

Introduction à la programmation et à la modélisation en Java

Automates

Partie 1 : Théorie

I. Exemple introductif

On considère le codage suivant :

- $a \rightarrow 0$
- $b \rightarrow 10$
- $c \rightarrow 110$
- $d \rightarrow 111$

On peut définir un certain nombre de situations et comportements dans le décodage :

- J'hésite entre dire a, b, c ou d. Dans ce cas, si je reçois 0, je peux dire a et hésiter à nouveau entre a, b, c ou d; sinon, c'est que j'ai reçu 1 et j'hésite maintenant entre dire b, c ou d.
- J'hésite entre dire b, c ou d. Dans ce cas, si je reçois 0, je peux dire b et hésiter à nouveau entre a, b, c ou d; sinon, c'est que j'ai reçu un 2ième 1 et j'hésite maintenant entre dire c ou d.
- J'hésite entre dire c ou d. Dans ce cas, si je reçois 0, je peux dire c et si je reçois 1 dire d, puis, dans les 2 cas, je vais hésiter à nouveau entre a, b, c ou d.

On observe que le dispositif de codage des caractères en entrée $\{0,1\}$, produit éventuellement des sorties $\{a,b,c,d\}$, possède des états $\{1,2,3\}$, dont un initial 1 et un satisfaisant pour une fin d'entrée 1 également.

Son fonctionnement est régi par des règles de transition.

Visualisons un automate correspondant à la situation précédente.

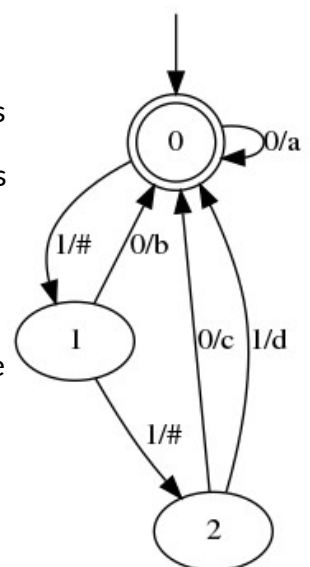
II. Définition d'un AEF

Les automates sont utiles à représenter / modéliser toutes sortes de systèmes météo, transport, population, jeux, etc.

La littérature désigne parfois par machine séquentielle les automates à états finis produisant une sortie.

Une telle machine séquentielle A est un sextuplet (V,O,E,I,F,T) , où :

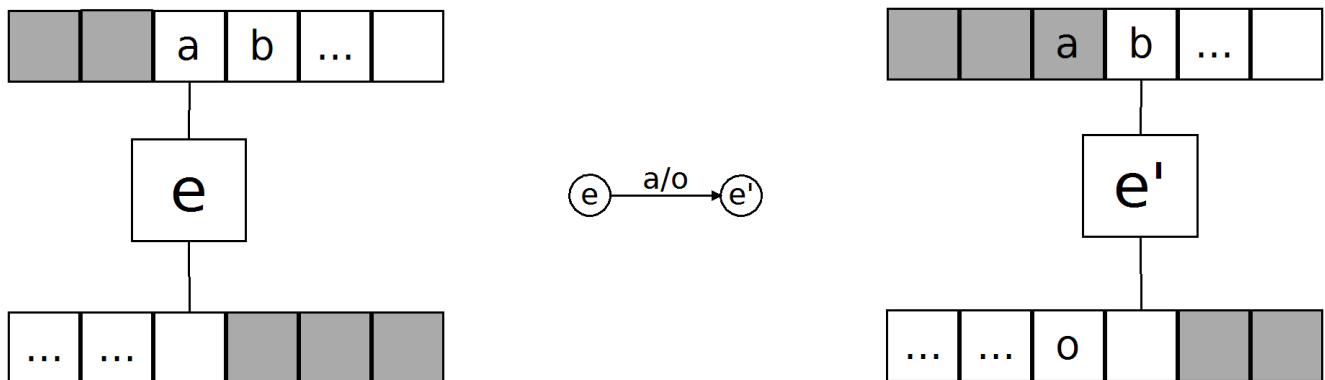
- $V = \{0,1, \dots\}$: alphabet d'entrée, est un ensemble fini non vide de symboles ou lettres
- O : alphabet de sortie, est un ensemble de symboles
- $E = \{e_1, e_2, \dots, e_s\}$: ensemble des états. E est fini, non vide.
- I : ensemble des états (partie de E) initiaux
- F : ensemble des états (partie de E) d'acceptation
- T : ensemble de quadruplets de $E \times V \times E \times O$, ensemble des transitions de $\backslash A$.



Dans la représentation graphique précédente :

- Le vocabulaire d'entrée est $\{0,1\}$.
- Les symboles de sortie sont : a,b,c,d.
- Il y a trois états : 0,1,2.
- Il y a un seul état initial 0, que l'on repère par la flèche entrante n'ayant rien à l'origine.
- Il y a un seul état acceptant, également 0, que l'on repère au double cercle.
- Les transitions sont symbolisées par des flèches orientées

III. Fonctionnement d'un AEF



Partie 2 : Programmation

Exemple : Automate.java Evolution d'un personnage de jeu

- Bonne santé, blessure, maladie, mort
- lister les états, les transitions, dessiner le graphe de l'automate
- Table de transition :

Etat/Transition	Combat Perdu ('c')	Attaque par un Virus ('v')	Prise d'une Potion ('p')
Bonne santé (0)	Blessé	Malade	Bonne santé
Blessé (1)	Mort	Mort	Bonne santé
Malade (2)	Mort	Malade	Bonne santé
Mort (3)	Mort	Mort	Mort

Pour créer un programme qui simule l'évolution d'un automate, il faut écrire

- une fonction de transition, qui reçoit l'état actuel et la transition, et qui renvoie le nouvel état du système.
- Un programme de simulation