

Definition 1 (constraint system) A constraint system \mathcal{C} is either \perp or a finite sequence of expressions $(T_i \Vdash u_i)_{1 \leq i \leq n}$, called constraints, where each T_i , called the left-hand side of the constraint, and each u_i is a term, called the right-hand side of the constraint, such that :

1. (monotonicity)
init $\in T_1$ and $T_i \subseteq T_{i+1}$ for every i such that $1 \leq i < n$;
2. (origination property)
if $x \in \text{vars}(T_i)$ then $\exists j < i$ such that $T_j = \min\{T \mid (T \Vdash u) \in \mathcal{C}, x \in \text{vars}(u)\}$ (for the inclusion relation) and $T_j \subsetneq T_i$.

A solution of \mathcal{C} is a closed substitution θ with $\text{dom}(\theta) = \text{vars}(\mathcal{C})$ such that for every $(T \Vdash u) \in \mathcal{C}$, we have that $T\theta \vdash u\theta$. The empty constraint system is always satisfiable whereas \perp denotes an unsatisfiable system.

Definition 2 (solved form) A constraint system is said solved if it is different from \perp and if each of its constraints is of the form $T \Vdash x$, where x is a variable.

Remark : Note that the empty constraint system is solved. Solved constraint systems always have a solution. Why ?

The *simplification rules* we consider are given below.

$$\begin{array}{ll}
 R_1 : & \mathcal{C} \wedge T \Vdash u \rightsquigarrow \mathcal{C} \quad \text{if } T \cup \{x \mid T' \Vdash x \in \mathcal{C}, T' \subsetneq T\} \vdash u \\
 R_2 : & \mathcal{C} \wedge T \Vdash u \rightsquigarrow_{\sigma} \mathcal{C}\sigma \wedge T\sigma \Vdash u\sigma \quad \text{if } \sigma = \text{mgu}(t, u) \text{ where } t \in \text{st}(T), t \neq u, \\
 & \text{and } t, u \text{ are not variables} \\
 R_3 : & \mathcal{C} \wedge T \Vdash u \rightsquigarrow_{\sigma} \mathcal{C}\sigma \wedge T\sigma \Vdash u\sigma \quad \text{if } \sigma = \text{mgu}(t_1, t_2), t_1, t_2 \in \text{st}(T) \setminus \mathcal{X}, t_1 \neq t_2, \\
 R_4 : & \mathcal{C} \wedge T \Vdash u \rightsquigarrow \perp \quad \text{if } \text{vars}(T \cup \{u\}) = \emptyset \text{ and } T \not\vdash u \\
 R_5 : & \mathcal{C} \wedge T \Vdash \langle u_1, u_2 \rangle \rightsquigarrow \mathcal{C} \wedge T \Vdash u_1 \wedge T \Vdash u_2 \\
 R_6 : & \mathcal{C} \wedge T \Vdash \{u_1\}_{u_2} \rightsquigarrow \mathcal{C} \wedge T \Vdash u_1 \wedge T \Vdash u_2
 \end{array}$$

All the rules are indexed by a substitution (when there is no index then the identity substitution is assumed). We write $\mathcal{C} \rightsquigarrow_{\sigma}^* \mathcal{C}'$ if there are constraint systems $\mathcal{C}_1, \dots, \mathcal{C}_n$ such that $\mathcal{C} \rightsquigarrow_{\sigma_0} \mathcal{C}_1 \rightsquigarrow_{\sigma_1} \dots \rightsquigarrow_{\sigma_n} \mathcal{C}'$ and $\sigma = \sigma_0\sigma_1 \dots \sigma_n$.

Lemma 1 The simplification rules transform a constraint system into a constraint system.

Theorem 1 Let \mathcal{C} be an unsolved constraint system.

