M2 Master Ingenierie Informatique 2013/2014
Programmation logique et par contraintes avancee Feuille de TP n° 2

Listes de différence et programmation dataflow en Oz

Exercice 1 (Listes de différence)
Une *liste imbriquée* est une liste dont les elements sont soit un atome, soit la liste vide, soit une liste de listes imbriquees. Par exemple,

```
[[a b] [[c] [d]] nil [e [f]]]
```

est une liste imbriquee (remarquez que cette liste ne serait pas legale en OCaml a cause du typage). Ecrire en Oz une fonction `Flatten` qui prend en entree une liste imbriquee, et qui envoie comme resultat l'aplatissement de cette liste. Sur l'exemple, le resultat doit être

```
[a b c d e f]
```

Utiliser les listes de di erence pour realiser la fonction Flatten (inspirez vous de la version de la fonction Append utilisant les listes de di erence vue en cours).

La bibliotheque standard contient une fonction qui est utile pour cet exercice : `{IsList X}` donne true quand la valeur de *X* est une liste (vide ou pas), false si *X* est liee a une valeur qui n'est pas une liste, et suspend si la variable *X* n'est pas liee a une valeur.

Exercice 2 (Les threads on Oz)
Le *crible d'Ératosthène* est une methode pour calculer la sequence des nombres premiers. La methode peut être decrite comme suit :

> Il y a un generateur qui engendre le ot de nombres a partir de 2.
>
> La sortie du generateur passe par une cha ne de ltres, dont chacun supprime tous les elements qui sont divisible par un certain nombre.

Le crible fonctionne comme suit :

> Il laisse passer le premier element (appele *x*).
>
> Pour tous les elements suivants de l'entree il elimine ceux qui sont divisible par *x*. Le ot des nombres qui restent apres cette elimination est passe a une nouvelle instance du crible.

Dans cet exercice vous programmerez le crible d'Eratosthene en utilisant la synchronisation par l'o re.

1. Ecrire un predicat *f*`DivFilter X L R`*g* qui, lie *R* a la liste de tous les elements de la liste *L* qui ne sont pas divisible par *X*.

2. Ecrire un predicat *f*`Sieve L R`*g* qui, si *L* est une liste d'entiers, lie *R* a la liste de tous les elements qui ne sont pas divisibles par un autre entier qui para^t plus tôt dans la liste. Utiliser le predicat `DivFilter`.

3. Ecrire un programme qui calcule la liste des tous les nombres premiers suivant la methode du crible d'Eratosthene. Utilisez le predicat *f*`Generate X`*g* vu en cours qui lie la variable *X* a la liste des nombres successives a partir de 2. Il y aura dans votre programme une procedure *f*`Sieve X`*g* qui lie X a la liste in nie de tous les premiers.

Exercice 3

Reprennez votre programme pour le crible d'Eratosthene obtenu dans l'exercice precedent, et changer la synchronisation par o re en une synchronisation par demande. La procedure `Sieve` sera maintenant une procedure a deux arguments, *f*`Sieve N X`*g*, qui lie *X* a la liste des *N* premiers nombre premiers.