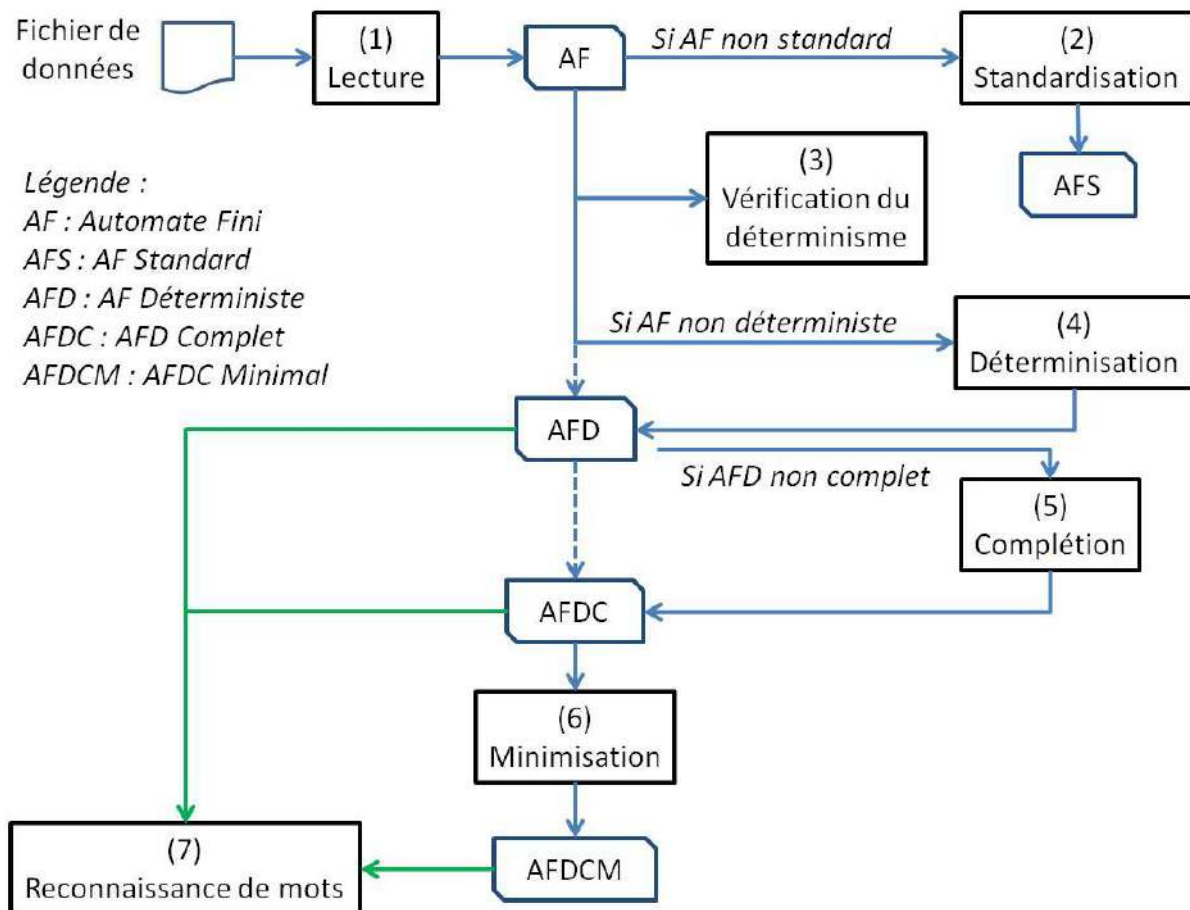


Traitement d'automate fini Reconnaissance de mot

Programme à développer

Votre programme doit effectuer les traitements indiqués dans le tableau ci-après, et illustrés dans le schéma suivant :



Les traitements doivent être exécutés dans l'ordre de leur numérotation.

Chaque traitement devant produire un automate (1, 2, 4, 5 et 6) doit impérativement se terminer par l'affichage d'une table de transitions, avec indication des états initiaux et terminaux.

- 1- Lire un automate contenu dans un fichier et le stocker en mémoire.
 Vous avez le choix de la syntaxe utilisée pour le fichier. La syntaxe doit cependant être suffisamment simple pour qu'on puisse très facilement, sur demande de l'enseignant pendant la soutenance, effectuer quelques modifications sur les automates existants, voire ajouter un automate simple.
 Le choix du fichier à lire se fait au travers de l'interface utilisateur (pendant l'exécution de votre programme). La saisie clavier de l'automate (sans fichier) n'est pas prévue.
- 2- Standardiser l'automate résultant de l'étape 1, s'il ne l'est pas (option).
 Votre programme doit tout d'abord indiquer si l'automate est

