

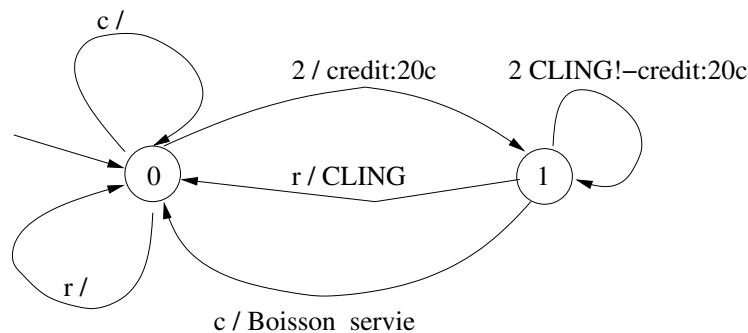
Automates

Les automates d'états finis sont des modèles permettant (entre autres) de simuler des systèmes qui interagissent avec leur environnement. Par exemple des systèmes de contrôle (d'un arroseur de plans de tomates, d'un distributeur, d'un pilote automatique ...).

- Le système reçoit des *entrées* (signal d'un capteur, action d'un utilisateur, données d'un flux ...)
- Il émet des *sorties* (déclenchement d'un arrosage, service d'une boisson, affichage d'un message pour l'utilisateur, calcul sur les données ...).

Il est défini par son *état* caractérisé par la séquence des entrées reçues depuis son initialisation. Dans cet état, il peut recevoir une entrée, et en fonction de celles-ci, émettre une sortie et changer d'état.

Exemple : Un distributeur de boisson.



On appelle automate d'états finis de Mealy un quintuplet $A = (Q, V, S, \delta, \sigma, q_0, F)$ dans lequel :

- Q est un ensemble fini d'états ;
- V est un ensemble fini de symboles (le vocabulaire d'entrée) ;
- S est un ensemble fini de symboles (le vocabulaire de sortie) ;
- $\delta \subseteq Q \times V \rightarrow Q$ est la fonction de transition ;
- $\sigma \subseteq Q \times V \rightarrow S$ est la fonction de sortie ;
- $q_0 \in Q$ est l'état initial ;
- $F \subseteq Q$ est l'ensemble des états terminaux (parfois vide).

Simuler le fonctionnement d'un automate

```

étatCourant <- étatInitial
tant que pas fini faire
  lire une entrée
  calculer étatSuivant en fonction de étatCourant et de entrée
  calculer la sortie en fonction de étatCourant et de entrée, et la traiter
  étatCourant <- étatSuivant
mettre à jour fini
  
```

Les exercices suivants portent tous sur le traitement de séquences de caractères (reconnaissance, transformation). Dans un premier temps, la séquence d'entrée sera lue depuis le clavier et la séquence de sortie sera écrite à l'écran. Le programme qui simule l'automate de la machine à café vous est donné sous deux formes :

- une première forme qui implémente les fonctions de transition et de sortie par des méthodes java :
`ExempleMachineCafe.java`;
- une seconde forme où les fonctions sont représentées sous forme de tableaux :
`ExempleMachineCafeTableau.java`.

Ces deux formes font exactement la même chose mais de manière différente.

1 Améliorer la machine à café

Modifiez l'automate de la machine à café afin d'étendre ses fonctionnalités, vous pourrez par exemple :

- lui faire accepter les pièces de 10c
- délivrer au choix du café ou du thé
- gérer des prix de boisson différents, par exemple 20c pour le café et 30c pour le thé
- ...

