Devoir maison n°3

À faire pour le mercredi 11/12

⋆ ⋆ ⋆

Problème du vide et automates de Büchi

1 Mots finis et langages reconnaissables

Un *automate fini non déterministe* sur un alphabet Σ est un quadruplet $A = (Q, I, F, T)$ où Q est un ensemble fini d'*états*, $I \subseteq Q$ est le sous-ensemble des *états initiaux*, $F \subseteq Q$ est le sous-ensemble des *états finaux* et l'ensemble $T \subseteq Q \times \Sigma \times Q$ est l'ensemble des *transitions*, étiquetées par les lettres de l'alphabet Σ.

Si $(q, a, q') \in T$, on note $q \xrightarrow{a} q'$ cette transition.

Pour représenter graphiquement un automate, on utilise une flèche entrante pour désigner un état initial et un double cercle pour désigner un état final, comme l'illustre l'exemple de la figure 1.

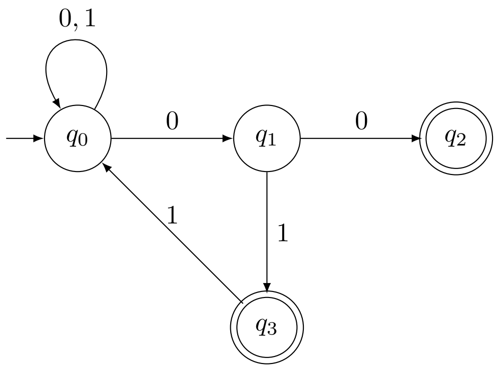

Figure 1 – L'automate A_1

Un mot $u = a_0 \ldots a_{n-1}$ est reconnu par l'automate $\mathcal{A}$ s'il existe une succession de transitions :

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} \ldots \xrightarrow{a_{n-2}} q_{n-1} \xrightarrow{a_{n-1}} q_n \qquad \text{avec } q_0 \in I \text{ et } q_n \in F$$

On dira que le mot u *étiquette un chemin* dans l'automate A allant de q_0 à q_n.

Le langage d'un automate A, noté L_A, est exactement l'ensemble des mots reconnus par l'automate A. On dit alors que A *reconnait* L_A. Un langage est dit *reconnaissable* s'il est le langage d'un automate fini.

Un automate $A = (Q, I, F, T)$ est dit *déterministe* si et seulement si $|I| = 1$ et pour tout $q \in Q$ et $a \in \Sigma$, l'ensemble $\{q' \in Q \mid (q, a, q') \in T\}$ est de cardinal au plus 1, c'est-à-dire il existe au plus une transition étiquetée par une même lettre sortant d'un état.

1.1 Premier exemple et miroir

Question 1 Décrire en français, sans justifier, le langage reconnu par l'automate A_1.

Question 2 Soit $A = (Q, I, F, T)$ un automate non déterministe tel que $|I| > 1$. Montrer qu'il existe un automate non déterministe à $|Q| + 1$ états reconnaissant le même langage que A et avec un seul état initial.

Question 3 Donner un exemple de langage reconnaissable L tel que tout automate (déterministe ou non) reconnaissant L possède au moins deux états initiaux ou au moins deux états finaux.

Pour $u = a_0 a_1 \ldots a_{n-1} \in \Sigma^*$ un mot, on définit le *miroir* de u, noté $\widetilde{u}$, par $\widetilde{u} = a_{n-1} a_{n-2} \ldots a_0$. Pour L un langage, on définit le *miroir* de L, noté $\widetilde{L}$, par $\widetilde{L} = \{\widetilde{u} \mid u \in L\}$.

Question 4 Représenter graphiquement, sans justifier, un automate déterministe reconnaissant $\widetilde{L}(A_1)$, où A_1 est l'automate de la figure 1.

Pour $A = (Q, I, F, T)$ un automate, on définit le *transposé* de A par $A^T = (Q, F, I, T')$, où $T' = \{(q', a, q) \mid (q, a, q') \in T\}$.

Question 5 Soit $A = (Q, I, F, T)$ un automate. Montrer que $L(A^T) = \widetilde{L(A)}$.

