Composition d'informatique n°3

Sujet 2 : corrigé

⋆⋆⋆

Question 1 On remarque qu'indépendamment de W, il n'existe qu'une seule stratégie pour le joueur 2 : $f_2 : s \in S_2 \mapsto 1$. De plus, il existe deux stratégies pour le joueur 1 : $f_1 : 1 \mapsto 2$ et $f_1' : 1 \mapsto 0$.

Si $W = \{\sigma \in \mathcal{C}(G) \mid |\sigma|_0 = 0\}$. Le jeu (G, S_1, S_2, W) est-il positionnel. En effet, On pose $R_1 = \{1, 2\}$ et $R_2 = \{0\}$, et on considère les stratégies f_1 et f_2.

- toute partie depuis 0 est gagnée par 2, car un chemin commençant par 0 ne peut pas être dans W. On en déduit que f_2 est bien une stratégie gagnante pour 2 depuis R_2 ;
- soit $\sigma = (s_i)_{0 \leqslant i < m}$ une f_1-partie depuis 1 ou 2. Alors σ alterne entre les sommets 1 et 2. Plus formellement :
  - ⋆ si $s_0 = 1$, alors pour tout $0 \leqslant i < m$, $s_i = 1 + (i \mod 2)$;
  - ⋆ si $s_0 = 2$, alors pour tout $0 \leqslant i < m$, $s_i = 2 - (i \mod 2)$.

  Dans les deux cas, f_1 est bien une stratégie gagnante pour 1 depuis R_1.

Question 2 Si $W = \{\sigma \in \mathcal{C}(G) \mid |\sigma|_0 = \infty \text{ et } |\sigma|_2 = \infty\}$, alors le jeu n'est pas positionnel. En effet, toute f_1-partie ne passera qu'au plus une fois par le sommet 0, et toute f_1'-partie ne passera qu'au plus une fois par le sommet 2. On en déduit qu'aucune stratégie n'est gagnante pour 1, quel que soit le sommet de départ. De plus, si on pose $\sigma = (0, 1, 2, 1, 0, 1, 2, \ldots)$, alors σ est une f_2-partie qui n'est pas gagnée par 2. On en déduit que le jeu n'est pas positionnel.

Question 3 Supposons que le jeu est positionnel et soient f_1 et f_2 les stratégies correspondantes. Supposons par l'absurde que $R_1 \cap R_2 \neq \emptyset$ et soit $s \in R_1 \cap R_2$. Sans perte de généralité, supposons $s \in S_1$. On définit $\sigma = (s_i)_{i \in \mathbb{N}}$ par :

- $s_0 = s$;
- pour $i \in \mathbb{N}$, $s_{i+1} = \begin{cases} f_1(s_i) & \text{si } s_i \in S_1 \\ f_2(s_i) & \text{sinon} \end{cases}$

Alors par définition, σ est une f_1-partie et une f_2-partie. Elle doit être donc gagnante pour 1 et pour 2, car $s \in R_1 \cap R_2$. On conclut par l'absurde.

Question 4 Un classique : pour chaque voisin t d'un sommet s, on ajoute s comme voisin de t dans le graphe transposé.

```
let transpose g =
    let n = Array.length g in
    let gt = Array.make n [] in
    for s = 0 to n - 1 do
        List.iter (fun t -> gt.(t) <- s :: gt.(t)) g.(s)
    done;
    gt;;
```

Question 5 On remarque qu'il existe une stratégie gagnante depuis un sommet s si et seulement s'il existe un chemin de s à un sommet $t \in T$. Pour construire la stratégie, on va plutôt travailler dans le graphe transposé et trouver un chemin de t à s. On écrit pour cela une fonction auxiliaire `dfs : int -> int -> unit` qui lance un parcours depuis le sommet t en indiquant que $f(t) = s$ (ou n si $t \in T$). On lance ensuite un parcours depuis chaque sommet de T.

```
let strategie g tab =
    let n = Array.length g in
    let gt = transpose g in
    let f = Array.make n (-1) in
    let rec dfs s t =
        if f.(t) < 0 then begin
           f.(t) <- if tab.(t) then n else s;
           List.iter (dfs t) gt.(t)
        end in
    for s = 0 to n - 1 do
        if tab.(s) then dfs n s
    done;
    f;;
```

