Licence d'Informatique

TD d'IP : Feuille 9

Listes chaînées

Objet

- Implémentation de listes chaînées.
- Type abstrait, sélecteurs, constructeurs.
- Programmation fonctionnelle.

Exercice 1 : On définit les listes de façon abstraites grâce au type abstrait ci-dessous, où `vide` est une constante, `liste-vide?` un prédicat, `cons` un constructeur, `car` et `cdr` des sélecteurs.

- constante `vide` $\in$ Listes
- `liste-vide?` : Listes $\rightarrow$ {`vrai`, `faux`}
  - `liste-vide?`(`vide`) = `vrai`
  - `liste-vide?`(`cons`(x,l)) = `faux`
- `cons` : Objets x Listes $\rightarrow$ Listes
- `car` : Listes \ `vide` $\rightarrow$ Listes
  - `car`(`cons`(x,l)) = x
- `cdr` : Listes \ `vide` $\rightarrow$ Listes
  - `cdr`(`cons`(x,l)) = l

On parle de *programmation fonctionnelle* lorsque les fonctions proposées ne modifient jamais les objets existants (pas d'effet de bord). L'opposé de *programmation fonctionnelle* est la *programmation impérative* où le programmeur manipule directement le contenu de la mémoire.
Donner une implémentation de type fonctionnel du type abstrait liste défini ci-dessus, en utilisant une structure de liste simplement chaînée sans entête.

Exercice 2 : Rajouter au module de liste fonctionnelle les fonctions `longueur(l)`, `concatene(l1, l2)`, `applique(f,l)`, `prefixe(i,l)`, `suffixe(i,l)`, et une fonction testant l'égalité de deux listes.

Exercice 3 : Rajouter une fonction qui supprime d'une liste `l` toutes les occurrences d'un objet `o` donné.
