

## Projet – Métasimulation

Version du 28 Février 2024

Le but de ce projet est de simuler des RAMs, d'écrire une simulation d'automates à pile tournant sur RAM et de proposer des méthodes d'optimisation des RAMs.

### 1 Simulation de l'exécution d'une machine RAM

Le code du projet est à faire en Python ou en C. 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 toutes les question. Par ailleurs, on doit pouvoir aussi lancer les tests de chaque question séparément le plus simplement et rapidement possible. Vous donnerez un fichier README qui explique comment utiliser votre code.

On utilisera le langage des machines RAM vu en cours. Vous pouvez restreindre ou étendre ce langage pour vous simplifier le travail. Dans ce cas, il faut spécifier très clairement ces modifications dans le README. On rappelle que les entrées d'une machine RAM sont contenus dans des registres spéciaux  $i_0, \dots, i_k$ ,  $i_0$  contient  $k$ , les registres  $i_0, \dots, i_k$  représentent l'entrée et tous les autres registres de la machine sont à zéro.

Un programme RAM sera stocké dans un fichier texte, une ligne par instruction.

**Question 1 :** Proposer une structure de données qui permet de représenter le programme d'une RAM. Écrire une fonction qui lit un fichier texte contenant le code d'une machine RAM et un mot d'entrée et qui initialise la structure de données pour représenter cette machine. Vous pourrez éventuellement utiliser lex et yacc pour cette question.

**Question 2 :** Proposer une structure de données pour représenter une configuration d'une Machine RAM. Donner une fonction qui prend en argument une machine RAM et une configuration et qui donne la configuration obtenue après un pas de calcul de la machine.

**Question 3 :** Écrire une fonction qui prend comme argument un mot et une machine RAM 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 RAM suivantes :

- Avec en entrée deux entiers  $a$  et  $b$ , calculer  $a^b$
- Avec comme entrée un tableau d'entier, écrire le tableau trié dans la sortie (par un tri à bulle)

Exécutez ces machines sur des exemples à l'aide de votre simulateur.

## 2 Comment faire rentrer la pile dans la RAM ?

On veut simuler n'importe quel automate à pile déterministe avec une machine RAM. On va coder les automates à pile de la manière suivante pour pouvoir les stocker dans une RAM. Les éléments des alphabets d'entrée, de pile et les états seront représentés par des entiers. On suppose que l'alphabet d'entrée est  $\{0,1\}$ , l'alphabet de pile  $\{0,1,2\}$  avec 0 le symbole de fond de pile. L'état initial est noté 0 et l'état final 1. On représente  $\varepsilon$  par 2. On considère uniquement des automates acceptant par état final.

On stocke un automate dans la RAM de la manière suivante, en stockant d'abord le nombre de transitions, puis dans l'ordre toutes les transitions. Une transition est un quintuplet  $(q, a, A, w, q')$  dont tous les membres se codent par des entiers sauf  $w$  qui est un mot de l'alphabet de pile. Pour coder  $w$ , on stocke dans un premier registre le nombre de lettres de ce mot, puis dans chacun des registres suivant, les lettres dans l'ordre. Ainsi, une RAM avec une seule transition  $(0, 1, 0, 11, 1)$  serait codé par  $i0 = 1$  (nombre de transitions),  $i1 = 0$  (état  $q$ ),  $i2 = 1$  (lettre  $a=1$  du mot d'entrée),  $i3 = 0$  (lettre  $A=0$  du haut de la pile),  $i4 = 2, i5 = 1, i6 = 1$  (codage du mot  $w=11$  à mettre sur le dessus de pile),  $i6 = 1$  (état  $q'$ ).

**Question 6 :** Écrire une machine RAM qui étant donné un automate à pile  $\mathcal{A}$  et un mot  $w$  en entrée écrit 0 en sortie si  $w$  est reconnu par  $\mathcal{A}$  et 1 sinon.

**Question 7 :** Faire tourner cette machine RAM sur un automate à pile reconnaissant le langage  $\{a^n b^n \mid n \in \mathbb{N}\}$ .

## 3 Optimisation de machine RAM

On veut optimiser le code des RAMs afin d'obtenir des machines plus petites et plus rapides.

**Question 8 :** On va représenter le code de la RAM de manière structurée par un graphe orienté. Chaque instruction est représentée par un sommet du graphe. Il y a un arc entre deux instructions si on peut passer de la première à la seconde en un pas de calcul. Donner une fonction qui crée ce graphe à partir du code d'une machine. Les instructions arithmétiques et le JUMP sont de degré sortant 1, tandis que les instructions conditionnelles sont de degré sortant 2.

**Question 9 :** On va appliquer une optimisation d'élimination du code mort. À partir du graphe représentant le code, calculer tous les sommets accessibles à partir de la première instruction. Tous les sommets non accessibles correspondent à des instructions qui ne seront jamais exécutées. Supprimer ces instructions dans votre code.

**Question 10 :** (Bonus) Proposer une méthode pour combiner plusieurs instructions en une seule. Par exemple, les deux instructions consécutives  $\text{ADD}(4,0,r1)$ ,  $\text{ADD}(r1,9,r1)$  peuvent être remplacées par  $\text{ADD}(13,0,r1)$ , si dans le graphe le sommet  $\text{ADD}(r1,9,r1)$  n'a comme prédécesseur que  $\text{ADD}(4,0,r1)$ .

## 4 Modalités pratiques

Merci de respecter les consignes suivantes :

- le projet est à faire en binôme ;
- le projet est à rendre sur ecampus, au plus tard le dimanche 5 mai à 23h59 ;
- une soutenance aura lieu la semaine du 6 mai.
- 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 (PC, YA, 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.