CPI1 2009-2010

Algorithmique fonctionnelle

Nga Nguyen – Stefan Bornhofen – Peio Loubière

TD5: Listes

Définir les fonctions...

1. **listSum** qui calcule la somme d'une liste de réels.

```
Fonction listSum (L: Liste de Réels): Réel
Début
        Si estVide(L) Alors
                Retourner 0
        Sinon
                Retourner premier(L) + listSum(reste(L))
        FinSi
Fin
```

2. **average** pour calculer la moyenne de tous les réels qui sont fournis dans une liste. Par exemple: average((2 -1)) = 0.5 et average((0 0 3 1)) = 1.

```
Fonction average (L: Liste de Réels): Réel
Début
        Retourner listSum(L)/longueur(L)
Fin
```

3. **listMin** qui renvoie le minimum d'une liste de réels donnée. Si la liste est vide, la fonction renvoie 0.

```
Fonction listMin(L: Liste de Réels): Réel
Début
        Si estVide(L) Alors
                Retourner 0
        SinonSi longueur(L)=1
                Retourner premier(L)
        Sinon
                Retourner min(premier(L), listMin(reste(L)))
        FinSi
Fin
```

4. **isEqual** qui vérifie l'égalité de deux listes. $(a_1,a_2,...,a_n) = (b_1,b_2,...,b_m)$ si et seulement si $n = m$ et $a_i = b_i$ pour tous les $i=1...n$.

```
Fonction isEqual(L1,L2: Liste de T): Booléen
Début
   Si longueur(L) != longueur(L2) Alors
      Retourner faux
   SinonSi estVide(L1)
      Retourner vrai
   Sinon
      Retourner (premier(L1) = premier(L2))
               et isEqual(reste(L1),reste(L2))
   FinSi
Fin
```

5. **rcons** qui ajoute un élément à la fin de la liste, contrairement à cons qui ajoute à la tête de la liste. Exemple: rcons (1, (2 3 4)) = (2 3 4 1).

```
Fonction rcons(x: T, L: Liste de T): Liste de T
Début
        Si estVide(L) Alors
                Retourner cons(x,())
        Sinon
                Retourner cons(premier(L),rcons(x,reste(L)))
        FinSi
Fin
```

6. **removeFirst** à deux arguments dont les valeurs sont respectivement un objet et une liste, qui retourne la liste privée de la première occurence de l'objet. Par exemple, removeFirst (1, (2 7 1 7 3 1)) = (2 7 7 3 1).

```
Fonction removeFirst(x: T, L: Liste de T): Liste de T
Début
        Si estVide(L) Alors
                Retourner L
        SinonSi premier(L) = x
                Retourner reste(L)
        Sinon
                Retourner cons(premier(L),removeFirst(x,reste(L)))
        FinSi
Fin
```

7. **mirror** qui renvoie une liste donnée dans l'ordre inverse. Par exemple, mirror ((1 2 3)) = (3 2 1).

```
Fonction mirror(L: Liste de T): Liste de T
Début
        Si estVide(L) Alors
                Retourner L
        Sinon
                Retourner rcons(premier(L),mirror(reste(L)))
        FinSi
Fin
```

8. **get** qui premet de récupérer l'élément à une position donnée. Par exemple: get (2, (“a” “b” “c” “d”)) = “c”

```
Fonction get(L: Liste de T, pos: Entier): T
Début
        Si pos = 0 Alors
                Retourner premier(L)
        Sinon
                Retourner get(reste(L),pos-1)
        FinSi
Fin
```

9. **subList** qui extrait d'une liste la sous-liste des éléments entre la position p1 (inclus) et p2 (exclus). Exemple: subList((“a” “b” “c” “d”), 1,3) = (“b” “c”)

```
Fonction subList(L: Liste de T, p1, p2: Entier): Liste de T
Début
        Si p1<0 ou p2<0 ou
         p1>longueur(L) ou p2>longueur(L) ou p1>p2 Alors
                Retourner () // erreur: retourner liste vide
        SinonSi p1>0
                Retourner subList(reste(L),p1-1,p2-1)
        SinonSi p2>0 {p1=0}
                Retourner cons(premier(L),subList(reste(L),0,p2-1))
        Sinon {p1=0, p2=0}
                Retourner ()
        FinSi
Fin
```

10. **listPalindrome** qui teste si les éléments d'une liste forment un palindrome. Par exemple: listPalindrome((1 2 3 3)) = faux, listPalindrome((“t” “a” “t”)) = vrai

```
Fonction listPalindrome(L: Liste de T): Booléen
Début
        Si longueur(L)<2
                Retourner vrai
        SinonSi premier(L) != get(L,longueur(L)-1)
                Retourner faux
        Sinon
                Retourner listPalindrome(subList(L,1,longueur(L)-1))
        FinSi
Fin
```

ou bien (plus malin!)

```
Fonction listPalindrome(L: Liste de T): Booléen
Début
        Retourner isEqual(L, mirror(L))
Fin
```

11. **explode** qui décompose une phrase en une liste de ses mots. Par exemple: explode(“Je suis un exo difficile”) = (“Je” “suis” “un” “exo” “difficile”).

```
Fonction explodeRec(c, mot: Chaîne): Liste de Chaînes
Début
  Si c=”” Alors
    Retourner cons(mot,())
  SinonSi extraire(c,0,1)=” ” // espace
    Retourner cons(mot,explodeRec(extraire(c,1,longueur(c)-1),””))
  Sinon
    Retourner explodeRec(extraire(c,1,longueur(c)-1),
                                        mot & extraire(c,0,1))
  FinSi
Fin

Fonction explode(c: Chaîne): Liste de Chaînes
Début
        Retourner exploreRec(c,””)
Fin
```