

LINFO1104 : TP4 Sémantique des procédures et Programmation d'ordre supérieur

1. Definition/Appel de procédure

- Expliquez brièvement que ce qu'il se passe lorsqu'une procédure est définie et lorsqu'elle est appelée.
- Pour le code suivant, décrivez l'état de la mémoire lorsque P est définie et les opérations effectuées sur la pile lors de l'appel de la procédure P.

```
local P in
  local Z in
    Z=1
    proc {P X Y} Y=X+Z end
  end
  local B A in
    A=10
    {P A B}
    {Browse B}
  end
end
```

2. Filtrer une liste Dans cette exercice, vous allez devoir implémenter des fonctions pour filtrer une liste d'entiers:

- Écrivez une fonction `{MakeMulFilter N}` qui renvoie une fonction à un argument I qui renvoie si I est une multiple de N.
- Ecrivez une fonction `{Filter L F}` qui renvoie une liste filtrée selon la fonction F.
- Pour une liste L, retournez les entiers pairs dans L.
- Pour une liste L, retournez les entiers multiple de 3 dans L.
- Maintenant pour une liste L, on ne veut garder que les nombres premiers. Que faut-t-il faire? Considérez qu'un entier N est premier s'il n'est divisible par aucun nombre k tel que $2 \leq k < N$

3. Réduction sur une liste Dans cette exercice, vous allez devoir implémenter des fonctions pour calculer un résultat à partir d'une liste. Essayez de généraliser votre solution de façon à ce qu'ajoutez une nouvelle fonction demande le moins d'efforts d'implémentation.

- Écrivez une fonction qui renvoie la multiplication des éléments d'une liste
- Écrivez un fonction qui renvoie la différence des éléments d'une liste.

4. Appliquer une fonction sur une liste Au TP2, vous avez écrit une fonction `{Applique L F}` qui applique la fonction a un argument F sur chaque élément de la liste L et retourne la liste des résultats. Sans modifier Applique, comment pourrait-on faire pour appliquer une fonction `{Pow N E}`, qui renvoie N exposé en E, sur une liste L.

5. Taux de conversion Définissez une fonction `{Convertir T V}`, qui renvoie T*V. Cette fonction est utile pour faire la conversion entre des livres et des kilogrammes ou des pieds et des mètres mais en pratique elle serait difficilement utilisable puisque même si on répète le même type de conversion (ex: pieds vers mètres) avec des valeurs différentes, il faut répéter le taux de conversion.

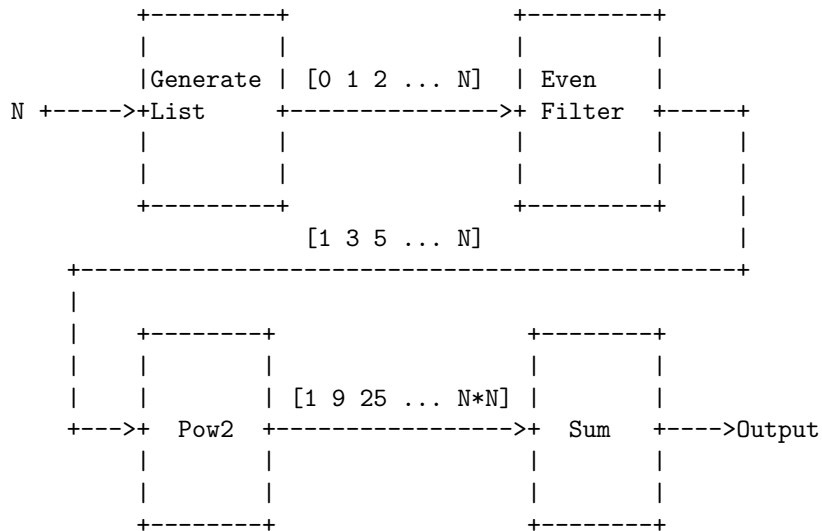
- Comment-pourrait on faire pour, tout en utilisant la fonction `Convertir`, éviter de devoir réécrire le même taux de conversion à chaque fois que l'on veut convertir des pieds en mètres par exemple?
- Il y a certain cas où multiplier par le taux de conversion n'est pas suffisant pour effectuer une conversion, par exemple pour convertir approximativement des degrés Fahrenheit en degrés Celsius, la formule est la suivante, $C = 0.56 * F - 17.78$. Modifiez la fonction `Convertir` pour qu'elle puisse effectuer les conversions de base et celles plus avancés.

6. Pipelining. Vous voulez calculer la somme des carrés des nombres impairs de 0 à n. Un moyen simple de le faire serait d'utiliser une boucle et un accumulateur :

```
fun {SumSquare N}
  fun{SumSquareAux N Acc}
    if N <= 0 then Acc
    elseif N mod 2 == 0 then {SumSquareAux N-1 Acc}
    else {SumSquareAux N-1 Acc + (N*N)} end
  end
in
  {SumSquareAux N 0}
end
```

Il est également possible d'utiliser la programmation fonctionnelle pour calculer la formule. L'idée est de séparer l'entièreté du calcul en sous-tâches. Chaque sous-tâche sera calculée par une fonction et renverra son resultat à la

fonction suivante. Le schéma représente le problème :



Le processus décrit par le schéma peut être implémenté de la manière suivante :

```

fun {PipeLine N}
  P1 P2 P3 in
    P1 = {GenerateList N}
    P2 = {MyFilter P1 fun {$ X} X mod 2 \= 0 end}
    P3 = {MyMap P2 fun {$ X} X * X end}

    {MyFoldL P3 fun {$ Acc X} X + Acc end 0}
  end
end
  
```

Implémentez les fonctions `GenerateList`, `MyFilter`, `MyMap` et `MyFoldL` (si ce n'est pas déjà le cas).

Certains langages de programmation utilisent beaucoup ce genre d'abstraction. Le morceau de code du dessus peut se réécrire en Scala comme :

```

def pipeline(n: Int): Int = {
  (0 to n)
    .filter(_ % 2 != 0)
    .map(a => a * a)
    .foldLeft(0)(_ + _)
}
  
```

7. **Environnement contextuel.** Quel est le résultat affiché par ce programme ? Utilisez la machine abstraite pour justifier votre réponse. *Attention, le programme suivant n'est pas tout à fait traduit en langage noyau. Quelles sont les instructions à traduire ?*

```

local Y LB in
  Y=10
  proc {LB X ?Z}
    if X>=Y then Z=X
    else Z=Y end
  end

  local Y=15 Z in
    {LB 5 Z}
    {Browse Z}
  end
end
end
  
```