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 $[\_Y|Ys]$
   $longer([\_X|Xs], [\_Y|Ys]) \leftarrow longer(Xs, Ys)$.

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]
(In our comments we abbreviate "answer of LONGER" by "answer"; a, b, c are constants.)

$S_1 = \{\, longer(l_1, l_2) \in \mathbf{B}_{\mathcal{A}} \mid l_2 \text{ is a list} \,\}$,
LONGER not complete w.r.t. S_1 ($longer(a, [\,]) \in S_1$ is not an answer)

$S_2 = \{\, longer(l_1, l_2) \in \mathbf{B}_{\mathcal{A}} \mid l_1 \text{ is a list} \,\}$,
LONGER not correct and not complete w.r.t. S_2
(an answer $longer([a|b], [\,]) \notin S_2$, and $longer([\,], a) \in S_2$ is not an answer)

$$S_3 = \{\, longer(l_1, l_2) \in \mathbf{B}_{\mathcal{A}} \mid l_1, l_2 \text{ are lists and } |l_1| > |l_2| \,\}$$
$$= \{\, longer([t_1, \dots, t_m], [u_1, \dots, u_n]) \in \mathbf{B}_{\mathcal{A}} \mid m > n \geq 0 \,\},$$
LONGER not correct w.r.t. S_3 (due to answer $longer([a|b], [\,]) \notin S_3$)

$$S_4 = \{\, longer(l_1, l_2) \in \mathbf{B}_{\mathcal{A}} \mid \text{if } l_1, l_2 \text{ are lists then } |l_1| > |l_2| \,\}$$
$$= S_3 \cup \{\, longer(l_1, l_2) \in \mathbf{B}_{\mathcal{A}} \mid l_1 \text{ or } l_2 \text{ is not a list} \,\},$$
LONGER not complete w.r.t. S_4 ($longer(a, b) \in S_4$ is not an answer)

$$S_5 = \left\{ longer(l_1, l_2) \in \mathbf{B}_{\mathcal{A}} \;\middle|\; \begin{array}{l} l_2 \text{ is a list and} \\ \text{if } l_1 \text{ is a list then } |l_1| > |l_2| \end{array} \right\}$$
$$= S_3 \cup \{\, longer(l_1, l_2) \in \mathbf{B}_{\mathcal{A}} \mid l_1 \text{ is not a list, } l_2 \text{ is a list} \,\},$$
LONGER not complete w.r.t. S_5 ($longer(a, [\,]) \in S_5$ is not an answer)

[1] For instance, let `a`,`b` be constants and `f` a two-argument function symbol. Then we have $||$`a`$|| = 1$, $||$`f(a,b)`$|| = 2$, $||$`[]`$|| = 1$, $||$`[a,b]`$|| = 3$, the latter because `[a,b]` is `.(a,.(b,[]))`. Also, $|$`[]`$| = 0$, $|$`[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`).

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

$$S_7 = \{\, longer(l_1,l_2) \in \mathbf{B}_\mathcal{A} \mid ||l_1|| > ||l_2|| \,\},$$

LONGER not correct w.r.t. S_7, (an answer $longer([a,b],[f(a,b,c)]) \notin S_7$), and not complete w.r.t. S_7, as $longer(f(a,b),[\,]) \in S_7$ is not an answer

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

LONGER not correct w.r.t. S_8, as an answer $longer([a],[\,]) \notin S_8$

Whenever it is not stated that the program is not correct (or not complete) then it *is* correct (respectively complete). This can be justified by appropriate correctness or completeness proof(s), and using the fact that correctness (respectively completeness) w.r.t. S implies correctness (completeness) w.r.t. any $S' \supseteq S$ ($S' \subseteq S$). In the context of the next problem, an effective approach is to construct (in a standard way) the least Herbrand model $\mathbf{M}_{\text{LONGER}}$, see that $\mathbf{M}_{\text{LONGER}} = S_6$, and show which of the specifications are its supersets, and which are subsets.

Note that, in a sense, we used the equivalence of $\alpha \to \beta$ and $\neg\alpha \vee \beta$ to describe each of S_4, S_5 in two ways.

2. Which of specifications $S_1,\dots,S_8$ is the least Herbrand model $\mathbf{M}_{\text{LONGER}}$ of the program? (For a solution see above.)

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.

Detailed solution. For each ground instance of a clause of LONGER, we have to show that if its body atoms are in S_5 then the head is in S_5.

The specification is

$$S_5 = \left\{ longer(l_1,l_2) \in \mathbf{B}_\mathcal{A} \;\middle|\; \begin{array}{l} l_2 \text{ is a list and} \\ \text{if } l_1 \text{ is a list then } |l_1| > |l_2| \end{array} \right\}$$

and the ground instances are

$$longer([t|u],[\,]), \tag{1}$$

$$\underbrace{longer([x|xs],[y|ys])}_{H} \leftarrow \underbrace{longer(xs,ys)}_{B}, \tag{2}$$

where $t,u,x,xs,y,ys \in \mathbf{U}_\mathcal{A}$ (i.e. they are ground terms).

