

Preuves assistées par ordinateur – TD n° 8

Définitions inductives de prédicats

Exercice 1 – Une première définition de la clôture réflexive-transitive

Sur un type de données $T : \text{Set}$ fixé, on cherche à définir la clôture réflexive-transitive d'une relation $R : T \rightarrow T \rightarrow \text{Prop}$, à savoir la plus petite relation réflexive et transitive R^* contenant R . Il est naturel de définir inductivement cette relation à partir des trois règles suivantes :

$$\frac{R(x, y)}{R^*(x, y)} \quad \frac{}{R^*(x, x)} \quad \frac{R^*(x, y) \quad R^*(y, z)}{R^*(x, z)}$$

1. Les règles ci-dessus expriment que la relation R^* contient R , est réflexive et transitive. Comment exprime-t-on que R^* est la plus petite relation satisfaisant ces propriétés ?

En Coq, cette définition inductive s'effectue de la manière suivante :

```
Inductive clos1 (R : T -> T -> Prop) : T -> T -> Prop :=
| c11_base : forall x y, R x y -> clos1 R x y
| c11_refl : forall x, clos1 R x x
| c11_trans : forall x y z, clos1 R x y -> clos1 R y z -> clos1 R x z.
```

La définition ci-dessus, qui est paramétrée par la relation R , définit en réalité un *constructeur de relation* $\text{clos1} : (T \rightarrow T \rightarrow \text{Prop}) \rightarrow T \rightarrow T \rightarrow \text{Prop}$ qui à chaque relation binaire R associe sa clôture réflexive-transitive $\text{clos1 } R$, définie au moyen des constantes `c11_base`, `c11_refl` et `c11_trans` (qui correspondent aux trois règles) et du principe d'induction `clos1_ind`, qui exprime que $\text{clos1 } R$ est la plus petite relation réflexive et transitive contenant R .

2. Comparer le principe d'induction `clos1_ind` avec celui formulé à la question 1. Lequel des deux est le plus général ? Le plus facile à utiliser ? Sont-ils équivalents ?
3. Montrer que si R est symétrique, alors $\text{clos1 } R$ est symétrique également.

Exercice 2 – Une autre définition de la clôture réflexive-transitive

Dans les démonstrations cependant, il est souvent plus commode de définir la clôture réflexive-transitive d'une manière un peu différente, à savoir comme la relation R_2^* engendrée par les deux règles suivantes :

$$\frac{}{R_2^*(x, x)} \quad \frac{R_2^*(x, y) \quad R(y, z)}{R_2^*(x, z)}$$

Cette définition alternative se modélise en Coq au moyen de la définition inductive suivante :

```
Inductive clos2 (R : T -> T -> Prop) : T -> T -> Prop :=
| c12_refl : forall x, clos2 R x x
| c12_next : forall x y z, clos2 R x y -> R y z -> clos2 R x z.
```

Le but de cet exercice est de montrer l'équivalence des deux définitions. Pour ce faire, on pourra suivre les étapes suivantes :

1. Montrer que `clos2 R x y` entraîne `clos1 R x y` (pour tous $x, y : T$).
2. Montrer que `R x y` entraîne `clos2 R x y` (pour tous $x, y : T$).
3. Montrer que `clos2 R` est une relation transitive.
4. En déduire que `clos1 R x y` entraîne `clos2 R x y` (pour tous $x, y : T$).
5. Montrer que l'opération de clôture réflexive-transitive est idempotente, c'est-à-dire que :
`clos1 (clos1 R) x y` \leftrightarrow `clos1 R x y` pour tous $x, y : T$.

Exercice 3 – Une troisième définition de la clôture réflexive-transitive

On travaille toujours avec type $T : \text{Set}$ fixé.

1. Définir la relation identité `id : T → T → Prop` ainsi que l'opérateur de composition de relations `comp : (T → T → Prop) → (T → T → Prop) → T → T → Prop`.
2. Définir une fonction `puiss : (T → T → Prop) → nat → T → T → Prop` telle que `puiss R n` calcule la puissance n -ième de la relation (par l'opération de composition).

Une troisième définition de la clôture réflexive-transitive d'une relation R est donnée par la réunion des puissances successives de R , c'est-à-dire par :

Definition `clos3 (R : T → T → Prop) (x y : T) := exists n, puiss R n x y`.

3. Montrer que cette définition est équivalente aux deux précédentes (`clos1` et `clos2`).