TD 5: Deducibility constraints in the symbolic setting

Margot Catinaud margot.catinaud@lmf.cnrs.fr

Recall the definition of deducibility constraint system:

Definition 1: Deducibility constraints system

A *deducibility constraint* is an expression of the form $T \vdash^?_{\mathcal{I}} u$ where T is a non-empty set of terms, u a term, $\mathcal{I}$ (often omitted) is the deduction system used.

A *deducibility constraint system* is either $\bot$ or a (possibly empty[a]) conjunction of deducibility constraints of the form:

$$\bigwedge_{i=1}^{n} (T_i \vdash^? u_i)$$

such that

- ***monotonicity:*** for all $i \in [\![1; n-1]\!]$, $T_i \subseteq T_{i+1}$;
- ***origination:*** for all i, $\mathsf{fv}(T_i) \subseteq \bigcup_{j=1}^{i-1} \mathsf{fv}(u_j)$

[a] an empty conjunction is equivalent to $\top$

The goal is to try to solve a deducibility constraint system:

Definition 2: Solution of a constraint system

A substitution σ is a *solution* of $\mathcal{C} = \bigwedge_{i=1}^{n} (T_i \vdash^? u_i)$, a deducibility constraint system if for all $i \in [\![1; n]\!]$, there exist a proof of $T_i\sigma \vdash u_i\sigma$

To do so, we want a class of constraint system where it is "easy" to show that they have a solution:

Definition 3: Solved constraint system

A constraint system $\mathcal{C}$ is said to be *solved* if it is in the form

$$\mathcal{C} = \bigwedge_{i=1}^{n} (T_i \vdash^? x_i)$$

where, for all $i \in [\![1; n]\!]$, x_i is a variable in $\mathcal{X}$.

And then, we propose the following simplification algorithm

Definition 4: Simplification rules for constraint system

We consider a set of simplification rules for constraint system: [a]

$$\mathcal{C} \wedge (T \vdash^? u) \rightsquigarrow \mathcal{C} \qquad \text{if } T \cup \left\{ x \in \mathcal{X} \mid (T' \vdash^? x) \in \mathcal{C}, T' \subseteq T \right\} \vdash u \qquad (R_1)$$

$$\mathcal{C} \wedge (T \vdash^? u) \rightsquigarrow_\sigma \mathcal{C}\sigma \wedge (T\sigma \vdash^? u\sigma) \qquad \text{if } t \in \mathsf{st}(T), \sigma = \mathsf{mgu}(t, u), t \neq u \text{ and } t, u \notin \mathcal{X} \qquad (R_2)$$

$$\mathcal{C} \wedge (T \vdash^? u) \rightsquigarrow_\sigma \mathcal{C}\sigma \wedge (T\sigma \vdash^? u\sigma) \qquad \text{if } t, v \in \mathsf{st}(T), \sigma = \mathsf{mgu}(t, v), t \neq v \qquad (R_3)$$

$$\mathcal{C} \wedge (T \vdash^? u) \rightsquigarrow \bot \qquad \text{if } \mathsf{fv}(T \cup \{u\}) = \emptyset \text{ and } T \not\vdash u \qquad (R_4)$$

$$\mathcal{C} \wedge (T \vdash^? f(u_1, \ldots, u_n)) \rightsquigarrow \mathcal{C} \wedge \bigwedge_{i=1}^{n} (T \vdash^? u_i) \qquad \text{if } (f/n) \in \Sigma \text{ is a constructor symbol} \qquad (R_f)$$

[a] In this exercise session the constructor symbols are senc, aenc, $(\_,\_)$, blind and sign

Exercise 1 (From protocols to constraint system)

Recall the Needham-Schroeder protocol:

$$\begin{aligned} B &\to A : \mathsf{pk}(B) \\ A &\to B : \{A, N_A\}_{\mathsf{pk}(B)} \\ B &\to A : \{N_A, N_B\}_{\mathsf{pk}(A)} \\ A &\to B : \{N_B\}_{\mathsf{pk}(B)} \end{aligned}$$

