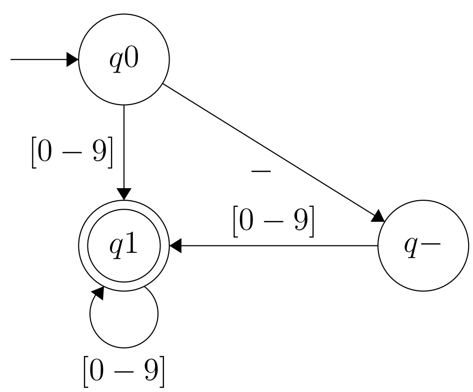
MAT115 - Exercices sur les automates à états finis

Manuel Lafond

Exercice 1: Pour chacun des ensembles de mots suivant, donnez un automate fini (déterministe ou non) qui accepte les mots de cet ensemble. À vous de décider d'un alphabet pertinent.

a. Les chaînes de caractères qui représentent des entiers, possiblement négatifs. Le caractère "-" pour le moins peut donc apparaître.

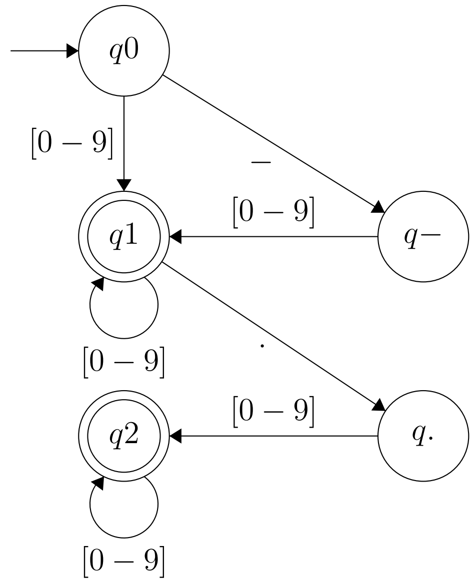
   Solution.



□

b. Les chaînes de caractères qui représentent des nombres réels, avec le point comme séparateur de décimales.

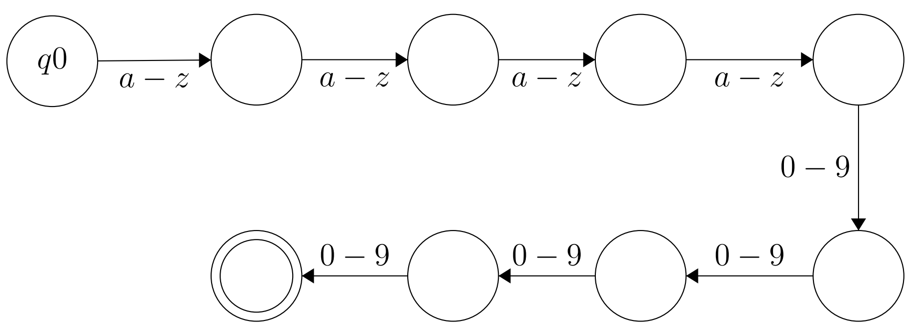
   Solution.



□

c. Les cip. Un cip est une chaîne contenant quatre lettres suivies de quatre chiffres.

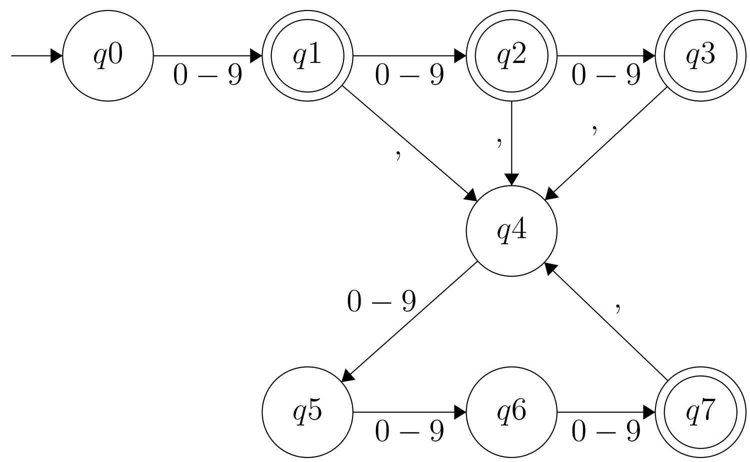
Solution.



