Fondements des systèmes de preuves – TP n° 4

Théorie des ensembles en Coq

Le but de ce TP est de formaliser en Coq un fragment de la théorie des ensembles [1], sous forme purement axiomatique. Pour cela, on introduit les paramètres suivants :

```
Parameter Ens : Type.               (* L'univers des ensembles *)
Parameter someset : Ens.            (* Un habitant de l'univers *)
Parameter elt : Ens -> Ens -> Prop. (* La relation d'appartenance *)
```

Afin de bénéficier des tactiques liées à l'égalité, on utilise l'égalité prédéfinie de Coq ("=") pour modéliser la relation d'égalité extensionnelle de la théorie des ensembles. Les axiomes de la théorie seront introduits au fur et à mesure dans les exercices qui suivent.[2]

Exercice 1 (Inclusion et extensionnalité)

1. Définissez en Coq la relation d'inclusion `subset : Ens -> Ens -> Prop`.
2. Montrez que cette relation est réflexive et transitive. Est-elle antisymétrique ?

On pose l'axiome suivant, qui exprime que deux ensembles ayant les mêmes éléments sont égaux :

```
Axiom extensionality :
  forall (a b : Ens), (forall x, elt x a <-> elt x b) -> a = b.
```

3. Vérifiez que l'implication réciproque est vraie en l'absence d'axiome.
4. Montrez à l'aide de l'axiome d'extensionnalité que l'inclusion est antisymétrique. Vérifiez que cette propriété est équivalente à l'axiome d'extensionnalité.

Exercice 2 (Axiome de la paire) L'axiome de la paire est introduit en Coq sous forme skolémisée à l'aide de la déclaration `Parameter pair : Ens -> Ens -> Ens` et de l'axiome :

```
Axiom pair_axiom :
  forall (a b x : Ens), elt x (pair a b) <-> x = a \/ x = b.
```

1. Montrez *sans utiliser* `rewrite` que : `a = a' -> b = b' -> pair a b = pair a' b'`.
2. Montrez que : `pair a b = pair c d -> (a = c /\ b = d) \/ (a = d /\ b = c)`.

On définit le singleton en posant : `Definition single (a : Ens) := pair a a`.

3. Vérifiez que : `forall (a x : Ens), elt x (single a) <-> x = a`.

Le couple (i.e. la *paire ordonnée*) est défini en théorie des ensembles par : $(a, b) = \{\{a\}; \{a; b\}\}$.

4. Définissez une constante `couple : Ens -> Ens -> Ens` qui modélise cette définition, et vérifiez que : `couple a b = couple c d <-> a = c /\ b = d`.

[1] Les axiomes présentés ici sont ceux de la théorie des ensembles de Zermelo sans axiome de la fondation.

[2] L'introduction de nouveaux axiomes dans un système tel que Coq pose naturellement la question de la cohérence logique du système obtenu. Il est possible de démontrer que les axiomes de Zermelo (tels que nous les introduisons dans ces exercices) ne mettent pas en danger la cohérence du formalisme de Coq.

Exercice 3 (Axiome de séparation) L'axiome de séparation est introduit en Coq à l'aide de la déclaration `Parameter separ : Ens -> (Ens -> Prop) -> Ens` et de l'axiome :

```
Axiom separ_axiom :
  forall (P : Ens -> Prop),
    forall (a x : Ens), elt x (separ a P) <-> elt x a /\ P x.
```

1. Définissez l'ensemble vide `emptyset : Ens`, et vérifiez qu'il est vide.
   (Indication : il est utile de considérer la constante `someset` introduite au début.)
2. Montrez que tout ensemble vide est égal à `emptyset`.
3. Montrez qu'il n'existe pas d'ensemble de tous les ensembles.
   (Indication : déduire de l'existence d'un tel ensemble le paradoxe de Russell.)

Exercice 4 (Axiome de la réunion) On introduit l'axiome de la réunion à l'aide de la déclaration `Parameter union : Ens -> Ens` et de l'axiome :

```
Axiom union_axiom :
  forall (a x : Ens), elt x (union a) <-> exists y, elt y a /\ elt x y.
```

1. Définissez l'opérateur d'union binaire `cup : Ens -> Ens -> Ens` et montrez qu'il vérifie la propriété : `forall (a b x : Ens), elt x (cup a b) <-> elt x a \/ elt x b`.

Exercice 5 (Axiome des parties) On introduit l'axiome des parties à l'aide de la déclaration `Parameter power : Ens -> Ens` et de l'axiome :

```
Axiom power_axiom :
  forall (a x : Ens), elt x (power a) <-> subset x a.
```

1. Vérifiez que l'ensemble vide et `a` sont des éléments de `power a`.
2. Démontrez que : `forall (a : Ens), union (power a) = a`.
3. Montrez que : `forall (a : Ens), exists (x : Ens), ~ elt x a`.
   (Indication : cet énoncé est classiquement équivalent au fait qu'il n'existe pas d'ensemble de tous les ensembles. On peut cependant en donner une preuve constructive en considérant l'objet `x = power (union a)`.)

Exercice 6 (Axiome de l'infini) En théorie des ensembles, on pose $0 = \varnothing$ et $s(x) = x \cup \{x\}$.

1. Définissez des constantes `zero` et `succ` correspondant à ces constructions, et montrez que `forall x : Ens, succ x <> zero`.

L'axiome de l'infini exprime qu'il existe un ensemble contenant `zero` et clos par successeur :

```
Parameter big : Ens.
Axiom big_axiom : elt zero big /\ forall x, elt x big -> elt (succ x) big.
```

L'ensemble des entiers naturels $\mathbb{N}$ est défini comme le plus petit sous-ensemble de `big` contenant 0 et clos par successeur.

2. Définissez une constante `N : Ens` correspondant à cette construction de $\mathbb{N}$, et vérifiez que l'ensemble `N` contient `zero` et est clos par `succ`.
3. Énoncez et prouvez le principe de récurrence sur les entiers naturels.
4. Montrez que la fonction successeur est injective sur `N`.