TME 4 Types inductifs

7 novembre 2017

1 Arbres binaires

1. Définir un type inductif des arbres binaires polymorphes en A, portant des A à leurs feuilles. Définir l'arbre $((1,2),5)$.
2. Définir la fonction `mirror` qui inverse les feuilles droites et gauches de l'arbre.
3. Prouver que `mirror` est involutive (i.e., `mirror (mirror t) = t`).
4. Définir la fonction `elements` qui retourne dans une liste le résultat du parcours de l'arbre de gauche à droite.
5. Prouvez que les éléments du miroir de t sont le renversement (au sens de `List.rev`) des éléments de t.
6. Prouvez que la taille (nombre de noeuds) d'un arbre t est comprise entre sa hauteur h (nombre de noeuds dans une chaîne) et $\sum_{i=0}^{h} i$. Attention : cette question nécessite plusieurs définitions et lemmes intermédiaires ; c'est votre travail que de les identifier.

2 Connecteurs logiques

1. Le connecteur logique "et" est défini en Coq par l'inductif à un constructeur :

   ```
   Inductive and (A B : Prop) :=
   | conj : A -> B -> and A B.
   ```

   que l'on peut lire comme : *"la seule façon de construire une preuve de* `and A B` *: à l'aide d'une preuve de* `A` *et d'une preuve de* `B`". Prouver les lemmes suivants :

   (a) `and A B -> A` (première élimination)

   (b) `and A B -> B` (deuxième élimination)

   (c) `and A B -> and B A` (symétrie)

2. De même, le connecteur logique "ou" est défini en Coq par l'inductif à deux constructeurs :

   ```
   Inductive or (A B : Prop) : Prop :=
   | or_introl : A -> or A B
   | or_intror : B -> or A B.
   ```

   que l'on peut lire comme *"il y a deux façons de construire une preuve de* `or A B` *: à l'aide d'une preuve de A, ou à l'aide d'une preuve de B*. Prouvez les lemmes suivants :

   (a) `or A B -> or B A` (symétrie)

   (b) `or A B -> (A -> C) -> (B -> C) -> C` (élimination)

   (c) `or A (and B C) -> and (or A B) (or A C)` (distributivité)