Logic Programming SuSe 20

Exercise Sheet 1 (due Monday, April 27, 2020, 12:30 pm)

Prof. Dr. Jürgen Giesl

Daniel Cloerkes, Marcel Hark, Dominik Meier

Notes:

- To solve the programming exercises you can use the Prolog interpreter **SWI-Prolog**, available for free on this website. For Debian and Ubuntu it suffices to install the `swi-prolog` package. You can use the command "`swipl`" to start it and use "`[exercise1].`" to load the facts from the file **exercise1.pl** in the current directory.
- Please solve these exercises in **groups of four**!
- Please upload a PDF with your solutions for the *non*-programming exercises **and** the source code of your solutions for the programming exercises in a single **ZIP**-archive via `RWTHmoodle` before the exercise course on Monday, April 27, 2020, 12:30 pm. Please name your archive `Sheet_i_Mat1_Mat2_Mat3_Mat4.zip`, where `i` is the number of the sheet and `Mat_1`. . . `Mat_4` are the immatriculation numbers of the group members. It is sufficient if **one** of the group members uploads your solution.
- Make sure that your solutions for the programming exercises are accepted by **SWI-Prolog**. Files, which are not accepted by **SWI-Prolog**, will not be marked.
- The exercise course will take place on Monday, April 27, 2020, 12:30 pm. Due to the coronavirus pandemic it will be held as a Zoom-Meeting. The details on how to join this meeting will be announced in `RWTHmoodle` shortly before its beginning.

Programming Exercise 1 (Simple Prolog): (1.5 + 2 + 1.5 = 5 points)

Consider the following Prolog program.

```
evolvedFrom(cat,miacis).
evolvedFrom(hyena,miacis).
evolvedFrom(weasel,miacis).
evolvedFrom(cynodictis,miacis).

evolvedFrom(raccoon,cynodictis).
evolvedFrom(bear,cynodictis).
evolvedFrom(tomarctus,cynodictis).

evolvedFrom(fox,tomarctus).
evolvedFrom(wolf,tomarctus).
evolvedFrom(dog,tomarctus).
```

a) Implement a predicate `evolvedFromSameCreature(A,B)` in Prolog which is true if both `A` and `B` evolved from the same creature according to the predicate `evolvedFrom` above. For example, the query `?- evolvedFromSameCreature(fox,wolf)` should yield the answer `true`, whereas `?- evolvedFromSameCreature(cat,dog)` should yield `false`.

b) Implement a predicate `descendsFrom(A,C)` in Prolog which is true if `A` is a descendant of `C`, i.e., `A` directly evolved from `C` or `A` evolved from a descendant `B` of `C`.

Make sure that the evaluation of all queries `?- descendsFrom(...,...)` terminates.

c) List all answers that Prolog gives for the following queries, in the order that Prolog gives them. Try to solve this part of the exercise without the help of a computer. Please write the answer for this subexercise into the PDF with your solutions for the *non-programming* exercises.

1. `?- evolvedFrom(X,tomarctus).`
2. `?- evolvedFromSameCreature(raccoon,X).`
3. `?- descendsFrom(wolf,X).`

Exercise 2 (Syntax): (2 + 1 = 3 points)

Consider the following Prolog program.

```
robot(wall_e).
robot(otto).
robot(c3po).
robot(r2d2).
robot(android(looks_like_a_human)).
robot(android(looks_like_a_machine)).

can_walk(c3po).
can_drive(r2d2).
can_drive(wall_e).
same_story(c3po,r2d2).
same_story(wall_e,otto).

can_move(X) :- can_walk(X).
can_move(X) :- can_drive(X).
same_story(X,Y):- robot(X), robot(Y), same_story(Y,X).
```

a) Construct the corresponding sets of formulas, predicate symbols, function symbols, and variables based on the program.

b) Give Prolog queries corresponding to the following questions:

- "Which robots can both walk and drive?"
- "Which pairs of robots can move, and are both part of the same story?"

Exercise 3 (Induction): (4 points)

Show by structural induction that for every term $t \in \mathcal{T}(\Sigma, \mathcal{V})$ and every substitution $\sigma : \mathcal{V} \to \mathcal{T}(\Sigma, \mathcal{V})$ we have

$$\mathcal{V}(\sigma(t)) = \bigcup_{X \in \mathcal{V}(t)} \mathcal{V}(\sigma(X)).$$

Exercise 4 (Semantics): (1.5 + 2.5 + 4 = 8 points)

Let (Σ, Δ) be a signature with $\Sigma = \Sigma_0 \cup \Sigma_1, \Delta = \Delta_3 = \{\mathsf{plus}\}$ and $\Sigma_0 = \{\mathsf{0}\}, \Sigma_1 = \{\mathsf{s}\}$.
Moreover, let $\Phi = \{\forall Y \quad \mathsf{plus}(\mathsf{0}, Y, Y), \quad \forall X, Y, Z \quad \mathsf{plus}(X, Y, Z) \to \mathsf{plus}(\mathsf{s}(X), Y, \mathsf{s}(Z))\}$,
$\varphi = \forall X, Y, Z \quad \mathsf{plus}(X, Y, Z) \leftrightarrow \mathsf{plus}(Y, X, Z)$,
$S = (\mathbb{N}, \alpha)$, $\alpha_{\mathsf{0}} = 0$, $\alpha_{\mathsf{s}}(x) = x + 1$ for all $x \in \mathbb{N}$, and $\alpha_{\mathsf{plus}} = \{(x, y, z) \in \mathbb{N}^3 \mid x + y = z\}$.
Prove or disprove the following statements.

a) $S \models \varphi$

b) $\models \varphi$

c) $\Phi \models \varphi$

Hints:

- For any $a, b, c \in \mathbb{N}$ let $S\,[\![X/a, Y/b, Z/c]\!]$ denote an interpretation $(\mathbb{N}, \alpha, \beta)$ where $\beta(X) = a$, $\beta(Y) = b$, and $\beta(Z) = c$.
- You may use that addition on natural numbers is commutative.