Institut National
Universitaire
Champollion

TD 03 - Révisions, terminaison

Types de données, preuves
L3 INFO - Semestre 6 - 2017

Exercice 1 - Fonction d'Ackermann

On rappelle que la fonction d'Ackermann est donnée par :

```
let rec ack = function
(0,y) -> y+1
|(x,0) -> ack(x-1, 1)
|(x,y) -> ack(x-1, ack(x,y-1)) ;;
```

Les paramètres x et y sont des entiers naturels ($\mathbb{N}$). Justifier que la fonction d'Ackermann termine.

Exercice 2 - Arbres binaires [Révisions + Terminaison]

On considère le type

```
type 'a arbre_bin =
Feuille of 'a
| Noeud of 'a * 'a arbre_bin * 'a arbre_bin ;;
```

1. Définir une fonction `echanger` qui échange partout les sous-arbres gauches et droits.

   ```
   echanger (Noeud(1, Feuille 2, Noeud(3, Feuille 4, Feuille 5))) ;;
   - : Noeud (1, Noeud (3, Feuille 5, Feuille 4), Feuille 2)}
   ```

2. Montrer par induction : $\forall t$:`arbre_bin` , `echanger(echanger(`t`))` = t
3. (a) **Écrire une fonction** `profondeur : 'a arbre_bin -> int` **qui calcule la profondeur d'un arbre binaire. Par convention, la hauteur d'une Feuille sera 0.**

   (b) **Montrer par induction que pour n'importe quel arbre binaire** `t`**,** `profondeur(t)` $\geqslant 0$**.**

   (c) Prouver la terminaison de votre fonction `echanger`.
4. Considérons la fonction suivante :

   ```
   let rec aplati = function
   [ ] -> [ ]
   | Feuille x :: q -> Feuille x :: (aplati q)
   | Noeud(x,t1,t2) :: q -> Feuille x :: aplati(t1 ::t2 ::q) ;;
   ```

   L'objectif est de prouver que cette fonction termine.

   (a) Quel est le type de cette fonction ? Que fait-elle ?

   (b) Rappelons que l'ordre sur les multiensembles défini en cours est bien fondé. En utilisant une mesure m à valeurs dans l'ensemble des multiensembles, prouver la terminaison de la fonction `aplati`.