

LOGICS EXERCISE

TU MÜNCHEN
INSTITUT FÜR INFORMATIK

PROF. TOBIAS NIPKOW
DR. PETER LAMMICH
SIMON WIMMER

SS 2016

EXERCISE SHEET 5

11.05.2016

Submission of Homework: Before tutorial on May 18

Exercise 5.1. [Unification]

1. Use the algorithm presented in the lecture to compute a most general unifier for the following set of formulas: $\mathbf{L} = \{P(x, y), P(f(a), g(x)), P(f(z), g(f(z)))\}$
2. Show that a naive implementation of the algorithm can have an exponential runtime.

Exercise 5.2. [Occurs Check]

What happens if one omits the occurs check in the unification algorithm? Find an example where a unification algorithm without occurs check diverges or returns the wrong result.

Exercise 5.3. [Resolution (Barber Paradox)]

Consider the following facts:

1. Every barber shaves those who do not shave themselves.
2. No barber shaves anyone who shaves himself.

Show that there are no barbers by resolution.

Exercise 5.4. [Simultaneous Substitution]

Recall that $F[t_1/x_1, \dots, t_n/x_n]$ is the *simultaneous* substitution of x_1, \dots, x_n by t_1, \dots, t_n .

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

Homework 5.1. [Natural Numbers and FOL] (5 points)

We consider the following axioms in an attempt to model the natural numbers in predicate logic:

1. $F_1 = \forall x \forall y (f(x) = f(y) \rightarrow x = y)$
2. $F_2 = \forall x (f(x) \neq 0)$
3. $F_3 = \forall x (x = 0 \vee \exists y (x = f(y)))$

Give a model with an *uncountable* universe for:

1. $\{F_1, F_2\}$
2. $\{F_1, F_2, F_3\}$

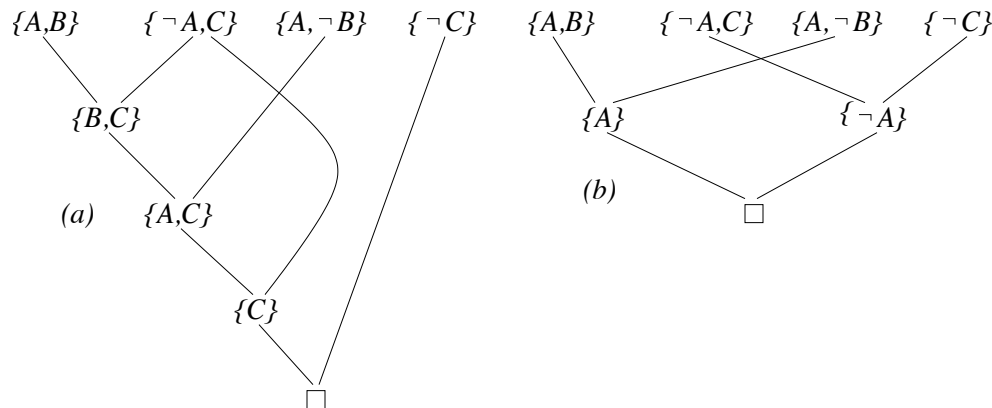
(Recall that a set S is uncountable if there is no bijection between S and \mathbb{N} . E.g. \mathbb{R} is an instance for S .)

Solution:

1. $U_A = \mathbb{R}_0^+$, $0^A = 0$, and $f^A(x) = x + 1$
2. Let $\mathbb{N}' = \{(x, 0) \mid x \in \mathbb{N}\}$ (i.e. we rename \mathbb{N} and \mathbb{R} apart.) Set $U_A = \mathbb{N}' \cup \mathbb{R}$, $f^A(x) = x + 1$ for all $x \in \mathbb{R}$, $f^A((x, 0)) = (x + 1, 0)$ for all $(x, 0) \in \mathbb{N}'$, and $0^A = (0, 0)$.

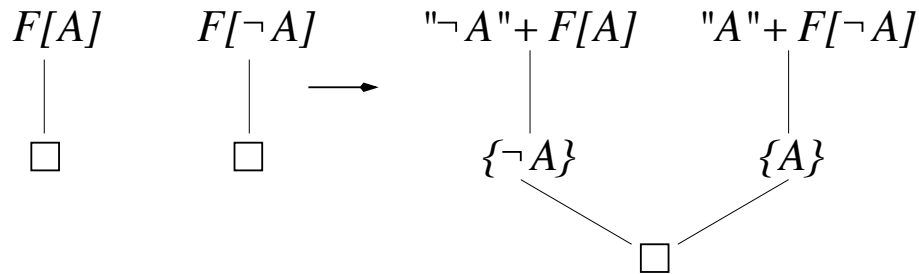
Homework 5.2. [Regular Resolution] (5 points)

We call an resolution proof for *propositional* logic *regular* if in every path from the leaves to the empty clause every literal is removed at most once by resolution. In the example below, (a) is regular because in the path on the very left, the literal A is removed two times. However, it is possible to give the equivalent regular proof (b).



Show that regular resolution is complete, i.e. there is a regular resolution proof for every unsatisfiable set of clauses. You can use the completeness proof for regular propositional resolution as a template for your proof.

Solution: The completeness proof for regular propositional resolution actually already constructs a *regular* resolution proof. This can easily be seen from the scheme that the proof is following:



By induction hypothesis, we can assume that the derivations on the left-hand side are regular. Adding A or $\neg A$ doesn't change their regularity. As a resolution involving A only appears in the last step on the rhs, the whole derivation on the rhs is regular. The base case is also given, as an unsatisfiable formula without atoms already has to be the empty clause and thus we talk about an empty resolution proof.

Homework 5.3. [Unifiable Terms] (5 points)

Specify the most general unifiers for the following sets of terms, if one exists:

$$L_1 = \{f(x, y), f(h(a), x)\}$$

$$L_2 = \{f(x, y), f(h(x), x)\}$$

$$L_3 = \{f(x, b), f(h(y), z)\}$$

$$L_4 = \{f(x, x), f(h(y), y)\}$$

Solution:

$$L_1 : \quad [h(a)/x, h(a)/y]$$

$$L_2 : \quad \text{No unifier, occurs check fails on } x \sim h(x)$$

$$L_3 : \quad [h(y)/x, b/z]$$

$$L_4 : \quad \text{No unifier, occurs check fails on } h(y) \sim y$$

Homework 5.4. [Anti-Unification] (5 points)

We want to explore the dual of unification, *anti-unification*. The goal of *anti-unification* is to find what is called the *least general generalization*. We will only consider the case of anti-unifying exactly two terms. Then the least general generalization of two terms can be informally be described as the least general term such that both terms under question are an instance of the least general generalization.

1. Give a formal definition of the least general generalization.
2. Specify an algorithm that can compute the least general generalization of two terms. We do *not* require a correctness proof.

Solution:

1. t is a generalization of s and u if there exist σ, σ' such that $s = t\sigma$ and $u = t\sigma'$. t is additionally a least general generalization if for every generalization t' of s and u there exists a σ such that $t = t'\sigma$.
2. Consider terms s and u . We compute $lgg(s, u)$ by a simple recursive procedure:
 - If $s = f(t_1, \dots, t_n)$ and $u = f(t'_1, \dots, t'_n)$ for some function symbol f and terms $t_1, t'_1, \dots, t_n, t'_n$, then return $f(lgg(t_1, t'_1), \dots, lgg(t_n, t'_n))$.
 - Else return $\mu(s, u)$ where $\mu(t, t')$ is a unique variable symbol for any two terms t, t' .

Note that the special case of constants (i.e. 0-ary function symbols) is covered by the first case already.