

## **Déterminisation / Minimalisation d'automate fini Reconnaissance de mot**

### **Programme à développer**

Votre programme doit :

- 1- Lire un automate contenu dans un fichier.
- 2- Vérifier si l'automate est déterministe ou non.
- 3- Rendre cet automate déterministe, s'il ne l'est pas déjà. Afficher les ensembles Q, E, I, T correspondant. (\*)
- 4- Rendre l'automate complet. Afficher les ensembles Q, E, I, T correspondant.
- 5- Rendre l'automate minimal. Afficher les ensembles Q, E, I, T correspondant. (\*)
- 6- Lire un mot (chaîne de symboles de l'alphabet).
- 7- Déterminer si le mot est ou non accepté par l'automate.

(\*) Vous devez mettre en œuvre au moins une des étapes 3 (déterminisation) et 5 (minimalisation). Pour la soutenance, des tests partant d'un automate déterministe (mais pas nécessairement complet) seront prévus pour ceux qui ne font que la minimalisation.

La note sera bien évidemment meilleure pour les groupes qui réalisent correctement les deux.

Chaque partie de votre programme doit non seulement afficher le résultat, mais aussi les étapes de son déroulement (traces d'exécution) permettant d'en vérifier le fonctionnement.

Pour la soutenance : prévoyez une boucle sur l'ensemble des traitements pour tester plusieurs fichiers, ainsi qu'une boucle sur les traitements 6 et 7 pour tester plusieurs chaînes sur chaque fichier.

Vous pouvez utiliser les langages C ou C++.

Le choix de la représentation mémoire pour un automate est libre.

Le choix de la représentation d'automate sur fichier est libre.

### **Automates à prendre en compte**

Les tests seront effectués sur des automates :

- ayant pour alphabet des lettres de 'a' à 'z' ;
- dont les états sont numérotés à partir de '1'

Les automates ayant des transitions étiquetées avec la chaîne vide ( $\epsilon$ ) ne sont pas au programme du TAI.

### **Déroulement du TAI**

#### **Constitution des équipes**

Par équipes de 3 étudiants appartenant à un même groupe de TD.

Si le nombre d'étudiants d'un groupe de TD ne le permet pas, il pourra y avoir soit une équipe de 2 étudiants, soit 1 ou 2 équipes de 4 étudiants. Bien entendu, le niveau d'exigence vis à vis du travail effectué sera modulé en conséquence.

**Date limite de constitution des équipes : 12 mars 2010.**

La constitution des équipes devra être communiquée à votre enseignant de TD.

Aucun changement dans les groupes ne sera accepté au delà de cette date, sauf cas de force majeure qui devra être explicitement accepté par les enseignants.

Tests (préparation à la soutenance)

Les automates de test qui seront utilisés pour la soutenance vous seront communiqués début mai 2010.

Ces automates vous seront fournis sous forme de schémas. Vous devrez donc les traduire sous forme de fichier en entrée de votre programme.

Remise de votre travail

Envoi de votre programme (code source) et de vos fichiers de test par email.

Date limite : sera fixée ultérieurement.

Adresse email : sera communiquée ultérieurement.

Les noms des fichiers doivent impérativement être préfixés par les noms des membres de chaque équipe, par exemple :

DupontDurandDupond-monprogramme.c

DupontDurandDupond-test1.txt

Si vous avez besoin d'archiver / compresser plusieurs fichiers : seul le format zip sera accepté ! Le nom de l'archive doit aussi respecter la règle de nommage ci-dessus.

Soutenance

Calendrier : sera fixée ultérieurement.

Durée : 30 minutes par équipe.

Date limite d'inscription des équipes dans les créneaux ouverts : sera communiquée ultérieurement.

Il n'y aura pas d'ordinateur mis à votre disposition. Vous devez impérativement venir avec le votre.

Déroulement : exposé, démonstration, discussion complémentaire éventuelle.

*Exposé*

Durée de l'exposé : 10 à 15 minutes.

Contenu :

- 1- représentation en mémoire des automates,
- 2- algorithme de détermination ou de minimalisation,
- 3- algorithme de reconnaissance de chaîne de symboles,

Chaque membre de l'équipe présentera une partie de l'exposé, sans intervention des autres.

Vous devez impérativement préparer un support écrit (à remettre sur papier en début de soutenance) pour votre exposé. Une page maximum pour chaque partie.

*Conseil : préférez un schéma clair et précis ou du pseudo-code bien structuré que vous commenterez, plutôt qu'un texte long qui ne pourra pas être lu durant la soutenance !*

### *Démonstration*

*Vous aurez préalablement placé dans votre environnement de travail les fichiers correspondant aux automates de test fournis.*

Vous compilerez votre programme, puis vous lancerez son exécution. L'examineur vous indiquera le ou les fichiers à lire, et les chaînes à vérifier.

Pour effectuer des tests complémentaires, des modifications des fichiers d'entrée pourront être demandées en séance.

*Donc : l'utilisation d'un fichier est obligatoire et les ensembles  $I$ ,  $T$  et  $E$  doivent être représentés de façon simple.*

### **Éléments de notation**

Vous serez bien entendu jugés sur le fonctionnement de votre programme. Sachez que les éléments suivants seront également pris en compte :

- clarté de l'interface de votre programme permettant un suivi simple de la démonstration (enchaînement des actions, trace d'exécution, ...) et une vérification aisée du bon fonctionnement de votre programme ;
- choix / justification des structures de données pour représenter les automates ;
- clarté de votre exposé et du support.

Une note globale sera donnée au groupe, mais pourra être modulée de quelques points notamment en fonction de la prestation orale de chacun.

### **Informations complémentaires**

Un forum de questions/réponse sera ouvert sur Campus. Vos questions devront être posées dans ce forum.