Programmation Fonctionnelle : feuille 3

Listes

Exercice 3.1

1. Écrire une version simplifiée de `mapcar(f l)` dont l'action est d'appliquer `f` sur chacun des éléments de l.
2. La tester avec $f(x) = x + 10$.

Exercice 3.2

1. Écrire une version simplifiée de `remove-if(pred l)`. Cette fonction reconstruit la liste `l` sans les éléments qui vérifient `pred`.
2. La tester avec $pred(x) \equiv (x = 0)$.

Exercice 3.3

1. Écrire une version de `make-list(n e)`. Cette fonction construit la liste qui contient n fois l'élément e.
2. Obtenir la liste `(glou glou glou glou glou)`.

Exercice 3.4

1. Écrire une version de `assoc(key l)` où `l` est une liste d'association (une liste de paires pointées). Cette fonction donne en résultat, soit la première paire dont la clef est `key`, soit “faux” s'il n'y en a pas.
2. La tester sur la liste d'association dont chaque paire est constituée d'une chaîne de caractères et d'un symbole de fonction arithmétique associé (e.g. `(‘‘add’’ . +)`).

N.B. Vous pouvez également tenter d'écrire des versions complètes des fonctions ci-dessus en vérifiant leurs spécifications sur la documentation Lisp que vous avez en ligne.