Le calcul de G^T se fait en temps $\mathcal{O}(|A|+|S|)$. De plus, on remarque qu'on fait au plus un appel à `List.iter` pour chaque liste d'adjacence, c'est-à-dire que le nombre total d'appels récursifs à `dfs` est au plus $\mathcal{O}(|A|)$. Étant donnée la taille de la boucle `for`, on a bien une complexité totale en $\mathcal{O}(|A|+|S|)$.

Question 6 $\mathrm{Attr}_i(X)$ est l'ensemble des sommets $s \in S$ tel qu'il existe une stratégie pour le joueur 1 lui permettant d'atteindre X en moins de i coups, quels que soient les choix du joueur 2.

Question 7 Dans le cas général, $\mathrm{Attr}_i(X)$ est l'ensemble des sommets $s \in S$ tel que le joueur 1 peut forcer une partie depuis s à arriver dans X en i coups ou moins.

Par définition, on remarque que la suite $(\mathrm{Attr}_i(X))_{i\in\mathbb{N}}$ est croissante. De plus, tous les $\mathrm{Attr}_i(X)$ sont inclus dans S. Étant donné que S est fini, on en déduit que la suite est ultimement stationnaire, ce qui conclut.

Question 8

1. On définit la stratégie f_1 de la manière suivante : si $s \in R_1 \cap S_1$, alors soit $i_0 = \min\{i \in \mathbb{N} \mid s \in \mathrm{Attr}_i(T)\}$. On pose $f_1(s) \in V(s) \cap \mathrm{Attr}_{i_0-1}(T)$ choisi arbitrairement (avec la convention $\mathrm{Attr}_{-1}(T) = S$). Par hypothèse, cet ensemble est non vide si $i_0 > 0$.

   Soit alors $\sigma = (s_i)_{0\leqslant i<m}$ une f_1-partie depuis un sommet $s \in R_1$. Soit $0 \leqslant i < m$ tel que $i+1 < m$ et j l'indice minimal tel que $s_i \in \mathrm{Attr}_j(T)$. Montrons que $s_{i+1} \in \mathrm{Attr}_{j-1}(T)$:

   - si $s_i \in S_1$, alors $s_{i+1} = f_1(s_i) \in V(s) \cap \mathrm{Attr}_{j-1}(T)$;
   - sinon, si $s_i \in S_2$, alors $V(s_i) \subseteq \mathrm{Attr}_{j-1}(T)$, donc $s_{i+1} \in V(s_i) \cap \mathrm{Attr}_{j-1}(T)$.

   On en déduit par une récurrence rapide que $s_{i_0} \in T$, donc σ est gagnante pour 1.
2. On définit la stratégie f_2 de la manière suivante : si $s \in R_2 \cap S_2$, alors par hypothèse, $V(s) \subsetneq \mathrm{Attr}(T)$. On pose alors $f_2(s) \in V(s) \cap R_2$ qui est bien non vide.

   Soit alors $\sigma = (s_i)_{0\leqslant i<m}$ une f_2-partie depuis un sommet $s \in R_2$. Montrons par récurrence que pour $0 \leqslant i < m$, $s_i \in R_2$:

   - $s_0 = s \in R_2$ par hypothèse ;
   - supposons le résultat vrai pour $0 \leqslant i < m$ tel que $i+1 < m$ et distinguons :
     * si $s_i \in S_1$, alors par définition de $\mathrm{Attr}(T)$, $V(s_i) \cap \mathrm{Attr}(T) = \emptyset$. On en déduit que $s_{i+1} \in V(s_i) \cap R_2$;
     * si $s_i \in S_2$, alors $s_{i+1} = f_2(s_i) \in V(s_i) \cap R_2$.

