Problem Set for Tutorial 7 — TDDD08

Włodek Drabent and Victor Lagerkvist

1. Consider a program from a former tutorial, let us call it LONGER:

```
% Any [_|_] is longer than 0
longer([_|_], []).
% If Xs is longer than Ys then [_X|Xs] is longer than [_Y|Ys]
longer([_X|Xs], [_Y|Ys]) ← longer(Xs, Ys).
```

Explain with respect to which of the specifications below the program is not correct. Do this by providing a counterexample (a ground atom A which is an answer of LONGER but is not in the specification). Do the same about the program being not complete (a ground atom in the specification which is not an answer of LONGER).

Below we denote by $|l|$ the length of a list l (and $|l|$ is undefined when l is not a list). Also, let $||t||$ stand for the number of constants in a term t.[1]

$$S_1 = \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid l_2 \text{ is a list} \,\},$$

$$S_2 = \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid l_1 \text{ is a list} \,\},$$

$$\begin{aligned} S_3 &= \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid l_1, l_2 \text{ are lists and } |l_1| > |l_2| \,\} \\ &= \{\, longer([t_1, \ldots, t_m], [u_1, \ldots, u_n]) \in \mathbf{B}_\mathcal{A} \mid m > n \geq 0 \,\}, \end{aligned}$$

$$\begin{aligned} S_4 &= \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid \text{if } l_1, l_2 \text{ are lists then } |l_1| > |l_2| \,\} \\ &= S_3 \cup \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid l_1 \text{ or } l_2 \text{ is not a list} \,\}, \end{aligned}$$

$$\begin{aligned} S_5 &= \left\{ longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \;\middle|\; \begin{array}{l} l_2 \text{ is a list and} \\ \text{if } l_1 \text{ is a list then } |l_1| > |l_2| \end{array} \right\} \\ &= S_3 \cup \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid l_1 \text{ is not a list, } l_2 \text{ is a list} \,\}, \end{aligned}$$

[1] For instance, let `a`, `b` be constants and `f` a two-argument function symbol. Then we have $||\texttt{a}|| = 1$, $||\texttt{f(a,b)}|| = 2$, $||\texttt{[]}|| = 1$, $||\texttt{[a,b]}|| = 3$, the latter because `[a,b]` is `.(a,.(b,[]))`. Also, $|\texttt{[]}| = 0$, $|\texttt{[b,f(b,b)]}| = 2$, and `[a|b]` is not a list.

Remember that $\mathbf{B}_\mathcal{A}$ is the Herbrand base of the considered alphabet $\mathcal{A}$, and writing $p(t, u) \in \mathbf{B}_\mathcal{A}$ is a compact way to state that p is a predicate symbol and t, u are ground terms. It is crucial that you understand the list notation of Prolog. See e.g. slide 9 from those for tutorial 2 (`le2.LPintro*.pdf`).

$$S_6 = \{\, longer([t_1, \ldots, t_{n+1}|t], [u_1, \ldots, u_n]) \in \mathbf{B}_\mathcal{A} \mid n \geq 0 \,\},$$
$$S_7 = \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid ||l_1|| > ||l_2|| \,\},$$

$$S_8 = \{\, longer(l_1, l_2) \in \mathbf{B}_\mathcal{A} \mid l_1, l_2 \text{ are lists and } |l_1| = |l_2| + 2 \,\}.$$

2. Which of specifications $S_1, \ldots, S_8$ is the least Herbrand model $\mathbf{M}_{\text{LONGER}}$ of the program?

3. Prove that the program is correct w.r.t. specification S_5. Use the standard method.
So you are proving that (in each answer of LONGER) the second argument is a list and if first one is a list then the first one is longer than the second.
A hint is provided by the informal comments in the program.

4. Out of specifications $S_1, \ldots, S_8$ choose one, let us call it S_{com}, for which you did not found program LONGER to be not complete. Prove that LONGER is semi-complete w.r.t. S_{com},

This implies that the program is complete w.r.t. S_{com}, as for any ground query $longer(t, u)$ the SLD-tree is finite. (The tree consists of a single branch in which for any non-leaf node A_i its child A_{i+1} contains fewer constants than A_i does. So such branch cannot be infinite.)

5. Modify the program so that it is also correct w.r.t. S_3. Provide at least an informal outline of a correctness proof.
So, speaking informally, the program should additionally assure that the first argument of *longer* is a list too.
It may be convenient to introduce an additional predicate.

6. Consider the main part of the efficient list reverse program (lecture 7).

$$\begin{array}{ll} \text{REV}: & rev([\,], X, X). \\ & rev([H|L], Y, X) \leftarrow rev(L, [H|Y], X). \end{array}$$

(Program REV reverses a given list l by answering query $rev(l, [\,], X)$.)

Describe (informally, in words) the relation defined by the program. Employ the notions of list and difference list. Do this more formally by describing a specification, for which REV is correct and complete, preferably in this style:

$$S_{\text{rev}} = \left\{ rev(s, t, u) \in \mathbf{B}_\mathcal{A} \mid \quad \ldots \text{is a list}, \ldots \text{the reverse of list} \ldots \right\}$$

Prove that REV is correct w.r.t. S_{rev}. Remember that list $[t_1, \ldots, t_n]$ represented as a difference list is a pair of terms $[t_1, \ldots, t_n|t]$ and t (where t is arbitrary).
If you are really stuck, the handouts associated with this tutorial (`corr.examples.pdf`) with example proofs contain a proof for basically the same program (Example 3.5) (Do not miss a different order of arguments!)