

Preuves de programmes – TP n° 5

Développement de fonctions partielles en Coq

On considère un développement en Coq paramétré par un type de données $A : \text{Type}$.

Exercice 1 – Accès à un élément d’une liste

1. Définir un prédicat $\text{Nth} : \text{list } A \rightarrow \text{nat} \rightarrow A \rightarrow \text{Prop}$ spécifiant le fait qu’un certain élément de A est le n -ème élément d’une liste. Montrer que ce prédicat est bien fonctionnel :

$$\text{forall } l \ n \ a \ a', \quad \text{Nth } l \ n \ a \rightarrow \text{Nth } l \ n \ a' \rightarrow a = a'$$

Montrer que la fonction correspondant à ce prédicat est bien définie lorsque n est inférieur à la taille de la liste :

$$\text{forall } l \ n, \quad n < \text{length } l \rightarrow \text{exists } a, \text{Nth } l \ n \ a$$

Montrer également la réciproque de cette dernière propriété.

2. Écrire une fonction $\text{nth_default} : \text{list } A \rightarrow \text{nat} \rightarrow A \rightarrow A$ dont le dernier argument sert de réponse par défaut lorsque la liste n’est pas assez grande. Montrer que :

$$\text{forall } l \ n \ d, \quad n < \text{length } l \rightarrow \text{Nth } l \ n \ (\text{nth_default } l \ n \ d)$$

3. Écrire une fonction $\text{nth_option} : \text{list } A \rightarrow \text{nat} \rightarrow \text{option } A$ et montrer que :

$$\text{forall } l \ n \ a, \quad \text{nth_option } l \ n = \text{Some } a \leftrightarrow \text{Nth } l \ n \ a$$

4. Écrire une fonction $\text{nth} : \text{forall } l \ n, \ n < \text{length } l \rightarrow \{ a \mid \text{Nth } l \ n \ a \}$, tout d’abord à l’aide de tactiques Coq, puis à l’aide de la méthode `Program`. Obtenir le programme Ocaml correspondant par extraction Coq.

Exercice 2 – Division euclidienne

En s’inspirant de l’exercice précédent, spécifier puis construire une fonction de division euclidienne pour les entiers naturels de Coq. On utilisera l’algorithme suivant :

$$\text{div } a \ b = \text{si } b \leq a \text{ alors } 1 + (\text{div } (a-b) \ b) \text{ sinon } 0$$

Il pourra être utile d’utiliser le lemme d’induction généralisé présent dans la librairie `Wf_nat` :

$$\begin{aligned} &\text{lt_wf_rec} \\ &: \text{forall } (n:\text{nat})(P:\text{nat} \rightarrow \text{Set}), \\ &\quad (\text{forall } m, (\text{forall } p, \ p < m \rightarrow P \ p) \rightarrow P \ m) \rightarrow P \ n \end{aligned}$$