

## TP de Typage n° 1

## Sémantique Opérationnelle

**Exercice 1** Évaluer les programmes suivants en call-by-value et call-by-name (aussi bien en big qu'en small step semantics) :

1. `let x = 3 in x + 1`
2. `let x = (if true then 1 else 2) in x + 1`
3. `let x = 4 in (let y = 3 in x + y)`
4. `let x = 4 in (let y = x + 1 in x * y)`
5. `let f = (λx.x + 1) in f (f 5)`
6. `let f = (λx.x + 1) in  
    (let g = (λy.y + 4) in (f 3) * (g 3))`
7. `let f = (λx.x + 10) in  
    (let g = (λy.(f y) + 4) in (f 3) * (g 3))`
8. `(fix(λadd.λx.  
    if fst x then snd x else add((fst x) - 1, snd x) + 1)) <1, 2>.`
9. `(fix(λfact.λx.if x then 1 else x * (fact (x - 1)))) 3`

**Exercice 2** Choisir une méthode de sémantique opérationnelle et l'implémenter.

**Exercice 3**

- Définir formellement une opération de  $\alpha$ -conversion.
- Implémenter une fonction qui prend en argument un terme  $M$  et qui renvoie un autre terme  $M'$   $\alpha$ -équivalent à  $M$ .
- Implémenter une fonction qui prend en argument deux termes  $M$  et  $M'$ , et qui renvoie `true` si et seulement  $M$  est  $\alpha$ -équivalent à  $M'$ .