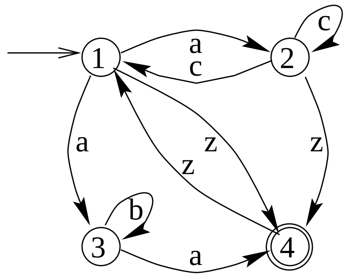
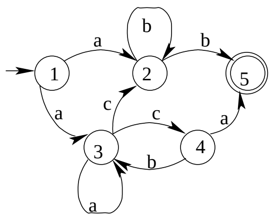
A.1 Automates : algorithmes

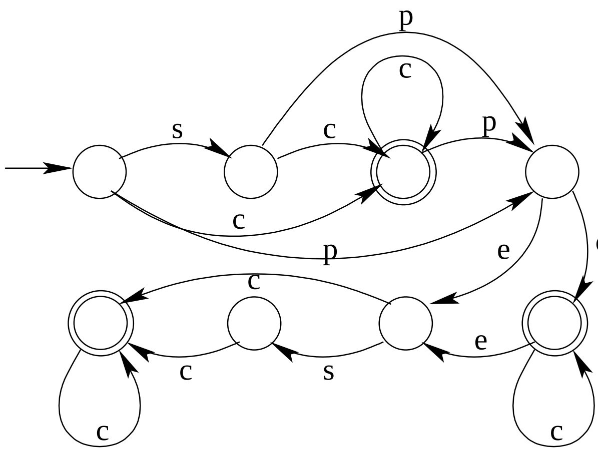


1. Déterminiser l'automate suivant. On donnera la table de transition ainsi que la représentation graphique, en précisant les classes qui forment les nouveaux états. Trouver une expression rationnelle caractérisant ce langage [*difficile*].
2. Soit $A = \{a, b, c\}$. Donner un automate déterministe complet qui reconnaît chacun des langage suivants :
   (a) L'ensemble des mots de longueur paire.
   (b) L'ensemble des mots où le nombre d'occurrences de b est divisible par 3.
   (c) L'ensemble des mots se terminant par b.
   (d) L'ensemble des mots ne se terminant pas par b.
   (e) L'ensemble des mots non vides ne se terminant pas par b.
   (f) L'ensemble des mots contenant au moins un b.
   (g) L'ensemble des mots contenant au plus un b.
   (h) L'ensemble des mots contenant exactement un b.
   (i) L'ensemble des mots ne contenant aucun b.
   (j) L'ensemble des mots contenant au moins un a et dont la première occurence de a n'est pas suivie par un c.
   (k) L'ensemble des mots comportant au moins 3 lettres et dont la troisième lettre à partir de la fin est un a ou un c.
3. Pour chacun des langages rationnels suivants sur l'alphabet $\{0, 1\}$, construire un automate non déterministe le reconnaissant. Déterminiser les automates obtenus, soit *de visu*, soit en utilisant l'algorithme de déterminisation.
   (a) Ensemble des mots dont l'avant-dernière lettre est 0.
   (b) Ensemble des mots contenant le facteur 001.
   (c) Ensemble des mots qui débutent et terminent par la même lettre.
4. Déterminiser l'automate suivant (on demande juste un automate déterministe reconnaissant le même langage) :



« Vérifier » que l'automate déterminisé reconnaît le même langage en testant 3 mots appartenant au langage.

5. Proposer un automate minimal (en nombre d'états) qui reconnaisse le langage décrit par l'expression rationnelle $a^*(c(ab|ba^*)|cab|cb)$. On déduira de l'automate une expression rationnelle plus simple. On ne demande pas nécessairement d'appliquer les algorithmes vus en cours.
6. Soit l'automate donné ici sous forme graphique. Donnez une expression rationnelle qui reconnaît le même langage.



7. Ecrire l'algorithme de parcours d'un automate déterministe mais non complet. On notera `q0` l'état initial, `x[1]...x[`n`]` la chaîne d'entrée, et on supposera que l'on dispose d'une fonction booléenne `existe_t`, à deux arguments `q` (état) et `c` (caractère) qui renvoie *true* si la fonction δ a une valeur pour (`q`,`c`). On dispose aussi d'une fonction `delta`, avec les mêmes arguments, qui retourne un état pour les valeurs pour laquelle elle est définie.
8. Proposer un algorithme de déterminisation d'un automate (sans ε-transition, mais pas nécessairement complet). On fera les hypothèses suivantes :
   – Le type `état` est défini
   – On dispose des types et primitives nécessaires pour manipuler (1) des listes, (2) des ensembles[1].
   – Un automate est déterminé par les fonctions suivantes :
     – `terminal(état q)` : renvoie `vrai` si `q` est un état terminal
     – `initial()` : renvoie l'état initial
     – `delta(état q, lettre x)` : renvoie un ensemble d'états

[1]En particulier, on suppose que l'on dispose d'une instruction qui permet de parcourir un ensemble : quelque chose comme `Pour tout x appartenant à X faire`...