COMP2411 Tutorial, Week 11

1. (Using the equational approach from lecture 17) Consider the following logic program.

\[
\begin{align*}
\text{parent}(X, Y) & : \neg \text{parents}(X, Y, Z). \\
\text{parent}(X, Z) & : \neg \text{parents}(X, Y, Z). \\
\text{Sibling}(X, Y) & : \neg \text{parent}(X, Z), \text{parent}(Y, Z). \\
\text{Brother}(X, Y) & : \neg \text{Male}(Y), \text{Sibling}(X, Y). \\
\text{Brother2}(X, Y) & : \neg \text{Sibling}(X, Y), \text{Male}(Y). \\
\text{Grandparent}(X, Y) & : \neg \text{parent}(X, Z), \text{parent}(Z, Y).
\end{align*}
\]

Construct SLD proof trees for the following queries, and describe the computed answer substitutions.

(a) \text{Brother}(\text{John}, \text{Y})? \\
(b) \text{Brother}(X, Y)? \\
(c) \text{Brother2}(X, Y)? \\
(d) \text{Grandparent}(X, Y)? \\
(e) \text{Grandparent}(\text{Rose}, \text{Joseph})?

Add rules for other family relationships, such as “sister”, “aunt”, “uncle”.

2. (Using the equational approach from lecture 17) What happens to the SLD proof tree for the path logic program considered in lectures if we add the fact \text{edge}(c, a)? What answer substitutions are found by a depth first search of this proof tree? Are there any other answer substitutions elsewhere in the tree that are not discovered by a depth first search?
3. Compute the composition $\theta_1 \circ \theta_2$ for the following pairs of substitutions:

(a) $\theta_1 = [X \mapsto f(a, Y), Y \mapsto Z]$ and $\theta_2 = [X \mapsto g(X), Y \mapsto h(Z), U \mapsto V, W \mapsto Z]$

(b) $\theta_1 = [X \mapsto Y, Y \mapsto U, U \mapsto X]$ and $\theta_2 = [X \mapsto Y, Y \mapsto U, U \mapsto X]$.

Let $A$ be the atom $p(X, f(U, V), Y, Z)$. For each of the above pairs of substitutions, compute $(A \theta_1) \theta_2$ and $A(\theta_1 \circ \theta_2)$ and compare.

4. Consider the unifiers $\theta_1 = [X \mapsto T, Y \mapsto T]$ and $\theta_2 = [X \mapsto Y]$ of $S = \{p(X), p(Y)\}$. Show that $\theta_1$ is as least as general as $\theta_2$ and that $\theta_2$ is at least as general as $\theta_1$. Conclude that the most general unifier of a set of expressions need not be unique.

Optional Advanced Problems:

1. Prove that composition of substitutions is associative, as claimed in the lecture.

2. A graph is said to be acyclic if there does not exist a path from any node back to itself. Suppose that $G = (E, V)$ is an acyclic graph and let $P$ be the logic program consisting of the rules

$$
\text{path}(X,Y) : \neg \text{edge}(X,Y).
\text{path}(X,Z) : \neg \text{edge}(X,Y), \text{path}(Y,Z).
$$

together with a fact $\text{edge}(u,v)$ for each tuple $(u,v) \in E$. Prove that a depth first search search of the SLD-proof tree for this program is guaranteed to terminate, and will return one exactly answer substitution corresponding to each path in the graph. Will it return exactly one answer substitution for each pair of vertices $u, v$ such that there is a path from $u$ to $v$?