est une solution de C_0, car elle satisfait la relation: $(X\sigma', \Lambda') = (X\theta, \Lambda^\theta) + (c_1 w_1 + \ldots + (c_j - \beta)w_j + \ldots + c_l w_l)$ (en utilisant que $c_j > \beta$ et le résultat sur les systèmes linéaires).

Soit maintenant $T_i \Vdash u_i \in C\backslash C_0$ et prouvons que σ' satisfait cette contrainte. On a $Var(u_i) \notin Var(C_0)$. En fait, si $Var(u_i) \in Var(C_0) = E_0$ alors $T_i \Vdash u_i \in C_0$ par la definition de C_0.

Supposons que $X \in Var(U_i) \cap Var(C_0)$. Comme σ est une solution de C, on a:

$$u_i\sigma = \lambda v\sigma[X\sigma] + t\sigma,$$

pour $v[X] \in U_i$ et t une combinaison linéaire de termes de U_i. En utilisant la definition de σ' on obtient:

$$u_i\sigma' = u_i\sigma = \lambda v\sigma[X\sigma' + \beta w_j^X] + t\sigma = \lambda v\sigma[X\sigma'] + t\sigma + \lambda'\beta w_j^X$$

Comme $X \in Var(T_i)$, la monotonie et l'origination nous garantissent que $T_{i_X} \subseteq T_i$. En plus, comme w_j est une solution de C_0^h, le lemme 3.20 nous dit que $\beta w_j^X = \beta'\beta_X w_j^X = \Sigma_j \mu_j t_j$, avec $t_j \in T_{i_X}$, t_j clos. En remplaçant ci-dessus on obtient:

$$u_i\sigma' = \lambda v\sigma[X\sigma'] + t\sigma + \Sigma_j \mu_j' t_j\sigma',$$

avec $v[X], t_j \in T_i$ et t une combinaison linéaire de termes de T_i.

En faisant ce remplacement pour chaque apparition d'une variable $X \in Var(U_i) \cap Var(C_0)$ dans U_i, on obtient finalement: $u_i\sigma' = \Sigma_j \mu_j t_j \sigma'$, avec $t_j \in T_i$. Donc $T_i \Vdash u_i$ est satisfaite par σ', pour chaque i.

On a obtenu donc une solution de C qui est plus petite que σ. Ceci contredit la minimalité de σ et donc on a que pour tout j: $c_j \leq \beta$. Soit $W = \{w_1, \ldots, w_m\}$ l'ensemble de solutions minimales de C_0^h et $w = \beta(w_1 + \ldots + w_m)$. Alors w est le vecteur $w(C_0)$ cherché. Effectivement, si $X\sigma \not\leq X\theta + w$, comme $X\sigma = X\theta + c_1 w_1 + \ldots + c_m w_m$, alors pour un j: $c_j > \beta$. Contradiction. □

Corrolaire 3.22 Soit C un système de contraintes bien-formé. Alors ils existent les ensembles $V = \{V_1, \ldots, V_k \mid V_i \subseteq Var(C)\}$ et $S = \{C_1, \ldots, C_n\}$ effectivement calculables, tels que:

- $\forall i : C_i$ est un système de contraintes bien-formé.
- $\forall i \exists j : Var(C_i) = Var(C) \backslash V_j$
- C a une solution si et seulement si $\exists i : C_i$ a une solution.

Preuve. On commence par considerer toutes les possibilités pour les termes utiles. Chaque choix nous donne un ensemble de variables V_i, qui est la classe minimale de $[=_{occ}]$. Prenons $V = \{V_1, \ldots, V_k\}$. Pour chaque ensemble de variables $V_i \in V$ on extrait du C un système linéaire C_0^i, comme expliqué dans le lemme précédent. On résout le système C_0^i et on obtient alors l'ensemble de solutions minimales de C_0^i M et l'ensemble de solutions minimales de $C_{0_h}^i$ M_h. Soit: $\beta = \prod_{X \in V_i} \beta_{i_X}$ et

$$S_i = \{\sigma \mid \exists \theta \in M, w_1, \ldots, w_l \in M_h, c_1, \ldots, c_l \leq \beta \in \mathbb{N} : \ \sigma = \theta + \Sigma_j c_j w_j\}$$

Le lemme 3.21 nous dit que pour toute solution minimale de C γ il existe $V_i \in V$ et $\sigma \in S_i$ t.q.: $\gamma|_{V_i} = \sigma$. Donc si pour chaque paire $(V_i \in V, \sigma \in S_i)$, on considère le systéme $C\sigma$ on obtient un ensemble de systèmes de contraintes $S = \{C_1, \ldots, C_n\}$ satisfaisant l'ennoncé du lemme. □

On conclut la preuve du théoreme 3.14 en observant que le Corrolaire 3.22 élimine un ensemble de variables de C. En répétant successivement cette procédure, on obtient finalement un système linéaire qu'on résout simplement.

