Master Parisien de Recherche en Informatique 2007–2008
Assistants de Preuve

TD/TP 4 - Programmation fonctionnelle

Mardi 18 décembre 2007

1 Monades

1. Proposer une traduction fonctionelle du programme suivant de multiplication des feuilles d'un arbre en s'arrêtant dès qu'un zéro est atteint :

```
type tree = Leaf of nat | Node of tree * tree
exception Zero
let rec multe t = match t with
   Leaf 0 ⇒ raise Zero | Leaf n ⇒ n
 | Node (t1,t2) ⇒ multe t1 * multe t2
let mult t = multe t with Zero ⇒ 0
```

2. On considère un langage avec du non-déterminisme fini (ie on a une construction $c\&d$ qui peut produire soit une évaluation de c soit une évaluation de d). Proposer une représentation fonctionelle monadique pour ces programmes, c'est-à-dire définir le type $M(A)$ les opérations `unit` et `bind` ainsi que l'interprétation de la construction non déterministe $\&$.

2 Preuve par réflexion et langage de tactiques

Cet exercice illustre la construction de tactiques par réflexion et l'utilisation du langage de tactiques pour construire la fonction de réification.

Le fichier `ltac_refl_ex.v` contient l'exemple de tactique par réflexion illustré pendant le cours (simplification de formules conjonctives).

1. Implémenter la fonction de simplification de votre choix.
2. Prouver sa correction.
3. Utiliser cette fonction pour implanter une tactique de simplification.
4. Améliorer la fonction de réification de manière à associer la même variable à 2 occurrences de la même formule.
   Hint : En Ltac, on peut tester l'égalité de 2 termes de la manière suivante :

```
match t=u with ?x=?x ⇒ ... end
```

3 Modules

1. Proposer un type de module `FinSet` pour des ensembles finis qui comporte le type des éléments vu comme un ensemble `O` ordonné. On modélisera les fonctions de base (ensemble vide, ajout, suppression, appartenance).
2. Proposer un foncteur pour implanter une structure d'ensembles finis paramétré par la structure d'ordre sur les éléments en utilisant des listes.
3. Proposer un foncteur définissant les opérations d'union et d'intersection paramétré par un module quelconque implantant les ensembles finis.