Master Parisien de Recherche en Informatique (MPRI)
Proof Assistants (2-7-2)

Self-evaluation

28 October 2024

You are invited to hand in your solutions to help us understand the current state. You are encouraged to write your name on the top right, but it is not mandatory – it would be very helpful for us, and we will send you individual feedback if we have your name.

Hint: Solve exercises that are easy for you first. Exercises are not necessarily in order of hardness.

Relevant constructors are

```
or_introl : ∀ {A} B, A → A ∨ B          or_intror : ∀ A {B}, B → A ∨ B
conj : ∀ {A B}, A → B → A ∧ B           I : True
inl : ∀ {A} B, A → A + B                inr : ∀ A {B}, B → A ∨ B
```

Question 1

Prove the following lemma using a proof script. Do not use `discriminate`, `congruence`, `inversion`.

```
Lemma O_neq_1 : 0 <> 1.
Proof.
```

Question 2

Given `A : Prop` and `B : Prop`, consider the two types `T1 := A ∨ B` and `T2 := A + B`.

Define a function of type `T1 → T2` or explain why it is not possible.

Define a function of type `T2 → T1` or explain why it is not possible.

Question 3

Give the type of the induction principle for the following predicate:

```
Inductive Forall {A : Type} (P : A → Prop) : list A → Prop :=
| Forall_nil : Forall P nil
| Forall_cons : ∀ (x : A) (l : list A), P x → Forall P l → Forall P (x :: l).
```

Question 4

Write down the goal (assumptions and conclusion) after executing the following proof script:

```
Inductive bintree A :=
| bt_leaf (a : A)
| bt_node (a : A) (l r : bintree A).

Fixpoint bt_map {A B} (f : A → B) (t : bintree A) : bintree B :=
  match t with
  | bt_leaf _ a ⇒ bt_leaf _ (f a)
  | bt_node _ a l r ⇒ bt_node _ (f a) (bt_map f l) (bt_map f r)
  end.

Lemma bt_map_id A (t : bintree A) :
    bt_map (fun x ⇒ x) t = t.
Proof.
  induction t as [a | a l ihl r ihr].
  - reflexivity.
  - cbn. rewrite ihl.
```

Question 5

Write down a proof term for `∀ A B : Prop, A ∨ B → ((A ∧ True) ∨ B) ∨ False.`