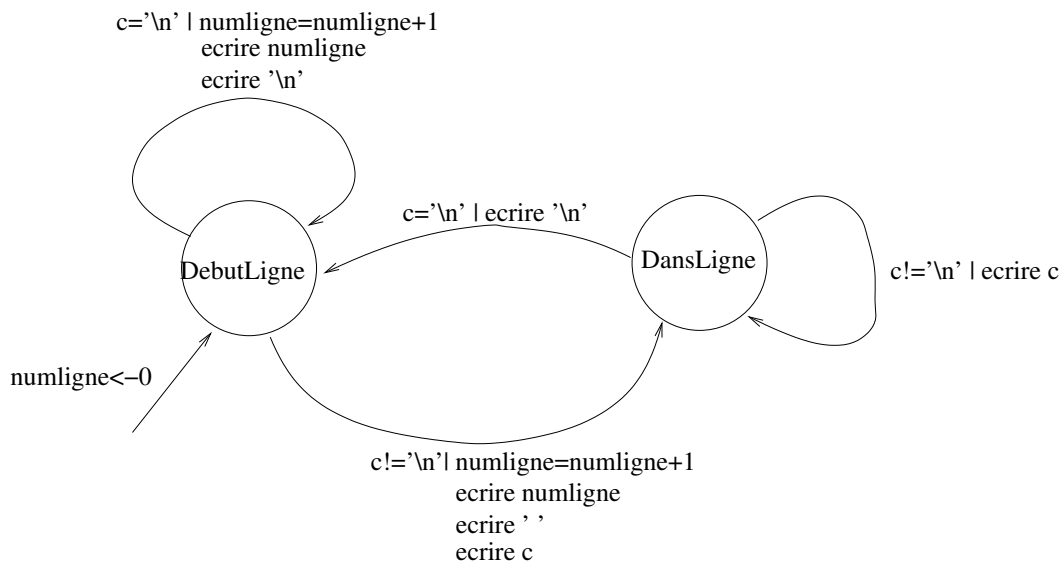
Commencez par dessiner l'automate modifié, puis implémentez votre automate modifié sur les deux variantes de programme.

2 Numéroté les lignes d'un fichier

Dans cette partie nous vous fournissons un programme, `ExempleCopie.java` qui lit un fichier dont le nom lui est donné en argument et qui affiche son contenu sur la sortie standard. Pour utiliser ce programme, il faut passer par la ligne de commande : d'une part parce que le programme va aller chercher le fichier dans le répertoire courant, d'autre part pour passer l'argument au programme de manière simple. Pour ce faire, il faut prévoir deux choses :

- permettre à java de trouver les fichiers compilés par Eclipse :
`export CLASSPATH=$HOME/eclipse-workspace/DU\ ISN\ -\ Automates/bin`
En remplaçant le chemin ci-dessus par l'emplacement du répertoire `bin` de votre projet Eclipse
- exécuter le programme en lui passant un argument :
`java ExempleCopie Mon_fichier`

Écrire un autre programme qui lit un fichier et, à l'aide d'un automate, le ré-affiche à l'écran en précédant chaque ligne de son numéro. L'automate réalisant cette tâche pourra être le suivant :



Par exemple, si le fichier lu contient :

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Pellentesque sapien sapien, vehicula non adipiscing et,  
pulvinar id diam.  
Aliquam erat volutpat.
```

```
Vestibulum metus quam, volutpat et suscipit ac,  
congue et lacus. Proin scelerisque nisi ut consectetur tincidunt.
```

Alors le programme devra afficher :

1. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2. Pellentesque sapien sapien, vehicula non adipiscing et,
3. pulvinar id diam.
4. Aliquam erat volutpat.
- 5.
6. Vestibulum metus quam, volutpat et suscipit ac,
7. congue et lacus. Proin scelerisque nisi ut consectetur tincidunt.
- 8.

3 Suppression des commentaires dans un programme Java

Ecrire un programme qui lit un fichier contenant un programme Java et, à l'aide d'un automate, le ré-affiche à l'écran sans les commentaires. On considèrera deux types de commentaires :

1. les commentaires sur une seule ligne, débutant par `//` et se terminant par une fin-de-ligne;
2. les commentaires multi-lignes, débutant par `/*` et se terminant par `*/`.

4 Digicode

Dans cette partie, nous souhaitons écrire un programme qui, à l'aide d'un automate, reconnaît un code de 5 chiffres. Le programme devra accepter des chiffres en entrée et devra afficher un joli message quand le bon code est saisi par l'utilisateur.

88591 : L'automate devra reconnaître le code 88591. Si l'utilisateur se trompe pour l'un des chiffres le programme devra afficher un message d'erreur et se terminer.

Un code paramétrable : L'automate devra reconnaître le code présent dans un fichier dont le nom est passé en argument. Ce code sert à construire l'automate qui servira à le reconnaître. Pour cela, nous vous conseillons d'utiliser la forme tabulée (qui utilise des tableaux pour représenter les fonctions de l'automate). Si l'utilisateur se trompe le programme ne devra pas s'arrêter. Le code devra être reconnu au moment où le programme identifiera dans les chiffres en entrée la bonne séquence de 5 chiffres. Essayez avec plusieurs codes, par exemple :

- 12345
- 12123
- ...