Structures Discrètes - T.M.E. Langages et automates

Partie I : création d'automates

Le logiciel JFLAP se trouve sous Unix ; si vous le souhaitez, vous pouvez le récupérer à cette adresse : `http://www.cs.duke.edu/ rodger/tools/jflap/`, où il y a des versions pour Unix, Windows et MacOS X. Lancer le logiciel **JFLAP** avec la commande

```
java -jar /usr/local/jflap/JFLAP.jar
```

et choisir **Finite Automaton**. Il apparaît alors une fenêtre d'édition, dans laquelle on peut dessiner un automate. La barre d'outils située sous l'onglet **Editor** contient quatre boutons ; de gauche à droite :

- **Attribute Tool** que l'on nommera ici bouton **A**
- **State Tool** (bouton **S**)
- **Transition Tool** (bouton **T**)
- **Deleter** (bouton **D**).

Exercice 1 Soit l'automate $\mathcal{A}$ défini sur l'alphabet $\{a, b\}$, d'états $0, 1, 2$, d'état initial 0, d'état final 2 et de transitions $0.a = 0$, $0.b = 1$, $1.a = 2$, $1.b = 2$, $2.a = 0$ et $2.b = 1$.

Création de l'automate $\mathcal{A}$

Pour dessiner l'automate $\mathcal{A}$:

– créer les états : cliquer sur le bouton **S** ; cliquer en trois endroits différents de la fenêtre d'édition, les états apparaissent avec les noms q_0, q_1, q_2

– choisir la nature des états (initial, final) : cliquer sur le bouton **A** ; faire un clic droit sur l'état q_0 et choisir *Initial* ; faire un clic droit sur l'état q_2 et choisir *Final*

– créer les transitions : cliquer sur le bouton **T**

  - transition $0.a = 0$: cliquer sur l'état q_0 et taper a dans le cadre qui apparaît
  - transition $0.b = 1$: promener la souris, bouton gauche enfoncé, de l'état q_0 à l'état q_1, lâcher le bouton et taper b dans le cadre qui apparaît
  - autres transitions : sur le modèle de $0.b = 1$
  - attention : il faut créer deux transitions pour $1.a = 2$ et $1.b = 2$ (ne pas taper les deux étiquettes a et b dans le même cadre).

– sauver le fichier en le nommant “exo1-tme”.

Vous découvrirez tout seul les autres utilisations du bouton **A** et l'utilisation du bouton **D**.

Reconnaissance de mots par $\mathcal{A}$

Pour vérifier si des mots sont acceptés, on utilise l'un des menus *Input/Step by State* ou *Input/Fast Run* ou *Input/Multiple Run.*

- Menu *Input/Step by State* : sous la fenêtre contenant l'automate apparaissent une fenêtre montrant les différentes configurations et une barre contenant six boutons. En cliquant sur le bouton **Step**, le mot est lu lettre à lettre, la progression dans le mot est visible dans la fenêtre des configurations et la progression dans l'automate est visible dans la fenêtre de l'automate. Après le dernier **Step**, la configuration est soit verte (le mot est accepté) soit rouge (le mot est rejeté). En cliquant sur une configuration puis sur le bouton **Trace**, on voit apparaître, dans une autre fenêtre, la suite des transitions qui ont amené à cette configuration.
- Menus *Input/Fast Run* et *Input/Multiple Run* : utilisation évidente.

Remarque : il y a aussi un menu *Input/Step by Closure* mais il n'a d'intérêt que pour les automates avec λ-transitions (transitions étiquetées par le mot vide).

En utilisant chacun des trois menus précédents, vérifier si les mots suivants sont acceptés ou non par l'automate $\mathcal{A}$: *ba*, *babb*, *aaabab*, *aaababa*, *abaabb*, *abaabaaab*. $\diamondsuit$

Exercice 2 On considère l'automate $\mathcal{B}$ obtenu en ajoutant la transition $2.b = 2$ à l'automate $\mathcal{A}$ défini dans l'exercice 1.
1. Dessiner l'automate $\mathcal{B}$ et sauver le fichier en le nommant "exo2-tme".
2. Choisir le menu *Test/Compare Equivalence* et tester si les automates $\mathcal{A}$ et $\mathcal{B}$ sont équivalents.
3. Utiliser le menu *Test/Highlight Nondeterminism* pour vérifier que l'automate $\mathcal{B}$ n'est pas déterministe.
4. Utiliser le menu *Input/Step by State* pour vérifier que le mot *ababbba* est accepté. Dans la fenêtre des configurations, il apparaît cinq configurations correspondant aux cinq lectures possibles du mot *ababbba* ; on remarque que certaines configurations sont acceptées et d'autres pas. Faire afficher la trace de chacune des cinq configurations.
5. Tester d'autres mots, en utilisant *Input/Step by State* ou *Input/Fast Run.* $\diamondsuit$

