Université de Versailles – Saint-Quentin-en-Yvelines 2022-2023
Licence 3 Informatique

Projet – Simulation de Machines de Turing

Version du 29 Décembre 2022

Pierre Coucheney – pierre.coucheney@ens.uvsq.fr
Franck Quessette – Franck.Quessette@uvsq.fr
Yann Strozecki – Yann.Strozecki@uvsq.fr
Sandrine Vial – Sandrine.Vial@uvsq.fr

Le but de ce projet est d'exécuter pas à pas des machines de Turing.

1 Simulation de l'exécution d'une machine de Turing

Le code du projet est à faire en Python. Vous devez automatiser votre travail (par exemple avec make, ou un script ou en le codant directement en Python) afin que l'utilisateur puisse en une ligne de commande lancer le code qui s'exécute pour répondre à chaque question. Par ailleurs, on doit pouvoir aussi lancer les tests de chaque question séparément en une ligne de commande.

On utilisera le langage de description de machine du site `https://turingmachinesimulator.com/`. Vous pouvez restreindre ou étendre ce langage pour vous simplifier le travail, par exemple interdire des espaces ou des sauts de ligne. Dans ce cas, il faut spécifier ces modifications avec un commentaire dans votre code.

Pour simplifier, on considérera uniquement des machines qui effectuent un calcul, avec un seul état final qui dénote la fin du calcul. Les états seront notés par des des chaînes de caractères, I sera l'état initial et F l'état final. L'alphabet d'entrée est $\{0, 1\}$, l'alphabet de travail $\{0, 1, \square\}$ et les machines ont k bandes.

Question 1 : Proposer une structure de données **MT** (une class en Python) qui permet de représenter une machine de Turing ainsi que sa configuration (état courant, état des bandes, position des têtes de lecture). Écrire une fonction qui initialise une instance de la structure **MT** à partir d'un mot d'entrée et d'un fichier contenant une description d'une machine de Turing.

Question 2 : Écrire une fonction qui étant donnée une machine de Turing, lui fait exécuter un pas de calcul.

Question 3 : Écrire une fonction qui prend comme argument un mot et une machine de Turing et qui simule le calcul de la machine sur le mot jusqu'à atteindre l'état final. Bravo, vous avez réalisé une machine universelle.

Question 4 : Modifier la fonction précédente pour que, à chaque pas de simulation, la configuration de la machine s'affiche de manière compréhensible (soit graphiquement, soit sur le terminal).

Question 5 : Donner le code des machines de Turing suivantes, chacun dans un fichier distinct :

- Positionner la tête de lecture sur la lettre la plus à droite de la bande numéro i : $LEFT_i$
- Positionner la tête de lecture sur le premier caractère $a \in \Sigma$ sur la bande i : $SEARCH_{i,a}$
- Effacer la bande numéro i : $ERASE_i$

— Copier la bande i sur la bande j : $COPY_{i,j}$

Exécutez ces machines de Turing sur des exemples.

2 Machine qui appelle

Question 6 : On considère maintenant le modèle de machine de Turing étendu par des instructions d'appel de machine de Turing (comme vu en cours). On rappelle que si une machine M a la transition spéciale (q, a, M', q'), alors dans l'état q et à la lecture de a elle lance le calcul de la machine M'. Quand M' a fini son calcul, alors M passe dans l'état q'.

On a en entrée le code de deux machines de Turing M_1 et M_2 tel que la machine M_1 fait des appels à M_2. Votre fonction doit produire une machine M_3, sans appel à M_2 qui réalise le même calcul que M_1. Bravo, vous venez de réaliser un linker.

3 Machine plus compliquée

Le binôme doit se partager les deux questions : chaque membre doit programmer seul une des deux machines de Turing. Essayez d'utiliser au moins un appel d'une machine de Turing simple donnée dans les questions précédentes ou que vous fournirez vous même. Exécutez ces machines de Turing sur des exemples.

Question 7 : Multiplier deux nombres en binaire x et y en utilisant la méthode égyptienne.

On initalise z (le résultat) à 0 et tant que x est supérieur strict à 0 on fait :

— si x est pair, $x = x/2$ et $y = y * 2$

— si x est impair, $x = x/2$, $z = z + y$ et $y = y * 2$

Vous devrez appeler la machine réalisant une addition.

Question 8 : Étant donnée une entrée de la forme 01#00#11#00#10 qui représente un tableau avec un nombre quelconque d'entiers sur deux bits, donner une machine de Turing qui trie ce tableau.

4 Bonus : Optimisation de machine de Turing

C'est une partie bonus, plus difficile. Vous pouvez la faire uniquement si vous avez fait tout le reste.

On veut optimiser le code des machines de Turing afin d'obtenir des machines plus petites et plus rapides. On propose deux types d'optimisation, mais vous pouvez aussi proposer les votres.

Question 9 : (Simplification) Écrire une fonction qui prend en entrée une machine de Turing et renvoie une machine équivalente dont toutes les instructions qui n'impliquent pas de déplacement ont été remplacées. Par exemple, si on a les transitions $A, 0 \to B, 0, -$ et $B, 0 \to C, 1, >$, on peut remplacer $A, 0 \to B, 0, -$ par $A, 0 \to C, 1, >$.

Question 10 : (Élimination du code mort) Écrire une fonction qui prend en entrée une machine de Turing et renvoie une machine équivalente dans laquelle les transitions qui ne sont jamais utilisées lors de l'exécution ont été éliminées.

5 Modalités pratiques

Merci de respecter les consignes suivantes :

— le projet est à faire en binôme ;

— chaque membre du binôme doit faire une des questions de la troisième partie, différente de celle faite par l'autre membre !

— le projet est à rendre sur ecampus, au plus tard le mardi 3 janvier à 23h59 ;

— une soutenance aura lieu le jeudi 5 janvier matin.

— vous devez déposer un fichier `XY_NOM1_Prenom1-NOM2_Prenom2.zip` qui est le zip du dossier `XY_NOM1_Prenom1-NOM2_Prenom2` contenant votre projet.
`XY` sont les initiales de votre chargé de TD (`SV`, `PC`, `FQ`, `YS`). Pour être évalué, votre travail doit s'exécuter quand on tape make dans le terminal. Si le nom de fichier remis sur ecampus ne respecte pas ces règles, le projet ne sera pas évalué.

— Un outil de détection de plagiat sera utilisé sur vos codes.

Le retard de la remise du projet entraîne 1 point de moins par heure de retard.