□

d. Les entiers positifs, avec une vigule avant chaque groupe de trois chiffres (quand on lit de droite à gauche). Par exemple, $1,000,000$ ou encore $22,542,367,754,576$. S'il y a trois chiffres ou moins, il ne devrait pas y avoir de virgule.

Solution.

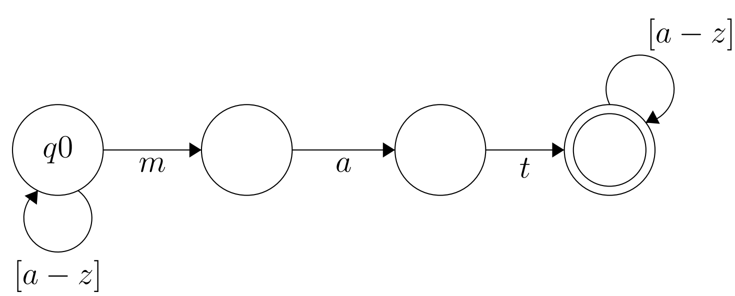


□

e. Les chaînes de caractères contenant la sous-chaîne “mat”. Essayez de faire un automate non-déterministe avec 4 états.

Solution.
J'ai présenté la version déterministe en classe. Voici un AFND pour la même tâche.

NOTE: l'état initial est $q0$. J'ai oublié de dessiner la flèche d'entrée vers l'état initial, et je n'ai pas envie de regénérer tout le dessin.



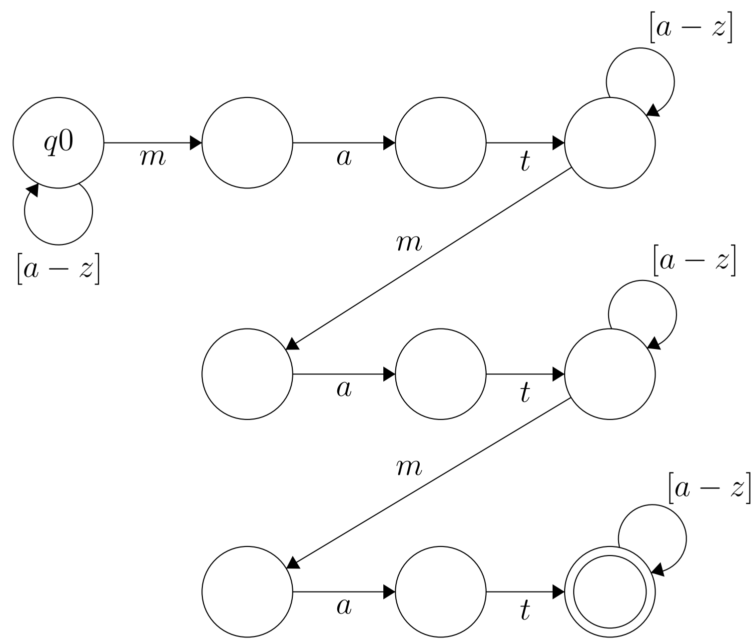
□

f. Les chaînes de caractères contenant au moins trois occurrences de la sous-chaîne “mat”.

Solution.
Il n'y a pas de compteur avec les automates. La seule façon que je connais

est d'avoir trois séries d'états pour chaque occurrence. Ma version est un AFND.

NOTE: l'état initial est $q0$. J'ai oublié de dessiner la flèche d'entrée vers l'état initial, et je n'ai pas envie de regénérer tout le dessin.

