### 4.3.2 Completeness

First, we show that proofs considered in solutions of constraints can be narrowed to so-called *simple proofs*. Let $T_1 \subseteq T_2 \subseteq \ldots \subseteq T_n$ . We say that a proof $\Pi$ of $T_i \vdash u$ is *left minimal* if, whenever there is a proof of $T_j \vdash u$ for some $j < i$, then $\Pi$ is also a proof of $T_j \vdash u$. In other words, the left-minimal proofs are those that can be performed in a minimal $T_j$ . We say that a proof is *simple* if all its subproofs are left minimal and there is no repeated label on any branch. Note that a subproof of a simple proof is simple.

**Example 4.4** *Assume* $T_1 = \{a, b\}$, $T_2 = \{a, b, \langle a, b\rangle\}$ *and* $T_3 = \{a, b, c, \langle a, b\rangle\}$.

- *The following proof is local but not simple:*

$$\frac{a \quad \langle a, b\rangle}{\langle a, \langle a, b\rangle\rangle}$$

  *Since the set of hypotheses is not contained in* $T_1$*, while there is a proof of* $T_1 \vdash \langle a, \langle a, b\rangle\rangle$.

- *The following proof is local but not simple:*

$$\frac{c \quad \langle a, b\rangle}{\langle c, \langle a, b\rangle\rangle}$$

  *Since there is a subproof, which is not simple.*

- $$\frac{c \quad \dfrac{a \quad b}{\langle a, b\rangle}}{\langle c, \langle a, b\rangle\rangle}$$

  *is a simple proof*

**Lemma 4.3** *Let* $T_1 \subseteq T_2 \subseteq \ldots \subseteq T_n$ *be a sequence of sets of terms and* $u$ *be a term such that* $T_i \vdash u$. *There exists a simple proof* $\Pi$ *of* $T_i \vdash u$.

*Proof :* Let $i$ be a minimal index for which there is a proof of $T_i \vdash u$. Thanks to Lemma 4.1, there is a local proof $\Pi_0$ of $T_i \vdash u$. We prove the lemma by induction on the size of $\Pi_0$.

*Base case:* $\Pi_0$ is reduced to a leaf. In such a case, $\Pi_0$ is a simple proof.
*Induction step:* Consider the last rule in the proof of $u$:

$$\Pi_0 = \left\{ \frac{\begin{array}{ccc}\Pi_1 & & \Pi_n \\ & \cdots & \\ u_1 & & u_n\end{array}}{u}\ \mathsf{R} \right.$$

For every $j = 1, ..., n$, we have that $\Pi_j$ is a proof of $T_i \vdash u_j$. By induction hypothesis, there are simple proofs $\Pi'_j$ of $u_j$. If $u$ appears as a node in some of these proofs, let $\Pi$ be the corresponding subproof and we get the desired result. Otherwise, let

$$\Pi = \left\{ \frac{\begin{array}{ccc}\Pi'_1 & & \Pi'_n \\ & \cdots & \\ u_1 & & u_n\end{array}}{u}\ \mathsf{R} \right.$$

The proof $\Pi$ is a simple proof of $u$. □

**Lemma 4.4** *Let $\mathcal{C}$ be an unsolved constraint system, $\theta$ be a solution of $\mathcal{C}$ and $T_i \overset{?}{\vdash} u_i$ 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 $t\theta = u$.*

*Proof :* Let $\Pi$ be a simple proof of $T_i\theta \vdash u$ such that its last rule is an axiom or a decomposition. Let $j$ be the minimal index such that $T_j\theta \vdash u$. Note that $j \leq i$ and by definition of a simple proof, we have that $\Pi$ is also a simple proof of $T_j\theta \vdash u$.

We prove the lemma, by induction on $\Pi$.

- The last rule is an axiom. Then $u \in T_j\theta$. There is $t \in T_j$ (thus $t \in st(T_j)$) such that $t\theta = u$. If $t$ is a variable then $T_t \overset{?}{\vdash} t$ is a constraint in $\mathcal{C}$ with $T_t \subsetneq T_j$ (see the definition of a constraint system). Hence $T_t\theta \vdash t\theta$, that is $T_t\theta \vdash u$, which contradicts the minimality of $j$. Thus, as required, $t$ is not a variable.
- The last rule is a decomposition. Suppose that it is a symmetric decryption. That is, there is $w$ such that $T_j\theta \vdash \mathsf{senc}(u, w)$, and $T_j\theta \vdash w$. By simplicity of the proof, the last rule applied when obtaining $\mathsf{senc}(u, w)$ is an axiom or a decomposition, otherwise the same node would appear twice. Then, applying the induction hypothesis we have that there is $t \in st(T_j) \setminus \mathcal{X}$ such that $t\theta = \mathsf{senc}(u, w)$. It follows that $t = \mathsf{senc}(t', t'')$ with $t'\theta = u$. If $t'$ is a variable then $T_{t'}\theta \vdash t'\theta$. That is $T_{t'}\theta \vdash u$, which again contradicts the minimality of $j$. Hence $t'$ is not variable, as required.

