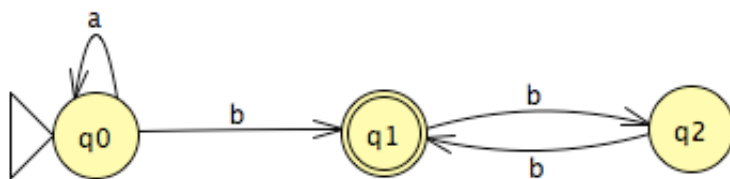


Exercices AF, MT et JFLAP

- [JFLAP](#)
- [Copie](#) locale de version 6
- AFD = automate fini déterministe
- AFND = automate fini déterministe
- MTD = machine de Turing déterministe
- MTND = machine de Turing non-déterministe
- $\Sigma = \{a, b\}$

Exercice 1

Considérer l'AFD A1 :



1. Définir formellement $A1 = (Q, \Sigma, \delta, s, F)$.
2. Décrire l'ensemble des mots accepté par A1 en français.
3. Donner une expression régulière décrivant le langage accepté.
4. Construire A1 à l'aide de JFLAP.
5. Utiliser l'item *Multiple Run* du menu *Input* pour le tester.
6. Utiliser l'item *Convert FA to RE* du menu *Convert* pour retrouver une expression régulière des mots acceptés par A1.

Exercice 2

Soit L l'ensemble des mots construits sur Σ qui se terminent par (au moins) un a.

1. Définir un AFD A2 qui accepte exactement L en donnant sa représentation graphique et formelle.
2. Le construire avec JFLAP.
3. Le tester avec JFLAP.
4. Le convertir en expression régulière.

Même énoncé avec l'ensemble L des mots construits sur Σ qui ne contiennent pas deux a consécutifs (indication : penser au complémentaire de L).

Exercice 3

1. Coder et tester l'automate fini non-déterministe de la page 41 de ce [PDF](#). Dessiner l'arbre des exécutions possibles démarrant à la configurations (q_0, aab) . Pour quelle raison le mot aab est-il accepté par cet AFND ?
2. Coder, tester et valider l'AFND de la page 40 de ce [PDF](#).

Exercice 4

1. Montrer que l'ensemble des automates finis (déterministes ou pas) est dénombrable.
2. Montrer que l'ensemble des langages n'est pas dénombrable.
3. En déduire qu'il existe des langages non réguliers.
4. Exhiber un langage non régulier.

Exercice 5

1. S'initier à la construction d'automate à pile via le [tutoriel JFLAP](#). Tester via l'item 'Step with closure' sur les mots abab et aabb.
2. Coder puis tester l'automate à pile de la page 95 de ce [PDF](#).
3. Comment compiler un automate fini quelconque vers un automate à pile reconnaissant le même langage ?

Exercice 6

Donner une définition aussi simple que possible d'une MTD M telle pour tout mot d'entrée w construit sur Σ , l'exécution de M sur w est une suite infinie de configuration. Tester avec JFLAP.

Exercice 7

Soit L l'ensemble des mots construits sur Σ qui se terminent par au moins un a .

1. Définir une MTD M_1 qui décide L .
2. Définir une MTD M_2 qui accepte L mais ne le décide pas.

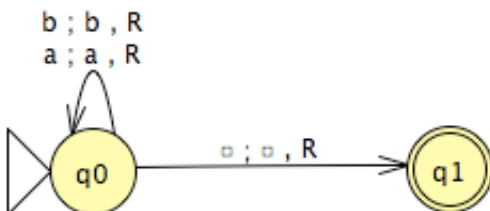
Dans chacun des cas, justifier. Tester avec JFLAP.

Exercice 8

Préciser comment compiler tout AFD en une MTD de comportement équivalent.
Application : exercice 1 et 2.

Exercice 9

Considérer la MTD M_1 :

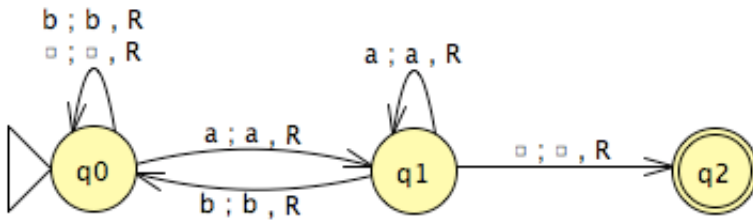


1. Définir formellement $M_1 = (Q, \Gamma, \Sigma, \delta, s, B, F)$.
2. Justifier la terminaison de M_1 pour tout mot d'entrée.
3. Quel est le langage décidé par M_1 ?

4. Construire M1 à l'aide de JFLAP.
5. Tester M1.

Exercice 10

Considérer la MTD M2 :



1. Construire M2 à l'aide de JFLAP.
2. Existe-t-il des mots qui génèrent une exécution infinie de M2 ? Le constater.
3. M2 peut-elle décider un langage ?
4. Quel est le langage accepté par M2 ?
5. Modifier M2 pour qu'elle décide ce langage.

Exercice 11

Soit la machine de Turing $M=(Q, \Gamma, \Sigma, \delta, q_0, \#, \emptyset)$ avec :

- $Q=\{q_0, q_1, q_2, q_3\}$,
- $\Gamma=\{a, b, A, A', B, B', \#\}$,
- δ définie par :

$(q_0, a) \rightarrow (q_1, A', R)$	$(q_0, b) \rightarrow (q_3, B', R)$
$(q_1, a) \rightarrow (q_1, a, R)$	$(q_3, a) \rightarrow (q_3, a, R)$
$(q_1, b) \rightarrow (q_1, b, R)$	$(q_3, b) \rightarrow (q_3, b, R)$
$(q_1, A) \rightarrow (q_1, A, R)$	$(q_3, A) \rightarrow (q_3, A, R)$
$(q_1, B) \rightarrow (q_1, B, R)$	$(q_3, B) \rightarrow (q_3, B, R)$
$(q_1, \#) \rightarrow (q_2, A, L)$	$(q_3, \#) \rightarrow (q_2, B, L)$
$(q_2, a) \rightarrow (q_2, a, L)$	$(q_2, b) \rightarrow (q_2, b, L)$
$(q_2, A) \rightarrow (q_2, A, L)$	$(q_2, B) \rightarrow (q_2, B, L)$
$(q_2, A') \rightarrow (q_0, A, R)$	$(q_2, B') \rightarrow (q_0, B, R)$

1. M est-elle déterministe ou non-déterministe ? Justifier.
2. Donner la suite des configurations de M obtenues pour le mot d'entrée abb.
3. Quel est le contenu du ruban après exécution de M sur le mot abab ?
4. Vérifier ces deux derniers résultats avec JFLAP.

Exercice 12

Construire une MTND à 4 états qui accepte les mots de Σ^* se terminant par ab. Dessiner les arbres

d'exécutions pour le mot vide, a, ab, aba et bab.

Exercice 13

Construire une MTD qui calcule l'application $f(n)=2n$, où les entiers sont codés :

1. en notation binaire ;
2. en notation unaire.

Exercice 14

- Définir une MTD qui décide $a^n b^n$, pour tout n entier
- Définir une MTD qui décide $a^n b^n c^n$, pour tout n entier
- Qu'illustrent ces deux machines relativement à la théorie des langages ?

Exercice 15

On peut définir des machines de Turing qui ont la possibilité supplémentaire d'écrire un symbole sans modifier la position de la tête de lecture.

1. Définir la relation de dérivation entre configurations pour ces machines (cf. la page 57 du support de cours).
2. Expliquer comment compiler de telles machines en machines de Turing standards.
3. Appliquer à la machine ci dessous.