Exemples. **1.** $C = \left\{ \begin{array}{rcl} 2a & \Vdash & X + a \\ 2a, X + b & \Vdash & 3Y + b \end{array} \right.$

Supposons que tous les termes sont utiles. Alors $X \prec_{occ} Y$ et $[=_{occ}] = \{\{X\}, \{Y\}\}$. La classe minimale de $[=_{occ}]$ est $\{X\}$ et alors $C_0 = \{2a \Vdash X + a\}$. Puisque C_0 est déjà un système linéaire, on n'a pas besoin de borner le vecteur Λ. La solution minimale de C_0 est $\{X \mapsto a\}$; la solution minimale de C_0^h est $\{X \mapsto 2a\}$. L'ensemble de solutions de C_0 est alors $\{X \mapsto (a + \lambda 2a) \mid \lambda \in \mathbb{N}\}$. Puisque $\beta = 1$, le lemme 3.21 nous dit qu'on peut borner λ par 1 en cherchant une solution minimale σ de C: $X\sigma \leq a + 2a$. On a donc deux possibilités pour X: $X = a$ ou $X = 3a$. En vérifiant (par remplacement) chacune de ces possibilités on obtient la solution (minimale) de C: $\{X \mapsto 3a, Y \mapsto a\}$.

2. $C = \left\{ \begin{array}{rcl} 3a & \Vdash & X \\ 3a, X+a & \Vdash & 2Y+a \\ 3a, X+a, Y & \Vdash & X+2a \\ 3a, X+a, Y & \Vdash & Z+a \\ 3a, X+a, Y, Z+b & \Vdash & 4a+b \end{array} \right.$

Supposons qu'en devinant les termes utiles on obtient:

$$C_U = \left\{ \begin{array}{rcl} 3a & \Vdash & X \\ 3a, X+a & \Vdash & 2Y+a \\ Y & \Vdash & X+2a \\ Y & \Vdash & Z+a \\ Z+b & \Vdash & 4a+b \end{array} \right.$$

Alors $X \prec_{occ} Y \prec_{occ} X$ et $Y \prec_{occ} Z$. La classe minimale est donc $\{X, Y\}$ et:

$$C_0 = \left\{ \begin{array}{rcl} 3a & \Vdash & X \\ 3a, X+a & \Vdash & 2Y+a \\ Y & \Vdash & X+2a \end{array} \right.$$

qui devient le système d'équations:

$$C_0 = \left\{ \begin{array}{rcl} \lambda_1 3a & = & X \\ \lambda_{21} 3a + \lambda_{22}(X+a) & = & 2Y+a \\ \lambda_3 Y & = & X+2a \end{array} \right.$$

Le lemme 3.17 nous dit qu'on peut borner λ_{22} par $2+1+2*2=7$ et λ_3 par $2+2+2*1=6$. Il nous faut considérer toutes les possibilités pour λ_{22}, λ_3. Supposons que $\lambda_{22}=2$ et $\lambda_3=1$. On obtient:

$$C_0 = \left\{ \begin{array}{rcl} \lambda_1 3a & = & X \\ \lambda_{21} 3a + 2(X+a) & = & 2Y+a \\ Y & = & X+2a \end{array} \right.$$

La solution minimale de C_0 est $\{X \mapsto 0, Y \mapsto 2a\}$. La solution minimale de C_{0_h} est $\{X \mapsto 3a, Y \mapsto 3a\}$. Les solution de C_0 sont alors $\{X \mapsto \lambda 3a, Y \mapsto 2a + \lambda 3a \mid \lambda \in \mathbb{N}\}$. Le lemme 3.21 nous dit qu'on peut borner λ par $1*2*1=2$ en cherchant une solution minimale de C. Effectivement, pour $\lambda=1$ on obtient la solution $\{X \mapsto 3a, Y \mapsto 5a, Z \mapsto 4a\}$.