   On conclut par récurrence que σ est gagnante pour 2, car $R_2 \cap T = \emptyset$.

Question 9 Cela correspond à l'algorithme vu en cours.

On utilise ici une sorte de parcours en profondeur dans G^T. Pour ce faire, on crée le tableau `attr`. On calcule un tableau `restant` qui détermine le nombre de voisins restants (c'est-à-dire pas encore dans l'attracteur) dans G pour un sommet s donné. Dans la fonction `dfs`, si s n'était pas déjà dans l'attracteur, on l'ajoute, puis on diminue de 1 la valeur de `restants.(t)` pour chacun de ses voisins t, puis, si $t \in S_1$, ou si $t \in S_2$ et que tous les voisins de t dans G sont dans l'attracteur (c'est-à-dire si `restants.(t)` vaut 0), on relance un parcours depuis t. Finalement, on lance un parcours depuis chaque sommet de T.

```
let attracteur g s1 tab =
    let n = Array.length g in
    let gt = transpose g and
        attr = Array.make n false and
        restant = Array.map List.length g in
    let rec dfs s =
        if not attr.(s) then begin
            attr.(s) <- true;
            let traiter t =
                restant.(t) <- restant.(t) - 1;
                if s1.(t) || restant.(t) = 0 then dfs t in
            List.iter traiter gt.(s)
        end in
    for s = 0 to n - 1 do
        if tab.(s) then dfs s
    done;
    attr;;
```

La complexité est la même que celle d'un parcours, soit $\mathcal{O}(|S| + |A|)$.

Question 10 Il existe une stratégie gagnante depuis s s'il existe $t \in T$ tel qu'il existe un chemin de s à t et un cycle (non vide) contenant t.

En effet, s'il existe une stratégie gagnante depuis s, soit σ la f-partie depuis s. Cette partie passe une infinité de fois par un sommet de T, donc une infinité de fois par le même sommet $t \in T$. Il existe donc un chemin de s à t et un cycle passant par t.

Réciproquement, la stratégie f qui consiste à atteindre un sommet du cycle, puis parcourir ce cycle passant par T (et définie arbitrairement en dehors de ce chemin et de ce cycle) est bien une stratégie gagnante.

En utilisant l'algorithme de Kosaraju, on peut calculer toutes les composantes fortement connexes du graphe en complexité $\mathcal{O}(|S| + |A|)$. Dès lors, un sommet $t \in T$ conviendra si sa composante fortement connexe est de taille > 1, ou s'il existe une boucle $(t, t) \in A$ (ce qui peut se tester pendant l'algorithme). On peut alors faire un parcours dans G^T depuis tous les sommets t qui conviennent pour trouver tous les sommets s cherchés. La complexité totale est en $\mathcal{O}(|S| + |A|)$.

Question 11 Montrons un résultat intermédiaire, à savoir que pour $i \in \mathbb{N}$, $\mathrm{Attr}_i^+(X) \cup X \subseteq \mathrm{Attr}_i(X)$, par récurrence sur i :

- c'est vrai pour $i = 0$ car $\mathrm{Attr}_0^+(X) \cup X = X = \mathrm{Attr}_0(X)$;
- supposons le résultat établi pour $i \in \mathbb{N}$ fixé. Soit alors $s \in \mathrm{Attr}_{i+1}^+(X)$. Si $s \in \mathrm{Attr}_i^+(X)$, alors par hypothèse, $s \in \mathrm{Attr}_i(X) \subseteq \mathrm{Attr}_{i+1}(X)$. Sinon, distinguons :
  - si $s \in S_1$, alors $V(s) \cap (\mathrm{Attr}_i^+(X) \cup X) \neq \emptyset$. On en déduit que $V(s) \cap \mathrm{Attr}_i(X) \neq \emptyset$, c'est-à-dire que $s \in \mathrm{Attr}_{i+1}(X)$;
  - si $s \in S_2$, alors $V(s) \subseteq \mathrm{Attr}_i^+(X) \cup X \subseteq \mathrm{Attr}_i(X)$, c'est-à-dire $s \in \mathrm{Attr}_{i+1}(X)$.