Exercice 3 Soit $A = \{a, b, c\}$. Dessiner des automates, non nécessairement déterministes, reconnaissant les langages suivants :
1) $\{a, ab, ca, cab, acc\}$
2) $[a(b + c)^*abc]^*$
3) l'ensemble des mots contenant un nombre impair de a
4) l'ensemble des mots contenant le facteur ab
5) l'ensemble des mots contenant le facteur ab et se terminant par b $\diamondsuit$

Partie II : Déterminisation d'automates

Exercice 4 Dessiner l'automate $\mathcal{A}$ sur l'alphabet $\{a, b, c\}$ d'états $0, 1, 2, 3$, d'état initial 0, d'état terminal 3 et de transitions $0.a = 1$, $0.a = 0$, $0.b = 0$, $0.c = 0$, $1.a = 2$, $1.b = 2$, $1.c = 2$, $2.a = 3$, $2.b = 3$, $2.c = 3$ (*cf.* ex. 7 du TD), en faisant en sorte que l'état i soit nommé q_i dans le dessin. Vérifier que $\mathcal{A}$ n'est pas déterministe.

Pour déterminiser l'automate $\mathcal{A}$, on choisit le menu *Convert/Convert to DFA*. La fenêtre se partage en deux :
– la partie gauche contient l'automate $\mathcal{A}$
– la partie droite est une sous-fenêtre de travail, dans laquelle on construit le déterminisé $\mathcal{A}'$ de $\mathcal{A}$, et dont la barre d'outils contient cinq boutons ; de gauche à droite :
- **Attribute Editor** (bouton **A**)
- **Expand Group on Terminal** (bouton **T**)
- **State Expander** (bouton **S**)
- **Complete**
- **Done?**.

Initialement, la sous-fenêtre de travail contient un seul état, l'état initial, nommé q_0, auquel est attaché un petit cadre contenant tous les indices des états initiaux de $\mathcal{A}$ (ici, le cadre contient 0). Il faut construire toutes les transitions de $\mathcal{A}'$, c'est-à-dire les transitions : $\{0\}.a = \{0, 1\}$, $\{0\}.b = \{0\}$, $\{0\}.c = \{0\}$, $\{0, 1\}.a = \{0, 1, 2\}$, $\{0, 1\}.b = \{0, 2\}$, etc...
Il y a plusieurs façons de construire les transitions :
— on peut les construire une à une : cliquer sur le bouton **T**
  – transition $\{0\}.a = \{0, 1\}$: promener la souris, bouton gauche enfoncé, de l'état q_0 de $\mathcal{A}'$ vers un endroit quelconque de la sous-fenêtre de travail ; lâcher le bouton ; une boîte de dialogue apparaît, demandant l'étiquette de la transition que l'on veut construire ("Expand on what terminal?") ; taper a ; une nouvelle boîte de dialogue apparaît, demandant l'état

but de la transition que l'on veut construire ; taper 0 et 1 en les séparant par un espace ; apparaissent alors l'état q_1 avec un petit cadre contenant 0,1 et la transition d'étiquette a qui va de q_0 à q_1.

- si l'état but de la transition existe déjà dans $\mathcal{A}'$, procéder comme dans la création d'automates ; par exemple, pour construire la transition $\{0\}.b = \{0\}$: cliquer sur l'état q_0 de $\mathcal{A}'$ puis taper b dans la boîte de dialogue

– on peut construire en une seule fois toutes les transitions issues d'un état q de $\mathcal{A}'$: cliquer sur le bouton **S** puis sur l'état q

– on peut construire toutes les transitions en une seule fois : cliquer sur le bouton **Complete**.