We know (from TD2) that there is an attack on N_B in this protocol, the goal of this exercise is to find it using constraint system. To do this, we need to translate the protocol to some constraint systems. For that, we first express the protocols as a set of rules. The idea behind those rules is to represent one possible transition of the protocol. For example, from the first two interactions of the protocol

$$\begin{aligned} B &\to A : \mathsf{pk}(B) \\ A &\to B : \{A, N_A\}_{\mathsf{pk}(B)} \end{aligned}$$

For the first interaction we write the rule:

$$\to \mathsf{pk}(B) \tag{B.1}$$

This rule represents the fact that without any input (there is nothing at the left side of the arrow), B sends its public key. We can now write (A.1), representing the message that A sends to answer this (x here represents the fact that A is willing to talk to anyone. So on input x, it sends back $\{A, N_A\}_x$).

$$x \to \{A, N_A\}_x \tag{A.1}$$

1. Write (B.2) and (A.2) representing the two other messages.
   Hint: You can restrict the input using pattern matching to caracterize the inputs. For example, B only answers back when the message it gets as input as this form: $\{(\mathsf{pk}(A), y)\}_{\mathsf{pk}(B)}$

Now, we want to transform this rules into a constraint system. For that, we need an ordering on rules.

2. Is there any restriction made by the protocol on the ordering of the rules?

3. Respecting those restrictions, find an ordering of the rules that leads to an attack.

To transform this ordering into a constraint system, we need to know the initial knowledge of the adversary T_0, and then use the ordering

$$\mathcal{O} : \ \forall\, i \in [\![1; n]\!],\ u_i \to v_i$$

Then, the ordering $\mathcal{O}$ leads to the constraint system $\mathcal{C}$ defined as follows:

$$\mathcal{C} \stackrel{\text{def}}{=} \bigwedge_{i=1}^{n} \Big(\big(T_0, (v_j)_{j=1}^{i-1}\big) \vdash^? u_i \Big) \ \wedge\ \Big(\big(T_0, (v_j)_{j=1}^{n}\big) \vdash^? N_B \Big).$$

since here, we want to show (actually break) the secrecy of N_B

4. Give the initial knowledge of the adversary T_0.

5. Give the corresponding constraint system $\mathcal{C}$ with the transformation given above.

6. Complete the subsitution $\sigma = \{x \to \mathsf{pk}(C),\ y \to N_A\}$ to make it a solution of the constraint system $\mathcal{C}$ and then prove that it is indeed a solution.

Exercise 2 (Examples of simplification)

On all those examples, try to apply the simplification rules as much as you can:

1. $\mathsf{senc}(n,k) \vdash^? \mathsf{senc}(x,k)$
2. $\mathsf{senc}\big(\mathsf{senc}(t_1,k),k\big) \vdash^? \mathsf{senc}(x,k)$
3. $T \vdash^? x \ \wedge\ (T,n) \vdash^? y \ \wedge\ \Big(T,n,\mathsf{senc}\big(m,\mathsf{senc}(x,k)\big),\mathsf{senc}(y,k)\Big) \vdash^? m$
4. $T \vdash^? x \ \wedge\ T \vdash^? (x,x)$
5. $n \vdash^? x \ \wedge\ n \vdash^? \mathsf{senc}(x,k)$

Exercise 3 (Needham-Schroeder simplification)

Apply the simplification rules on the constraint system given for the Needham-Schroeder protocol at the previous exercise to get back σ.

Exercise 4 (Wide-mouthed frog)

The *wide-mouthed-frog* protocol is defined as follows:

$$A \to S : A, \{B, s, m_1\}_{\mathsf{k}_{AS}}$$
$$S \to B : \{A, s, m_2\}_{\mathsf{k}_{BS}}$$

Here, we are interested in the case where A agrees to talk to both B and C (a dishonest agent).

1. Write the protocol as rules with input and output.
2. Show that the associated constraint system does not have a solution, you can use the simplification rules to do so.

Next, we propose a variant of the *wide-mouthed-frog* protocol:

$$A \to S : A, B, \{s\}_{\mathsf{k}_{AS}}$$
$$S \to B : A, \{s\}_{\mathsf{k}_{BS}}$$

3. Write down the protocol as rules with input and output.
4. Show that the associated constraint system does have a solution. Exhibit a solution using the constraint solving simplification. What can you conclude?