  Comme $X \subseteq \mathrm{Attr}_{i+1}(X)$, on en déduit que $\mathrm{Attr}_{i+1}^+(X) \cup X \subseteq \mathrm{Attr}_{i+1}(X)$.

On conclut par récurrence. On peut maintenant répondre à la question :

1. soit $s \in \mathrm{Attr}^+(X) \cap S_1$ et soit $i \in \mathbb{N}^*$ l'indice minimal tel que $s \in \mathrm{Attr}_i^+(X)$. Par hypothèse, $V(s) \cap (\mathrm{Attr}_{i-1}^+(X) \cup X) \neq \emptyset$. On en déduit qu'il existe un voisin de s dans $\mathrm{Attr}_{i-1}^+(X) \cup X \subseteq \mathrm{Attr}_{i-1}(X) \subseteq \mathrm{Attr}(X)$.
2. soit $s \in \mathrm{Attr}^+(X) \cap S_2$ et soit $i \in \mathbb{N}^*$ l'indice minimal tel que $s \in \mathrm{Attr}_i^+(X)$. Par hypothèse, $V(s) \subseteq \mathrm{Attr}_{i-1}^+(X) \cup X \subseteq \mathrm{Attr}_{i-1}(X) \subseteq \mathrm{Attr}(X)$.

Notons que l'inclusion réciproque étant vraie par définition de $\mathrm{Attr}^+(X)$, on a :

$$\mathrm{Attr}^+(X) = \{s \in S_1 \mid V(s) \cap \mathrm{Attr}(X) \neq \emptyset\} \cup \{s \in S_2 \mid V(s) \subseteq \mathrm{Attr}(X)\}$$

Question 12 Reformulons en français ce que veulent dire ces ensembles :

- $\mathrm{Attr}_i^+(X)$ est l'ensemble des sommets $s \in S$ tel que le joueur 1 peut forcer une partie depuis s à arriver dans X en un nombre de coups compris entre 1 et i ;
- $\mathrm{Attr}^+(X)$ est l'ensemble des sommets $s \in S$ tel que le joueur 1 peut forcer une partie depuis s à arriver dans X en un nombre de coups strictement positif ;
- $E_i(T)$ est l'ensemble des sommets $t \in T$ tel que le joueur 1 peut forcer une partie depuis t à passer $i+1$ fois par un sommet de T ;
- $E(T)$ est l'ensemble des sommets $t \in T$ tel que le joueur 1 peut forcer une partie depuis t à passer une infinité de fois par un sommet de T.

Remarquons tout d'abord que si $X \subseteq Y$, alors $\mathrm{Attr}^+(X) \subseteq \mathrm{Attr}^+(Y)$. Montrons que $E_{i+1}(T) \subseteq E_i(T)$ par récurrence sur $i \in \mathbb{N}$:

- pour $i = 0$, $E_1(T) = \mathrm{Attr}^+(T) \cap T \subseteq T = E_0(T)$;
- supposons le résultat vrai pour $i \in \mathbb{N}$ fixé. Alors $E_{i+2}(T) = \mathrm{Attr}^+(E_{i+1}(T)) \cap T \subseteq \mathrm{Attr}^+(E_i(T)) \cap T = E_{i+1}(T)$.

On conclut par récurrence.

Comme les $E_i(T)$ sont des ensembles finis, on en déduit qu'il existe $k \in \mathbb{N}$ tel que $E_{k+1}(T) = E_k(T) = E(T)$. On définit alors la stratégie f_1 suivante pour $s \in \mathrm{Attr}(E(T)) \cap S_1$:

- si $s \in E(T)$, alors $s \in E_{k+1}(T) = \mathrm{Attr}^+(E_k(T)) \cap T$. On en déduit que $V(s) \cap \mathrm{Attr}(E_k(T)) \neq \emptyset$, donc et on pose $f(s)$ un voisin de s dans $\mathrm{Attr}(E(T))$;
- si $s \notin E(T)$, alors par la question 8, il existe une stratégie f telle que toute f-partie depuis s rencontre $E(T)$. On pose $f_1(s) = f(s)$.