3.4 Augmentation du pouvoir de l'intrus

Dans cette section nous considérons un modèle comme dans la section précédente, mais pour un système de déduction de l'intrus étendu: $I = \{u, v \vdash u+v;\ u, u+v \vdash u\}$. En plus de faire la somme de deux termes, se système permet de soustraire un terme v d'un terme $t = u+v$, qui est plus grand que v. Observons qu'il est réaliste d'ajouter cette règle, puisque dans la majorité des applications (e.g. le porte-monnaie électronique invoqué dans l'introduction) l'operateur $+$ a la propriété d'inverse.

L'ensemble de termes (T) est le même qu'avant. La relation de déductibilité se définit de manière similaire pour le nouveau système d'inférence:

Definition 3.23 (relation de déductibilité) Soit $U \subseteq T$ et $t \in T$. On note $U \vdash t$ si t peut être obtenu par une combinaison linéaire de termes dans U, avec des coefficients dans $\mathbb{Z}$: $t = \lambda_1 u_1 + \ldots + \lambda_k u_k,\ u_i \in U,\ \lambda_i \in \mathbb{Z}$. On dit alors que t est déductible a partir de U.

Exemple. Soit $C = \left\{ \begin{array}{rcl} 2a & \Vdash & X + a \\ 2a, X + b & \Vdash & 2b \end{array} \right.$
C n'a pas de solution dans le système d'inférence précédent, mais $\{X \mapsto a\}$ est une solution de C dans le système d'inférence étendu, puisque $2*(a+b)-2a = 2b$.

Notation. Pour une contrainte $C = v_1, \ldots, v_k \Vdash u$, une substitution σ et un vecteur $\Lambda = (\lambda_1, \ldots, \lambda_k)$ on definit $Diff(C, \sigma, \Lambda) = \Sigma_i \lambda_i v_i \sigma - u\sigma$.

Dans la suite, nous prouvons (pour le nouveau système d'inférence) le théorème suivant:

Théorème 3.24 La satisfaisbilité de systèmes de contraintes bien-formés est décidable.

L'idée de la preuve est de borner les coefficients λ de la première contrainte. Observons que l'origination et le déterminisme assurent que la première contrainte contient une seule variable.

Lemme 3.25. Soit S un système de contraintes bien-formé et $v_1, \ldots, v_k \Vdash \alpha X + \gamma$ la première contrainte dans S. Si S est satisfaisable, il a une solution σ t.q: $\alpha X\sigma + \gamma = \Sigma_i \lambda_{1i}^{\sigma} v_i$, avec $0 < \lambda_{1i} < \alpha + max_c(\gamma)$, pour tout $1 \leq i \leq k$.

Preuve. Premièrement, nous montrons que s'il existe une solution (θ, Λ^θ), il existe une solution (δ, Λ^δ) avec $\lambda_{1i}^\delta > 0$, pour tout $1 \leq i \leq k$. Soit une solution θ telle que pour un j: $\lambda_{1j}^\theta < 0$. Alors on obtient une solution δ avec $\lambda_{1j}^\delta = \lambda_{1j}^\theta + \alpha$ et $\lambda_{1i}^\delta = \lambda_{1i}^\theta$, pour $i \neq j$, en prenant $X\delta = X\sigma + v_j$ (v_j est clos par origination). Vérifions que δ est une solution de S.

Soit $C = v_1, \ldots, v_k, T \Vdash u \in S$. Comme θ est une solution de S, il existe Λ^θ t.q $Diff(C, \theta, \Lambda^\theta) = 0$. Puisque $\delta - \theta$ est un multiple de v_j, on déduit que $Diff(C, \delta, \Lambda^\theta) = mv_j$, pour un entier m. Donc, en prenant $\lambda_j^\delta = \lambda_j^\theta - m$ et $\lambda_i^\delta = \lambda_i^\theta$ pour $i \neq j$, on obtient $Diff(C, \delta, \Lambda^\delta) = 0$.

Observons que si C est la première contrainte, $m = -\alpha$ et on a effectivement $\lambda_{1j}^\delta = \lambda_{1j}^\theta + \alpha$. Puisque cette transformation de solution accroît λ_{1j} sans affecter λ_{1i} pour $i \neq j$, on peut l'appliquer successivement pour obtenir une solution δ avec $\lambda_{1i}^\delta > 0$ pour tout i.

