Assignment 4
Semantics, WS 2011-2012

Prof. Dr. Gert Smolka, Dr. Chad Brown
www.ps.uni-saarland.de/courses/sem-ws11/

Exercise 4.1 Define a family of inductive propositions that behave like disjunctions and prove that your disjunctions are equivalent to Coq's predefined disjunctions.

Exercise 4.2 Consider the following inductive proposition.

```
Inductive decp : Prop -> Prop :=
| decp0 : forall X:Prop, ~X -> decp X
| decp1 : forall X:Prop, X -> decp X.
```

Prove the following.

```
Goal forall X:Prop, decp X <-> X \/ ~X.
```

Exercise 4.3 Define a family of inductive propositions that behave like existential quantifications. Arrange your definition such that you obtain a constructor *Ex* for which you can prove

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

Exercise 4.4 Prove the following goal.

```
Goal forall (X : Type) (x y : X) (p : X -> Prop),
Eq X x y -> p x -> p y.
```

Exercise 4.5 Prove the following goals.

```
Goal forall n, even n -> even (pred (pred n)).

Goal forall m n, even m -> even n -> even (m+n).

Goal forall m n, even (m+n) -> even m -> even n.
```

Exercise 4.6 One can define evenness with a boolean function.

```
Fixpoint evenb (n : nat) : bool :=
match n with
| 0 => true
| 1 => false
| S (S n') => evenb n'
end.
```

Prove that the boolean and the inductive definition agree. The proof goes through if you generalize the claim as follows.

```
Goal forall n,
(evenb n = true <-> even n) /\
(evenb (S n) = true <-> even (S n)).
```

Exercise 4.7 Prove that *leq* is transitive.

```
Goal forall x y z, leq x y -> leq y z -> leq x z.
```

Exercise 4.8 Prove that *leq* agrees with a boolean definition of the natural order.

```
Fixpoint leqb (x y : nat) : bool :=
match x,y with
| 0, _ => true
| S _, 0 => false
| S x', S y' => leqb x' y'
end.
```

```
Goal forall x y, leqb x y = true <-> leq x y.
```

Exercise 4.9 Define a recursive function

```
eq_nat : nat -> nat -> bool
```

and prove

```
Goal forall x y, eq_nat x y = true <-> x = y.
```

Exercise 4.10 Define a induction predicate

```
odd : nat -> Prop
```

and prove

```
Goal forall x, odd x <-> even (S x).
```

Give the inference rules for odd. You may need to formulate and prove a lemma to use.

Exercise 4.11 Define an inductive predicate

```
rel : exp -> nat -> Prop
```

and prove

```
Goal forall e n, rel e n <-> evalExp e = n.
```

Give the inference rules for *rel* (write $e \Downarrow n$ for *rel e n*).