La stratégie f_1 est bien gagnante pour 1 : depuis un sommet de $E(T)$, on peut forcer un coup dans $\mathrm{Attr}(E(T))$, et depuis un sommet de $\mathrm{Attr}(E(T)) \setminus E(T)$, la stratégie permet de revenir dans $E(T)$ en un nombre fini de coups. Une f_1-partie passe donc une infinité de fois par un sommet de $E(T) \subseteq T$.

Question 13 On pose, par convention, $E_{-1}(T) = S$. Remarquons que pour $s \notin \mathrm{Attr}(E(T))$, il existe $i \in \mathbb{N}$ tel que $s \in \mathrm{Attr}(E_{i-1}(T))$ et $s \notin \mathrm{Attr}(E_i(T))$ (sinon $s \in \mathrm{Attr}(E(T))$). Pour $s \in S_2$, on définit la stratégie f_2 en distinguant :

- si $i = 0$, alors $s \in S \setminus \mathrm{Attr}(T)$, donc il existe une stratégie f pour le joueur 2 tel qu'une f-partie n'atteint jamais un sommet de T. On pose $f_2(s) = f(s)$;
- sinon, si $s \in T$, alors $s \in E_{i-1}(T)$, et on a $V(s) \setminus \mathrm{Attr}(E_{i-1}(T)) \neq \emptyset$ (sinon $s \in \mathrm{Attr}^+(E_{i-1}(T)) \cap T = E_i(T)$). On pose $f_2(s)$ un tel sommet dans $V(s) \setminus \mathrm{Attr}(E_{i-1}(T))$.
- sinon, $V(s) \setminus \mathrm{Attr}(E_i(T)) \neq \emptyset$, car $s \notin \mathrm{Attr}(E_i(T))$. On pose $f_2(s)$ un tel sommet dans $V(s) \setminus \mathrm{Attr}(E_i(T))$.

Montrons que f_2 est une stratégie gagnante pour 2 depuis tout sommet s de $S \setminus \mathrm{Attr}(E(T))$. Soit $\sigma = (s_i)_{i \in \mathbb{N}}$ une f_2-partie depuis s. Il existe un indice $i_0 \in \mathbb{N}$ tel que $s \in \mathrm{Attr}(E_{i_0-1}(T))$ et $s \notin \mathrm{Attr}(E_{i_0}(T))$. Alors σ passe au plus i_0 fois par un sommet de T : tant qu'on ne passe pas par T, on reste dans un $\mathrm{Attr}(E_{j-1}(T)) \setminus \mathrm{Attr}(E_j(T))$, et dès qu'on passe par T, par construction, le sommet suivant sera dans $\mathrm{Attr}(E_{j-2}(T)) \setminus \mathrm{Attr}(E_{j-1}(T))$.

Question 14 Un mot peut être considéré comme un chemin dans le graphe $G_\Sigma = (\Sigma, \Sigma \times \Sigma)$.

Question 15 On pose $u = a_1 a_2 \ldots a_{m-1}$, pour $m > 1$. On définit $u^\omega = (b_i)_{i \in \mathbb{N}^*}$ tel que pour $i \in \mathbb{N}^*$, $b_i = a_{1+((i-1) \mod (m-1))}$, c'est-à-dire le mot constitué de répétitions du mot u. Alors $u^\omega \in \Sigma^\omega$ vérifie bien $u^\omega = uu^\omega$. Supposons qu'il existe un mot $v = (c_i)_{i \in \mathbb{N}^*} \in \Sigma^\omega$ tel que $v = uv$. Alors $a_1 \ldots a_{m-1} = c_1 \ldots c_{m-1}$ et $v = (c_i)_{m \leqslant i}$. On en déduit par récurrence que $v = u^\omega$.

