Logique
TD n°11

Luc Chabassier
chabassier@lsv.fr
Amélie Ledein
ledein@lsv.fr

April 13, 2022

Semantic trees

Exercise 1 : Draw me a tree

Let $\mathcal{L}$ be the language containing a constant function symbol a, a unary function symbol f, and two binary predicate symbols P, Q.

Consider the set of clauses $E = \{ P(x) , \neg P(f(x)) \vee \neg Q(a) , Q(a) \vee \neg P(f(a)) \}$ and enumeration of closed terms $P(a), Q(a), P(f(a)), Q(f(a)), \dots$.

Build the associated semantic tree.

Exercise 2 : Negative strategy

In this exercise, we focus on propositional formulas. The language contains only the constant predicate symbols $P_1, P_2, \dots$.

A clause is *negative* if it contains only negative literals. We study the following resolution strategy, called *negative strategy*: the application of the resolution rule is restricted to the case where one of the premisses is negative. We write $\vdash_\neg$ the associated deduction relation.

1. Let $E = \{ \neg P \vee Q , P \vee Q , P \vee \neg Q , \neg P \vee \neg Q \}$. Show that $E \vdash_\neg \bot$.

We will call a *partial interpretation* a node of a semantic tree associated to the enumeration $P_0, P_1, \dots$, represented by the set of literals chosen by this node.

If I and J are partial interpretations, we write $I >_{lex} J$ when there is $k \geq 1$ such that:

- for every $j < k$, $P_j \in I$ and $P_j \in J$, or $\neg P_j \in I$ and $\neg P_j \in J$;
- $P_k \in I$ and $\neg P_k \in J$.

As a reminder, $I \leq J$ if $I \subseteq J$.

2. What is the partial interpretation associated
   (a) to the root?

(b) to the left and right children of the root?

(c) to the left and right child of a node represented by partial interpretation I of size k?

3. Show that $\geq_{lex}$ is an order and that for all partial interpretations I et J, either $I \leq J$, $J \leq I$, $I \leq_{lex} J$, or $J \leq_{lex} I$.
4. Let A be the semantic tree of a set of clauses E. Assuming that A is finite and nonempty, show there exists a unique maximal partial interpretation for $\leq_{lex}$ being a leaf and not falsifying any negative clause of E.
5. Prove the refutational completeness of $\vdash_\neg$ using semantic trees.

   Hint: consider the maximal leaf for $\leq_{lex}$ not falsifying any negative clauses of E^* in the semantic tree of $E^* = \{ C \ : \ E \vdash_\neg C \}$.

   Another hint: I does not satisfy some formula of the form $P_i \vee C$. Consider the tree rooted in partial interpretation $I \cap \{ P_j, \neg P_j \ : \ 0 \leq j < i \} \cup \{ P_i \}$.

Herbrand and applications

Exercise 3 : An example

Let $\mathcal{L}$ be the language containing the binary predicate P and no function symbols, and S the set of the two formulas

$$\forall x \exists y P(x, y)$$
$$\exists x \forall y (P(x, y) \Rightarrow \exists z (P(x, z) \wedge \neg P(z, y)))$$

Build a language $\mathcal{L}'$ and a model over $\mathcal{L}'$ which is a model of S.

Exercise 4 : Löwenheim-Skolem

Let $\mathcal{L}$ be a countable language and S a set of formulas written over $\mathcal{L}$ which has a model. Show that S has a countable model.

The selection strategy and an application to security

Exercise 5 : Selection strategy

Let f be a function which, given a clause, returns one of its literals, called the *selected literal* of the clause. The *selection strategy* of function f restricts the resolution rule so that the literals on which resolution is performed are selected in their respective clauses.

1. Show that this strategy is not refutationally complete.
2. Is it refutationally complete if f chooses a negative literal whenever possible?

A *Horn clause* is a clause containing at most one positive literal (it can be seen as an implication). The selection strategy is refutationally complete for Horn clauses.

Exercise 6 : Security

We want to represent cryptographic protocols using Horn clauses. We will proceed using the following signature:

- Terms represent messages exchanged by participants.
- Cryptographic primitives are represented by functions:
  - pair(2) and aenc(2) are binary function symbols representing respectively pairs of messages and encryption of a message using a key.
  - pk(1) is a unary function symbol representing the public key of a participant.
  - $s(0)$ is a constant function symbol representing a secret.
  - $a(0)$, $b(0)$, $i(0)$ are constant function symbols representing the secret keys of the three participants Alice, Bob, and Impostor (the attacker).
- The attacker and her abilities are represented by a unary predicate att(1).

