

I5 : Langages formels, automates et grammaires - TP n° 2

1 Automates déterministes

Écrivez 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'AEFD 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];
```

`N` désigne le nombre maximum de caractères d'une chaîne et `Q` le nombre maximum d'état dans un automate. Si une variable `delta` est de type `transitions` alors `delta[i][c] == j` signifie qu'en lisant le caractère dont le code ascii (compris entre 0 et 127) est `c` à partir de l'état q_i on arrive à l'état q_j , sauf si `j` est égal à `-1`, qui signifie qu'il n'y a pas de transition avec ce caractère à partir de cet état.

Voici comment on peut alors utiliser la fonction avec l'exemple du coffre (vu en cours) :

```
for(i = 0; i < Q; i++)
    for(j = 0; j < 128; j++)
        delta[i][j] = -1;
delta[0]['o'] = 1;
delta[1]['f'] = 0; delta[1]['r'] = 2;
delta[2]['f'] = 3; delta[2]['p'] = 1;
delta[3]['o'] = 2;
final[0] = 0; final[1] = 0; final[2] = 0; final[3] = 1;

scanf("%s", mot);
if(accepte(mot, 0, delta, final, 0) == 1)
    printf("oui\n");
else
    printf("non\n");
```

2 Recherche (efficace) d'un facteur dans un mot

Une façon de déterminer si un facteur appartient à un mot est de déterminer l'automate qui reconnaît tous les mots qui contiennent ce facteur, puis de vérifier que le mot est accepté par l'automate.

Définissez un automate déterministe permettant de reconnaître efficacement le langage des mots contenant le facteur `baaa`. En utilisant la fonction `accepte` écrite précédemment, écrivez un programme qui détermine si un texte entré au clavier contient la suite de caractères `aaab`. Si le texte entré a une longueur de `n` caractères, combien y aura-t-il de comparaisons de caractères dans le pire des cas ?

Répondez aux mêmes questions que ci-dessus dans le cas où le facteur est `ababac`.

Si le facteur à rechercher était une suite de `m` symboles `a` suivi d'un symbole `b` et que le texte entré avait une longueur de `n` caractères, combien y aurait-il de comparaisons de caractères dans le pire des cas ? Comparez avec les résultats obtenus dans le TP 1.