Question 16 Supposons par l'absurde qu'il existe $v, w \in \Sigma^*$ tels que $u = vw^\omega$. Notons $k_0 = |w|_0$ le nombre de 0 dans w et $k_1 = |w|_1$ le nombre de 1 dans w. Distinguons :

– si $k_1 = 0$ ou $k_0 = 0$, alors vw^ω ne contient que des 0 ou que des 1 à partir d'un certain rang, donc $vw^\omega \neq u$;

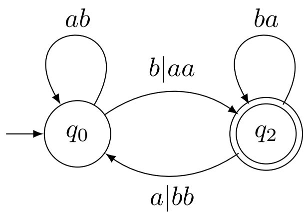
– sinon, notons p_i le préfixe de taille i de vw^ω. Alors $\dfrac{|p_i|_1}{|p_i|_0} \underset{i \longrightarrow \infty}{\longrightarrow} \dfrac{k_1}{k_0}$. C'est absurde car si p_i est un préfixe de u, $\dfrac{|p_i|_1}{|p_i|_0} \underset{i \longrightarrow \infty}{\longrightarrow} 0$.

Dans tous les cas, on arrive à une contradiction.

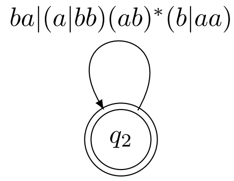
Question 17

1. Les mots infinis qui ne contiennent qu'un nombre fini de b : $(a|b)^*a^\omega$.
2. Les mots infinis où toute position paire contient un a : $(aa|ba)^\omega$.
3. Les mots infinis tel qu'un a est toujours suivi d'un b : $(ab|b)^\omega$.
4. Les mots infinis contenant aucun a **ou** contenant une infinité de a. : $b^\omega|(b^*a)^\omega$.

Question 18 On utilise l'algorithme d'élimination des états pour trouver une expression régulière correspondant à un cycle de q_2 vers lui-même. En éliminant l'état q_1, on obtient :

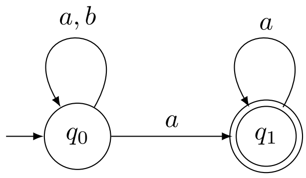


En supprimant l'état q_0, on obtient :



Dès lors, un mot est dans $L_B(A_1)$ si et seulement s'il est de la forme uv où $q_2 \in \delta^*(q_0, u)$ et $v = (v_i)_i$ avec $q_2 \in \delta^*(q_2, v_i)$. On en déduit que $L_B(A_1) = \mathcal{L}(ef^\omega)$ où $e = (ab)^*(b|aa)$ et $f = ba|(a|bb)(ab)^*(b|aa)$.

Question 19 On considère l'automate :



Question 20 Supposons qu'il existe un automate déterministe $A = (Q, \delta, q_0, F)$ tel que $L_B(A) = \mathcal{L}(e_0) = L$.

Comme $a^\omega \in L$, il existe $n_1 \in \mathbb{N}$ tel que $q_1 = \delta^*(q_0, a^{n_1}) \in F$.

Comme $a^{n_1}ba^\omega \in L$, il existe $n_2 \in \mathbb{N}$ tel que $q_2 = \delta^*(q_1, ba^{n_2}) \in F$.

On construit de même par récurrence des états q_i, $i \geqslant 3$, … Comme F est fini, il existe $1 \leqslant i < j$ tels que $q_i = q_j$. Dès lors, $\delta^*(q_i, ba^{n_{i+1}}ba^{n_{i+2}}b \ldots ba^{n_j}) = q_j = q_i$.

On en déduit que $a^{n_1}b \ldots ba^{n_i}\,(ba^{n_{i+1}}ba^{n_{i+2}}b \ldots ba^{n_j})^\omega \in L_B(A)$, ce qui est absurde car ce mot contient une infinité de b.