- standard ou non puis, le cas échéant, le standardiser.
- 3- Vérifier si l'automate résultant de l'étape 1 est déterministe ou non.
Cette étape doit se faire indépendamment du processus de déterminisation.
Le résultat, « déterministe » ou « non déterministe », doit s'afficher clairement.
 - 4- Rendre déterministe l'automate résultant de l'étape 1, s'il ne l'est pas déjà.
Entre autres choses, votre programme doit indiquer clairement à quels états de l'automate initial correspond chacun des états de l'automate déterministe, ce qui veut dire qu'il vaut mieux garder, pour les états composés de l'AFD, la notation correspondant à l'ensemble d'états de l'AF non déterministe auxquels ils correspondent.
 - 5- Rendre complet l'automate issu de l'étape 4 (ou de l'étape 1 s'il est déjà déterministe).
Votre programme doit d'abord indiquer si l'automate est déjà complet ou non.
 - 6- Rendre minimal l'automate issu des étapes précédentes (1, 4 ou 5 selon les cas).
Entre autres choses, votre programme doit indiquer clairement à quels états de l'automate déterministe correspond chacun des états de l'automate minimal. Ici, il peut ne pas être pratique de les noter par des ensembles d'états de l'AFDC qu'on minimise, si ceux-ci sont déjà notés comme des ensembles. Il peut être préférable d'utiliser des lettres, ou de chiffres romains. Une table de correspondance doit alors apparaître sur l'écran.
Les partitions successives du processus de minimisation doivent être affichées, et pas seulement le résultat final.
 - 7- Lire un mot (chaîne de symboles de l'alphabet).
La saisie des mots à reconnaître se fait au clavier, de façon simple (on tape le mot, on presse Entrer), et non pas caractère par caractère. Attention : prévoir une façon (de votre choix) de saisir le mot vide, car celui-ci sera certainement testé pour les automates retenus par l'examineur.
 - 8- Déterminer si le mot est ou non accepté par l'automate.
Affiche 'ok' ou 'non ok'.
La reconnaissance peut s'effectuer au choix sur l'AFD, l'AFDC ou l'AFDCM. Il est vivement déconseillé de se lancer dans une reconnaissance de mots sur la base de l'AF initial s'il est non déterministe.
Votre programme peut aussi afficher l'endroit dans le mot à partir duquel il peut être décidé comme n'appartenant pas au langage. Plusieurs cas peuvent être détectés, certains en fonction de l'automate utilisé pour la reconnaissance, par exemple : mot trop court (fin d'analyse sur un état non terminal), symbole non attendu (transition vers l'état poubelle de l'AFDC, ou absence de transition dans un AFD non complet).

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 7 et 8 pour tester plusieurs chaînes sur chaque fichier.

L'interface utilisateur doit être assez simple et guidée par des messages écran de façon à ce que l'enseignant puisse ne pas vous demander comment faire ceci ou cela.

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

Le choix de la représentation d'automate sur fichier est libre à condition d'être immédiatement compréhensible.

Vous pouvez utiliser les langages C ou C++.

Votre programme doit pouvoir être compilé et s'exécuter sous Unix/Linux et Windows. Evitez des instructions n'appartenant pas au langage standard (non reconnu par le compilateur gcc). A vous de vérifier sur les machines de l'école, et de n'utiliser que des mécanismes suffisamment standards.

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 (ϵ) ne sont pas au programme du TAI.

Il n'y a pas de limite concernant le nombre d'états que les automates de test pourront contenir. Rien ne vous empêche de prévoir cette information dans chaque fichier de donnée (traitement '1') si vous le jugez utile.

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 : vendredi 16 mars 2011.

La constitution des équipes devra être envoyée par les délégués des groupes de TD par email aux adresses suivantes :

herve.barbot@gmail.com

velikson.efrei@gmail.com

Dès réception complète, la constitution de tous les groupes sera diffusée sur Campus. Aucun changement 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 2012.

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 uniquement, bien commenté) et de vos fichiers de test par email.

Date limite : sera fixée ultérieurement.

Adresse email : voir section concernant la constitution des groupes.

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

DupontDurandDupond-monprogramme.cpp

DupontDurandDupond-monprogramme.h

DupontDurandDupond-test1.txt

Pas de fichier archive autre qu'au format zip. Vous pouvez aussi envoyer autant de pièces jointes que nécessaire.

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.

Exposé structuré en 3 parties distinctes :

- 1- format du fichier, mécanisme de lecture, représentation en mémoire des automates, standardisation ;
- 2- déterminisation et minimalisation ;
- 3- reconnaissance de chaîne de symboles.

Chaque membre de l'équipe présentera une partie de l'exposé, sans intervention des autres. Sachez à l'avance qui présente quoi !

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

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 ! Votre structure de données et vos algorithmes doivent en sortir de façon absolument claire, et vous n'utiliserez pas directement le code C ou C++ pour la présentation.

Si vous voulez utiliser une présentation formelle de type « powerpoint », vous devrez vous équiper d'un vidéoprojecteur. Ceci n'est cependant pas conseillé dans la mesure où sa mise en place prend du temps.

Démonstration

Vous aurez préalablement placé dans votre environnement de travail tous 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 ou fournira un autre automate, et les chaînes à vérifier. Il pourra également demander la modification d'un automate existant, voire la saisie d'un nouvel automate.

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.

Assurez-vous que votre ordinateur est opérationnel (batterie chargée, système lancé, ...) avant d'entrer dans la salle de soutenance !

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

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

Informations complémentaires

Une séance de TD sera utilisée pour répondre à vos questions concernant le TAI. Elle sera programmée durant la semaine du 02 au 06 avril. Chaque trinôme devra préalablement envoyer ses questions par écrit aux adresses indiquées plus haut au plus tard le 26 mars.

Un forum de questions/réponse sera également ouvert sur Campus pour les questions que vous pourrez avoir après cette date.