



# Projet Algo 2

## Résumé

Le projet porte sur la recherche d'occurrences de motifs (séquence de lettres correspondant à une expression régulière) dans une chaîne de caractères formée à partir d'un alphabet de 4 lettres. Chaque occurrence trouvée rapporte des points(\*). Différents traitements et calculs seront réalisés sur les occurrences et les points.

Etape 1 :

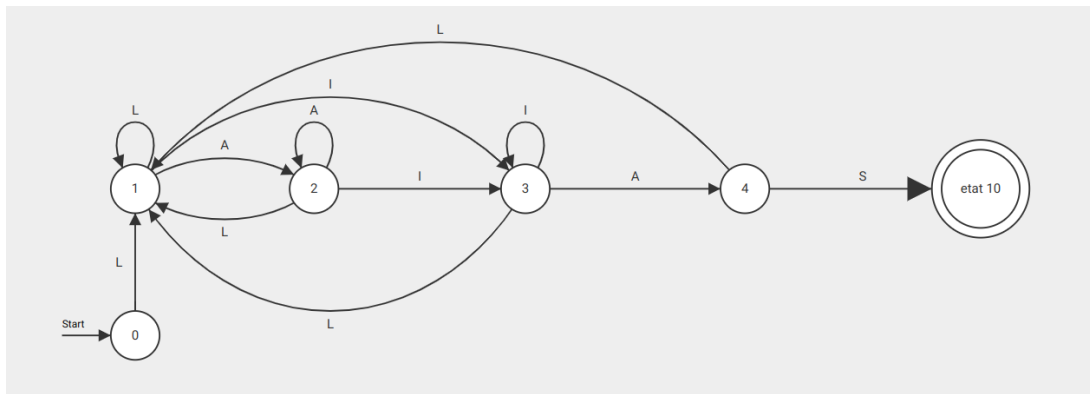


FIGURE 1 – Automate de mon motif 1 :  $L+A*I+AS$ .

```
// Tableau de transition pour le premier motif
int ttransit[NBETATS][NBTYPES] = {{1, 0, 0, 0, 0},
                                   {1, 2, 3, 0, 0},
                                   {1, 2, 3, 0, 0},
                                   {1, 4, 3, 0, 0},
                                   {1, 0, 0, 10, 0}};
```

FIGURE 2 – table de transition pour le motif 1.

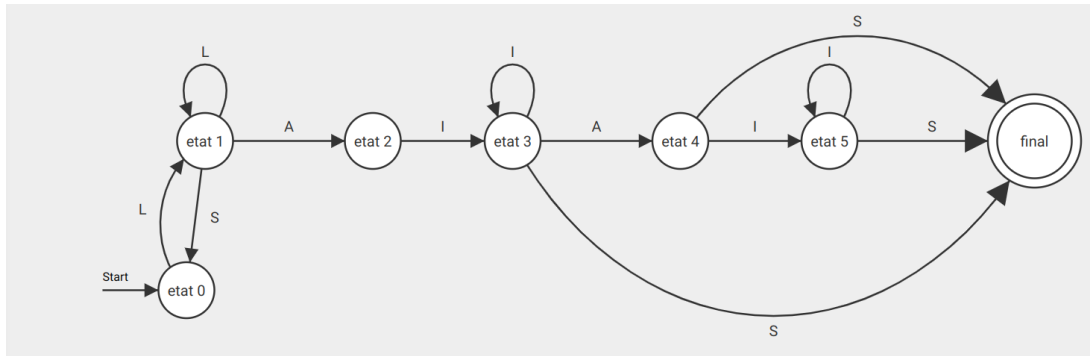


FIGURE 3 – Automate de mon motif 2 :  $L+AI+A?I^*S$ .

```
int ttransit_2[NBETATS_2][NBTYPES] = {
    /* etat 0 */    { 1, 0, 0, 0, 0 } ,
    /* etat 1*/    { 1, 2, 0, 0, 0},
    /* etats 2*/    { 1, 0, 3, 0,0},
    /* etat 3*/    {1, 4, 3, 6 ,0},
    /* etat 4*/    {1, 0, 5, 6 ,0},
    /*etat 5 */    {1, 0, 5, 6 ,0}};
```

FIGURE 4 – table de transition pour le motif 2.

```

Alphabet : LAIS
Motif 1 : L+A*I+AS
Motif 2 : L+AI+A?I*S
Tableau : de 82 caractères
LIASLLLLAIASLLLIASLIASLLLLLLAIASLAIIIIISLIIIIIASLIASLAIIIIIISLAIASLAIAS
Affichage table des occurrences du motif 1 :
Occ [pos pts]
-----
LIAS [ 0 40]
LLLIAS [ 13 70]
LIAS [ 19 40]
LIAS [ 23 40]
LIIIIIAS [ 46 52]
LIAS [ 54 40]
6 occ du motif trouvees
Affichage table des occurrences du motif 2:
Occ [pos pts]
-----
LLLLAIAS [ 4 89]
LLLLLLAIAS [ 27 134]
LAIIIIS [ 39 49]
LAIIIIIIS [ 58 61]
LAIAS [ 69 44]
LAIAS [ 75 44]
6 occ du motif trouvees
Ensemble des occurrences differentes (sans doublons)
Occ          Nb d'occurrences
-----
LAIIIIIIS    1
LLLLAIAS     1
LAIIIIS      1
LAIAS        2
LIIIIIAS     1
LLLLLLAIAS   1
LIAS         4

```

FIGURE 5 – début de ma trace d'exécution.

```

Ensemble des occurrences differentes (sans doublons)
Occ          Nb d'occurrences
-----
LAIIIIIIIIS    1
LLLLAIAIS      1
LAIIIIS        1
LAI AIS        2
LIIIIIIAS      1
LLLLLLAIAIS    1
LIAS           4
LLLIAS         1
Ensemble des occurrences differentes (sans doublons)
Occ          Nb d'occurrences
-----
LAIIIIIIIIS    1
LLLLAIAIS      1
LAIIIIS        1
LIIIIIIAS      1
LLLLLLAIAIS    1
LLLIAS         1
LAI AIS        2
LIAS           4
Nombre moyen de points par occurrence : 1.50
Tableau des points uniques :
40 70 52 89 134 49 61 44
8
Moyenne des valeurs du tableau : 33.69

```

FIGURE 6 – suite de ma trace d'exécution.

