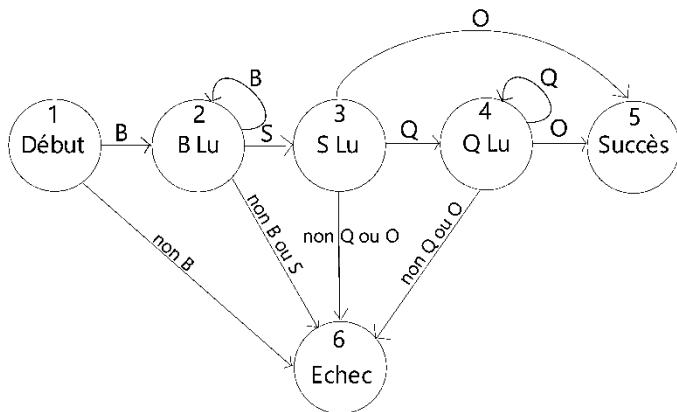


Algorithmique et structures de données 2

Automates

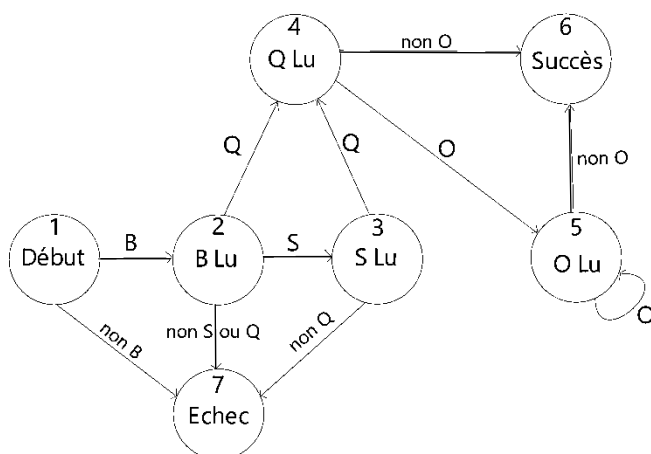
Schémas et table de transition :

Motif 1 : B+SQ*O



	Etat 1	Etat 2	Etat 3	Etat 4
B	2	2	6	6
S	6	3	6	6
Q	6	6	4	4
O	6	6	5	5

Motif 1 : BS ? QO*



	Etat 1	Etat 2	Etat 3	Etat 4	Etat 5
B	2	7	7	6	6
S	7	3	7	6	6
Q	7	4	4	6	6
O	7	7	7	5	5

Sophie Bousquet
L2-Y

Trace d'exécution :

Etape 2 :

L'étape 2 consiste à créer un automate en utilisant l'alphabet BSQO ainsi que le modèle $B+SQ^*O$.

```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
t de donnée 2/Automate_BousquetSophie/Etape 2$ gcc automate1.c
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 2$ ./a.out
Motif 1 : B+SQ*O

Affichage table des occurrences
BSQO [0 3]
BBBSQO [13 19]
BSO [21 23]
BBSQO [39 44]
BSQO [52 56]
BSQO [61 65]
BSQO [71 74]
BBSQO [81 85]
BBSO [88 91]
BBSO [99 102]
BSQO [110 113]
BBBSO [118 122]
BSO [129 131]
BBBSQO [137 143]
BSQO [144 147]
BBBSQO [150 157]
BBBSO [159 164]
BSQO [173 176]
BBSQO [183 187]
19 motifs 1 trouvés
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 2$
```

Etape 3 :

L'étape 3 consiste à créer un automate en utilisant l'alphabet BSQO ainsi que le modèle $BS ? QQ^*$.

```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 3$ ./a.out
Motif 2 : BS?QQ*

Affichage table des occurrences
BSQO [0 3]
BQO [7 10]
BSQ [15 17]
BQ [26 27]
BQO [30 33]
BQO [48 50]
BSQ [52 54]
BSQ [61 63]
BSQO [71 75]
BSQ [92 94]
BSQO [110 114]
BQO [123 125]
BQO [133 135]
BSQO [144 147]
BSQO [173 177]
BQO [194 196]
16 motifs 2 trouvés
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 3$
```

Sophie Bousquet
L2-Y

Etape 4 :

Dans l'étape 4 on doit créer l'union entre les 2 automates créés précédemment. On doit aussi afficher le nombre de doublons de chaque occurrence.

