

TP n°5

Monades avec plus

Résolvez les problèmes suivants en utilisant la monade des listes avec somme vue en cours.

Exercice 1 Étant donnée une liste `l` représentant un ensemble S , construire les listes représentant tous les combinaisons de n éléments de S .

Exercice 2 Étant donnée une liste `l`, construire toutes les permutations de cette liste. (Suggestion : commencer par construire une fonction `ndinsert : 'a -> 'a list -> 'a list list` qui insère un élément dans une liste dans toutes les positions possibles).

Remarque : Les solutions qu'on vient de donner sont extrêmement simples et élégantes, mais passent mal à l'échelle : si vous avez vraiment besoin d'utiliser les combinaisons ou les permutations d'un ensemble de taille plus grande que 10, il vaut en général mieux les énumérer une à la fois, en utilisant un des algorithmes décrits dans

Donald E. Knuth. 2005. *The Art of Computer Programming, Volume 4, Fascicle 3 : Generating all Combinations and Partitions*. Addison-Wesley Professional.

Exercice 3 [Kangourou CE2-CM1-CM2 2012] Laura, Romain, Bernard et Patricia s'alignent pour une photo. Patricia et Laura veulent être à côté l'une de l'autre et Romain veut être à côté de Laura. En respectant tous ces vœux, de combien de manière peuvent-ils s'aligner tous les quatre pour la photo ?

Exercice 4 [Les meilleurs séances autour d'un horaire donné] Reprétons l'exercice 3 de la session précédente, toujours avec les horaires donnés pour les trois films qui nous intéressent.

```
let movies = ["Lincoln",149,[12,15;15,15;18,25];  
             "Django",164,[17,40;21,0];  
             "Mobius",103,[12,50;15,0;17,20;19,50;22,10]];
```

Maintenant, nous voulons trouver la combinaison de séances permettant d'avoir le moins de temps d'attente, *avec une sortie autour de 21h, plus ou moins 45m*, pour aller diner ensemble après.

Il serait tentant d'écrire une fonction `inInterval` qui garde seulement les séances permettant une sortie dans l'intervalle donné, et conclure en réutilisant le code de la semaine passée :

```
(mftimes |> combinations |> maxwait1 |> rank |> (inInterval 1215 1305) ) movies;;
```

Mais ce n'est pas très efficace : on calcule toutes les combinaisons d'abord, pour en jeter plein après.

Proposez une ou plusieurs solutions alternatives permettant de ne pas engendrer toutes les combinaisons.