Etate 2 et 3 :

La fonction automate : Ceci est la fonction la plus importante du code, elle parcourt dans le tableau passé en paramètre et fait passer chaque lettre dans la table de transition, si le caractère retourne un état 1 et que l'état précédent n'était pas 1, on peut dire que c'est le début du motif car dans mes 2 motifs, le début est un L+, ce qui veut dire une infinité de L, mais on doit savoir à partir de quand est le début du motif.

Et si la lettre avant le L est différente de lui, donc la variable etat\_pre qui s'occupe de stocker l'état précédent à chaque fois est importante. Ensuite quand on connaît la première lettre du motif, on a gardé une trace de l'indice actuelle, cet indice sera stocké dans une variable premier\_indice qui ensuite va être stocker dans un tableau pour le début de chaque indice.

Quand on aura trouvé l'état sortie du motif, on reprend la variable i qui représente l'indice actuelle et on la stocke dans notre tableau fin. A la fin la fonction automate stocke le debut et la fin de chaque motif dans 2 tableau, et le nombre total de motif trouvé dans nb\_motif

La fonction imprimer motif ne fait que reprendre la même logique mais au lieu d'imprimer elle utilise la fonction snprintf pour extraire le motif du tableau en s'aidant comme arguments pour savoir depuis quelle indice à quelle autre indice extraire des tableaux fin et début précédemment stocké grâce à la fonction automate.

Etape 4 :

Dans un premier temps j'ai rempli un tableau avec une fonction qui la rempli de manière aléatoire mais pour vérifier correctement j'ai écrit un tableau en dur ou j'ai rempli le tableau de plus petite taille avec des motifs corrects pour vérifier si mes fonctions sont correctes.

Etape 5 :

Pour pouvoir ajouter mes motifs dans une hashtable j'ai du légèrement changer une fonction "add\_nom" créé par notre professeur en cours, En ajoutant une fonction si le mot est déjà présent et vérifier avec la fonction search\_nom, si le motif existe déjà on incrémente une variable "occurences" ajouté dans la struct linked dans le free-list.c. A noter que cette fonction a une partie ou les nombres de points unqiues seront ajouté pour l'exercice 8.

Etape 6 :

fonction print\_occurences \_\_with \_\_nb \_\_croissant(); Cette fonction parcourt une table de hachage contenant des listes chaînées d'éléments, où chaque élément représente une occurrence d'un certain motif. Elle collecte toutes ces occurrences dans un tableau occurrences\_tab, puis trie ce tableau en fonction du nombre d'occurrences de chaque élément. Ensuite elle imprime le nom de chaque occurrence et son nombre d'occurrences associé dans l'ordre croissant des occurrences. Elle utilise un tri à bulles simples pour cela.

Etape 7 :

La fonction fonction calculer\_nombre \_\_moyen \_\_points \_\_par \_\_occurence(); est similaire aux étapes précédentes, elle parcourt la hashtable et incrémente 2 variables, total\_points qui stocke tous les points quand on parcourt les occurrences ( int occurrences dans la struct linked a été ajouté pour stoker le nombre d'occurrences ) et total\_points qui est une variable qui stocke la somme des points, je retourne donc la division des points par le nombre d'occurrences.

Etape 8 :

J'ai créé un tableau tableau\_point\_unique pour stocker la nombre de point unique depuis ma fonction afficher\_motifs.

Etape 9 :

Le nombre moyen de points à partir de l'ensemble constitué à l'étape 8 est 33, après avoir relancé mon code plusieurs fois avec plusieurs tableaux de tailles différentes, il est toujours entre 29 et 33 points.

Question 10 :

Le nombre d'éléments de l'ensemble calculé à l'étape 5 est toujours légèrement supérieur à celui calculé à l'étape 9.

Question 11 : La moyenne de points trouvé à l'étape 9 (33 points) est toujours supérieur à au nombre de points moyens de toutes les occurrences trouvé dans l'étape 7, cela est du au fait que on prend que les points uniques. Dans l'étape 7 on prend la moyenne des points de toutes les occurrences et donc beaucoup de doublons de petits motifs avec peu de points font baisser la moyenne.

Question 12 :

Le nombre de point minimal pour mes 2 motifs est le même.

Pour le motif suivant  $L+A*I+AS$ , LIAS me rapporte 40 points.

Pour le motif 2 :  $L+AI+A ?I*S$  c'est LIAS qui me rapporte le point de points.

c'est le minimum de caractères que je peux avoir dans mon motif dans mes 2 motifs.

Car  $L = 15$  points,  $A = 1$  point,  $I = 3$  points,  $S = 21$  points, d'après mes 2 motifs je dois au moins avoir chaque lettre une fois dans c'est le minimum possible. Le maximum de point est difficile à prédire mais ça dépend en fonction de mon nombres de consonnes.

Question 13 :

Le motif pour pour alphabet qui me rapporterait le plus de point sera un motif qui renvoie 1 ou une infinité de consommées.

Exemple :  $L+A*I*S+$