IFT2105–Introduction à l'informatique théorique (Devoir #2 – Été 2021)

Louis Salvail[1]

Université de Montréal (DIRO), QC, Canada
salvail@iro.umontreal.ca

1 Remise

La remise est vendredi le 18 juin avant 23h 59m. Aucun retard ne sera accepté. Vous pouvez (et vous êtes encouragés) faire votre devoir en équipe d'au plus deux.

Ce devoir vise à vous préparer pour l'examen intra. D'autres questions semblables au sujet des langages réguliers peuvent être facilement trouvées dans le livre de Sipser. Si vous savez comment répondre à toutes les questions alors vous êtes en très bonne position pour réussir l'examen intra. Le devoir #1 est également un élément important pour votre préparation à l'examen intra.

2 Questions

Répondez à 6 questions de votre choix parmi les suivantes:

1. Au théorème 3.10 des diapos, on indique comment une machine de Turing peut *simuler* un programme `TANTQUE`. Les instructions de base des programmes `TANTQUE` sont $\texttt{inc}(r_i)$ et $r_i \leftarrow r_j$ en plus des boucles `TANTQUE` $r_i \neq r_j$ `FAIRE` $[\langle \text{Bloc} \rangle]$. La machine de Turing utilise la première portion de son ruban pour ranger les valeurs des registres pendant la simulation du programme `TANTQUE`. Si le programme `TANTQUE` utilise les registres $r_0, r_1, \ldots, r_n$ alors le ruban contient, avant la simulation de la prochaine instruction, la chaîne $\langle r_0 \rangle \# \langle r_1 \rangle \ldots \# \langle r_n \rangle$, où $\langle r_i \rangle$ dénote la valeur de r_i écrite en binaire. Les portions du ruban à gauche et à droite de la chaîne contiennent que le symbole blanc $\sqcup \in \Gamma$. Si la prochaine instruction à simuler du programme `TANTQUE` est $\texttt{inc}(r_i)$ alors la machine de Turing utilise le bout de code qui *incrémente* un nombre en binaire vu au cours et en démo (i.e. après avoir positionné la tête de lecture sur le premier bit de $\langle r_i \rangle$). Cette question est au sujet de la simulation de l'instruction $r_i \leftarrow r_j$. Donnez une machine de Turing qui *simule* l'instruction $r_0 \leftarrow r_1$ en supposant que la tête de lecture/écriture est initialement sur le premier bit de $\langle r_0 \rangle$. À la fin du calcul, la tête de lecture/écriture doit retourner sur le premier bit de $\langle r_0 \rangle$, qui encode maintenant la valeur r_1. Votre machine doit laisser inchangées les valeurs de tous les autres registres. Elle devrait donc produire la chaîne suivante sur le ruban:

$$\ldots \sqcup \sqcup \langle r_1 \rangle \# \langle r_1 \rangle \# \langle r_2 \rangle \# \ldots \# \langle r_n \rangle \sqcup \sqcup \ldots$$

avec la tête de lecture/écriture positionnée sur le premier bit à gauche. Vous pouvez maintenant imaginer comment *simuler* n'importe quelle affectation $r_i \leftarrow r_j$. Donnez votre machine de Turing sous la forme d'un graphe tel que vu en démo. Supposez que $\Sigma = \{0, 1, \#\}$ et donnez explicitement l'alphabet de ruban Γ dont vous avez besoin pour le fonctionnement de votre machine.

2. Donnez une λ–expression F telle que $F\,\underline{n}$ "calcule" $\underline{n!}$ (par l'application de α et β–réductions), où $\underline{n} := \lambda f.\lambda x.\underbrace{f(f(\ldots(f}_{n \text{ fois}}\ x)\ldots))$ est l'encodage de $n \in \mathbb{N}$ par une λ–expression (rappelez-vous que $n! = n \cdot (n-1) \cdot \ldots \cdot 1$ avec $0! = 1$). Vous pouvez utiliser les fonctions introduites sur les diapos du cours (mais seulement celles-là) lorsque vous donnez votre λ–expression pour F.
3. Soit L_1 et L_2 deux langages. Nous définissons la différence entre deux langages $L_1 - L_2$ comme $L_1 - L_2 = \{w | w \in L_1 \text{ mais } w \notin L_2\}$. Est-ce que la classe des langages réguliers est fermée pour la différence entre deux langages? Prouvez votre réponse.
4. Donnez un AFD ou un AFN qui reconnaît $L_4 = \{\mathsf{a}^i\mathsf{b}^j\mathsf{c}^h | (i + j + h) \mod 3 = 1\}$. Évidemment, l'alphabet du langage L_4 est $\Sigma = \{\mathsf{a}, \mathsf{b}, \mathsf{c}\}$.
5. Donnez un AFD ou un AFN qui reconnaît

$$L_5 = \{w_1 w_2 \cdots w_n w_1 w_n | n > 1, \text{ et } w_i \in \{\mathsf{a}, \mathsf{b}\} \text{ pour } 1 \leq i \leq n\} \ .$$

Évidemment, l'alphabet du langage L_5 est $\Sigma = \{\mathsf{a}, \mathsf{b}\}$.

6. Soit $S = \{n_1, n_2, n_3, \ldots\}$ un ensemble fini ou infini d'entiers naturels. Nous dirons de S qu'il est un ensemble *régulier* d'entiers naturels s'il existe un automate fini déterministe $M_S = (Q, \Sigma, \delta, q_0, F)$ avec $\Sigma = \{1\}$ tel que $n \in S \Leftrightarrow 1^n \in L(M_S)$. Est-ce que $S = \{n \in \mathbb{N} | (\exists i \in \mathbb{N})[n \mod 3 = 2^i \mod 3]\}$ est un ensemble régulier d'entiers naturels? Montrez votre réponse.
7. Soit $\Sigma = \{\mathsf{0}, \mathsf{1}, \#\}$ et dénotons la valeur de $x \in \{\mathsf{0}, \mathsf{1}\}^*$ écrite en binaire par $\text{val}(x)$. Par exemple, $\text{val}(\mathsf{101}) = 5$ et $\text{val}(\mathsf{00111}) = 7$. Soit

$$L_7 = \{x\#y \,|\, x, y \in \{\mathsf{0}, \mathsf{1}\}^* \text{ avec } |x|, |y| > 0 \text{ et } \text{val}(y) = \text{val}(x) + 1\}.$$

Montrez que L_7 n'est pas un langage régulier.

8. Montrer que $L_8 = \{w \in \{\mathsf{a}, \mathsf{b}\}^* \,|\, |w|_{\mathsf{a}} > |w|_{\mathsf{b}}\}$ n'est pas régulier.