Construire l'automate $\mathcal{A}'$, déterminisé de $\mathcal{A}$, en utilisant uniquement le bouton **T**. ♢

Exercice 5 Déterminiser l'automate $\mathcal{B}$ défini dans l'exercice 2 (sans utiliser le bouton **Complete**). ♢

Exercice 6 Tester si les automates construits dans l'exercice 3 sont déterministes et déterminiser ceux qui ne le sont pas (sans utiliser le bouton **Complete**). ♢

Partie III : Minimisation d'automates

Pour minimiser un automate $\mathcal{A}$, on utilise l'algorithme suivant :

– initialement l'ensemble des états est partagé en deux sous-ensembles : l'ensemble des états terminaux et l'ensemble des états non terminaux

– on réitère le processus suivant :

  - parmi les ensembles d'états déjà construits Q_1, Q_2 ... Q_m, on choisit un ensemble Q_i
  - on choisit une lettre x
  - on partage l'ensemble Q_i en plusieurs sous-ensembles : deux états p et q de Q_i appartiennent au même sous-ensemble ssi les états $p.x$ et $q.x$ de $\mathcal{A}$ appartiennent à un même ensemble Q_j

– et ce jusqu'à ce qu'aucune lettre ne puisse plus partager l'un des ensembles d'états.

Exercice 7 Soit l'automate $\mathcal{A}$ défini sur l'alphabet $\{a, b\}$, d'états 0, 1, 2, 3, 4, 5, d'état initial 0, d'état final 4 et de transitions $0.a = 1$, $0.b = 2$, $1.a = 3$, $1.b = 3$, $2.a = 3$, $2.b = 3$, $3.a = 4$, $3.b = 5$, $4.a = 4$, $4.b = 4$, $5.a = 5$ et $5.b = 4$.

Pour minimiser l'automate $\mathcal{A}$, on peut dérouler l'algorithme de plusieurs manières.

Une première manière :

– initialement : $\{0, 1, 2, 3, 5\}$ et $\{4\}$
– on partage $\{0, 1, 2, 3, 5\}$ en utilisant a : $\{0, 1, 2, 5\}$, $\{3\}$ et $\{4\}$
– on partage $\{0, 1, 2, 5\}$ en utilisant b : $\{0\}$, $\{1, 2\}$, $\{5\}$ et $\{3\}$, $\{4\}$
– on ne peut pas partager $\{1, 2\}$ (ni en utilisant a, ni en utilisant b).

Une deuxième manière :

– initialement : $\{0, 1, 2, 3, 5\}$ et $\{4\}$
– on partage $\{0, 1, 2, 3, 5\}$ en utilisant a : $\{0, 1, 2, 5\}$, $\{3\}$ et $\{4\}$
– on partage $\{0, 1, 2, 5\}$ en utilisant a : $\{0, 5\}$, $\{1, 2\}$ et $\{3\}$, $\{4\}$
– on partage $\{0, 5\}$ en utilisant a : $\{0\}$, $\{5\}$ et $\{1, 2\}$, $\{3\}$, $\{4\}$
– on ne peut pas partager $\{1, 2\}$ (ni en utilisant a, ni en utilisant b).

Et il y a encore d'autres manières...

Minimisation de l'automate $\mathcal{A}$

Dessiner l'automate $\mathcal{A}$, en faisant en sorte que l'état i soit nommé q_i dans le dessin. Pour construire l'automate $\mathcal{A}'$, minimisé de l'automate $\mathcal{A}$, on choisit le menu *Convert/Minimize DFA*. La fenêtre se partage en deux :

– la partie gauche contient l'automate $\mathcal{A}$ (ou le complété de $\mathcal{A}$ si $\mathcal{A}$ n'est pas complet)

– la partie droite est une sous-fenêtre de travail qui contient un arbre dont les feuilles représentent des ensembles d'états.

Initialement, l'arbre a seulement deux feuilles : l'une représente les états non terminaux (ici 0, 1, 2, 3, 5) et l'autre les états terminaux (ici 4). Pour compléter cet arbre, on partage les ensembles d'états comme il est dit dans l'algorithme de minimisation.

En suivant la première manière :

– pour partager $\{0,1,2,3,5\}$ en utilisant a : cliquer sur le cadre contenant 0, 1, 2, 3, 5 ; cliquer sur le bouton **Set Terminal** ; taper a dans la boîte de dialogue ; sous le cadre contenant 0, 1, 2, 3, 5, apparaissent deux sous-arbres, dont les feuilles sont vides ; on remplit l'une des deux feuilles avec 0, 1, 2, 5 et l'autre feuille avec 3 ; pour cela, cliquer sur l'une des deux feuilles puis, successivement, sur les états q_0, q_1, q_2, q_5 de $\mathcal{A}$; ensuite, cliquer sur l'autre feuille puis sur l'état q_3 de $\mathcal{A}$; remarque : la même démarche (cliquer sur la feuille de l'arbre puis sur l'état de l'automate $\mathcal{A}$) permet d'enlever un état déjà présent dans une feuille ; cliquer sur le bouton **Check Node** pour soumettre la partition ;