Ici, l'union entre nos 2 automates ne contient qu'une seule possibilité, BSQO.

```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
t de donnée 2/Automate_BousquetSophie/Etape 4$ ./a.out
Motif 1 : B+SQ*O

Affichage table des occurrences
BSQO [0 3]
BBBSQO [13 19]
BSO [21 23]
BBSQO [39 44]
BSQO [52 56]
BSQO [61 65]
BSQO [71 74]
BBSQO [81 85]
BBSO [88 91]
BBSO [99 102]
BSQO [110 113]
BBBSO [118 122]
BSO [129 131]
BBBSQO [137 143]
BSQO [144 147]
BBBSQO [150 157]
BBBSO [159 164]
BSQO [173 176]
BBSQO [183 187]
19 motifs 1 trouvés

Motif 2 : BS?QO*

Affichage table des occurrences
BSQO [0 3]
BQO [7 10]
BSQ [15 17]
BQ [26 27]
BQO [30 33]
BQO [48 50]
BSQ [52 54]
BSQ [61 63]
BSQO [71 75]
BSQ [92 94]
BSQO [110 114]
BQO [123 125]
BQO [133 135]
BSQO [144 147]
BSQO [173 177]
BQO [194 196]
16 motifs 2 trouvés

Union des motifs trouvés sans doublons
BSQO 10

--> 1 motifs uniques
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 4$
```

Sophie Bousquet
L2-Y

Etape 5 :

L'étape 5 consiste à utiliser l'union créée précédemment et de la ranger par ordre décroissant par rapport au nombre de doublons de chaque occurrence.

Ici, étant donné que l'on n'a qu'une seule possibilité l'ordre des occurrences ne changera pas.

```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
t de donnée 2/Automate_BousquetSophie/Etape 5$ ./a.out
Motif 1 : B+SQ*O

Affichage table des occurrences
BSQO [0 3]
BBBSQO [13 19]
BSO [21 23]
BBBSQO [39 44]
BSQO [52 56]
BSQO [61 65]
BSQO [71 74]
BBBSQO [81 85]
BBBSO [88 91]
BBBSO [99 102]
BSQO [110 113]
BBBSO [118 122]
BSO [129 131]
BBBSQO [137 143]
BSQO [144 147]
BBBSQO [150 157]
BBBSO [159 164]
BSQO [173 176]
BBBSQO [183 187]
19 motifs 1 trouvés

Motif 2 : BS?QO*

Affichage table des occurrences
BSQO [0 3]
BQO [7 10]
BSQ [15 17]
BQ [26 27]
BQO [30 33]
BQO [48 50]
BSQ [52 54]
BSQ [61 63]
BSQO [71 75]
BSQ [92 94]
BSQO [110 114]
BQO [123 125]
BQO [133 135]
BSQO [144 147]
BSQO [173 177]
BQO [194 196]
16 motifs 2 trouvés

Union des motifs trouvés sans doublons
BSQO 10

--> 1 motifs uniques

Motifs triés
BSQO 10
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 5$
```

Sophie Bousquet
L2-Y

Etape 9 :

Dans l'étape 9 on doit implémenter dans notre programme les 2 fonctions données, affiche_motifs et affiche_un_motif.

```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
Motif 1 : B+SQ*O
Affichage table des occurrences
BSQO [0 3]
BBBSQO [13 19]
BSO [21 23]
BBSQO [39 44]
BSQO [52 56]
BSQO [61 65]
BSQO [71 74]
BBSQO [81 85]
BBSO [88 91]
BBSO [99 102]
BSQO [110 113]
BBBSO [118 122]
BSO [129 131]
BBBSQO [137 143]
BSQO [144 147]
BBBSQO [150 157]
BBBSO [159 164]
BSQO [173 176]
BBSQO [183 187]
19 motifs 1 trouvés

Motif 2 : BS?QO*
Affichage table des occurrences
BSQO [0 3]
BQO [7 10]
BSQ [15 17]
BQ [26 27]
BQO [30 33]
BQO [48 50]
BSQ [52 54]
BSQ [61 63]
BSQO [71 75]
BSQ [92 94]
BSQO [110 114]
BQO [123 125]
BQO [133 135]
BSQO [144 147]
BSQO [173 177]
BQO [194 196]
16 motifs 2 trouvés
```

```
affiche_motifs

Motif 1 : B+SQ*O
BSQO  BBBSQO BSO  BBSQO BSQO  BSQO  BSQO  BBSQO BBSO  BBSO  B
SQO  BBBSO  BSO  BBBSQO BSQO  BBBSQO  BBBSO BSQO  BBSQO
BSQO

Motif 2 : BS?QO*
BSQO  BQO  BSQ  BQ  BQO  BQO  BSQ  BSQ  BSQO  BSQ  B
SQO  BQO  BQO  BSQO  BSQO  BQO
BSQO
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 9$
```

