TD et TP de Scheme

Pierre Boulet, Jean-Marc Nicod, Frédéric Vivien

9 – 21 février 1997

1 Évaluation d'expressions

Donner la valeur retournée par les expressions ci-dessous:

1. (+ 2 3) ⇒ ...
2. (+ (* 2 3 5) (- 6 8)) ⇒ ...
3. (and (> 12 45) (=3 (/ 12 4))) ⇒ ...
4. (and (> 12 45) (=3 (/ 12 0))) ⇒ ...
5. (or #t (< 12 3)) ⇒ ...
6. (not #f) ⇒ ...
7. (symbol? 'tigre) ⇒ ...
8. (symbol? "tigre") ⇒ ...
9. Supposez que les définitions suivantes aient été faites dans l'ordre donné:
   - (define puce 0.01)
   - (define baleine 99999)
   - (define bestiole 'baleine)
   - (define dauphin 500)
   - (define tigre 'félin)
   - (define animal baleine)

   Trouvez quels seront les résultats des évaluations suivantes:
   - baleine ⇒ ...
   - 'baleine ⇒ ...
   - (+ animal puce) ⇒ ...
   - dauphin ⇒ ...
   - puce ⇒ ...
   - bestiole ⇒ ...
   - tigre ⇒ ...
   - 'tigre ⇒ ...
   - "animal ⇒ ...
   - (+ bestiole baleine) ⇒ ...
10. ((lambda (x y z) (+ x y z)) 22 3 9) ⇒ ...

11. ((lambda (f x y) (+ (f x) (f y))) (lambda (x) (* x x)) 3 4) ⇒ ...
12. (car (cdr '(a b c d))) ⇒ ...
13. (cdr '(a (b c d))) ⇒ ...
14. (cdr '((a b c d))) ⇒ ...
15. (cdr (car (cdr '(a (b c) (d e))))) ⇒ ...

2 Premières fonctions

Écrire:

- une fonction qui rend le nombre entier suivant du nombre entier passé en argument;
- une fonction constante égale à 7;
- une fonction qui rend le signe d'un nombre (versions avec et sans ***cond***;
- une fonction qui renvoie le second élément d'une liste d'au moins deux éléments;
- une fonction qui rend vrai si et seulement si une liste n'a qu'un seul élément.

3 Fonctions récursives sur les listes

Écrire une fonction:

- qui calcule la longueur d'une liste;
- qui calcule la somme des éléments d'une liste de nombres;
- qui concatène deux listes;
- qui renverse une liste;
- qui rend le dernier élément d'une liste non vide;
- qui prend une liste et un élément en arguments et renvoie ***vrai*** si et seulement si l'élément appartient à la liste.
- qui rend la liste des symboles de la liste argument: (flatten (a (b c) d)) ⇒ (a b c d).