

GRAPHE TP1

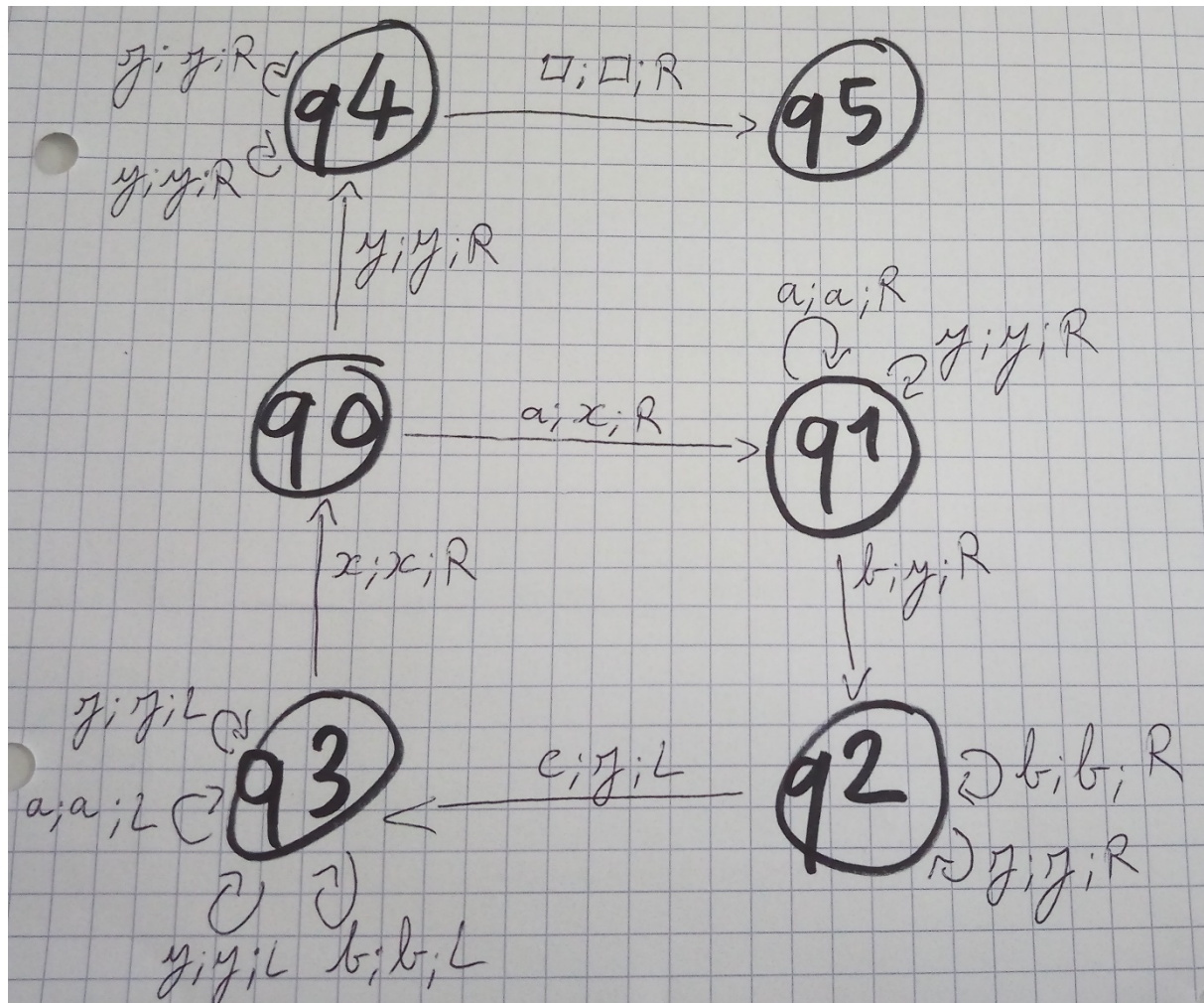
<https://www.irit.fr/~Cassia.Trojahn/graphes/tp1.pdf>

EXERCICE 1

QUEL LANGAGE L LA MACHINE M EST CAPABLE DE RECONNAÎTRE ?

Le langage L de la machine M reconnait (a, b, c).

DONNEZ LA REPRÉSENTATION MACHINE D'ÉTAT DE M.



EXERCICE 2

INDIQUEZ L'ORDRE DE COMPLEXITÉ DE CHAQUE INSTRUCTION ;

Somme () = $1 + 1 + 1 + (n * n * 1) + (n * 1) + 1 + 1$

Factorielle () = $1 + 1 + (n * (1 + 1)) + 1$

INDIQUEZ L'ORDRE DE COMPLEXITÉ GLOBAL DE L'ALGORITHME.

Somme () = $O(n^2)$

Factorielle () = $O(n)$

EXERCICE 3

DÉTERMINEZ LA COMPLEXITÉ DE L'ALGORITHME BUBBLESORT POUR LE MEILLEUR DES CAS ET POUR LE PIRE DES CAS. POUR CELA, INDIQUEZ L'ORDRE DE COMPLEXITÉ DE CHACUNE DES INSTRUCTIONS NÉCESSAIRES POUR TRIER UN TABLEAU DE TAILLE N ;

BubbleSort () = $\Omega(1), O(n^2)$

MergeSort () = $O(n \log(n))$

COMPAREZ LA COMPLEXITE DE BUBBLESORT AVEC CELLE DE L'ALGORITHME MERGESORT POUR LE PIRE DES CAS. QUEL ALGORITHME EST LE MOINS EFFICACE ?

BubbleSort est moins efficace.

EXERCICE 4

IMPLÉMENTEZ LA VERSION ITÉRATIVE DE LA SUITE DE FIBONACCI ;

```
fonction Fibonacci_iter(n : entier) : entier
variable compteur, val1, val2, val3, retour : entier
debut
  si n < 2 alors
    retour <- n
  sinon
    val1 <- 0
    val2 <- 1
    pour compteur de 2 à n par pas de 1 faire
      val3 <- val1 + val2
      val1 <- val2
      val2 <- val3
    fin pour
    retour <- val2
  fin si
  retourner retour
fin
```

IMPLÉMENTEZ LA VERSION RECURSIVE DE CETTE SUITE ;

```
Fonction Fibonacci(n: entier): entier;

Var
    résultat: entier;

Debut

    Si(n = 0) Alors
        résultat := 0;
    Sinon Si(n = 1) Alors
        résultat := 1;
    Sinon
        résultat := Fibonacci(n - 1) + Fibonacci(n - 2);
    FinSi

    Renvoyer résultat;

Fin
```

EVALUEZ LA COMPLEXITE DE CHAQUE VERSION. POUR CELA, EXECUTEZ-LES AVEC DIFFERENTES VALEURS DE N (N=10, N=100, N=10000).