Logic Programming SS17
Exercise Sheet 4 (due Tuesday, May 30th, 2017 (8:30 am))

Prof. Dr. Jürgen Giesl — Jera Hensel

Notes:

- Please solve these exercises in **groups of three or four**!
- The solutions must be handed in **directly before (very latest: at the beginning of)** the exercise course on **Tuesday, May 30th, 2017 (8:30 am)**, in lecture hall **AH 2**. Alternatively you can drop your solutions into a box which is located right next to Prof. Giesl's office (this box is emptied **15 minutes before** the exercise course starts).
- Please write the **names** and **immatriculation numbers** of all students on your solution. Also please staple the individual sheets!

Exercise 1 (Resolution): (2 points)

Consider again the following logic program from Exercise Sheet 3

```
equals(0,0).
equals(s(X),s(Y)) :- equals(X,Y).
```

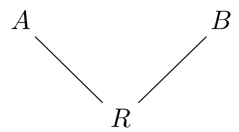
and the query

```
? - equals(s(s(0)),s(s(0))).
```

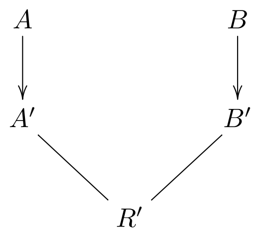
Show that the formulas φ_1 and φ_2 corresponding to the logic program entail the formula φ corresponding to the query (i.e., $\{\varphi_1, \varphi_2\} \models \varphi$) using the resolution algorithm in predicate logic.

Exercise 2 (Lifting Lemma): (3 points)

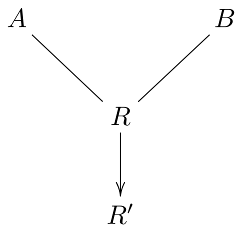
Consider the clauses $\underbrace{\{\mathtt{inc}(X, \mathtt{s}(X))\}}_{=:A}, \underbrace{\{\neg\mathtt{inc}(Y, Z), \mathtt{inc}(\mathtt{s}(Y), \mathtt{s}(Z))\}}_{=:B}$. These clauses can be resolved to $R := \{\mathtt{inc}(\mathtt{s}(X), \mathtt{s}(\mathtt{s}(X)))\}$ as follows:



For this resolution step, find all ground instances A', B', and R' of A, B, and R (using substitution with ground terms built from the function symbols `s` and `0`), such that we have



(i.e., R' is a resolvent of A' and B') and by the lifting lemma (Lemma 3.4.8) we get:



If there is an infinite number of such ground instances for A, B, and R, give a suitable finite description of these ground instances.

Exercise 3 (Restrictions of Resolution): (2 + 3 + 3 + 2 + 2 + 1 = 13 points)

Consider the set of clauses

$$\mathcal{K} = \{\{\mathsf{p}(\mathsf{f}(\mathsf{a}), X)\}, \{\mathsf{p}(Y, \mathsf{f}(Z)), \neg\mathsf{p}(\mathsf{f}(Y), Z)\}, \{\neg\mathsf{p}(\mathsf{a}, X), \mathsf{q}(\mathsf{f}(X), X)\}, \{\neg\mathsf{p}(\mathsf{a}, Z), \neg\mathsf{q}(\mathsf{f}(\mathsf{f}(Y)), Z), \neg\mathsf{q}(\mathsf{f}(Z), \mathsf{f}(Y))\}\}$$

with $\mathsf{a} \in \Sigma_0$, $\mathsf{f} \in \Sigma_1$, $\mathsf{q} \in \Delta_2$, and $\mathsf{p} \in \Delta_2$.

a) Derive the empty clause from $\mathcal{K}$ using full but not linear resolution (i.e., there must be at least one non-linear resolution step). For each step denote the substitutions used.

b) Derive the empty clause from $\mathcal{K}$ using linear but not input resolution. For each step denote the substitutions used.

c) Derive the empty clause from $\mathcal{K}$ using input resolution but not SLD resolution. For each step denote the substitutions used.

d) Derive the empty clause from $\mathcal{K}$ using SLD resolution but not binary SLD resolution. For each step denote the substitutions used. In addition, also give the answer substitution. Here, the answer substitution is computed as follows. Consider an SLD resolution proof from a negative clause N of the form $N, R_1, R_2, \dots, R_m$, where R_m is the empty clause $\square$ and where no variable renamings have been applied to $N, R_1, R_2, \dots, R_m$ during the resolution proof (i.e., variable renamings are only applied to definite Horn clauses. Here, we use variable renamings to ensure that the variables in the definite parent clause are disjoint from all variables occurring in clauses that have already been used in the resolution proof.) So R_1 is the resolvent of N and a definite clause from the input set. Similarly, R_2 is the resolvent of R_1 and a definite clause from the input set, etc. In the first resolution step, let σ_1 be the used mgu. In the second resolution step, one used the mgu σ_2, etc. Then the answer substitution is $\sigma_m \circ \ldots \circ \sigma_2 \circ \sigma_1$, i.e., σ_1 is applied first in this composition of substitutions. Moreover, the answer substitution is restricted to those variables that occur in the original negative clause N.

e) Derive the empty clause from $\mathcal{K}$ using binary SLD resolution. For each step denote the substitutions used. In addition, also give the answer substitution.

f) Express $\mathcal{K}$ as queries, facts, and rules of a logic program.

Exercise 4 (Unrenamed Resolution): (4 points)

A clause R is an *unrenamed resolvent* of two clauses K_1 and K_2 iff the following two conditions are satisfied:

- There are literals $L_1, \dots, L_m \in K_1$ and $L'_1, \dots, L'_n \in K_2$ with $m, n \geq 1$ such that $\{\overline{L_1}, \dots, \overline{L_m}, L'_1, \dots, L'_n\}$ is unifiable with some mgu σ.
- $R = \sigma((K_1 \setminus \{L_1, \dots, L_m\}) \cup (K_2 \setminus \{L'_1, \dots, L'_n\}))$

Unrenamed resolution is, thus, defined like resolution in predicate logic, but without renaming the clauses first such that they do not have any variables in common.
Please prove or disprove the following statements:

a) Unrenamed resolution is *sound*, i.e., there is no satisfiable clause set $\mathcal{K}$ from which one can derive $\square$ by unrenamed resolution.

b) Unrenamed resolution is *complete*, i.e., from any unsatisfiable clause set $\mathcal{K}$ one can derive $\square$ by unrenamed resolution.