Assignment 11
Introduction to Computational Logic, SS 2011

Prof. Dr. Gert Smolka, Dr. Chad Brown
www.ps.uni-saarland.de/courses/cl-ss11/

Read in the lecture notes: Chapter 6

Note: It is very important to do all the examples in the lecture notes and the exercises below in the system Coq.

Exercise 11.1 Prove the correctness of K.

```
Lemma K_correct (A B : Type) (b : B) :
 K A b O = b /\ forall n x, K A b (S n) x = K A b n.
```

Exercise 11.2 Recall the definition of *allb*:

```
Definition allb (p : bool -> bool) : bool := andb (p false) (p true).
```

Prove *allb* is correct:

```
Lemma allb_correct (p : bool -> bool) : allb p <-> forall a, p a.
```

Then prove the following:

```
Lemma allb3 : forall g:bool ^ 3 --> bool, forall x y,
   comp allb 2 g x y <-> forall z, g x y z.
```

Exercise 11.3 Formulate each of the equations below as a lemma in Coq and then prove the lemma.

a) $g\,a\,\vec{a} = g\,(a :: \vec{a})$ where $g : A^{S\,n} \to B$, $a : A$ and $\vec{a}$: *ilist* $A\,n$.

b) $(f \circ g)\,\vec{a} = f(g\,\vec{a})$ where $f : B \to C$, $g : A^n \to B$ and $\vec{a}$: *ilist* $A\,n$.

c) $(f \circ^2 (g, h))\,\vec{a} = f(g\,\vec{a})(h\,\vec{a})$ where $f : B \to C$, $g, h : A^n \to B$ and $\vec{a}$: *ilist* $A\,n$.

Exercise 11.4 For cascaded functions $g, h : A^n \to B$, we define $g \equiv h$ to mean $\forall \vec{a}$: *ilist* $A\,n. g\,\vec{a} = h\,\vec{a}$. This is defined as *Feq* in Coq in the lecture notes, and the infix notation == is given. Formulate the following equivalences as lemmas in Coq and prove them using the lemmas from Exercise 11.3.

a) $f \circ K_b^{A,n} \equiv K_{fb}^{A,n}$ where $f : B \to C$ and $b : B$.

b) $f \circ^2 (K_b^{A,n}, h) \equiv (f b) \circ h$ where $f : B \to C$, $b : B$ and $h : A^n \to B$.

Exercise 11.5 Consider the following boolean definition of implication (predefined in Coq).

```
Definition implb (b1 b2:bool) : bool := if b1 then b2 else true.
```

a) Prove the following lemma.

```
Lemma implb_negb_orb (a b : bool) : implb a b = orb (negb a) b.
```

b) Use the lemma from part (a) to prove the following equivalence of functions $bool^2 \to bool$.

```
Lemma Feq_implb_negb_orb (n : nat) (g h : bool ^ n --> bool):
  Feq n (comp2 implb n g h) (comp2 orb n (comp negb n g) h).
```

Exercise 11.6 In this exercise you will modify the construction of the certifying first function *firstc* so that it gives a function of type

$$\forall p : nat \to bool \,.\, ex\, p \;\to\; sig\, p$$

Start by assuming p is given and defining the *safe* predicate:

```
Variable p:nat -> bool.

Inductive safe (n : nat) : Prop :=
| safeI : p n -> safe n
| safeS : safe (S n) -> safe n.
```

a) Define a function *somec'* of type $\forall n : nat \,.\, safe\, n \;\to\; sig\, p$. (You may find it helpful to first construct the definition using a proof script.)

```
Definition somec' : forall n, safe n -> sig p :=
```

b) Prove the following and end the script with Defined:

```
Lemma safe_O : forall n, safe n -> safe O.
```

c) Prove the following and end the script with Defined:

```
Lemma ex_safe : ex p -> safe O.
```

d) Define a function *somec* of the required type:

```
Definition somec : ex p -> sig p :=
 ...
```