Assignment 4
Introduction to Computational Logic, SS 2012

Prof. Dr. Gert Smolka, Dr. Chad Brown
www.ps.uni-saarland.de/courses/cl-ss12/

Read in the lecture notes: Chapters 3 and 4

Note: Most of the exercises concern Chapter 4 (Untyped Lambda Calculus). The last exercises concern the material from Chapter 3 (Propositions and Proofs). We may still include such problems on the tests. Make sure you understand this material. All the material is important.

Exercise 4.1 Use beta reduction to derive normal forms of the following terms.

a) $(\lambda xy.fyx)ab$
b) $(\lambda fxy.fyx)(\lambda xy.yx)ab$
c) $(\lambda xy.y)((\lambda x.xx)(\lambda x.xx))a$
d) $(\lambda xx.x)yz$

Exercise 4.2 Find terms *true*, *false*, and *if* such that *if true* $x\ y \to_\beta^* x$ and *if false* $x\ y \to_\beta^* y$. Hint: Represent *true* and *false* as two-argument functions returning their first and second argument, respectively.

Exercise 4.3 Find terms *pair*, *fst*, and *snd* such that *fst* (*pair* $x\ y$) $\to_\beta^* x$ and *snd* (*pair* $x\ y$) $\to_\beta^* y$. Hint: Represent a pair (x, y) as the function $\lambda f.fxy$.

Exercise 4.4 Let Y be the term $\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$. Let f be a variable. Find a term to which both $f(Yf)$ and Yf reduce.

Exercise 4.5 Draw the nameless tree representations of the following terms.

a) $\lambda xyz.x$
b) $\lambda xyz.z(\lambda x.xz)$

Exercise 4.6 Make sure you understand the definitions of *subst* and *shift* well enough to reconstruct the essential parts of the defnitions.

Exercise 4.7 Compute the normal form of the following terms. Prove your answers are correct in Coq.

a) $(\lambda xy.x)y$
b) $(\lambda f.fxy)(\lambda uv.u)$

Exercise 4.8 Write a function $power : Nat \to Nat \to Nat$ that for two numbers m and n yields the power m^n (numbers represented as Church numerals).

Exercise 4.9 The following type represents pairs as functions:

```
Definition Prod (X Y : Prop) : Prop :=
forall Z : Prop, (X -> Y -> Z) -> Z.
```

Write functions *pair*, *fst*, and *snd* that construct and decompose pairs.

Exercise 4.10 Write a function $fac : Nat \to Nat$ that computes the factorial $n!$ of a number n (numbers represented as Church numerals). Hint: Iterate on pairs $(n, n!)$ starting with $(0, 0!)$.

Exercise 4.11 Prove the following without using *tauto*. (See Exercise 3.15.4 for an explanation for why such propositions are generally provable.)

```
Goal forall X : Prop,
~~(X \/ ~X).
```

```
Goal forall X Y : Prop,
~~(~(X /\ Y) -> ~X \/ ~Y).
```

Exercise 4.12 (Decidable Propositions) A proposition s is **decidable** if the proposition $s \vee \neg s$ is provable. Show that the following propositions are decidable.

a) `forall X : Prop, ~(X \/ ~X)`

b) `exists X : Prop, ~(X \/ ~X)`