For the other decomposition rules the same reasoning holds. □

**Lemma 4.5** *Every simple proof is local*

The proof is left as an exercise.

**Lemma 4.6** *Let $\mathcal{C} = T_0 \overset{?}{\vdash} x_0, \ldots, T_{i-1} \overset{?}{\vdash} x_{i-1}, T_i \overset{?}{\vdash} u, \ldots$ be a constraint system and $\sigma$ be a solution of $\mathcal{C}$ such that*

1. *$T_i$ does not contain two distinct subterms $t_1, t_2$ with $t_1\sigma = t_2\sigma$,*
2. *$u$ is a non-variable subterm of $T_i$.*

*Then $T_i' \vdash u$, where $T_i' = T_i \cup \{x \mid (T \overset{?}{\vdash} x) \in \mathcal{C}, T \subsetneq T_i\}$.*

*Proof :* Let $j$ be minimal such that $T_j\sigma \vdash u\sigma$. Thus $j \leq i$ and $T_j \subseteq T_i$. Consider a simple proof $\Pi$ of $T_j\sigma \vdash u\sigma$. We reason by induction on the depth of $\Pi$.

*Base case:* $\Pi$ is reduced to a leaf. Then there is $t \in T_j$ such that $t\sigma = u\sigma$. By hypothesis 1, we deduce that $t = u$. Hence, we have that $u \in T_j$ and thus $T_i' \vdash u$, as required.

*Induction step:* We analyse the different cases, depending on the last rule $\mathsf{R}$ of $\Pi$:

- *Case* $\mathsf{R}$ is a composition rule. Assume for example that $\mathsf{R} = \mathsf{SE}$. In such a case, we have that:

$$\Pi = \left\{ \begin{array}{cc} \Pi_1 & \Pi_2 \\ v_1 & v_2 \\ \hline \mathsf{senc}(v_1, v_2) \end{array} \right.$$

with $u\sigma = \mathsf{senc}(v_1, v_2)$. Since $u$ is not a variable, $u = \mathsf{senc}(u_1, u_2)$, $u_1\sigma = v_1$, and $u_2\sigma = v_2$. If $u_1$ (resp. $u_2$) is a variable then $u_1$ (resp. $u_2$) belongs to $fv(T_i)$ since $u \in st(T_i)$. Again, this implies $u_1 \in T_i'$ (resp. $u_2 \in T_i'$). Otherwise, $u_1$ (resp. $u_2$) is not a variable. Then, by induction hypothesis, $T_i' \vdash u_1$ (resp. $T_i' \vdash u_2$). Hence in both cases we have that $T_i' \vdash u_1$ and $T_i' \vdash u_2$. This allows us to conclude that $T_i' \vdash u$.

- *Case* $\mathsf{R} = \mathsf{SD}$. In such a case, there is $w$ such that $T_j\sigma \vdash \mathsf{senc}(u\sigma, w)$, and $T_j\sigma \vdash w$:

$$\Pi = \left\{ \begin{array}{cc} \Pi_1 & \Pi_2 \\ \mathsf{senc}(u\sigma, w) & w \\ \hline u\sigma \end{array} \right.$$

  By simplicity, the last rule of the proof $\Pi_1$ is a decomposition or an axiom. By Lemma 4.4, there is $t \in st(T_j) \setminus \mathcal{X}$ such that $t\sigma = \mathsf{senc}(u\sigma, w)$. Let $t = \mathsf{senc}(t_1, t_2)$ with $t_1\sigma = u\sigma$, and $t_2\sigma = w$. By induction hypothesis, $T_i' \vdash t$. Since $t_1\sigma = u\sigma$, by hypothesis 1, we have that $t_1 = u$.

  Now, if $t_2$ is a variable, and since $t_2 \in fv(T_i)$, we have that $T_{t_2} \subsetneq T_i$ and thus $t_2 \in T_i'$. If $t_2$ is not a variable, then, from $T_j\sigma \vdash t_2\sigma$ and by induction hypothesis, $T_i' \vdash t_2$. So, in any case, $T_i' \vdash t_2$.

  Hence, we have both that $T_i' \vdash \mathsf{senc}(u, t_2)$ and $T_i' \vdash t_2$, from which we conclude that $T_i' \vdash u$, by symmetric decryption.

