Foundations of Logic Programming

Exam Winter Semester 2013

Examiner: Sebastian Rudolph

11.02.2014, start: .... , duration: 60 minutes

Please write your student registration number on every sheet.

Name: Student registration number:

Task	Possible Points	Achieved Points
1	2	
2	12	
3	11	
4	13	
5	8	
6	14	
Sum	60	

Task 1. (2 points)

Explain the difference between ASP and Prolog.

Task 2. (12 points)

Compute the composition of the following substitutions

a) $\{x/y\} \cdot \{y/x\}$ (1 point)

b) $\{x/y, y/x\} \cdot \{y/7\} \cdot \{x/y, y/x\}$ (1 point)

c) $\{x/f(x)\} \cdot \{x/f(x)\}$ (1 point)

d) $\{x/f(y)\} \cdot \{y/g(x)\}$ (1 point)

Determine for each of the following term pair, if a unifier exists and if so write down the most general unifier:

e) $g(f(x), h(y))$ and $g(y, x)$ (2 points)

f) $f(x, y, y)$ and $f(z, z, c)$ (2 points)

g) $f(g(x), y)$ and $f(f(g(x), y), y)$ (2 points)

h) $g(y, f(x))$ and $g(x, f(y))$ (2 points)

Task 3. (11 points)

Consider the following program P:

```
p ←.
t ← p.
q ← r.
p ← r, ¬q.
r ← q.
t ← ¬q.
```

a) Construct the dependency graph D_P of P. (3 points)

b) Is P hierarchical? (1 point)

c) Is P stratified? (1 point)

c) Give a stratification of P and use it to compute the standard model M_P of P. (6 points)

Task 4. (13 points)

Consider the following program together with the query `?- p(X).`

```
r(a).
r(b).
s(b).
p(X) ← s(X).
p(X) ← q(X),!,r(X).
p(f(X)) ← r(X).
q(g(X)) ← s(X).
q(X) ← p(X),!,r(X).
```

a) Show with the help of the Prolog tree how the *cut* is used. (10 points)

b) Indicate explicitly, if branches are eliminated from the tree. (2 points)

c) Give the output in the order of the computation. (1 point)

Task 5. (8 points)

Consider the following program:

```
    p(X) ← r([a|X])
r([Y|X]) ← s(X)
s([Y|X]) ← p(X)
```

a) Provide a level mapping for which the program is recurrent. (6 points)

b) Provide a bounded query for this level mapping containing at least one variable. (1 point)

c) Provide an unbounded query for this level mapping. (1 point)

Task 6. (14 points)

In answer-set programming (ASP) the semantics of a program is given by its stable models. These are defined via the *Gelfond-Lifschitz reduct*.

a) How is the Gelfond-Lifschitz reduct defined? (3 points)

b) Consider the program Π, check whether there are stable models by applying the reduct.

$$\begin{aligned}\Pi = \{&a \leftarrow a;\\ &b \leftarrow not\ c, d;\\ &c \leftarrow not\ b, d;\\ &d \leftarrow not\ a\}\end{aligned}$$

(6 points)

c) Explain the difference between *stable*, *supported*, *minimal* and *classical models*. (5 points)