

Projet PO2 (2017 – 2018)

Simulation de la Machine de Turing

But du projet : développer une interface graphique en utilisant MVC permettant de simuler la machine de Turing.

1. Sujet du projet

(1) Machine de Turing

Dans un article qui fera date, « On Computable Numbers with an Application to the Entscheidungsproblem » publié en 1936, Alan Turing (1912-1954), jeune mathématicien britannique, jette les bases de ce qui deviendra la théorie de la calculabilité en répondant par la négative à la question de savoir si toute proposition peut être démontrée dans un système formel. Pour ce faire, il imagine une machine à calcul universelle qui est appelé la Machine de Turing.

La machine de Turing est composé de :

- un ruban divisé en cases consécutives. Chaque case contient un symbole parmi un alphabet fini ;
- une tête de lecture/écriture qui peut lire et écrire les symboles sur le ruban, et se déplacer vers la gauche ou vers la droite du ruban ;
- un registre d'état qui mémorise l'état courant de la machine de Turing
- une table de règles sous forme (étatCourant, symboleLu)-> (étatSuivant, symboleEcrit, direction), qui indique à la machine en fonction de son état courant et du symbole lu sur le ruban, quel est son état suivant, quel symbole écrire sur le ruban, comment déplacer la tête de lecture (à droite, à gauche, pas déplacement).

Le symboleLu peut être vide : c'est le cas où il n'y a pas de symbole (représenté par un caractère vide) dans la case actuelle sous la tête de lecture/écriture.

De même, symboleEcrit peut être vide pour indiquer que la tête de lecture/écriture écrit un caractère vide.

La machine commence par le premier symbole d'une chaîne de caractères sur le ruban et avec l'état courant 0. Quand la machine ne trouve pas une règle correspondant au état courant et le symbole lu, la machine s'arrête.

(2) Exemple : Additionner un nombre de 1

Entrée : un nombre sous forme binaire

Sortie : additionner ce nombre de 1

Voici la table de règles:

$(0,) \Rightarrow (1, <)$

$(0,0) \Rightarrow (0,0,>)$

$(0,1) \Rightarrow (0,1,>)$
 $(1,) \Rightarrow (2,1,)$
 $(1,0) \Rightarrow (2,1,)$
 $(1,1) \Rightarrow (1,0,<)$

L'état 0 représente le départ, la tête de lecture/écriture déplace vers le droite jusqu'à la fin du nombre, et change l'état à 1. A l'état 1, si la tête de lecture/écriture rencontre 1, elle écrit 0 et déplace vers le gauche; si elle rencontre 0 ou le caractère blanc ' ', elle écrit 1 et change l'état à 2. L'état 2 représente l'arrêt d'exécution.

On peut dérouler pour le nombre 1011 sous forme binaire.

```

ruban initial  1011
               règle appliquée : (0,1) ⇒ (0,1,>)
ruban  1011
               règle appliquée : (0,0) ⇒ (0,0,>)
ruban  1011
               règle appliquée : (0,1) ⇒ (0,1,>)
ruban  1011
               règle appliquée : (0,1) ⇒ (0,1,>)
ruban  1011
               règle appliquée : (0, ) ⇒ (1, ,<)
ruban  1011
               règle appliquée : (1,1) ⇒ (1,0,<)
ruban  1010
               règle appliquée : (1,1) ⇒ (1,0,<)
ruban  1000
               règle appliquée : (1,0) ⇒ (2,1, )
ruban  1100
ruban final   1100

```

2. Interface avec MVC

Réaliser une interface en utilisant MVC. L'interface ci-après peut servir comme une base de départ. Mais nous vous encourageons à concevoir votre interface.

Vous devez obligatoirement :

- Utiliser le modèle MVC de votre réalisation. La réalisation sans MVC ne sera pas pris en compte.
- Proposer la visualisation du ruban et de la position de la tête de lecture/écriture
- Proposer l'édition de la table de règles ainsi que sa sauvegarde/restauration

Vous pouvez proposer d'autres options comme l'historique.

The interface is divided into three main sections: REGLES, EXECUTION, and RUBAN.

- REGLES:** Contains buttons for 'charger', 'sauvegarder', and 'ajouter'. Below these is a list of rules:
 - $(0,) \Rightarrow (1, , <)$
 - $(0,0) \Rightarrow (0,0,>)$
 - $(0,1) \Rightarrow (0,1,>)$
 - $(1,) \Rightarrow (2,1,)$
 - $(1,0) \Rightarrow (2,1,)$
- EXECUTION:** Contains controls for the simulation:
 - 'ruban initial' field with value '1011'.
 - 'initialiser' and 'faire un pas' buttons.
 - 'lancer' and 'stopper' buttons.
 - 'vitesse' slider with a play button icon and a '+' sign.
- RUBAN:** A visual representation of the tape as a grid of cells. The current state shows the sequence '1 1 0 0' with a head position '0' below it.
- HISTORIQUE:** A log of the simulation steps, showing the state of the tape and the rule applied at each step, matching the text in the first block of the document.

3. Organisation du projet

Le projet est à effectuer en binôme. Une séance de soutenance des projets aura lieu durant la semaine d'examen en Janvier (la date exacte sera précisée ultérieurement en fonction du planning des examens). Les soutenances seront individuelles et de durée variable : elles permettront de vérifier que chaque étudiant est bien auteur de son projet. Une mauvaise soutenance peut conduire à une note inférieure à 10/20.

Un rapport imprimé devra être rendu au moment de la soutenance. Ce rapport doit détailler le modèle (MVC) choisi. Le rapport ne doit pas comporter de code.

Vous enverrez, la veille de la soutenance, par mail, le zip du code source.