

Preuves assistées par ordinateur – TD n° 4

Mathématiques élémentaires en Coq**Exercice 1 – Relations d'ordre**

En Coq, on considère un type $E : \text{Set}$ muni d'une relation binaire R dont on suppose qu'elle satisfait aux axiomes des relations d'ordre :

```
Parameter E : Set.
Parameter R : E -> E -> Prop.

Axiom refl : forall x : E, R x x.
Axiom trans : forall x y z : E, R x y -> R y z -> R x z.
Axiom antisym : forall x y : E, R x y -> R y x -> x = y.
```

On définit les notions de plus petit élément et d'élément minimal de la façon suivante :

```
Definition smallest (x0 : E) := forall x : E, R x0 x.
Definition minimal (x0 : E) := forall x : E, R x x0 -> x = x0.
```

Quels sont les types des objets `smallest` et `minimal` ?

Énoncer en Coq puis démontrer les lemmes suivants :

1. Si R admet un plus petit élément, alors celui-ci est unique.
2. Le plus petit élément, s'il existe, est un élément minimal.
3. Si R admet un plus petit élément, alors il n'y a pas d'autre élément minimal que celui-ci.

Indications : En Coq, une définition s'utilise en remplaçant le *definiendum* par son *definiens* à l'aide de la tactique `unfold <definiendum>` (*unfold* = déplier). L'égalité se traite à l'aide des tactiques `reflexivity`, `symmetry`, `transitivity <terme>` et `rewrite <hypothèse>`.

Exercice 2 – Logique classique

Dans cet exercice, on suppose la règle de raisonnement par l'absurde, que l'on déclare en Coq de la manière suivante :

```
Axiom not_not_elim : forall A : Prop, ~~A -> A.
```

1. Montrer en Coq que cet axiome entraîne le tiers-exclus : `forall A : Prop, A \/ ~ A`.

On se propose maintenant de formaliser le paradoxe des buveurs, dû à Smullyan :

*Dans toute pièce non vide on peut trouver une personne ayant la propriété suivante :
Si cette personne boit, alors tout le monde dans la pièce boit.*

2. Déclarer en Coq les divers éléments du problème (en s'inspirant de l'exercice 1).
3. Énoncer le paradoxe et en effectuer la preuve (laquelle repose sur le tiers-exclus).

Exercice 3 – Sous-ensembles

Étant donné un type $E : \mathbf{Set}$ dans Coq, on s'intéresse aux sous-ensembles de E , qu'il est commode de représenter par des prédicats unaires sur E , c'est-à-dire par des objets de type $E \rightarrow \mathbf{Prop}$.

1. Définir en Coq un prédicat binaire `subset` : $(E \rightarrow \mathbf{Prop}) \rightarrow (E \rightarrow \mathbf{Prop}) \rightarrow \mathbf{Prop}$ exprimant l'inclusion entre deux sous-ensembles. Montrer que cette relation est réflexive et transitive. Est-elle antisymétrique ?
2. Définir en Coq un prédicat binaire `eq` : $(E \rightarrow \mathbf{Prop}) \rightarrow (E \rightarrow \mathbf{Prop}) \rightarrow \mathbf{Prop}$ exprimant l'égalité extensionnelle de deux ensembles. Montrer qu'il s'agit d'une relation d'équivalence.
3. Définir en Coq les opérateurs d'union et d'intersection binaire sur les sous-ensembles de E . Montrer que ces deux opérations sont associatives, commutatives, idempotentes, et distributives l'une par rapport à l'autre.