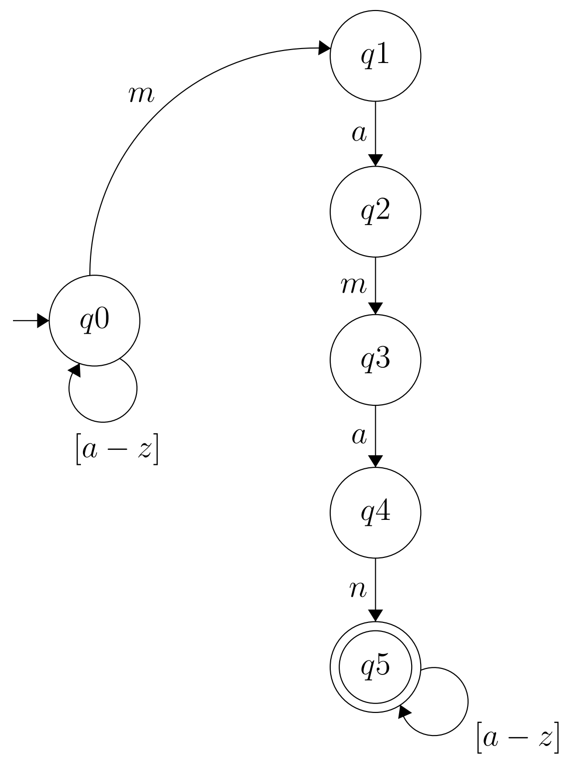


□

g. Les chaînes de caractères contenant la sous-chaîne "maman".

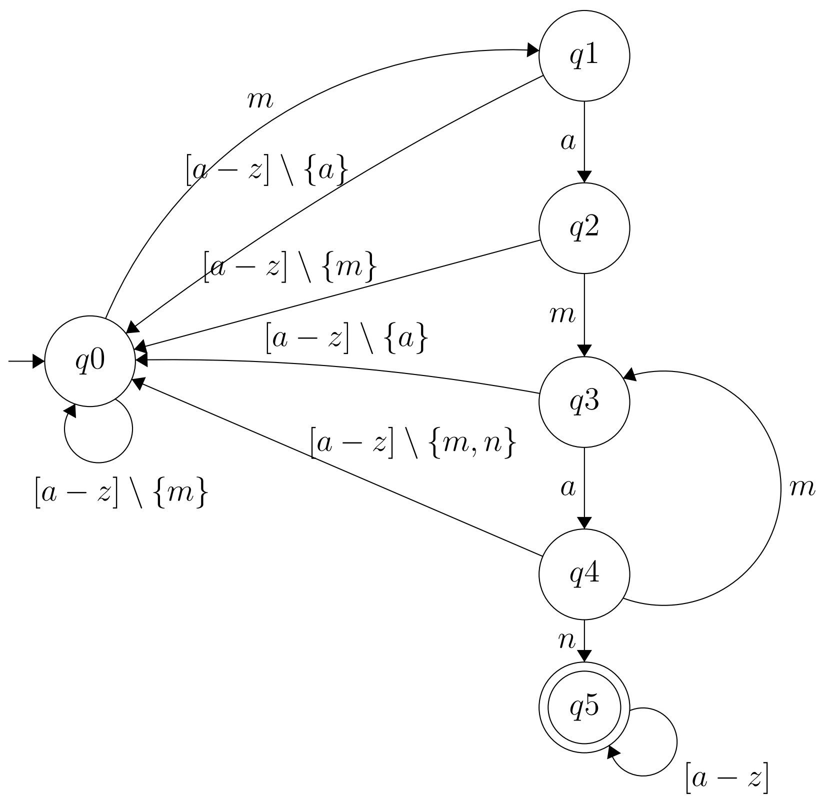
Faites un AFND et un AFD.

Solution.
En AFND, c'est facile: on branche sur "maman" quand il y a une occurrence.



En AFD, c'est plus subtil. Que fait-on sur le mot "mamaman"? Après avoir parcouru "mamama", il ne faut pas revenir au départ, mais plutôt au deuxième m de *maman*.

C'est plutôt horrible car dans la plupart des cas, il faut revenir à $q0$, mais l'important ici est de comprendre pourquoi la transition de $q4$ à $q3$ est requise.

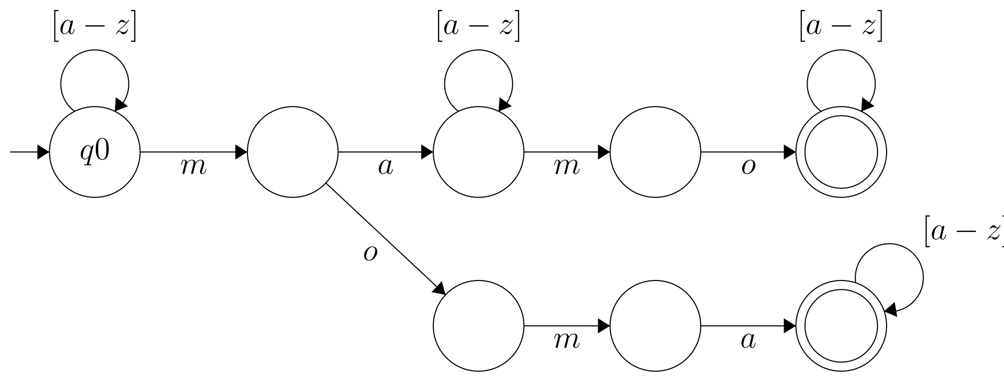


□

h. Les chaînes de caractères contenant au moins une occurrence de “ma” et au moins une occurrence de “mo”.