Soit maintenant une solution δ de S avec $\lambda^{\delta}_{1i} > 0$ pour tout i, et prouvons qu'il existe une solution σ avec $0 < \lambda_{1i} < \alpha + max_c(\gamma)$. Supposons que $\lambda^{\delta}_{1j} \geq \alpha + max_c(\gamma)$. Puisque δ est une solution, on a $\alpha X\delta + \gamma = \lambda^{\delta}_{1j} v_j + \Sigma_{i \neq j} \lambda^{\delta}_{1i} v_i$. Comme $\lambda^{\delta}_{1j} \geq \alpha + max_c(\gamma)$ et $\lambda^{\delta}_{1i} > 0$ pour tout i, on déduit que $X\delta > v_j$.

On peut donc construire σ: $X\sigma = X\theta - v_j$ pour obtenir une solution avec $\lambda^{\sigma}_{1j} = \lambda^{\delta}_{1j} - \alpha$ et $\lambda^{\sigma}_{1i} = \lambda^{\delta}_{1i}$ pour tout $i \neq j$. La vérification que σ est une solution est effectuée comme pour δ ci-dessus, puisque ici on a aussi $\sigma - \delta$ multiple de v_j. En repetant cette construction on obtient la solution σ cherchée. □

Corrolaire 3.26 Soit S un système de contraintes bien-formé et X la première variable apparaissant dans S. Il existe un ensemble de systèmes de contraintes bien-formés $C = \{S_1, \ldots, S_m\}$ tel que:

- S satisfaisable ssi $\exists S_i \in C :$ S_i satisfaisable.
- $\forall S_i \in C :$ $Var(S_i) = Var(S) \backslash \{X\}$.

Preuve. Soit $v_1, \ldots, v_k \Vdash \alpha X + \gamma$ la première contrainte de S et l'ensemble d'équations: $E_X = \{c_1 v_1 + \ldots + c_k v_k = \alpha X + \gamma \mid c_1, \ldots, c_k < \alpha + max_c(\gamma)\}$. Soit $S_X = \{\sigma \mid \exists e \in E_X : X\sigma \textit{ solution de } e\}$. Puisque tous les équations dans E_X ont une seule variable, S_X est calculable. Le lemme 3.25 nous dit que C est satisfaisable ssi $\exists \sigma \in S_X$ tq $C\sigma$ est satisfaisable. Donc $S = \{C\sigma \mid \sigma \in S_X\}$ satisfait l'enoncé du lemme. □

On conclut la preuve du Théorème 3.24 en observant qu'on peut eliminer toutes les variables d'un système de contraintes bien-formé C, par l'application successive du Corrolaire 3.26. On obtient alors un système linéaire qu'on sait résoudre.

4 Conclusion et perspectives

Le pouvoir de déduction d'un intrus ainsi que les propriétés algébriques de primitives cryptographiques peuvent varier suivant les protocoles considérés et les applications en vue. Pour vérifier automatiquement des propriétés dans ces modèles, une approche consiste à réduire la théorie équationelle à une théorie plus simple: AC [CLD05].

Suivant cette voie, dans ce rapport nous nous sommes intéressés à un modèle de l'intrus simplifié, en présence d'un opérateur associatif et commutatif. Nous avons montré 3 résultats:

- Indécidabilité dans le cas général.
- Un algorithme de décision avec des hypothéses sur les occurences de variables.
- Un algorithme de décision pour un système de déduction simple et des contraintes bien-formées.

L'étude de la résolution de systèmes de contraintes dans ce modèle est une brique de base pour un travail dont l'objectif est de pouvoir ensuite traîter systématiquement des systèmes plus complexes, en combinant des procédures de décision sur des systèmes simples.

Un instance de ce problème est le cas du porte-monnaie électronique mentionné dans l'introduction [Del06]. La modélisation donne le système d'inférence suivant:

$$(\Sigma)\frac{x_1 \quad \dots \quad x_n}{x_1 + \ldots + x_n} \qquad (h)\frac{x}{h(x)} \qquad (\pi)\frac{h(x_1) \quad \dots \quad h(x_n)}{h(x_1 + \ldots + x_n)}$$

$$(exp_+)\frac{h(h(x)) \quad h(y_1) \quad \dots \quad h(y_n)}{h(h(x + y_1 \ldots + y_n))}$$

$$(\Pi)\frac{x_1 \quad \dots \quad x_n}{x_1 * \ldots * x_n} \qquad (exp_*)\frac{h(x) \quad y}{h(x * y)},$$

où $+$ est AC et $*$ AG (groupe abélien).