For example, the attacker can construct and deconstruct pairs, represented by Horn clauses in the following way:

$$\begin{aligned} \mathrm{att}(x) \wedge \mathrm{att}(y) &\Rightarrow \mathrm{att}(\mathrm{pair}(x,y)) \\ \mathrm{att}(\mathrm{pair}(x,y)) &\Rightarrow \mathrm{att}(x) \\ \mathrm{att}(\mathrm{pair}(x,y)) &\Rightarrow \mathrm{att}(y) \end{aligned}$$

She can also, given a public key, encrypt messages:

$$\mathrm{att}(m) \wedge \mathrm{att}(k) \Rightarrow \mathrm{att}(\mathrm{aenc}(m,k))$$

To decrypt messages, she needs the secret key of the associated participant:

$$\mathrm{att}(\mathrm{aenc}(m,\mathrm{pk}(p))) \wedge \mathrm{att}(p) \Rightarrow \mathrm{att}(m)$$

We also assume that the attacker has access to the public keys of other participants, represented by clauses $\mathrm{att}(\mathrm{pk}(a))$ and $\mathrm{att}(\mathrm{pk}(b))$, and that she has her own secret and public keys, represented by clauses $\mathrm{att}(i)$ and $\mathrm{att}(\mathrm{pk}(i))$. We name A the set of 9 clauses we just described.

1. Prove that if the attacker has access to an encrypted secret but not to the associated secret key, she cannot get the secret, i.e. one cannot derive $\bot$ from $A \cup \{\, \mathrm{att}(\mathrm{aenc}(s,\mathrm{pk}(a))) \,,\, \neg\mathrm{att}(s) \,\}$ using resolution.

   Hint: use resolution by selection, with selection function choosing literals of the form $att(t)$ or $\neg att(t)$ where t is not a variable when possible, else a positive literal, else an arbitrary literal.

Here is a cryptographic protocol:

- Participant A contacts participant B, encrypting with B's public key both her public key and a secret already encrypted with B's public key:

$$A \to B \;:\; \mathrm{pair}(\mathrm{pk}(a), \mathrm{aenc}(s, \mathrm{pk}(b)))$$

- Participant B responds with the secret encoded with A's public key:

$$B \to A \quad : \quad \mathrm{aenc}(s, \mathrm{pk}(a))$$

The attacker can intercept messages exchanged during this protocol, and transmit messages to Alice and Bob, which is represented by the Horn clauses:

$$\begin{array}{rcl} & & \mathrm{att}(\mathrm{pair}(\mathrm{pk}(a), \mathrm{aenc}(s, \mathrm{pk}(b)))) \\ \mathrm{att}(\mathrm{pair}(x, \mathrm{aenc}(y, \mathrm{pk}(b)))) & \Rightarrow & \mathrm{att}(\mathrm{aenc}(y, x)) \end{array}$$

We call P the set of 11 clauses containing A and the two above clauses.

2. Prove that an attack is possible on this protocol, i.e. one can derive $\bot$ from $P \cup \{ \neg\mathrm{att}(s) \}$.

3. A way to prevent this attack is to encrypt pair of the identity of A and the secret in the first message, yielding protocol

$$\begin{array}{rcl} A \to B & : & \mathrm{aenc}(\mathrm{pair}(\mathrm{pk}(a), s), \mathrm{pk}(b)) \\ B \to A & : & \mathrm{aenc}(s, \mathrm{pk}(a)) \end{array}$$

and associated clauses

$$\begin{array}{rcl} & & \mathrm{att}(\mathrm{aenc}(\mathrm{pair}(\mathrm{pk}(a), s), \mathrm{pk}(b))) \\ \mathrm{att}(\mathrm{aenc}(\mathrm{pair}(x, y), \mathrm{pk}(b))) & \Rightarrow & \mathrm{att}(\mathrm{aenc}(y, x)) \end{array}$$

Let P' be the set of eleven clauses containing A and the two above clauses. Show that we cannot derive $\bot$ from $P' \cup \{ \neg\mathrm{att}(s) \}$.

Hint: use resolution by selection with the same selection function as in the question 1 hint.