Exercises

MAPi

2007

1 Logical Reasoning

1. Prove the following theorem:

   Theorem $ex4 : \forall\ (X : \mathsf{Set})\ (P : X \rightarrow \mathsf{Prop}), \sim(\forall\ x, \sim(P\ x)) \rightarrow (exists\ x, P\ x)$.

2. Assuming the *Excluded Middle* axiom, prove:

   (a) **Theorem** $Pierce : \forall\ P\ Q, ((P \rightarrow Q) \rightarrow P) \rightarrow P$.

   (b) **Theorem** $NNE : \forall\ P, \sim\sim P \rightarrow P$.

2 Reasoning about lists

The following exercises require the library `Lists`. You can load that library executing the command Require Import *Lists*.

1. Consider the following inductive relation:

   Inductive $last\ (A : \mathsf{Set})\ (x : A) : list\ A \rightarrow \mathsf{Prop} :=$
   $\mid last\_base : last\ x\ (cons\ x\ nil)$
   $\mid last\_step : \forall\ l\ y, last\ x\ l \rightarrow last\ x\ (cons\ y\ l)$.

   (a) Use `inversion` to prove that $\forall\ x, \sim(last\ x\ nil)$.

   (b) (Difficult) Try to avoid using that tactic.

2. Consider the following definition for the `Even` predicate:

   Inductive $Even : nat \rightarrow \mathsf{Prop} :=$
   $\mid Even\_base : Even\ 0$
   $\mid Even\_step : \forall\ n, Even\ n \rightarrow Even\ (S\ (S\ n))$.

   (a) Define the `Odd` predicate.

   (b) (Difficult) Prove that, for every number n, $Even\ n \rightarrow Odd\ (S\ n)$. (*Hint: you should strength the property to* $(Even\ n \leftrightarrow Odd\ (S\ n)) \wedge (Odd\ n \leftrightarrow Even\ (S\ n))$)

   (c) Define the function `rev` that reverses a list.

   (d) Prove that, for every list l, $rev\ (rev\ l) = l$.

   (e) Recall the definition for the function `app` (concatenation of lists). Prove that for every lists l_1 and l_2, $rev\ (app\ l1\ l2) = app\ (rev\ l2)\ (rev\ l1)$.

3. Consider the inductive predicate:

$$\textbf{Inductive } InL\ (A : \mathsf{Type})\ (a : A) : list\ A \to \mathsf{Prop} :=$$
$$\mid InHead : \forall\ (xs : list\ A), InL\ a\ (cons\ a\ xs)$$
$$\mid InTail : \forall\ (x : A)\ (xs : list\ A), InL\ a\ xs \to InL\ a\ (cons\ x\ xs).$$

Prove the following properties:

(a) $\forall\ (A : \mathsf{Type})\ (a : A)\ (l1\ \ l2 : list\ A), InL\ a\ l1 \lor InL\ a\ l2 \to InL\ a\ (app\ l1\ l2)$.

(b) $\forall\ (A : \mathsf{Type})\ (a : A)\ (l1\ \ l2 : list\ A), InL\ a\ (app\ l1\ l2) \to InL\ a\ l1 \lor InL\ a\ l2$.

4. Define the function `elem` that checks if an element belongs a list of integers (*Hint: we might import the `ZArith` to use the `Z_eq_dec` that tests for integer equality*).

5. Prove the correctness of `elem`, that is, $\forall\ (a : Z)\ (l1\ \ l2 : list\ Z), elem\ a\ (app\ l1\ l2) = orb\ (elem\ a\ l1)\ (elem\ a\ l2)$ (the function `orb` is the boolean-or function defined in Library `Bool`).