- *Case* $\mathsf{R} = \mathsf{PKD}$. In such a case, there is $w$ such that $T_j\sigma \vdash \mathsf{sk}(w)$ and $T_j\sigma \vdash \mathsf{aenc}(u\sigma, w)$. As in the previous case, there is $t \in st(T_j) \setminus \mathcal{X}$ such that $t\sigma = \mathsf{aenc}(u\sigma, w)$. By induction hypothesis, $T_i' \vdash t$. Let $t = \mathsf{aenc}(t_1, t_2)$. As in the previous case, we have that $t_1\sigma = u\sigma$, and thus $t_1 = u$ (thanks to hypothesis 1).

  The last rule in the proof of $T_j\sigma \vdash \mathsf{sk}(w)$ is a decomposition (no composition rule can yield a term headed with $\mathsf{sk}(\_)$). Then, by Lemma 4.4 ($T_j$ satisfies the hypotheses of the lemma since $T_j \subseteq T_i$), there is a non-variable subterm $w_1 \in st(T_j)$ such that $w_1\sigma = \mathsf{sk}(w)$. Let $w_1 = \mathsf{sk}(w_2)$. By induction hypothesis, $T_j' \vdash \mathsf{sk}(w_2)$. Moreover, since $w_2\sigma = t_2\sigma$, by hypothesis 2, we have that $w_2 = t_2$,

  Finally, from $T_i' \vdash \mathsf{aenc}(u, w_2)$ and $T_i' \vdash \mathsf{sk}(w_2)$, we conclude that $T_i' \vdash u$.

The proof is similar for the other decomposition rules. □

**Proposition 4.2 (Completeness for one step)** *If $\mathcal{C}$ is an unsolved deducibility constraint system and $\theta$ is a solution of $\mathcal{C}$, then there is a deducibility constraint system $\mathcal{C}'$, a substitution $\sigma$, and a solution $\theta'$ of $\mathcal{C}'$ such that $\mathcal{C} \rightsquigarrow_\sigma \mathcal{C}'$ and $\theta = \sigma\theta'$.*

*Proof :* Let $\mathcal{C}$ be an unsolved constraint system and $\theta$ be a solution of $\mathcal{C}$. We show that there is a constraint system $\mathcal{C}'$ and a solution $\theta'$ of $\mathcal{C}'$ such that $\mathcal{C} \rightsquigarrow_\sigma \mathcal{C}'$ and $\theta = \sigma\theta'$.

Consider a minimal unsolved constraint $T_i \overset{?}{\vdash} u_i$ such that $u_i$ is not a variable. We have that $T_i\theta \vdash u_i\theta$. Consider a simple proof $\Pi$ of $T_i\theta \vdash u_i\theta$. We analyse the different cases depending on the last rule of $\Pi$.

1. *The last rule is a composition.* Suppose that it is the pairing rule. That is, there are $w_1, w_2$ such that $T_i\theta \vdash w_1$, $T_i\theta \vdash w_2$ and $\langle w_1, w_2\rangle = u_i\theta$. Since $u_i$ is not a variable there exists $u', u''$ such that $u_i = \langle u', u''\rangle$. Hence we can apply the simplification rule $\mathsf{R_f}$ in order to obtain $\mathcal{C}'$. Since $u'\theta = w_1$ and $u''\theta = w_2$, the substitution $\theta$ is also a solution to $\mathcal{C}'$. For the other composition rules the same reasoning holds.

2. *The last rule is an axiom or a decomposition.* Applying Lemma 4.4 we obtain that there is $t \in st(T_i) \setminus \mathcal{X}$ such that $t\theta = u_i\theta$. We can have the following two possibilities:

   (a) If $t \neq u_i$ then we apply the simplification rule $\mathsf{R_2}$.

   (b) Otherwise, if $t = u_i$, then $u_i \in st(T_i)$ and we already know that $u_i$ is not a variable. We consider two cases:

i. There are two distinct terms $t_1, t_2 \in st(T)$ such that $t_1\theta = t_2\theta$. Then we apply the simplification rule $\mathsf{R}_3$.

ii. Otherwise, the simplification rule $\mathsf{R}_1$ can be applied (Lemma 4.6). □

### 4.3.3 Complexity

The termination stated in Theorem 4.1 does not provide with tight complexity bounds. In fact, applying the simplification rules may lead to branches of exponential length in the size of the constraint system [113]. Inspecting the completeness proof, there is still some room for choosing a strategy to ensure that the length of each branch is polynomially bounded in $\mathcal{C}$ (while keeping completeness). Note that correctness is independent of the order of the rules application.

Moreover, for any suitable representation of terms, we have that $|u\sigma, v\sigma| < |u, v|$ where $\sigma = \mathsf{mgu}(u, v)$. Hence, if we use a DAG representation of terms, when $\mathcal{C} \rightsquigarrow_\sigma^* \mathcal{C}'$, we have that the size of $\mathcal{C}'$ is polynomially bounded in the size of $\mathcal{C}$. As a consequence, the security problem is in co-NP and it is actually co-NP-complete [234]. The NP-hardness can be established with a reduction from 3-SAT.