Question 21 Soit $A = (Q, \delta, q_0, F)$. On considère $G = (Q, A_G)$ où $A_G = \bigcup_{q \in Q, a \in \Sigma} \{q\} \times \delta(q, a)$, c'est-à-dire le graphe dont les sommets sont les états de l'automate et les arêtes sont données par les transitions (on enlève les étiquettes). Alors $L_B(A) \neq \emptyset$ si et seulement s'il existe une stratégie gagnante depuis q_0 pour atteindre une infinité de fois les sommets de F, c'est-à-dire si $q_0 \in \mathrm{Attr}(E(F))$. On a vu à la question 10 que ce problème peut être résolu en complexité $\mathcal{O}(|Q| + |A_G|)$, donc il est décidable. Par réduction, `LANGUAGE NON VIDE` est décidable.

Question 22 Soit $A = (Q, \delta, q_0, F)$ un automate fini. Remarquons que pour $(p, q) \in Q^2$, le langage $L_{p,q} = \{u \in \Sigma^* \mid q \in \delta^*(p, u)\}$ est rationnel : il est reconnu par l'automate fini $(Q, \delta, p, \{q\})$.

Montrons que $L_B(A) = \bigcup_{q \in F} L_{q_0,q}(L_{q,q} \setminus \{\varepsilon\})^\omega$ par double inclusion. Soit $u = (a_i)_{i \in \mathbb{N}^*} \in \Sigma^\omega$.

- Si $u \in \bigcup_{q \in F} L_{q_0,q}(L_{q,q} \setminus \{\varepsilon\})^\omega$, alors il existe $q \in F$ tel que $u \in L_{q_0,q}(L_{q,q} \setminus \{\varepsilon\})^\omega$. On en déduit qu'il existe un mot $u_0 \in L_{q_0,q}$ et une suite de mots $(u_j)_{j \in \mathbb{N}^*} \in (L_{q,q} \setminus \{\varepsilon\})^{\mathbb{N}^*}$ tels que $u = u_0 u_1 u_2 \ldots$. Dès lors, il existe bien un calcul de u qui passe une infinité de fois par q, donc $u \in L_B(A)$.
- Si $u \in L_B(A)$, alors il existe un état $q \in F$ et un calcul de u qui passe une infinité de fois par q. Notons $i_0, i_1, \ldots$ les indices des transitions de ce calcul menant vers q. Alors $q \in \delta^*(q_0, u_1 \ldots u_{i_0})$ et pour $j \in \mathbb{N}$, $q \in \delta^*(q, u_{i_j+1} \ldots u_{i_{j+1}})$. On en déduit $u \in \bigcup_{q \in F} L_{q_0,q}(L_{q,q} \setminus \{\varepsilon\})^\omega$.

On conclut en remarquant que $\bigcup_{q \in F} L_{q_0,q}(L_{q,q} \setminus \{\varepsilon\})^\omega$ est bien l'interprétation d'une expression ω-régulière.

Question 23 L'idée est de créer une boucle depuis les états finaux vers l'état initial (mais sans ε-transitions). On a déjà montré en classe qu'on peut choisir A comme un automate standard (pas de transition vers l'état initial). Dès lors, on considère l'automate $A^\omega = (Q, \delta', q_0, F)$ dont la fonction de transition est définie par :

- pour tout $q \in Q$, $a \in \Sigma$, $\delta(q, a) \subseteq \delta'(q, a)$;
- pour tout $q \in F$, $a \in \Sigma$ $\delta(q_0, a) \subseteq \delta'(q, a)$.

L'idée est de rajouter une transition depuis chaque état final vers chaque voisin de l'état initial, étiquetée par la bonne lettre. Alors $L_B(A^\omega) = L(A)^\omega$.

