

012345678901234567890123

Université d'Aix-Marseille III - Licence Math-Info, 2e année

I5 : Langages formels, automates et grammaires - TP n° 3

1 Automates non déterministes

Ecrivez la fonction `int accepte(chaine mot, int i, transitions delta, int final[Q], int etat)` *qui détermine si le facteur droit de mot à partir de la position* `i` *est accepté par l'AEFND dont l'état initial est* `etat`, *la fonction de transition est* `delta` *et l'ensemble des états terminaux est mémorisé par le tableau de booléens* `final`. On pourra définir les types `chaine` et `transitions` ainsi :

```
typedef char chaine[N];
typedef int transitions[Q][128][Q];
```

`N` désigne le nombre maximum de caractères d'une chaîne et `Q` le nombre maximum d'états dans un automate. Si une variable `delta` est de type `transitions` alors `delta[i][c][j] == k` signifie qu'en lisant le caractère dont le code ascii (compris entre 0 et 127) est `c` à partir de l'état `qi` on arrive à l'état `qk` et que c'est la $(j - 1)^e$ façon d'atteindre un état à partir de `qi` en lisant le caractère `c`. Si `delta[i][c][j]` est égal à -1, cela signifie qu'il y a `j` transitions possibles à partir de `qi` en lisant le caractère `c`.

Voici comment on peut alors utiliser la fonction avec un automate reconnaissant le langage des mots sur $\{a, b\}^*$ dont l'avant-dernier symbole est `a` :

```
for(i = 0; i < Q; i++)
    for(j = 0; j < 128; j++)
        delta[i][j] = -1;
delta[0]['a'][0] = 0; delta[0]['a'][1] = 1; delta[0]['a'][2] = -1;
delta[0]['b'][0] = 0; delta[0]['b'][1] = -1;
delta[1]['a'][0] = 2; delta[1]['a'][1] = -1;
delta[1]['b'][0] = 2; delta[1]['b'][1] = -1;
final[0] = 0; final[1] = 0; final[2] = 1;
scanf("%s", mot);
if(accepte(mot, 0, delta, final, 0) == 1)
    printf("oui\n");
else
    printf("non\n");
```

2 Automates avec ϵ -transitions

Ecrivez la fonction `int accepte(chaine mot, int i, transitions delta, int final[Q], int etat)` pour traiter les ϵ -AEFND. La structure de données permettant de représenter la fonction de transition est la même que précédemment mais une ϵ -transition est représenté par le caractère `&` (qui ne fait pas partie de l'alphabet).

Voici comment on peut alors mémoriser la fonction de transition de l'automate reconnaissant le langage des chiffres romains de I à VIII :

```
delta[0]['&'][0] = 1; delta[0]['&'][1] = 5; delta[0]['&'][2] = -1;
delta[0]['I'][0] = 5; delta[0]['I'][1] = -1; delta[0]['V'][0] = 1; delta[0]['V'][1] = -1;
delta[1]['&'][0] = 2; delta[1]['&'][1] = -1; delta[1]['I'][0] = 2; delta[1]['I'][1] = -1;
delta[2]['&'][0] = 3; delta[2]['&'][1] = -1; delta[2]['I'][0] = 3; delta[2]['I'][1] = -1;
delta[3]['&'][0] = 4; delta[3]['&'][1] = -1; delta[3]['I'][0] = 4; delta[3]['I'][1] = -1;
delta[5]['V'][0] = 4; delta[5]['V'][1] = -1;
final[0] = 0; final[1] = 0; final[2] = 0; final[3] = 0; final[4] = 1; final[5] = 0;
```

3 Recherche de correspondance entre un motif et une ligne de texte

Sous Unix, la commande `grep` permet de sélectionner les lignes d'un fichier correspondant à un motif donné, c'est-à-dire, un nom avec des caractères spéciaux qui représentent des groupes de caractères :

- le caractère spécial `.` représente n'importe quel caractère. Ex : `a.b` représente une chaîne de trois caractères dont celui du milieu est quelconque.
- le caractère spécial `*` indique que le caractère précédent est répété zéro, une ou plusieurs fois. Ex : `a*b` représente un nom commençant par une suite de `a` et finissant par `b`.
- le caractère spécial `+` indique que le caractère précédent est répété une ou plusieurs fois. Ex : `a+b` représente un nom commençant par une suite non vide de `a` et finissant par `b`.
- une suite de caractères entre crochets représente un seul caractère parmi cette suite. Ex : `a[bcd]e` représente `abe`, `ace` ou `ade`.

Un motif peut contenir autant de caractères spéciaux différents qu'on veut et à n'importe quel endroit.

Pour chaque caractère spécial, déterminez la forme qu'a un automate modélisant les mots correspondant au motif contenant ce caractère spécial. Déduisez-en la façon dont on peut construire n'importe quel automate correspondant à n'importe quel motif.

Ecrivez un programme qui saisit un motif puis une ligne de caractères et indique si la ligne contient le motif. Pour cela, on réutilisera la fonction `accepte` après avoir rempli le tableau `delta` à partir du motif.