**DEVOIR MAISON 2 : AUTOMATES : IMPLEMENTATION EN C**

**I. RECONNAISSANCE DE L’HEURE AU FORMAT HH:MM:SS**

**1) PRENDRE CONNAISSANCE DES RESSOURCES**

Les fichiers .h contiennent des suites d’instructions qui représentent des constantes, des fonctions ou encore des structures qui permettront de pouvoir lire un fichier l’analyser et afficher la sortie en nous indiquant si elle est bonne ou pas.

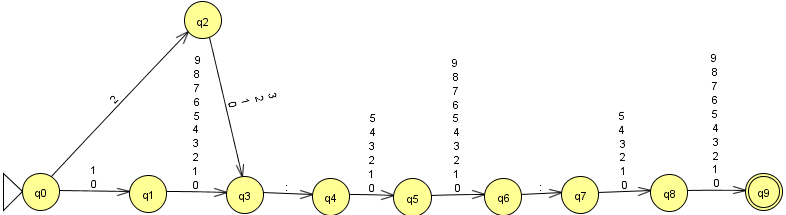
Les #define sont des constantes qui permettent de déterminer une taille ou un nombre maximal qui sera accepté par l’automate : dans notre cas, il s’agit de la taille du tableau de caractères pour la lecture des mots, du nombre maximal d’états de l’automate et du nombre maximum de tests.

On a ensuite la structure lTransitions, qui représente une transition avec un caractère et un état suivant : en d’autres termes, elle permet, en fonction d’un caractère de passer à un état suivant.

Maintenant on a différentes fonctions :

* lireFichier permet de lire le fichier et de renseigner la table de transition. On voit que cette fonction fait appel à la structure lTransitions, à la table de transitions de l’automate mais intègre aussi l’état initial et l’état final de l’automate.
* analyse permet d’analyser le mot entré : on met un pointeur sur le mot à analyser, on intègre la table de transition, l’état initial et l’état final.
* libererTransitions permet de faire passer la structure à l’état suivant : on aura donc une avancée dans la table de transition de l’automate.
* lireFichierDates permet de lire le fichier qui contient les dates à tester et de créer un tableau de pointeurs vers des chaînes de caractères.
* ecrireResultat permet de sortir le résultat en fonction du mot analysé : si le mot est correct alors il va sortir un OK sinon, il va sortir un KO.

La représentation graphique du fichier texte de l’automate se définit comme suit :



**2) COMPRENDRE LA STRUCTURE DES DONNEES**

La variable « char \*tableHeuresTests[NBTEST] » représente les inputs que va faire l’utilisateur et qui seront analysés par l’automate. Le NBTEST permet de limiter les inputs de l’utilisateur à 100 entrées. Nous sommes obligés de faire une chaîne de caractères à cause des « : » présents dans les entrées.

Notre variable sera représentée alors comme suit :

|  |  |
| --- | --- |
| HEURES VALIDES | HEURES NON VALIDES |
| 20:15:16 | 24:15:14 |
| 12:14:07 | 18:55:60 |
| 01:06:54 | 19:60:52 |
| 10:56:02 | 2:25:2 |
| 15:15:15 | 1:1:1 |
| 06:00:00 | 0:25:54 |
| 18:17:16 | 12:13:14:15 |
| 23:59:59 | 1f:52:24 |
| 21:54:27 | 21:0:21 |

**3) FICHIER ENTREESORTIE.C**

 Maintenant que nous avons ouvert le fichier demandé, nous pouvons remplir les fonctions en fonctions des indications qui nous sont données. On obtient alors les deux fonctions suivantes :



**4) FICHIER MAIN.C**

 Dans notre fichier main.c, nous devons remplacer tous les TODO par des boucles ou des fonctions déjà prédéfinies dans les autres fichiers qui nous sont fournies. On obtient alors le programme suivant :

**5) FICHIER ACTIONS.C**