Sophie Bousquet
L2-Y

Trace d'exécution des programmes de vérification :

Etape 6 :

L'étape 6 correspond à vérifier si nos 2 automates, de l'étape 2 et 3, nous affichent les bons résultats.

Ici, la vérification des 2 automates a été faite dans 2 fichiers distincts, mais on a utilisé les mêmes noms de variables.

Pour la vérification, on a utilisé un tableau de 66 lettres afin d'obtenir un nombre d'occurrences pas trop élevé pour la vérification.

Ensuite, on a créé nb_occ_obtenir qui contient le nombre d'occurrences à obtenir, tab_occ_obtenir qui contient toutes les occurrences que l'automate doit nous afficher, ainsi que tab_indice_obtenir qui contient les indices de début et de fin des occurrences qu'on doit obtenir par rapport au tableau de 66 lettres.

Enfin, on a assert les variables des automates avec celle que l'on a créée afin de savoir si on obtient les bons résultats, si ce n'est pas le cas le programme s'arrête, mais ici tout fonctionne.

Vérification du premier automate :

```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 6$ gcc verif_automate1.c
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 6$ ./a.out
Motif 1 : B+SQ*0

Tableau des occurrences obtenu :

BSQ0 [0 3]
BBBSQ00 [13 19]
BSQ [21 23]
BBBSQ00 [39 44]
BSQ00 [52 56]
BSQ00 [61 65]

Tableau des occurrences à obtenir :

BSQ0 [0 3]
BBBSQ00 [13 19]
BSQ [21 23]
BBBSQ00 [39 44]
BSQ00 [52 56]
BSQ00 [61 65]

Nombre d'occurrence obtenu = 6
Nombre d'occurrence à obtenir = 6

sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 6$
```

Vérification du deuxième automate :

```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
t de donnée 2/Automate_BousquetSophie/Etape 6$ ./a.out
Motif 2 : BS?Q0*

Tableau des occurrences obtenu :

BSQ0 [0 3]
BQ00 [7 10]
BSQ [15 17]
BQ [26 27]
BQ00 [30 33]
BQ0 [48 50]
BSQ [52 54]
BSQ [61 63]

Tableau des occurrences à obtenir :

BSQ0 [0 3]
BQ00 [7 10]
BSQ [15 17]
BQ [26 27]
BQ00 [30 33]
BQ0 [48 50]
BSQ [52 54]
BSQ [61 63]

Nombre d'occurrence obtenu = 8
Nombre d'occurrence à obtenir = 8

sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 6$
```

Sophie Bousquet
L2-Y

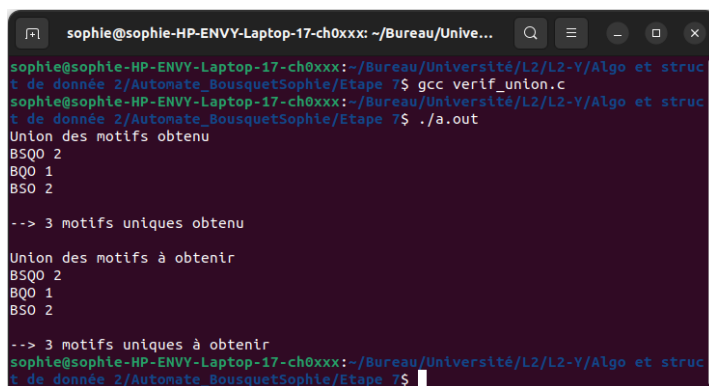
Etape 7 :

L'étape 7 correspond à vérifier si les occurrences créées dans l'étape 4 sont correctes.

Pour la vérification, on a initialisé 2 variables, automate1 et automate2, avec des occurrences prédéfinies étant donné que l'union de nos 2 automates ne contient qu'une seule possibilité.

