# Problem Set for Tutorial 5 — TDDD08

Wlodek Drabent*[1] and Victor Lagerkvist†[1]

[1]Department of Computer and Information Science, Linköping University, Linköping, Sweden

1. Consider the following definite program $P$:

$$o(s(z)). \tag{1}$$
$$o(s(s(N))) \leftarrow o(N). \tag{2}$$
$$e(z). \tag{3}$$
$$e(s(s(N))) \leftarrow e(N). \tag{4}$$
$$n(N) \leftarrow o(N). \tag{5}$$
$$n(N) \leftarrow e(N). \tag{6}$$

   (a) Assume that the vocabulary $\mathcal{A}$ contains only the constant $z$ and the unary function symbol $s$. What is the Herbrand universe $\mathbf{U}_{\mathcal{A}}$?

   (b) What is the Herbrand base?

   (c) Find the least Herbrand model of the program.

   (d) Give an example of a model of the program which is *not* an Herbrand model.

Solution.

   (a) $\{z, s(z), s(s(z)), \ldots\} = \{s^i(z) \mid i \geq 0\}$ (we let $s^i(z)$ be $i$ repeated applications of $s$, and let $s^0(z) = z$).

   (b) $\{o(t), e(t), n(t) \mid t \in \mathbf{U}_{\mathcal{A}}\}$.

   (c) $\{o(s^i(z)) \mid i \text{ is odd}\} \cup \{e(s^i(z)) \mid i \text{ is even}\} \cup \{n(t) \mid t \in \mathbf{U}_{\mathcal{A}}\}$. How can we arrive at this answer? The general idea is to construct the elements of the least Herbrand model, i.e., the ground, logical consequences of the program, in a bottom-up manner. Hence, we start by adding all ground instantiations of the facts of the program. Then, if we have a ground instantiation of a rule whose body is true in the current set, we add the head of the rule to the current set, too. Once all rules are satisfied in this way, then we do not have to add anything further, and we are done. Hence, conceptually, we keep track of the current set of true statements in a set, and in each iteration go through each rule in the program and check whether we need to add anything new. For our current program $P$ this can be expressed via the following iterations, where the set $I_i$ is the set of "true" statements in iteration number $i$. Note that $I_0 = \emptyset$. If you do not recall the meaning of the $T_P^i$-operator, then see either the course book or the corresponding lecture slides.

      - $I_0 = \emptyset$.

*wlodzimierz.drabent@liu.se

†victor.lagerkvist@liu.se

- $I_1 = T_P(\emptyset) = \{o(s(z)), e(z)\}$.
- $I_2 = T_P^2(\emptyset) = T_P(I_1) = \{o(s(s(s(z)))), e(e(z)), n(s(z)), n(z)\} \cup I_1$.

  $\vdots$

- $I_i = T_P^i(\emptyset) = T_P(I_{i-1}) = \{o(s(s(t))), e(s(s(t'))), n(t), n(t') \mid o(t), e(t') \in I_{i-1}\} \cup I_{i-1}$.

Here we see a clear pattern and conclude that the least Herbrand model is indeed $\{o(s^i(z)) \mid i \text{ is odd}\} \cup \{e(s^i(z)) \mid i \text{ is even}\} \cup \{n(t) \mid t \in \mathbf{U}_\mathcal{A}\}$.

(d) This is very straightforward. Note that we do not have to provide a "minimal" model, simply a model which is not a Herbrand model. Thus, as the underlying domain we could simply choose the natural numbers, map $s$ to the successor function, and simply say that $o/1$, $e/1$, and $n/1$ are true for all natural numbers. Then we do not have a Herbrand model since the universe is not the Herbrand universe $\mathbf{U}_\mathcal{A}$. We could of course also choose a more natural model where $o/1$ is true for all odd numbers, $e/1$ is true for all even numbers, and where $n/1$ is true for all natural numbers.

2. Consider the following definite program $P$:

$$p(\,X, Y\,) \leftarrow r(\,g(X), X\,). \tag{1}$$
$$r(\,g(Z), f(Z)\,). \tag{2}$$
$$r(\,g(X), Y\,) \leftarrow r(\,X, f(Y)\,). \tag{3}$$

(a) Assume that the vocabulary $\mathcal{A}$ contains one constant $a$ and two one-argument function symbols $f, g$. What is the Herbrand universe $\mathbf{U}_\mathcal{A}$ corresponding to $\mathcal{A}$?

(b) Which of the following Herbrand interpretations are models of the program?

$$\begin{aligned}
&I_0 = \emptyset\\
&I_1 = \{\, r(g(t), f(t)) \mid t \in \mathbf{U}_\mathcal{A}\,\}\\
&I_2 = I_1 \cup \{\, r(g^i(f^j(t)), t) \mid i, j \geq 0,\ t \in \mathbf{U}_\mathcal{A}\,\}\\
&I_3 = I_2 \cup \{\, p(t, u) \mid t, u \in \mathbf{U}_\mathcal{A}\,\}
\end{aligned}$$

(c) Find the least Herbrand model of the program.

(d) Give an example of a ground atom which is a logical consequence of $P$, but is not an instance of $r(g(Z), f(Z))$.

(e) Give an example of a ground atom which is not a logical consequence of $P$, but it is an instance of $r(\,g(X), Y\,)$.

