

Projet de TP Grammaire et Langages

Réalisé par

KALAI Mohamed Hedi

SATOURI Khaireddine

L3 Informatique

Algorithme de transformation AFN sans epsilon transition vers AFD

Tache réalisée :

- Création d'un programme en C, qui prend en entrée dans un fichier un automate fini non déterministe (et sans epsilon transition) de taille raisonnable (i.e. : on peut limiter à l'avance le nombre d'états si on le souhaite), et qui construit un automate AFD qui reconnaît le même langage. On considère un alphabet avec seulement 2 caractères.
- Une présentation et une description du principe de l'algo et du choix des structures utilisées

Ouverture de fichier (en mode lecture) de l'automate non déterministe :

```
FILE * fich;  
fich = fopen("./fichentree", "r"); //automate non deterministe
```

Ouverture de fichier (en mode écriture) de l'automate déterministe :

```
FILE * fich_sortie;  
fich_sortie = fopen("./fichsortie", "w+"); //sortie : automate deterministe
```

Initialisation d'état actuel à 0 puis on importe l'automate non déterministe :

```
char Automate_D[50][4][4]; //pour enregistrer les valeurs de l'automate deterministe  
int pnt = 1, pntf = 0; //pnt : pointe vers l'etat qui suit et pntf: pointe vers l'etat final  
for (int i = 0; i < 5; i++) {  
    Etat_Actuel[i] = '0';  
}  
int pos = 0;  
strcpy(Automate_D[0][0], "10000"); //initialisation  
while (fscanf(fich, "%s %s %d", Automate_N[pos].symb1, Automate_N[pos].symb2, &  
    Automate_N[pos].final) != EOF && pos != 128) {  
    pos++; // importation de l'automate non deterministe  
}
```

Boucle pour faire le traitement pour les deux caractères en entrée :

- Si l'état (this_etat) déjà existe on ajoute la position de cet état à l'automate final déterministe.
Sinon on crée un nouvel état dans l'automate final déterministe.
- Remplissage du fichier de sortie.

```
for (int j = 0; j < 49 && !pntf; j++) {
    for (int k = 1; k <= 2; k++) {
        for (int l = 0; l < 5; l++) {
            Etat_Actuel[l] = '0';
        }
        pntf = this_etat(k, Automate_D[j][0]);
        pnt = 1;
        for (int m = 0; m <= j; m++) {
            if (!strcmp(Automate_D[m][0], Etat_Actuel)) {
                snprintf(Automate_D[j][k], 5, "%d", m);
                pnt = 0;
            }
        }
        if (pnt) {
            strcpy(Automate_D[j + 1][0], Etat_Actuel);
            snprintf(Automate_D[j][k], 5, "%d", j + 1);
        }
    }
    fprintf(fich_sortie, "%d : %s    %s    %s\n", j, Automate_D[j][0],
        Automate_D[j][1], Automate_D[j][2]);
}
```

Fermeture de fichier (fich) d'entrée de l'automate final non-déterministe.

Fermeture de fichier de sortie (fich_sortie) d'entrée de l'automate final déterministe.

```
fclose(fich);
fclose(fich_sortie);
```

Exécution :

Contenu du fichier d'entrée : (AFN)

```
0000 0010 0
1000 1100 0
0000 0001 0
0000 0000 1
```

Contenu du fichier de sortie : (AFD)

```
0 : 10000    1    1
1 : 00100    2    2
2 : 00010    3    3
```

Sources :

<https://codes->

sources.commentcamarche.net/source/view/50684/1207507#browser

<https://codes-sources.commentcamarche.net/source/38214-transformation-afn-en-afd>

<https://www.youtube.com/>

<https://stackoverflow.com/>