

# Programmation fonctionnelle

TD 11 – Piles

L1 – Université Nice Sophia Antipolis

30 avril 2018

## Exercice 1 Animation réversible

Reprennez le code de l’animation “Monte-Carlo réversible” vue en cours, et utilisez un monde avec deux piles pour la rendre “bi-réversible”, de sorte que l’on puisse rétablir un lancer annulé en pressant  $\rightarrow$ .

## Exercice 2 Parcours suffixe

Donnez le parcours suffixe de l’arbre `'(+ (* 1 2) (/ (- 3 4) 5))`

## Exercice 3 Profondeur

Écrivez une fonction (`profondeur A`) qui renvoie la profondeur de l’arbre A (la longueur d’un plus long chemin de la racine à une feuille).

## Exercice 4 Nombre d’opérateurs

À l’aide d’une récurrence enveloppée, programmez une fonction (`nb-op A`) prenant un arbre binaire A renvoyant le nombre d’opérateurs de A. En utilisant une pile, donnez une autre version itérative de cette fonction.

## Exercice 5 Évaluation postfixe avec variables

Étendez la fonction (`postfixe->valeur L`) vue en cours en une nouvelle fonction (`postfixe->valeur L AL`) prenant un parcours suffixe L d’arbre pouvant contenir des variables et une liste d’association AL permettant de leur associer une valeur. Par exemple, (`postfixe->valeur '(x 1 y + *) '((x 2) (y 3))`) renvoie  $2 \cdot (1 + 3) = 8$ .

## Exercice 6 Du postfixe à l’arbre

En vous inspirant de la fonction d’évaluation d’arbres arithmétiques vue en cours, écrivez une fonction (`postfixe->arbre L`) prenant un parcours postfixe et renvoyant l’arbre correspondant.

## Exercice 7 Profondeur postfixe

Écrivez une fonction (`postfixe->profondeur L`) qui renvoie la profondeur de l’arbre de parcours postfixe L, sans reconstruire cet arbre.