Solution.
Un des problèmes est qu’on ne sait pas lequel des deux sous-mots arrive en premier. Il faut tester les deux ordres possibles, je ne connais pas d’autre façon. Ma version est non-déterministe.

NOTE: il manque une transition dans le premier état du bas (où on arrive quand on suit “m, o”). Il devrait transitionner sur lui-même sur $[a - z]$.



$\square$

Exercice 2: Déterminisez des automates. Faites les exercices des notes de cours de Marc Frappier, Chapitre 4 section exercices, numéro 9.

Exercice 3: Minimisez des automates. Faites les exercices des notes de cours de Marc Frappier, Chapitre 4 section exercices, numéro 10.

Exercice 4: Soit Σ un alphabet et $M \subseteq \Sigma^*$ un ensemble de mots. On dit que M est *régulier* s'il existe un automate fini A tel que $L(A) = M$.

a. Montrez que si M est régulier, alors $\Sigma^* \setminus M$ est régulier.

Solution.

Pour montrer que $\Sigma^* \setminus M$ est régulier, il faut argumenter qu'il existe un AFD ou un AFND dont le langage est $\Sigma^* \setminus M$.

Sachant que M est régulier, alors par définition il existe un AFD $A = \langle Q, \Sigma, \delta, q_0, F\rangle$ tel que $L(A) = M$.

Considérez l'AFD A' tel que $A' = \langle Q, \Sigma, \delta, q_0, Q \backslash F\rangle$. En d'autres termes, A' est identique à A, mais on a inversé les états finaux et non-finaux.

On argumente que $L(A') = \Sigma^* \setminus M$. Soit w un mot de M. Alors sur A, w termine dans un état $q \in F$. Sur A', w termine dans le même état q, mais q n'est pas final dans A'. Donc w n'est pas accepté dans A'. Soit w un mot de $\Sigma^* \setminus M$. Sur A, w termine dans un état $q \notin F$, et sur A', w termine dans le même état q, qui est final sur A'. Donc A' accepte w.

On conclut que les mots acceptés par A' sont précisément les mots de $\Sigma^* \setminus M$.

□

b. Soit M_1, M_2 deux ensembles de mots réguliers, ainsi que l'ensemble

$$M_1{}^\frown M_2 = \{w \mid w = u^\frown v \wedge u \in M_1 \wedge v \in M_2\}$$

où $^\frown$ est le symbole de concaténation. Donc $u^\frown v$ est le mot obtenu en ajoutant v à la fin de u. Par exemple, si $u = allo$ et $v = bonjour$, alors $u^\frown v = allobonjour$.

Montrez que $M_1{}^\frown M_2$ est régulier.

Solution.
Soit A_1 un AFD tel que $L(A_1) = M_1$ et A_2 un AFD tel que $L(A_2) = M_2$. On combine A_1 et A_2 en un AFND A' comme suit.

Les états de A' sont l'union des états de A_1 et A_2, avec les mêmes transitions. L'état initial de A' est l'état initial de A_1. Les états finaux de A' sont les états finaux de A_2. Finalement, pour chaque état final q de A_1, on ajoute une λ-transition de q vers l'état initial de A_2.

On argumente que $L(A') = M_1{}^\frown M_2$. Si $w \in M_1{}^\frown M_2$, on peut diviser w en mots $w_1 w_2$ tel que $w_1 \in M_1$ et $w_2 \in M_2$. En lisant d'abord w_1, notre AFND peut terminer dans un état qui est final de A_1. On peut ensuite prendre la transition λ pour arriver dans l'état initial de A_2, puis lire w_2 et terminer dans un état final de A_2, et donc de A'. Donc, w est bel et bien accepté.