Consider (1). $longer([t|u],[\,]) \in S_5$, as $[\,]$ is a list, and if $[t|u]$ is a list then $|[t|u]| > 0 = |[\,]|$.

Consider (2) and assume $B \in S_5$. So ys is a list, thus $[y|ys]$ is a list. Assume $[x|xs]$ is a list; then xs is a list, hence $|xs| > |ys|$ (as $B \in S_5$), hence $|xs|+1 > |ys|+1$, this means $[x|xs] > [y|ys]$. The last sentence proves that if $[x|xs]$ is a list then $[x|xs] > [y|ys]$. (This completes our proof that $H \in S_5$.)

Another solution (using the other description of S_5, and slightly less detailed).

$$\begin{aligned} S_5 &= S_3 \cup S_5' \\ S_3 &= \{\, longer(l_1,l_2) \in \mathbf{B}_\mathcal{A} \mid l_1,l_2 \text{ are lists and } |l_1| > |l_2| \,\} \\ S_5' &= \{\, longer(l_1,l_2) \in \mathbf{B}_\mathcal{A} \mid l_1 \text{ is not a list, } l_2 \text{ is a list} \,\}, \end{aligned}$$

For each ground instance of a clause of LONGER, we show that if its body atoms are in S_5 then the head is in S_5 (by showing that it is in S_3 or in S_5').

Consider (1). Either $[t|u]$ is not a list and thus $longer([t|u],[\,]) \in S_5'$. Or $[t|u]$ is a list and thus $|[t|u]| > 0 = |[\,]|$, hence $longer([t|u],[\,]) \in S_3$.

Consider (2), and assume that $B \in S_5$. We have two cases.

(i) $B \in S_3$. As xs, ys are lists and $|xs| > |ys|$, we have that $[x|xs]$, $[y|ys]$ are lists too, and $|[x|xs]| > |[y|ys]|$. Thus $H \in S_3$.

(ii) $B \in S_5'$, so xs is not a list and ys is. Hence $[x|xs]$ is not and $[y|ys]$ is a list. Thus $H \in S_5'$.

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

This implies that the program is complete w.r.t. $S_{\rm com}$, 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.)

Solution. Let us choose

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

(Completeness w.r.t. S_3 is more useful, a proof is left for you.)

To prove that LONGER is semi-complete w.r.t. S_8, we show that each atom A from S_8 is covered by LONGER w.r.t. S_8. This means A is the head of a ground instance of a clause from LONGER, such that all its body atoms are in S_8.

Take an $A = longer(l_1,l_2) \in S_8$. Obviously, $|l_1| > 0$, so l_1 is of the form $[t|u]$.

If $|l_2| = 0$ then A is of the form (1), i.e. A is covered by ground instance (1) of the first clause of LONGER.

If $|l_2| > 0$ then $A = longer([x|xs],[y|ys])$ (for some x, xs, y, ys, where xs, ys are lists). We have $|[x|xs]| = |[y|ys]| + 2$ and thus $|xs| = |ys| + 2$. Now we see that A is the head of a ground instance (2) of a clause from LONGER, and its body atom $longer(xs,ys)$ is in S_8. So A is covered by the program w.r.t. S_8. This completes the proof

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.

Two solutions. The last clause of each of them is the same as that of LONGER.

```
% list(L) – L is a list
list([ ]).
list([_X|Xs]) ← list(Xs).                longer([_X], [ ]).
longer([_|Xs], [ ]) ← list(Xs).          longer([_|Xs], [ ]) ← longer(Xs, [ ]).
              longer([_X|Xs], [_Y|Ys]) ← longer(Xs, Ys).
```

A correctness proof of the second program is, briefly, as follows. The specification is

$$S_3 = \{\, longer(l_1,l_2) \in \mathbf{B}_\mathcal{A} \mid l_1, l_2 \text{ are lists and } |l_1| > |l_2| \,\}$$

The first clause is a fact, and each its ground instance $longer([x],[\,])$ is in S_3 (as $1 > 0$). For any ground instance of the second clause, if its body $longer(xs,[\,])$ is in S_3 then its head $longer([x|xs],[\,])$ is in S_3 too (as xs is a list, hence $[x|xs]$ is a list, longer than 0). For any ground instance (2) of the third clause, if its body atom $longer(xs,ys) \in S_3$ then the head $longer([x|xs],[y|ys]) \in S_3$ (because $[x|xs], [y|ys]$ are lists as xs, ys are, and $|[x|xs]| > |[y|ys]|$ as $|xs| > |ys|$).

Note that for a correctness proof of the first program we need to extend specification S_3 so it also describes $list/1$.

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:

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

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!)