Logic Exercises

Technical University of Munich
Chair for Logic and Verification

Prof. Tobias Nipkow
Kevin Kappelmann

SS 2022 — Exercise Sheet 8 — 17.06.2022

Exercise 8.1. **[Simultaneous substitution]**
Recall that $[t_1/x_1, \ldots, t_n/x_n]$ is the *simultaneous* substitution of $x_1, \ldots, x_n$ by $t_1, \ldots, t_n$.

1. Can we always express $[t_1/x_1, \ldots, t_n/x_n]$ as a series of one-variable substitutions?
2. Can we always summarise a series of one-variable substitutions to a single simultaneous substitution?

Exercise 8.2. **[Most General Unifier]**
Consider the unification problem $x \stackrel{?}{=} f(y)$. Without running the unification algorithm, prove that

1. $\sigma_1 = [f(y)/x]$ is a most general unifier.
2. $\sigma_2 = [f(c)/x, c/y]$ is unifier, but not a most general unifier.

Exercise 8.3. **[Occurs check]**
What happens if one omits the occurs check in the unification algorithm? Find an example where the unification algorithm without occurs check diverges or returns the wrong result.

Exercise 8.4. **[Unifiable terms]**
Specify the most general unifiers for the following sets of terms, if one exists:

$$
\begin{aligned}
L_1 &= \{f(x, y), f(h(a), x)\} \\
L_2 &= \{f(x, y), f(h(x), x)\} \\
L_3 &= \{f(x, b), f(h(y), z)\} \\
L_4 &= \{f(x, x), f(h(y), y)\}
\end{aligned}
$$

Homework 8.1. [Unification] (+)
Use the algorithm presented in the lecture to compute a most general unifier for the following set of formulas: $\{P(g(x), f(a)), P(y, x), P(g(f(z)), f(z))\}$

Homework 8.2. [Untangling simultaneous substitution] (++)
Recall Exercise 8.1. Demonstrate how to "untangle" a simultaneous substitution that has been obtained by consolidating one-variable substitutions back into one-variable substitutions.

Homework 8.3. [Anti-Unification] (+++)
A term t is a *generalisation* of a list of terms S if for each $s \in S$ there is a substitution σ_s such that $t\sigma_s = s$. A term t is a *most specific generalisation* (msg) of S if for any generalisation t' of S, there is a substitution $\sigma_{t'}$ such that $t'\sigma_{t'} = t$.

Give a recursive procedure that computes the msg of a finite list S. Apply your algorithm to the list $S \coloneqq \left[f(g(x), x, d, x), f(x, g(x), d, g(x)), f(h(c), h(c), d, h(c))\right]$ (where c, d are constants) and prove that the returned msg is indeed an msg of S.

Hint: design an algorithm that operates recursively on the structure of terms.

Optional: Prove that your algorithm always returns the msg.

Homework 8.4. [We're Far From The Shallow Now] (+++)
In this exercise, we consider FOL without constants.

A term is called *shallow* if it contains no nested function. For example, x and $f(x)$ are shallow while $f(f(x))$ is not.

An atom is called *simple* if it only contains shallow terms. For example, $R(x)$ and $R(f(x))$ are simple while $R(f(f(x)))$ is not.

An atom is *covering* if every functional subterm of it contains all variables of the atom. For example, $R(x_1, x_2)$ and $R(f(x_1, x_2), x_2)$ are covering while $R(f(x_1), x_2)$ is not.

Let $A \coloneqq R(t_1, \dots, t_n)$ and $B \coloneqq R(t'_1, \dots, t'_n)$ be atoms that are simple and covering, $\mathsf{vars}(A) \cap \mathsf{vars}(B) = \emptyset$, and assume θ is an mgu of A, B. Show that $C \coloneqq A\theta = B\theta$ is simple.

> Nature will always maintain her rights and prevail in the end over any abstract reasoning whatsoever.
>
> — David Hume