

TP : travail à réaliser sur 4 séances (4x3h)

Objectif : Créer un projet Java permettant d'encoder des automates à états finis déterministes et l'appliquer à un cas concret.

Détail du travail demandé

1. Créer les classes permettant d'encoder des automates à états finis déterministes.

Un automate est constitué d'un ensemble d'états. Les automates à implémenter comporteront un seul état initial, un ou plusieurs états finaux, et un ensemble de transitions. Chaque transition possède un état origine, un état final et un symbole étiquetant la transition. L'alphabet sera constitué de caractères uniquement.

Les méthodes à développer (en plus des méthodes usuelles : constructeurs, getters et setters, ...) :

- `boolean appartient(String mot)` : méthode de la classe Automate. Renvoie *true* si l'automate reconnaît le mot en entrée de la méthode, *false* sinon. Dans le cas où le mot n'est pas reconnu, la méthode affiche la raison de l'échec de la reconnaissance (symbole n'appartenant pas à l'alphabet de l'automate ; absence de transition à partir de l'état courant avec le symbole lu ; fin du mot avant d'atteindre un état final)
- `Automate(String nomDeFichier)` : un constructeur de la classe Automate à partir de données lues dans un fichier texte. Un exemple est fourni sur Moodle (fichier "automate1.txt").

2. Appliquer ce code à la simulation d'un automate de gestion d'un distributeur de café très simple :

- l'appareil ne distribue que deux types de boissons : un café court (0,30€) et un café long (0,50€). Chaque boisson se commande par un bouton spécifique ;
- il n'accepte que des pièces de 10c, 20c et 50c ;
- dès qu'il reçoit plus que 50c, il rend la monnaie dans le bac de récupération de la monnaie (on supposera que le distributeur dispose toujours de la monnaie suffisante) ;
- l'appareil ne comporte pas de bouton d'annulation (pour rendre l'argent) ;
- après avoir inséré les pièces, le client appuie sur un des deux boutons pour indiquer quel type de café il veut (court ou long).
- si un paiement suffisant a été fait, le café est délivré, sinon la machine attend d'autres pièces et "oublie" la commande ;
- une fois le café délivré, la machine attend le prochain client.

Il y a un seul état final, correspondant à une distribution réussie.

Pour les transitions, les symboles suivants sont imposés :

- les symboles 1, 2, 5 correspondent respectivement à l'introduction de 10c, 20c, 50c par le client,
- les symboles S, L (majuscules) correspondent à l'appui sur les boutons café court et café long,
- les symboles s et l (minuscules) correspondent à la distribution d'un café court et d'un café long,
- les symboles d,v,c correspondent au rendu d'une pièce de 10c, 20c ou 50c,

La simulation doit permettre de vérifier que le comportement du distributeur, c'est-à-dire qu'il n'accepte que les séquences correctes d'actions.

Par exemple :

- la séquence représentée par le mot 21Ss doit être acceptée. Elle correspond aux actions suivantes : le client introduit 20c puis 10c, puis appuie sur le bouton café court, qui est distribué ;
- la séquence 5Svs doit être acceptée. Elle correspond aux actions suivantes : le client introduit une pièce de 50c, appuie ensuite sur le bouton café court, la machine lui rend 20c et distribue un café court.

- la séquence L22L1Ll doit être acceptée. Elle correspond aux actions suivantes : le client presse le bouton café long, puis introduit 20c, puis 20c, puis ré-appuie sur le bouton café long, puis remet 10c, ré-appuie sur café long, qui est distribué ;
- 2Ss, 5S5s et d51l sont des exemples de séquences qui ne doivent pas être acceptées.

Le programme doit renvoyer un message informatif : la séquence d'actions a été acceptée ou n'a pas été acceptée, et dans ce dernier cas le symbole de la séquence qui a déclenché un échec.

Le programme doit être testé à partir d'un ensemble de séquences (suites de symboles) lus à partir d'un ou plusieurs fichiers de test.

Production demandée, à déposer sur Moodle à la fin de la dernière séance dans la zone de dépôt de votre groupe :

1. un projet Java répondant au problème ;
2. le fichier de construction de l'automate et les fichiers de test utilisés ;
3. un compte-rendu écrit (10 pages maxi) comprenant :
 - une représentation graphique de l'automate représentant le comportement du distributeur ;
 - des explications sur les erreurs détectées et les tests correspondants qui ont été réalisés.