Question 24 On pose $A = (Q_A, \delta_A, q_A, F_A)$ et $B = (Q_B, \delta_B, q_B, F_B)$. On définit $A_\cup = (Q, \delta, q_0, F)$ par :

- $Q = Q_A \cup Q_B \cup \{q_0\}$;
- $F = F_A \cup F_B$;
- pour $q \in Q$ et $a \in \Sigma$, $\delta(q, a) = \begin{cases} \delta_A(q, a) & \text{si } q \in Q_A \\ \delta_B(q, a) & \text{si } q \in Q_B \\ \delta_A(q_A, a) \cup \delta_B(q_B, a) & \text{si } q = q_0 \end{cases}$

L'idée est de rajouter un état initial et de faire des transitions depuis cet état vers chacun des voisins des états initiaux des deux automates. La lecture d'un mot infini se fera alors dans l'un ou dans l'autre.

Question 25 On pose $A = (Q_A, \delta_A, q_A, F_A)$ et $B = (Q_B, \delta_B, q_B, F_B)$. On définit $C = (Q, \delta, q_0, F)$ par :

- $Q = Q_A \cup Q_B$;
- $F = F_B$;
- pour $q \in Q$ et $a \in \Sigma$, $\delta(q, a) = \begin{cases} \delta_A(q, a) & \text{si } q \in Q_A \setminus F_A \\ \delta_A(q, a) \cup \delta_B(q_B, a) & \text{si } q \in F_A \\ \delta_B(q, a) & \text{si } q \in Q_B \end{cases}$

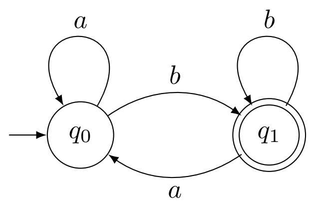
On rajoute une transition depuis les états finaux du premier automate vers chacun des voisins de l'état initial du deuxième automate pour considérer la concaténation des langages.

Question 26 Soit $e = e_1 f_1^\omega | e_2 f_2^\omega | \ldots | e_k f_k^\omega$ une expression ω-régulière. Par la question 23, les $\mathcal{L}(f_i^\omega)$ sont Büchi-reconnaissables. Par la question 25, les $\mathcal{L}(e_i f_i^\omega)$ sont Büchi-reconnaissables. Par la question 24, $\mathcal{L}(e)$ est Büchi-reconnaissable, ce qui conclut.

Question 27 Soit $u \in \Sigma^\omega$. On considère $(p_A^0, p_B^0, i^0), (p_A^1, p_B^1, i^1), (p_A^2, p_B^2, i^2)\dots$ un calcul de u dans $A_\cap$. Alors par construction, $p_A^0, p_A^1, p_A^2, \dots$ et $p_B^0, p_B^1, p_B^2, \dots$ sont des calculs de u dans A et B respectivement. Montrons l'égalité voulue par double inclusion :

- si $u \in L_B(A_\cap)$, alors il existe une infinité d'indices $j \in \mathbb{N}$ tels que $(p_A^j, p_B^j, i^j) \in F$, c'est-à-dire tels que $p_B^j \in F_B$ et $i^j = 2$. On en déduit que $i^{j+1} = 1$ par définition de la fonction de transition. De plus, comme il existe $k > j$ tel que $i^k = 2$, on en déduit qu'il existe $j + 1 \leqslant \ell < k$ tel que $i^\ell = 1$ et $i^{\ell+1} = 2$. Par définition de la fonction de transition, cela implique que $p_A^\ell \in F_A$. On en déduit que u est l'étiquette d'un calcul dans $L_B(A)$ et dans $L_B(B)$.
- La réciproque se fait de manière similaire.

Question 28 On rappelle que $\mathcal{L}(e_0)$ est l'ensemble des mots infinis contenant un nombre fini de b. Son complémentaire est l'ensemble des mots contenant un nombre infini de b. Ce langage est reconnu par l'automate déterministe :



On a montré en question 20 que $\mathcal{L}(e_0)$ n'était Büchi-reconnu par aucun automate déterministe, ce qui conclut.

⋆⋆⋆