Assignment 7
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: Chapter 7

Exercise 7.1 Prove the following goal twice: Once with *discriminate* and once with *change* and without *discriminate*.

```
Goal forall (X : Type) (x : X) (xs : list X),
cons x xs <> nil.
```

Exercise 7.2 Prove the following goals twice: once with *injection* and once with *change* and without *injection*.

(a)
```
Goal forall (X : Type) (x y x' y' : X),
pair x y = pair x' y' -> y = y'.
```

(b)
```
Goal forall (X : Type) (x x' : X) (xs xs' : list X),
cons x xs = cons x' xs' -> xs = xs'.
```

Exercise 7.3 Prove the following goals.

(a) `Goal forall x, negb x <> x.`

(b) `Goal forall x, S x <> x.`

(c) `Goal forall x y z, x + y = x + z -> y = z.`

Exercise 7.4 Write boolean equality tests for the type *list nat* and prove that it agrees with Coq's equality.

Exercise 7.5 Prove if there is a boolean equality test for a type X, then equality on X is decidable.

```
Definition decidable (X:Prop) : Prop := X \/ ~X.

Goal forall X:Type, forall f:X -> X -> bool,
 (forall x y, f x y = true <-> x = y)
 -> forall x y:X, decidable (x = y).
```

Exercise 7.6 Prove the following goal.

```
Goal bool <> option bool.
```

Exercise 7.7 Prove the following generalized diagonalisation theorem. All diagonalisation results stated so far can be obtained as instances of the general result. The theorem shows that diagonalisation can be formulated as a purely logical result not depending on the inductive types *bool* and *nat*.

```
Definition strong (X : Type) : Prop :=
exists f : X -> X, forall x, f x <> x.

Theorem Cantor (X Y : Type) :
strong Y -> smaller X (X -> Y).
```

Exercise 7.8

a) Explain why $s \to t$ is a proposition if s is a type and t is a proposition.
b) Prove the following goal.

```
Goal forall (X : Type) (Y : Prop) ,
X -> Y <-> (exists x : X, True) -> Y.
```

Exercise 7.9 Prove the following goals.

(a) `Goal forall x,  ~ x < x.`

(b) `Goal forall x y,  x <= y -> x < y \/ x = y.`

(c) `Goal forall x y,  negb (x <= y) = (x > y).`

(d) `Goal forall x y,  x < y \/ x = y \/ x > y.`

Exercise 7.10 (Boolean Reflection) Prove the following goals.

(a) `Goal forall x y : bool,  x /\ y <-> andb x y.`

(b) `Goal forall (b : bool) (X : Prop),  (b <-> X) -> (~b <-> ~X).`

Exercise 7.11 Prove the following fact.

```
Goal forall x y, x <= y <-> exists z, x + z = y.
```

Exercise 7.12 Prove complete induction using size induction.

Exercise 7.13 Prove the following proposition in two ways.

```
Goal forall p : nat -> Prop,
p 0 -> p 1 -> (forall n, p n -> p (S (S n))) -> forall n, p n.
```

a) With a proof term using fix and match.
b) With complete induction. After a few steps you will be left with the claim $n \le Sn$. Prove this claim inline with the induction tactic. Use the tactic *clear* to clear away unnecessary assumptions before you apply the induction tactic.

Exercise 7.14 Specify multiplication and prove that Coq's predefined function satisfy the specifications. Also prove that two functions agree on all arguments if they satisfy the specification. Write a multiplication function using primitive recursion. Prove your function satisfies the specification.

Exercise 7.15 Prove the following variant of Kaminski's equation.

```
Goal forall (f g : bool -> bool) (x : bool),
f (f (f (g x))) = f (g (g (g x))).
```