En OCaml, on suppose que l'alphabet Σ est un ensemble d'entiers $\Sigma = [\![0, p-1]\!]$ avec $p = |\Sigma|$. On représente un mot comme une liste de ses lettres. Par exemple, le mot $u = 0110110$ peut être créé par la commande `let u = [0; 1; 1; 0; 1; 1 ;0];;`.

On représente en OCaml automate par le type suivant :

```
type automate = {
    init : int list;
    final : int list;
    delta : int list array array
};;
```

tel que si `aut` est un objet de type `automate` représentant $A = (Q, I, F, T)$, alors :

- pour $n = |Q|$, on a $Q = [\![0, n-1]\!]$;
- `aut.init` et `aut.final` sont les listes, sans doublons, des éléments de I et F respectivement ;
- `aut.delta` est une matrice de taille $n \times p$ telle que pour $q \in Q$ et $a \in \Sigma$, `aut.delta.(q).(a)` est la liste des éléments q' tels que $(q, a, q') \in T$.

Par exemple, l'automate A_1 de la figure 1 peut être créé par :

```
let a1 = {
    init = [0];
    final = [2; 3];
    delta = [| [| [0; 1]; [0] |];
               [| [2]; [3] |];
               [| []; [] |];
               [| []; [0]|] |]
};;
```

On rappelle qu'une matrice de taille $n \times p$ dont toutes les cases contiennent un élément x peut être créée avec la commande `Array.make_matrix n p x`.

Question 6 Écrire une fonction `transpose (aut : automate) : automate` qui prend en argument un automate A et renvoie son transposé A^T.

Question 7 Déterminer la complexité temporelle de la fonction précédente dans le pire cas en fonction de $|Q|$ et $|\Sigma|$.

1.2 Reconnaissance de mot

On souhaite écrire ici une fonction permettant de tester l'appartenance d'un mot à un langage.

Question 8 Écrire une fonction `delta (aut : automate) (ens : int list) (a : int) : int list` qui prend en argument un automate $A = (Q, I, F, T)$, un ensemble $X \subseteq Q$ donné par la liste de ses éléments et une lettre $a \in \Sigma$ et calcule l'ensemble $\Delta(X, a) = \{q' \in Q \mid \exists q \in X, (q, a, q') \in T\}$. On attend une complexité en $\mathcal{O}(|Q|^2)$.

Question 9 En déduire une fonction

```
delta_etoile (aut : automate) (ens : int list) (u : int list) : int list
```

qui prend en argument un automate $A = (Q, I, F, T)$, un ensemble $X \subseteq Q$ donné par la liste de ses éléments et un mot $u \in \Sigma^*$ et calcule l'ensemble $\Delta^*(X, u) = \{q' \in Q \mid \exists q \in X, u$ étiquette un chemin de q à $q'\}$

Question 10 En déduire une fonction `reconnu (aut : automate) (u : int list) : bool` qui détermine si un mot est reconnu ou non par un automate.

Question 11 Déterminer la complexité temporelle de la fonction précédente dans le pire cas en fonction de $|Q|$, $|\Sigma|$ et $|u|$.

1.3 Problème du vide

Dans cette partie, on souhaite déterminer si un langage est vide ou non.

Question 12 Écrire une fonction `accessibles (aut : automate) : int list` qui prend en argument un automate $A = (Q, I, F, T)$ et renvoie la liste des états accessibles de A.

Question 13 En déduire une fonction `est_vide (aut : automate) : bool` qui détermine si le langage reconnu par un automate est vide ou non.

2 Mots infinis et langages de Büchi

2.1 Mots infinis

Un *mot infini* sur Σ est une suite $u = (a_i)_{i\in\mathbb{N}}$ où $a_i \in \Sigma$ pour tout $i \in \mathbb{N}$. On note Σ^ω l'ensemble des mots infinis sur Σ.

Si $u = a_0 \ldots a_{n-1} \in \Sigma^*$ et $v = (b_i)_{i\in\mathbb{N}} \in \Sigma^\omega$, on appelle *concaténation* de u et v le mot infini $uv = (c_i)_{i\in\mathbb{N}}$ défini par :

- pour $i < n$, $c_i = a_i$;
- pour $i \geqslant n$, $c_i = b_{i-n}$.

Question 14 Soit $u \in \Sigma^+$ un mot fini non vide. Montrer qu'il existe un unique mot infini $v \in \Sigma^\omega$ tel que $v = uv$. Pour la suite, on notera u^ω cet unique mot.