Un premier pas de décomposition est de considérer les deux systèmes $I_+ = \{\Sigma, h, \pi, exp_+\}$ et $I_* = \{\Pi, h, exp_*\}$. Puisque I_+ et I_* sont d'une certaine manière disjoints, l'espoir est de pouvoir combiner facilement une procédure de décision pour I_+ avec une autre pour I_*. Le système I_* est connu décidable et a été traité récemment par des techniques de combinaison [CR06]. L'approche présentée dans [CR06] ne s'applique pas pour I_+. Un thème de recherche actuel consiste à généraliser ce résultat pour pouvoir traîter I_+, éventuellement en le réduisant à un système de déduction étudié dans ce rapport. Le schéma utilisé

pour se ramener a un système avec seulement des constantes et le symbole + consiste à borner le nombre des termes de la forme $h(v)$ utilisés dans les preuves, à deviner un ordre de déduction de ces termes et les remplacer ensuite par des constantes.

Un objectif plus général est d'obtenir des conditions suffisantes pour combiner deux procédures de decision. Dans [CR05] est montré qu'on peut combiner les procédures de décision des systèmes sur des signatures disjointes. Mais comme l'example du porte-monnaie et d'autres applications le montrent, cette condition est trop restrictive et ne permet pas de traîter beaucoup des cas pratiques. Il nous faut donc un résultat de combinaison plus général pour réaliser la vérification automatique des protocoles cryptographiques en tenant compte des propriétés algébriques.

Bibliographie

[BGW01] N. Borisov, I. Goldberg, and D. Wagner. Intercepting mobile communications: the insecurity of 802.11. *MOBICOM*, pages 180–189, 2001.

[CKR+03] Y. Chevalier, R. Küsters, M. Rusinowitch, M. Turuani, and M. Vigneron. Extending the Dolev-Yao intruder for analyzing an unbounded number of sessions. *CSL*, pages 128–141, 2003.

[CKRT03] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. An NP decision procedure for protocol insecurity with xor. *LICS*, pages 261–270, 2003.

[CKRT04] Y. Chevalier, R. Küsters, M. Rusinowitch, and M. Turuani. Deciding the security of protocols with commuting public key encryption. *ARSPA*, 2004.

[CLC03] H. Comon-Lundh and V. Cortier. New decidability results for fragments of first-order logic and application to cryptographic protocols. *RTA*, pages 148–164, 2003.

[CLD05] H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. *RTA*, pages 294–307, 2005.

[CLS03] H. Comon-Lundh and V. Shmatikov. Intruder deductions, constraint solving and insecurity decision in presence of exclusive or. *LICS*, pages 271–280, 2003.

[CR05] Y. Chevalier and M. Rusinowitch. Combining intruder theories. *ICALP*, pages 639–651, 2005.

[CR06] Y. Chevalier and M. Rusinowitch. Hierarchical combination of intruder theories. *RTA*, pages 108–122, 2006.

[Del06] Stéphanie Delaune. *Vérification des protocoles cryptographiques et propriétés algébriques*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, June 2006.

[DLLT06] S. Delaune, P. Lafourcade, D. Lugiez, and R. Treinen. Symbolic protocol analysis in presence of a homomorphism operator and exclusive or. *ICALP*, pages 132–143, 2006.

[LLT05] P. Lafourcade, D. Lugiez, and R. Treinen. Intruder deduction for AC-like equational theories with homomorphisms. *RTA*, pages 308–322, 2005.

[MS03] J. Millen and V. Shmatikov. Symbolic protocol analysis with products and Diffie-Hellman exponentiation. *CSFW*, pages 47–61, 2003.

[MS05] J. Millen and V. Shmatikov. Symbolic protocol analysis with an abelian group operator or Diffie-Hellman exponentiation. *Journal of Computer Security*, 13:515–564, 2005.

[RT03] M. Rusinowitch and M. Turuani. Protocol insecurity with finite number of sessions and composed keys is NP-complete. *Theoretical Computer Science*, 299:451–475, 2003.

[Sch86] A. Schrijver. *Theory of Linear and Integer Programming*. Wiley, 1986.

[Sim94] G. Simmons. Cryptoanalysis and protocol failures. *Communications of the ACM*, 37:56–65, 1994.