1. (Termination) There is no infinite chain $\mathcal{C} \rightsquigarrow_{\sigma_1} \mathcal{C}_1 \dots \rightsquigarrow_{\sigma_n} \mathcal{C}_n$.
2. (Correctness) If $\mathcal{C} \rightsquigarrow_{\sigma}^* \mathcal{C}'$ for some constraint system \mathcal{C}' and some substitution σ and if θ is a solution of \mathcal{C}' then $\sigma\theta$ is a solution of \mathcal{C} .
3. (Completeness) If θ is a solution of \mathcal{C} , then there exist a solved constraint system \mathcal{C}' and substitutions σ, θ' such that $\theta = \sigma\theta'$, $\mathcal{C} \rightsquigarrow_{\sigma}^* \mathcal{C}'$ and θ' is a solution of \mathcal{C}' .

Some definitions and lemmas to establish completeness.

Definition 3 (simple proof) Let $T_1 \subseteq T_2 \subseteq \dots \subseteq T_n$. We say that a proof π of $T_i \vdash u$ is left-minimal if for any $j < i$ such that $T_j \vdash u$, π' is a proof of $T_j \vdash u$ where π' is obtained from π by replacing T_i with T_j in the left-hand side of each node of π . We say that a proof is simple if any subproof is left-minimal and on any branch there are no two equal nodes.

Remark : a subproof of a simple proof is also simple.

Lemma 2 If $T_i \vdash u$ then there is a simple proof of it.

Lemma 3 Let \mathcal{C} be an unsolved constraint system, θ be a solution of \mathcal{C} and $T_i \Vdash u_i$ be a minimal unsolved constraint of \mathcal{C} . Let u be a term. If there is a simple proof of $T_i\theta \vdash u$ having the last rule an axiom or a decomposition then there is $t \in st(T_i) \setminus \mathcal{X}$ such that $t\theta = u$.

Lemma 4 Let \mathcal{C} be an unsolved constraint system, θ be a solution of \mathcal{C} and $T_i \Vdash v_i$ be a minimal unsolved constraint of \mathcal{C} such that for all $t_1, t_2 \in st(T_i)$ such that $t_1 \neq t_2$

$$t_1\theta = t_2\theta \text{ implies } t_1 \text{ or } t_2 \text{ is a variable}$$

Assume $u \in st(T_i) \setminus \mathcal{X}$ and $T_i\theta \vdash u\theta$. Then $T_i \cup \{x \mid T \Vdash x \in \mathcal{C}, T \subsetneq T_i\} \vdash u$.

Exercice 1

Démontrez les lemmes et le théorème.

Exercice 2

Montrer que le problème de savoir si un système de contraintes est satisfaisable est :

1. décidable,
2. NP-difficile. *Indication : on pourra coder le problème 3-SAT*

Exercice 3

Soit le protocole suivant :

$$\begin{aligned} \text{Message 1. } A \rightarrow B &: \langle \{ \langle N_A, k \rangle \}_{K_{AB}}, A \rangle \\ \text{Message 2. } B \rightarrow A &: \langle \{ \langle N_A, N_B \rangle \}_{K_{AB}}, \langle \{ N_B \}_k, N_B \rangle \rangle \end{aligned}$$

On suppose que N_A et N_B sont des nonces frais et que k est une clé fraîche. On suppose également que K_{AB} est une clé partagée uniquement entre A et B . Supposons que les capacités de l'intrus sont modélisées par le système d'inférence de Dolev-Yao vu au cours.

1. Décrivez (informellement) comment un intrus peut compromettre le secret de la clé k (du point de vue du participant B).
2. Modélisez les rôles nécessaires à cette attaque.
3. Donnez le système de contraintes représentant l'entrelacement qui mène à cette attaque. Appliquez la procédure de résolution vue au cours pour montrer que ce système a une solution.

Exercice 4

Montrer que les règles de simplifications ne sont pas complètes si l'on considère du chiffrement asymétrique.

Indication : Le lemme 4 est faux en présence de chiffrement asymétrique.