Question 15 On considère $u = (a_i)_{i\in\mathbb{N}}$ tel que $a_i = 0$ si i est un carré, et $a_i = 1$ sinon. Montrer qu'il n'existe pas de mot $x, y \in \Sigma^*$ tels que $u = xy^\omega$.

Pour $L \subseteq \Sigma^*$ un langage de mots finis et $L' \subseteq \Sigma^\omega$ un langage de mots infinis, on appelle *concaténation* de L et L' le langage de mots infinis $LL' = \{uu' \mid u \in L, u' \in L'\}$. Si $\varepsilon \notin L$, on note L^ω l'unique langage de mots infinis tel que $L^\omega = LL^\omega$.

Question 16 Montrer que dans le cas général, $L^\omega \neq \{u^\omega \mid u \in L\}$.

2.2 Langage de Büchi

Pour $A = (Q, I, F, T)$ un automate, et $u = (a_i)_{i\in\mathbb{N}} \in \Sigma^\omega$, on dit que u est *Büchi-reconnu* par A s'il existe une lecture de u qui passe une infinité de fois par un état de F. Autrement dit, pour tout $k \in \mathbb{N}$, il existe un entier $m \in \mathbb{N}$ et une lecture de $(a_i)_{0\leqslant i<m}$ qui passe k fois par un état de F.

On appelle *langage de Büchi* de A, noté $\mathcal{B}(A)$, l'ensemble des mots infinis Büchi-reconnu par A.

Par exemple, le langage de Büchi de l'automate A_2 de la figure 2 est le langage des mots sur $\Sigma = \{0, 1\}$ contenant une infinité de 0.

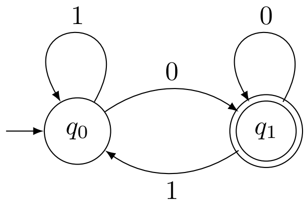

Figure 2 – L'automate A_2.

Question 17 Représenter graphiquement, sans justifier, des automates à 2 états dont les langages de Büchi sont les langages suivants sur l'alphabet $\Sigma = \{0, 1\}$:

1. $L_1 = \{0, 1\}^*\{0\}^\omega$.
2. L_2 : l'ensemble des mots infinis où toute position paire contient un 0.
3. L_3 : l'ensemble des mots infinis tel qu'un 0 est toujours suivi d'un 1.

Question 18 Déterminer, en justifiant succinctement, le langage de Büchi de l'automate A_3 de la figure 3.

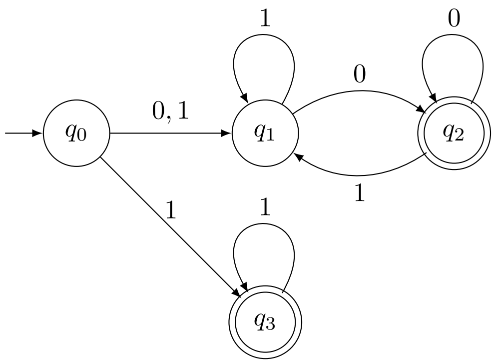

Figure 3 – L'automate A_3.

Question 19 Montrer qu'il n'existe pas d'automate fini déterministe A tel que $\mathcal{B}(A) = L_1$, où L_1 est le langage défini à la question 17.

2.3 Problème du vide et clôture

Question 20 Décrire en français un algorithme permettant de déterminer si le langage de Büchi d'un automate est vide ou non. Quelle est sa complexité temporelle ?

Soit A et A' deux automates. Pour les questions suivantes, on attend une description précise des automates demandés, et une preuve succincte de leur correction.

Question 21 Montrer qu'il existe un automate $A_\cup$ tel que $\mathcal{B}(A_\cup) = \mathcal{B}(A) \cup \mathcal{B}(A')$.

Question 22 Montrer qu'il existe un automate $A_\bullet$ tel que $\mathcal{B}(A_\bullet) = L(A)\mathcal{B}(A')$.

Question 23 On suppose qu'aucun état initial de A n'est final. Montrer qu'il existe un automate A_ω tel que $\mathcal{B}(A_\omega) = L(A)^\omega$.

⋆ ⋆ ⋆