(f) Give an example of a non-ground atom which is a logical consequence of $P$, but is not an instance of $r(g(Z), f(Z))$.

Solution.

(a) $\mathbf{U}_\mathcal{A} = \{a, f(a), g(a), f(f(a)), f(g(a)), g(g(a)), g(f(a)), \ldots\}$.

(b) We take each case in turn. First, $I_0$ is clearly not a model since it does not satisfy (2). To see that $I_1$ is not a model, consider e.g. a ground instantiation $r(g^2(a), a) \leftarrow r(g(a), f(a))$ of (3). Clearly, $r(g(a), f(a)) \in I_1$, but $r(g^2(a), a) \notin I_1$. For $I_2$, consider the ground clause $p(a, f(a)) \leftarrow r(g(a), a)$. Then $r(g(a), a) \in I_2$ but $p(a, f(a)) \notin I_2$. However, we claim that $I_3 \models P$. First, since $I_3$ contains both $I_1$ and $I_2$, we will perform a case analysis with respect to the third clause, depending on whether a ground instance contains a body atom from $I_1$, or from $I_2$ (but not from $I_1$, to make the two cases distinct). Thus:

i. Assume a ground instance of the third clause with the body atom from $I_1$. Then it has to be of the form $r(g^2(t), t) \leftarrow r(g(t), f(t))$ $(t \in \mathbf{U}_{\mathcal{A}})$. But then the head is in $I_2 \subseteq I_3$.

ii. Assume a ground instance of the third clause where the body atom is included in $I_2$, but not in $I_1$ (in $I_3 \setminus I_1$). Then it is of the form $r(\, g(g^i(f^j(f(t')))), t') \leftarrow r(\, g^i(f^j(f(t'))), f(t'))$ $(t' \in \mathbf{U}_{\mathcal{A}})$. But then the head is in $I_2 \subseteq I_3$.

Thus, $P$ is correct w.r.t. $I_3$

(c) We try to construct the least Herbrand model bottom-up in a systematic manner.

- $I_1 = \{r(g(t), f(t)) \mid t \in \mathbf{U}_{\mathcal{A}}\}$.
- $I_2 = I_1 \cup \{r(g(g(t)), t) \mid t \in \mathbf{U}_{\mathcal{A}}\}$.
- $I_3 = I_2 \cup \{r(g(g(g(f(t)))), t) \mid t \in \mathbf{U}_{\mathcal{A}}\}$.

  $\vdots$

- $I_i = I_{i-1} \cup \{r(g^i(f^{i-2}(t)), t) \mid t \in \mathbf{U}_{\mathcal{A}}\}$.

Let us explain the step from $I_2$ to $I_3$ in greater detail. We already know that $r(g(g(t)), t) \in I_2$ for *every* ground term $t \in \mathbf{U}_{\mathcal{A}}$. Hence, in particular, $r(g(g(f(t))), f(t)) \in I_2$ for every ground term $t \in \mathbf{U}_{\mathcal{A}}$. But if $r(g(g(f(t))), f(t))$ is true, $r(g(g(g(f(t)))), t)$ must be true, too, since $r(g(g(g(f(t)))), t) \leftarrow r(g(g(f(t))), f(t))$ is a ground initialisation of the clause (3).

Note also that in no step of the iteration are we going to be able to find a ground instantiation of (1) where the body is true in the previous iteration. Now the general pattern is clear and we conclude that the least Herbrand model is the set $\{r(g(t), f(t)) \mid t \in \mathbf{U}_{\mathcal{A}}\} \cup \{r(g^{i+2}(f^i(t)), t) \mid i \geq 0, t \in \mathbf{U}_{\mathcal{A}})\}$.

(d) For example, $r(g(g(a)), a)$.

(e) For example, $r(g(f(a)), a)$.

(f) For example, $r(g(g(X)), X)$.

3. Write a DCG which recognises whether a string is a *palindrome*, i.e., whether the string reads the same forwards and backwards. Translate your DCG to Prolog using the translation described in Section 10.5 of the course book.

   Solution. See the source file associated with this tutorial.

4. Consider the following fragment of the syntax of a programming language:

```
<exp> ::= begin <exp> end | skip | if <b_exp> exp | <id> := <num>
<b_exp> ::= <id> < <id> | <id> = <id>
```

First, write a DCG which recognises the above language, under reasonable assumptions with respect to `<id>` and `<num>` (e.g., you may assume that `<id> ::= x | <id> ::= y`, to avoid specifying exactly what strings constitute variables). Second, write a DCG `exp(T)` which recognises the language and where `T` is a term corresponding to the accepted string. For example, to parse strings of the form `if <b_exp> exp` your program should include a DCG rule along the lines of:

```
exp(if(B_term, Exp_term)) --> [if], b_exp(B_term), exp(Exp_term).
```

Solution. (Partial) See the source file associated with this tutorial.

5. Say that a string of left and right parenthesis (i.e., a string consisting of '(', and ')')) is *balanced* if each left parenthesis has a matching right parenthesis. For example, (())() is balanced but (()() is not. Write a DCG which recognises the language of all balanced strings of parenthesises, i.e., $\{\alpha \in \{(,)\}^* \mid \alpha \text{ is balanced}\}$. Would your solution work as expected in Prolog (under the standard translation)?

6. Write a DCG which recognises the language $\{a^n b^n c^n \mid n \geq 1\}$.

   Solution. See the source file associated with this tutorial.