Thibaud LAURENS

Rapport du projet Automates

Introduction :

Ce travail a été réalisé par Thibaud LAURENS, étudiant en 2e année de BUT Informatique, dans le cadre de ses études.

Le programme a pour but de permettre aux utilisateurs d'instancier différents automates accepteurs et de vérifier, pour chacun d'eux, si une chaîne de caractères est reconnue (valide) ou non.

La dernière modification en date du programme remonte au 14 février 2026.

Définitions :

Un **automate** est un modèle mathématique utilisé pour reconnaître ou générer des chaînes de caractères appartenant à un langage formel. Il est constitué d'un ensemble d'états, de symboles et de transitions pouvant se représenter par un graphe orienté valué :
- Les états de l'automate sont les nœuds du graphe.
- Les transitions de l'automate sont les arcs du graphe, permettant de passer d'un état à un autre.
- Les symboles de l'automate sont les valeurs des arcs du graphe, indiquant quel symbole l'automate doit rencontrer pour passer d'un état à un autre.

Un **automate accepteur** est un type d'automate spécialisé dans la reconnaissance des chaînes de caractères (symboles) : il décide si la chaîne est acceptée ou rejetée en se basant sur son alphabet, ses états et ses transitions.
L'automate accepteur commence dans un état initial. A chaque symbole lu, il vérifie si le symbole permet de passer de l'état courant à un autre ou non. Si oui il change d'état, si non la chaîne est rejetée. L'automate doit finir la lecture de la chaîne de caractères sur un état final pour accepter la chaîne.

Un **automate déterministe** n'a qu'un seul état initial, ne contient pas de transition sans symbole et n'autorise pas l'état courant à disposer de plusieurs transitions avec le même symbole. Les accepteurs de ce projet sont déterministes

Réalisations :

Le code du projet se réparti dans trois fichiers :

- ***State.kt*** contient une classe définissant les informations relatives à un état : son nom et ses voisins (quels symboles sont autorisés pour quitter l'état et vers quel état suivant cela mène). La classe dispose de méthodes pour ajouter un couple (symbole, état) dans ses voisins (on parle d'ajouter une transition), vérifier qu'une transition est autorisée par l'état courant et retourner l'état suivant, réinitialiser l'ensemble de ses informations et se comparer à un autre état en se basant sur les noms.

- ***Automate.kt*** contient la classe définissant les données d'un automate accepteur déterministe : un nom, la liste de ses états, son état initial, son / ses état(s) final / finaux, la liste des symboles autorisés dans son alphabet. Il n'a pas besoin de recenser ses transitions, considérant que les états le font déjà. Ses nombreuses permettent de : ajouter un ou plusieurs états, ajouter des symboles à l'alphabet, définir l'état initial (si existe déjà dans la liste de ses états), définir un ou plusieurs états finaux (si existent déjà dans la liste des états), ajouter des transitions (si les états et symboles impliqués sont reconnus), réinitialiser les éléments précédents, écrire l'ensemble de ses informations dans un fichier out.dot, remplir ses éléments à partir d'un fichier texte, et évaluer si une chaîne est reconnue ou non.

- ***Main.kt*** est le fichier démarrant le programme. Il définit les interactions avec l'utilisateur, instancie les automates, récupère les chaînes à vérifier.

Les automates construits sont les suivants :

- **Smileys** (exercice 3 du cahier d'exercices) : reconnaît les smileys ':)', ':(', ';)' et ';-)'.

- **Entiers pairs** (exercice 4) : reconnaît si un entier naturel écrit en binaire est pair ou non. Par exemple, 010111010100100 est reconnu et 0101101 est rejeté.

- **HH:MM** (exercice 5) : reconnaît les heures de la journée allant 00:00 à 23:59.

- **Entiers multiples de 3** (exercice 6) : reconnaît si un nombre entier naturel est un multiple de 3 ou non. Par exemple, 2664 est reconnu et 2665 est rejeté.

- **Adresses électroniques** (exercice 8) : reconnaît si une adresse mail est valide ou non. Par exemple, thibaud.laurens@etu.univ-nantes.fr est reconnu, tout comme BLAbla@gmail.com, mais not.email@@zone.tld est rejeté.

- **Polynômes carrés** (exercice 9) : reconnaît les équations de la forme aX**2+bX+c, où a,b et c sont des entiers naturels entre 1 et 25 compris et que les + peuvent être substitués par des -. Par exemple, 0X**2-25X+9 est reconnu mais 0X*2-26X+9X ne l'est pas pour trois raisons : il manque un '*', 26 n'est pas un nombre autorisé et il ne doit pas y avoir de 'X' à la fin.

- **Instructions de programme** (exercice 10) : reconnaît les expressions simple de la forme (x+x+x+x+…), où les ‘x’ peuvent être substitués par ‘y’ et les ‘+’ par des ‘-’ ou ‘/’. Par exemple, ‘(y/x+y-x)’ est reconnu mais pas ‘(x++x)’.

Pour aller plus loin :

Permettre aux utilisateurs de passer en entrée un fichier bien formater pour instancier leur propre automate.