**MB University of M'sila** | **Master 2 IDO 2020/2021**
**Faculty de M & I** | **Optimization methods**
**Department of informatics** | **Recitation 2 / genetic programming**

## Exercise 1

Convert in prefix form:

1) f(x)= 3x+2
2) f(x)= $2x^3$-x+3
3) f(x)= $x^2$ +1 si $x \geq 0$ et f(x)=1/x sinon
4) IF (NOC = 2) AND (S > 800) THEN return true ELSE return false
5) $(x \wedge true) \rightarrow ((x \vee y) \vee (z \leftrightarrow (x \wedge y)))$
6) i =1; while (i < 20) { i = i +1 }
7) x=n !
8) n!=1 si n=0 et n ! =n*(n-1)! sinon
9) $2 \cdot \pi + \left( (x+3) - \frac{y}{5+1} \right)$

Write the corresponding algorithm

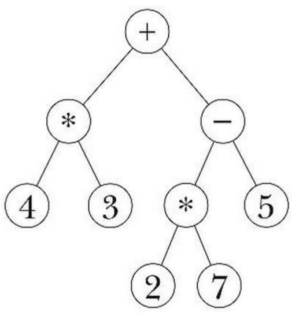
## Exercise 2

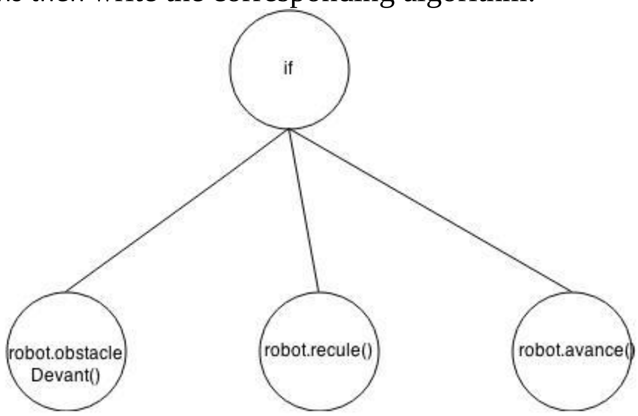
Write in java :

1) (+ 1 2 (IF (> TIME 10) 3 4))
2) (define (factorial x) (if (= x 0) 1 (* x (factorial (- x 1)))))
3) (* 2 (cos 0) (+ 4 6))
4) (defun fibonacci (N)
   "Compute the N'th Fibonacci number."
   (if (or (zerop N) (= N 1))
   1
   (+ (fibonacci (- N 1)) (fibonacci (- N 2)))))

## Exercise 3

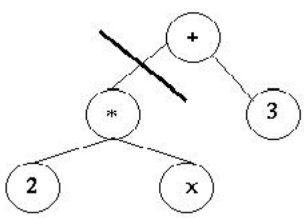
Find the corresponding s-expressions then write the corresponding algorithm.

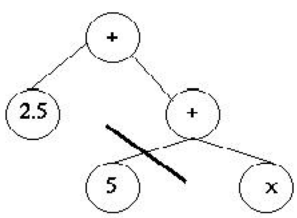



## Exercise 4

Give the 2 child for the following parents crossover then write the corresponding algorithm:



**Exercise 5**

Let terminal set = {a,b,intrandom(1-5)}
Let function set = {+,*,sqrt}
Give an initial population of 6 individuals using :

- Grow approach ;
- Full approach ;
- Ramped half and half ;

Write the corresponding algorithm in each case.

**Exercise 6 Even-parity-4 problem using GP.** In its general formulation, the Boolean even-parity k function of k Boolean arguments returns true if an even number of its Boolean arguments evaluates true, otherwise it returns false. For the special case where k = 4, 16 fitness cases must be checked to evaluate the fitness of an individual.
Formulate the objective function for this problem. Our goal is to design a genetic programming approach to solve the problem. Propose a tree-like representation for this problem. For this purpose, the operators and the terminal set of the tree have to be defined.

**Exercise 7 Distance measure for GP trees.** Genetic programming use parse trees to encode solutions. Defining a distance between trees is not a trivial task.
Propose such a distance to deal with tree-based representations. As shown in the previous chapter, the distance measure may be important in the landscape analysis of the problem and the design of diversification or intensification search mechanisms. Is the distant proposed coherent with the mutation operators?

**Exercise 8 Decision tree generation using GP.** Decision trees represent a classical model in data mining for classification tasks. The problem consists in the induction using GP of the "optimal" decision tree from a training set (or learning set) of data. Each record from the training set is composed of several attributes and one goal attribute that is the class attribute. The decision tree will be used to predict the class of new records with an unknown class (label).
A decision tree is a tree where the terminal nodes are defined by classes ci, i = 1, n, and the nonterminal nodes represent the different attributes aj, j = 1,m. Each edge of the tree is defined by a test on the attribute values aj = vj where vj represents the possible values of the attribute aj.
Design a GP algorithm for this problem.

**Exercise 9**

Retrieve the function f (x) = 0,75x2-5x + 3 (values chosen completely randomly)
knowing 25 points of the function distributed in the interval [0,10]

| | |
|---|---|
| **objective** | To determine a program that approximates the function f (x) = 0.75x2-5x + 3, we will not give any help to the algorithm to facilitate its search (ie the algorithm will not look for a polynomial and will not know the degree of the higher term) |
| **Terminal set** | x or float constant |
| **functions** | +, -, * |
| **Regression set** | 25 values between 0 et 10 |
| **Fitness** | Mean absolute difference or deviation |
| **Stopping criterion** | Fitness < = 1,25 |
| **Parameters** | 1000 individuals , 100 generations |

**Références bibliographiques :**


- Brameier, M. (2004), *On Linear Genetic Programming*

- Koza, J.R. (1990), Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems, Stanford University Computer Science Department technical report STAN-CS-90-1314. A thorough report, possibly used as a draft to his 1992 book.
- Koza, J.R. (1992), Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press
- Koza, J.R. (1994), Genetic Programming II: Automatic Discovery of Reusable Programs, MIT Press
- Koza, J.R., Bennett, F.H., Andre, D., and Keane, M.A. (1999), Genetic Programming III: Darwinian Invention and Problem Solving, Morgan Kaufmann
- Koza, J.R., Keane, M.A., Streeter, M.J., Mydlowec, W., Yu, J., Lanza, G. (2003), Genetic Programming IV: Routine Human-Competitive Machine Intelligence, Kluwer Academic Publishers
- Langdon, W. B., Poli, R. (2002), *Foundations of Genetic Programming*, Springer-Verlag
- Poli, R., Langdon, W. B., McPhee, N. F. (2008), *A Field Guide to Genetic Programming*, freely available via Lulu.com.
- Weise, T., Global Optimization Algorithms - Theory and Application