– pour partager $\{0,1,2,5\}$ en utilisant b : cliquer sur le cadre contenant 0, 1, 2, 5 ; cliquer sur le bouton **Set Terminal** ; taper b dans la boîte de dialogue ; sous le cadre contenant 0, 1, 2, 5, apparaissent deux sous-arbres ; comme la lettre b partage $\{0,1,2,5\}$ en trois sous-ensembles ($\{0\}$, $\{1,2\}$ et $\{5\}$), il faut ajouter un sous-arbre (bouton **Add Child**) ; remplir les trois feuilles ; cliquer sur le bouton **Check Node** pour soumettre la partition ; si elle n'est pas correcte, un message d'erreur s'affiche et il faut alors modifier la composition des feuilles ; si elle est correcte, le message "The expansion is correct!" s'affiche

– comme il n'y a plus d'ensemble d'états à partager, le seul bouton accessible est le bouton **Finish** ; cliquer dessus ; dans la sous-fenêtre de travail, l'arbre est remplacé par l'ensemble des états de l'automate minimal ; construire les transitions de l'automate minimal

– cliquer sur le bouton **Done?** ; s'il manque des transitions, un message le signale.

Refaire la minimisation de l'automate $\mathcal{A}$ en suivant la deuxième manière de dérouler l'algorithme.
$\diamondsuit$

Exercice 8 Soit l'automate $\mathcal{A}$ d'états 0, 1, 2, 3, 4, 5, d'état initial 0, d'état terminal 5 et de transitions : $0.a = 1$, $1.a = 2.a = 2$, $3.a = 4.a = 4$, $5.a = 5$, $0.b = 3$, $1.b = 2.b = 3.b = 4.b = 5.b = 5$. Dessiner l'automate $\mathcal{A}$. Minimiser $\mathcal{A}$. $\diamondsuit$

Exercice 9

1) Soit l'automate $\mathcal{A}$ d'états 0, 1, 2, 3, 4, 5, d'état initial 0, d'état terminal 3 et de transitions : $0.a = 1$, $1.a = 1$, $2.a = 4$, $3.a = 5$, $4.a = 4$, $5.a = 5$, $0.b = 2$, $1.b = 3$, $2.b = 2$, $3.b = 3$, $4.b = 5$, $5.b = 5$. Dessiner l'automate $\mathcal{A}$. Minimiser $\mathcal{A}$.

2) Soit l'automate $\mathcal{B}$ d'états 0, 1, 2, 3, 4, 5 d'état initial 0, d'états terminaux 3, 4, 5 et de même transitions que $\mathcal{A}$. Dessiner l'automate $\mathcal{B}$. Minimiser $\mathcal{B}$. $\diamondsuit$

Exercice 10

1) Soit l'automate $\mathcal{A}$ d'états 0, 1, 2, 3 d'état initial 0, d'état terminal 2 et de transitions : $0.a = 0$, $0.a = 1$, $1.a = 1$, $3.a = 0$, $1.b = 1$, $1.b = 2$, $3.b = 0$, $3.b = 2$, $1.c = 1$, $1.c = 3$,$3.c = 0$.

1) Dessiner l'automate $\mathcal{A}$.

2) Déterminiser $\mathcal{A}$.

3) Minimiser $\mathcal{A}$. $\diamondsuit$

Partie IV : À découvrir seul

1) Le menu *Convert/Convert FA to RE* de la fenêtre **Editor**. Il permet de calculer une expression rationnelle pour le langage reconnu par un automate. Pour chacun des exercices 22, 24, 25, 26 de la feuille de TD, on pourra comparer l'expression rationnelle calculée par le logiciel JFLAP à l'expression rationnelle calculée en TD. On pourra au préalable traiter l'exercice suivant :

EXERCICE 11 Soient X et Y des langages sur un alphabet A. Montrer que

$$(X + Y)^* = (X^*Y)^*X^* = X^*(YX^*)^*$$

$\diamondsuit$

2) Le bouton **Regular Expression** de la fenêtre **New Document**. Il permet de dessiner l'automate reconnaissant un langage donné par une expression rationnelle (ne contenant que des +, . et *). Pour chacune des expressions rationnelles de l'exercice 21 de la feuille de TD, on pourra comparer l'automate construit par le logiciel JFLAP à l'automate construit en TD.