On doit ensuite montrer que si $w \notin M_1{}^\frown M_2$, alors w est rejeté par A'. On montre la contraposée, i.e. si w est accepté, alors $w \notin M_1{}^\frown M_2$. Si w est accepté, il prend une λ-transition à un certain moment, après avoir lu un sous-mot w_1 de w. Puisque ces transition sont accessibles seulement dans un état final de A_1, on sait que $w_1 \in M_1$. Ensuite, on est dans la copie de A_2 dans son état de départ pour lire le reste de w. Puisqu'on finit dans un état acceptant de A', et donc de A_2, on déduit que le reste de w est dans M_2. C'est donc bien la concaténation d'un mot de M_1 et d'un mot de M_2.

□

c. Soit $M_1, M_2 \subseteq \Sigma^*$. Montrez que si M_1 et M_2 sont réguliers, alors $M_1 \cap M_2$ est régulier.

Solution.
Soit $A_1 = \langle Q_1, \Sigma, \delta_1, q_1, F_1\rangle$ et $A_2 = \langle Q_2, \Sigma, \delta_2, q_2, F_2\rangle$ des AFD pour

M_1 et M_2, respectivement. On créé un nouvel automate A dans lequel l'ensemble des états est

$$Q = \{(p, q) \mid p \in Q_1 \land q \in Q_2\}$$

Donc Q représente les combinaisons d'état de A_1 et de A_2. L'idée est que l'on sera dans un état de A_1 et A_2 en "même temps". Être dans l'état (p, q) signifie qu'après avoir lu un certain nombre de symbole d'un mot, on serait dans l'état p si on exécutait sur A_1, et on serait dans l'état q si on était dans A_2. L'état initial est (q_1, q_2). On ajoute une transition $((p, q), a, (p', q'))$ à A si et seulement si $(p, a, p') \in \delta_1$ et $(q, a, q') \in \delta_2$. Le but est que si nous sommes dans p sur A_1 et q sur A_2, sur symbole a on regarde où p mène sur symbole a (disons p') et où q mène sur symbole a (disons q'), et on se retrouve alors dans (p', q'). Finalement, les états finaux sont

$$F = \{(p, q) \mid p \in F_1 \land q \in F_2\}$$

Si un mot w est dans $M_1 \cap M_2$, alors il sera accepté par A_1 et par A_2. Donc sur A, w va terminer dans un état (p, q) tel que $p \in F_1$ et $q \in F_2$, et il sera correctement accepté. Si w n'est pas dans M_1 ou bien n'est pas dans M_2, il sera rejeté par M_1 ou rejeté par M_2 et terminera dans un état (p, q) tel que $p \notin F_1$ ou bien $q \notin F_2$. Dans les deux cas, w ne sera pas accepté. On déduit que l'ensemble des mots acceptés par notre A est bel et bien $M_1 \cap M_2$. □

d. Soit $w \in \Sigma^*$. On écrit $rev(w)$ pour dénoter l'inverse de w, c'est-à-dire le mot w lu de droite à gauche. Par exemple, si $w = allo$, alors $rev(w) = olla$. Pour $M \subseteq \Sigma^*$, on définit

$$rev(M) = \{w \mid w \in \Sigma^* \land rev(w) \in M\}$$

Montrez que si M est régulier, alors $rev(M)$ est régulier.

e. (défi) Soit M un langage. On définit

$$suffixes(M) = \{w \mid w \in \Sigma^* \land \exists u \cdot (u \in \Sigma^* \land concat(u, w) \in M)\}$$

Dit autrement, $suffixes(M)$ contient tous les mots de M, ainsi que tous les suffixes des mots de M.

Montrez que si M est un langage régulier, alors $suffixes(M)$ est un langage régulier.

Idée: considérez un automate A pour M minimisé. Donc chaque état de A est accessible par un mot u, et donc chaque peut être vu comme u "producteur de suffixes".

Exercice 5: Démontrez que la procédure vue en classe pour déterminiser un automate fini non-déterministe ne donne pas nécessairement un automate minimal.

Exercice 6: (défi) Soit M l'ensemble des mots de la forme $a^n b^n$ pour un entier n. Dit autrement, M est contient les mots constitués d'une série de a, suivie d'une série de b, tels qu'il y a autant de a que de b. Par exemple, *aaabbb* ou *ab* sont dans M, mais pas *aaaabbb* ni *aababb*.
Démontrez qu'il n'existe pas d'automate fini dont le langage est M.