## 4.4 Further Readings

Many parts of this section are borrowed from [113]. Hence, more details can be found in this paper. Another decision procedure based on constraint simplification rules has been proposed by J. Millen and V. Shmatikov [205]. Many results (*e.g.* [114, 41]) have been obtained within this framework. In particular, this framework has been extended by several authors to deal with algebraic properties of cryptographic primitives.

## 4.5 Exercises

**Exercice 9**
Say whether each couple of terms are unifiable or not. If so, give a most general unifier ($\mathsf{mgu}$).

1. $\langle x, b\rangle$ and $\langle a, y\rangle$,
2. $\mathsf{aenc}(x, a)$ and $\mathsf{aenc}(b, x)$,
3. $\langle x, y\rangle$ and $\langle\langle y, y\rangle, a\rangle$,
4. $z$ and $\langle x, y\rangle$.

**Exercice 10 ($\star$)**
Consider the following inference system:

$$\frac{x \quad y}{\langle x, y\rangle} \qquad \frac{\langle x, y\rangle}{x} \qquad \frac{\langle x, y\rangle}{y} \qquad \frac{x \quad y}{\mathsf{senc}(x, y)} \qquad \frac{\mathsf{senc}(x, y) \quad y}{x}$$

Let $T = \{\mathsf{senc}(s, \langle k_1, k_2\rangle),\ \mathsf{senc}(k_1, k_3),\ k_3,\ k_2\}$.

1. Enumerate all the subterms of $T$.
2. The term $s$ is deducible from $T$. Give a derivation witnessing this fact.
3. Among the subterms of $T$, give those that are deducible.
4. Give a term $u$ that is not a subterm of $T$ and such that $T \vdash u$.

**Exercice 11 ($\star\star\star$)**
Consider the following inference system:

$$\frac{x \quad y}{\langle x, y\rangle} \qquad \frac{\langle x, y\rangle}{x} \qquad \frac{\langle x, y\rangle}{y} \qquad \frac{x \quad y}{\mathsf{senc}(x, y)} \qquad \frac{\mathsf{senc}(x, y) \quad y}{x}$$

In order to decide whether a term $s$ is deducible from a set of terms $T$ in the inference system described above, we propose the following algorithm:

*Algorithm:*

1. Apply as much as possible the decryption and the projection rules. This leads to a set of terms called $\mathsf{analz}(T)$.

2. Check whether $s$ can be obtained by applying the encryption and the pairing rules. The (infinite) set of terms obtained by applying the composition rules is denoted $\mathsf{synth}(\mathsf{analz}(T))$.

If $s \in \mathsf{synth}(\mathsf{analz}(T))$ then the algorithm return *yes*. Otherwise, it returns *no*.

1. Show that this algorithm terminates.

2. Show that this algorithm is sound, *i.e.* if the algorithm returns yes then $T \vdash s$.

3. The algorithm is not complete, *i.e.* there exist $T$ and $s$ such that $T \vdash s$, and for which the algorithm returns no. Find an example illustrating this fact.

4. Give an hypothesis on $T$ that allows one to restore completeness.

5. Show that the algorithm is complete when this hypothesis is fulfilled.

**Exercice 12 ($\star$)**
We consider the following inference system allowing us to model asymmetric encryption.

$$\frac{x \quad y}{\mathsf{aenc}(x, y)} \qquad \frac{\mathsf{aenc}(x, \mathsf{pk}(z)) \quad \mathsf{sk}(z)}{x} \qquad \frac{z}{\mathsf{pk}(z)}$$

Is this inference system local, or not? If so, give a proof. If not, give a derivation witnessing this fact.

**Exercice 13 ($\star\star$)**
Consider the following inference system allowing us to model digital signature.

$$\frac{x \quad \mathsf{sk}(z)}{\mathsf{sign}(x, \mathsf{sk}(z))} \qquad \frac{\mathsf{sign}(x, \mathsf{sk}(z)) \quad \mathsf{vk}(z)}{x} \qquad \frac{z}{\mathsf{vk}(z)}$$

1. This inference system is not local according to Definition 4.2. Give an example witnessing this fact.

2. Show that the intruder deduction problem is decidable.
   *You can use the technique described in this chapter and extend the notion of subterm to restore the locality property.*

**Exercice 14 ($\star$)**
We consider the signature and the inference system given in Example 4.1. Let $T_0 = \{a,\ b,\ c,\ \mathsf{sk}(c),\ \mathsf{aenc}(\langle a, \mathsf{aenc}$
and $\mathcal{C} = \{T_0 \overset{?}{\vdash} \mathsf{aenc}(\langle a, \mathsf{aenc}(x_1, b)\rangle, b)\}$. What are the solutions of $\mathcal{C}$?