

# TP6 : Fonctions d'ordre supérieur

Semaine du 16 Octobre

## Exercice 1. Ordre supérieur

1. Proposer une fonction `appliquer_liste` de signature  $('a \rightarrow 'a) \text{ list} \rightarrow 'a \rightarrow 'a$  de sorte à ce que `appliquer_liste [f1; f2; ... ; fn] x` soit égal à `fn ( ... (f2 (f1 x)))`.
2. Proposer une fonction `appliquer_liste_inv` de signature  $('a \rightarrow 'a) \text{ list} \rightarrow 'a \rightarrow 'a$  de sorte à ce que `appliquer_liste [f1; f2; ... ; fn] x` soit égal à `f1 ( f2 ( ... (fn x)))`.

## Exercice 2.

1. Proposer une fonction `curryfier` de signature  $('a * 'b \rightarrow 'c) \rightarrow a \rightarrow b \rightarrow c$  telle que `(curryfier f) x y` soit égal à `f (x, y)` pour tout `f`, `x` et `y`.
2. Proposer une fonction de `dec Curryfier` de signature  $('a \rightarrow 'b \rightarrow 'c) \rightarrow a * b \rightarrow c$  telle que `(dec Curryfier f) (x, y)` soit égal à `f x y` pour tout `f`, `x` et `y`.
3. Soit  $A, B$  et  $C$  trois ensembles. En utilisant la curryfication, montrer l'existence d'une bijection entre les fonctions de  $A \times B$  dans  $C$  noté  $C^{A \times B}$  et l'ensemble des fonctions de  $A$  dans les fonctions de  $B$  dans  $C$  noté  $(C^B)^A$ .

*Correction :*

1. `let curryfier f a b = f (a,b)`

2. `let dec Curryfier f (a, b) = f a b`

3. ...

## Exercice 3. Continuation et récursivité terminale

## Exercice 4. D'autres manières de faire de la récursion

1 `let rec combinateur f x = f (combinateur f) x`

1. Que fait la fonction `g` ?

2. Sur ce model, proposer une implémentation d'une fonction qui calcule la suite de syracuse.

```
1 let f n =  
2     let r = ref (fun x -> 0) in  
3     (r := fun x -> if x = 0 then 1 else x * (!r)(x-1)) ;  
4     !r n
```

3. Que fait la fonction g2 ?
4. Proposer une implémentation de combinateur sans `rec` et avec des références.

### Exercice 5. Quine en OCaml

```
1 (fun s -> Printf.printf "%s %S;;" s s) "(fun s -> Printf.printf \"%s %S;;\" s s)";;
```