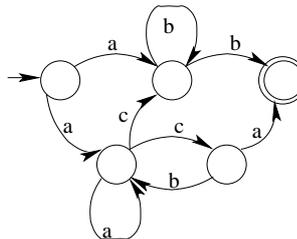
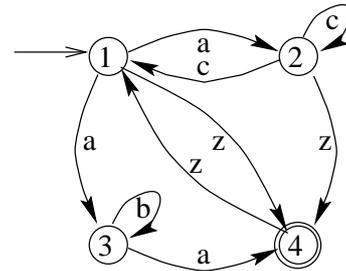


### A.1 Automates : algorithmes

- Déterminiser l'automate suivant. On donnera la table de transition ainsi que la représentation graphique, en précisant les classes qui forment les nouveaux états.
- Proposer un automate **minimal** qui reconnaisse le même langage que l'expression rationnelle  $a(b|bc)^*c$ .
- 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.

- Ecrire l'algorithme de parcours d'un automate déterministe mais non complet. On notera  $q_0$  l'état initial,  $x[1] \dots x[n]$  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  $\delta$  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.
- Proposer un algorithme de déterminisation d'un automate sans  $\epsilon$ -transition (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<sup>1</sup>.
  - Un automate est déterminisé 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
- Proposer une grammaire régulière qui engendre le même langage que l'automate suivant :

	a	b	c	$\epsilon$
$\rightarrow$ 1	1,2	3	5	5
2	3	2	1	
$\leftarrow$ 3			5	4,6
4	3		6	2
5	5	4,6	6	
$\leftarrow$ 6				

- Proposer un automate sans  $\epsilon$ -transition qui reconnaît le même langage que l'automate proposé à l'exercice n° 2 (on demande la table de transition).
- 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.

<sup>1</sup>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...`