Programmation Fonctionnelle : feuille 4

Structures de données

Exercice 4.1 *type abstrait*

Écrire la base d'une implémentation de la notion d'ensemble à l'aide de listes (utiliser un argument mot-clé pour l'égalité):

1. Définir une constante `empty-set` pour représenter l'ensemble vide.
2. Écrire la fonction `set-emptyp` qui teste si un ensemble est vide.
3. Écrire la fonction `set-adjoin` qui ajoute un élément à un ensemble.
4. Écrire la fonction `set-member` qui teste si un élément appartient à un ensemble.
5. Écrire la fonction `set-from-list` qui construit l'ensemble des éléments contenus dans une liste.

Remarque : si on choisit d'implémenter les ensembles par des liste, on pourra utiliser les fonctions Lisp `adjoin`, `member`, `remove-duplicates`, `member`.

Exercice 4.2 *tableaux*

Pour tester le générateur de nombres aléatoires de Lisp (accessible grâce à la fonction `random`), on veut imprimer un histogramme de m valeurs aléatoires comprises entre 0 et $n-1$ produites par le générateur. pour compter le nombre de fois qu'une valeur est produite, on utilisera un tableau à n entrées. Voici un exemple d'histogramme produit pour les valeurs $m = 200$ et $n = 11$.

```
*  (print-histogram *histogram*)
 0 [ 20] ********************
 1 [ 20] ********************
 2 [ 21] *********************
 3 [ 19] *******************
 4 [ 14] **************
 5 [ 15] ***************
 6 [ 22] **********************
 7 [ 13] *************
 8 [ 17] *****************
 9 [ 17] *****************
10 [ 23] ***********************
NIL
*
```

On pourra mémoriser m et n dans des variables globales.