

Programmation fonctionnelle

TP n° 9 : Flots

Dans ce TP, on manipule les flots paresseux définis en cours :

```
type 'a item = Item of 'a * 'a flot
and 'a flot = 'a item Lazy.t
let scons x f : 'a flot = lazy(Item(x, f))
let get_item = function (lazy(Item(hd, tl)) : 'a flot) -> (hd, tl)
```

Tous les flots que vous aurez à définir sont des flots infinis d'entiers.

Partie 1 : Préliminaires

1. Considérons les deux fonctions suivantes :

```
let rec ints_from_strict n = scons n (ints_from_strict (n + 1))
let rec ints_from n = lazy (Item (n, (ints_from (n + 1))))
```

- (a) Que se passe-t-il si on réalise l'appel `ints_from_strict 0` ?
 - (b) Que se passe-t-il si on réalise l'appel `ints_from 0` ?
 - (c) Pourquoi le comportement diffère alors qu'on a simplement remplacé l'appel à `scons` par sa définition ?
2. Définissez une fonction `sprint` telle que `sprint n s` affiche les `n` premiers entiers du flot `s`. Par exemple, `sprint 10 (ints_from 0)` affiche `[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...]`.

Partie 2 : Fonctions utilitaires

Les flots infinis peuvent être vus comme une généralisation des listes. La plupart des fonctions sur les listes peuvent donc être adaptées aux flots.

1. Définissez les fonctions `shhead` et `stail` qui renvoient la tête et la queue d'un flot.
2. Définissez la fonction `snth` telle que `snth s n` renvoie l'élément d'indice `n` du flot `s`.
3. Définissez la fonction `sfilter` telle que `sfilter p s` renvoie le flot contenant les éléments de `s` vérifiant le prédicat `p`.
4. Définissez la fonction `smap2` telle que `smap2 f s1 s2` renvoie le flot contenant les images élément par élément des flots `s1` et `s2` par la fonction `f`.
5. Déduisez-en les opérateurs `++` et `**` qui additionnent et multiplient deux flots élément par élément.

Partie 3 : Flots par récurrence

A l'aide d'une fonction récursive, définissez les flots suivants :

1. le flot des entiers impairs,
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, ...]
2. le flot des factorielles,
[1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880, ...]

3. le flot des entiers de Fibonacci.
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...]

Partie 4 : Flots par point fixe

A l'aide d'une équation du point fixe, définissez les flots suivants :

1. le flot dont tous les éléments valent 1,
[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...]
2. le flot des entiers naturels,
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, ...]
3. le flot des factorielles,
[1, 1, 2, 6, 24, 120, 720, 5040, 40320, 362880, ...]
4. le flot des entiers de Fibonacci.
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...]

Partie 5 : Nombres premiers

1. Définissez une fonction `eratosthene` telle que `eratosthene s` réalise le crible d'Ératosthène sur le flot `s`. La tête du flot est préservée, on retire tous ses multiples dans la queue du flot et on applique récursivement `eratosthene`.
2. Déduisez-en une définition du flot des nombres premiers.
[2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...]