Ensuite on a créé 3 variables, union_ob qui contient les valeurs de l'union qu'on doit obtenir, occurrence_ob qui contient le nombre de doublons de chaque occurrence que l'on doit obtenir, ainsi que nbmotif_ob qui contient le nombre de motifs dans l'union des 2 automates, il va permettre de créer une boucle for afin d'afficher les valeurs de union_ob et nbmotif_ob et de savoir si on a le bon nombre de motif dans l'union.

Enfin, on a assert les variables de l'union avec celle que l'on a créée afin de savoir si on obtient les bons résultats, si ce n'est pas le cas le programme s'arrête, mais ici tout fonctionne.



```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 7$ gcc verif_union.c
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 7$ ./a.out
Union des motifs obtenu
BSQO 2
BQO 1
BSO 2

--> 3 motifs uniques obtenu

Union des motifs à obtenir
BSQO 2
BQO 1
BSO 2

--> 3 motifs uniques à obtenir
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 7$
```

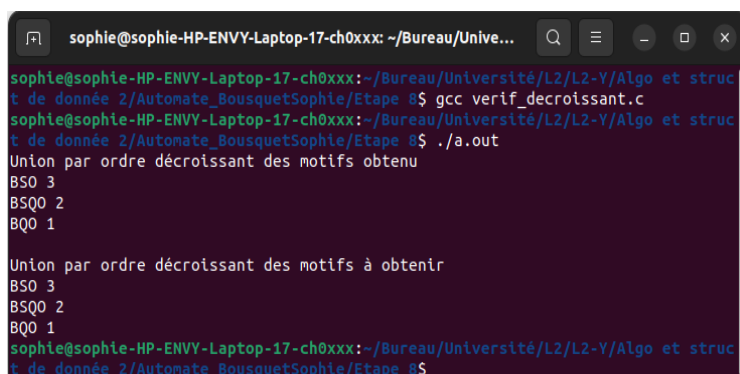
Etape 8 :

L'étape 8 correspond à vérifier si les occurrences créées dans l'ordre décroissant de l'étape 5 sont correctes.

Pour la vérification, on a gardé l'initialisation des 2 automates de l'étape précédente.

Ensuite on a créé les variables union_decroissant qui contient les occurrences de l'union entre les 2 automates mais rangé par ordre décroissant en fonction du nombre de doublons de chaque occurrence, occurrence_decroissant qui contient le nombre de doublons de chaque occurrence par ordre décroissant, ainsi que nbmotif_ob qui contient le nombre de motifs dans l'union des 2 automates afin de savoir si on a le bon nombre de motif et qui va être utilisé de la même manière que l'étape précédente.

Enfin, on a assert les variables de l'union ranger par ordre décroissant avec celle que l'on a créée afin de savoir si on obtient les bons résultats, si ce n'est pas le cas le programme s'arrête, mais ici tout fonctionne.



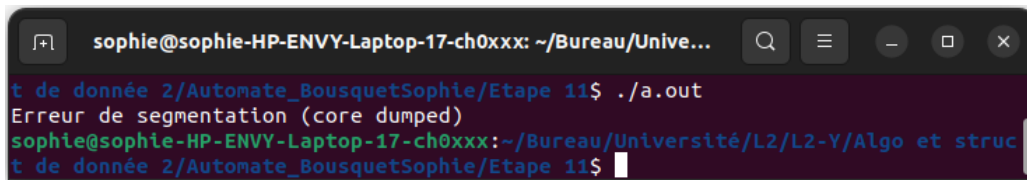
```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 8$ gcc verif_decroissant.c
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 8$ ./a.out
Union par ordre décroissant des motifs obtenu
BSO 3
BSQO 2
BQO 1

Union par ordre décroissant des motifs à obtenir
BSO 3
BSQO 2
BQO 1
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 8$
```

Sophie Bousquet
L2-Y

Question de l'étape 10 :

Le programme tourne avec un tableau ayant une taille de 2 000 000 maximum, avec une taille 20 000 000 il ne tourne plus et affiche :



```
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx: ~/Bureau/Unive...
t de donnée 2/Automate_BousquetSophie/Etape 11$ ./a.out
Erreur de segmentation (core dumped)
sophie@sophie-HP-ENVY-Laptop-17-ch0xxx:~/Bureau/Université/L2/L2-Y/Algo et struc
t de donnée 2/Automate_BousquetSophie/Etape 11$
```