Composition d'informatique n°3

Sujet 2 : informatique fondamentale

(Durée : 4 heures)

L'utilisation de la calculatrice **n'est pas autorisée** pour cette épreuve.

⋆⋆⋆

Jeux et automates de Büchi

Consignes

Les questions de programmation doivent être traitées en langage OCaml ou C selon ce qui est demandé par l'énoncé. En OCaml, on autorisera toutes les fonctions des modules `Array` et `List`, ainsi que les fonctions de la bibliothèque standard (celles qui s'écrivent sans nom de module, comme `max`, `incr` ainsi que les opérateurs comme `@`). Sauf précision de l'énoncé, l'utilisation d'autres modules sera interdite.

Lorsque le candidat écrira une fonction, il pourra faire appel à des fonctions définies dans les questions précédentes, même si elles n'ont pas été traitées. Il pourra également définir des fonctions auxiliaires, mais devra préciser leurs rôles ainsi que les types et significations de leurs arguments. Les candidats sont encouragés à expliquer les choix d'implémentation de leurs fonctions lorsque ceux-ci ne découlent pas directement des spécifications de l'énoncé. Si les paramètres d'une fonction à coder sont supposés vérifier certaines hypothèses, il ne sera pas utile dans l'écriture de cette fonction de tester si les hypothèses sont bien satisfaites.

On identifiera une même grandeur écrite dans deux polices de caractères différentes, en italique du point de vue mathématique (par exemple n) et en Computer Modern à chasse fixe du point de vue informatique (par exemple `n`).

Sans précision supplémentaire, lorsqu'une question demande la complexité d'une fonction, il s'agira de la complexité temporelle dans le pire des cas. La complexité sera exprimée sous la forme $\mathcal{O}(f(n,m))$ où n et m sont les tailles des arguments de la fonction, et f une expression la plus simple possible. Les calculs de complexité seront justifiés succinctement.

1 Jeux dans un graphe

Un **graphe** (fini et orienté) est un couple $G = (S, A)$ tel que S est un ensemble fini non vide et $A \subseteq S \times S$. Pour $s \in S$ un sommet, on note $V(s)$ l'ensemble des **voisins** de s, c'est-à-dire $V(s) = \{t \in S \mid (s,t) \in A\}$. On appelle **chemin** σ une suite non vide, finie ou infinie, de sommets $(s_i)_{0 \leqslant i < m}$, avec $m \in \mathbb{N}^* \cup \{\infty\}$, telle que $s_{i+1} \in V(s_i)$ si $0 \leqslant i$ et $i+1 < m$. Si $m \neq \infty$, on dit que σ est **fini de longueur** $m-1$. Sinon, il est dit **infini**. Il est à noter qu'un chemin σ peut passer plusieurs fois par le même sommet. On notera $\mathcal{C}(G)$ l'ensemble des chemins (finis ou infinis) de G.

Pour $s \in S$, le **nombre d'occurrences de s dans σ**, noté $|\sigma|_s$, correspond aux nombres de fois qu'un sommet de T apparaît dans σ. Formellement, $|\sigma|_s = \text{Card}\,\{i \in [\![0, m[\![\mid s_i = s\}$, ce cardinal pouvant être fini ou infini. On note de même, pour $T \subseteq S$, $|\sigma|_T = \text{Card}\,\{i \in [\![0, m[\![\mid s_i \in T\}$.

On considère un jeu à deux joueurs dans un graphe. Informellement, une partie se déroule comme suit : un jeton est placé sur un sommet du graphe G et déplacé par les joueurs de sommet en sommets, par une succession de coups. Un coup consiste à déplacer le jeton en suivant une arête : lorsque le jeton est sur un sommet s, le joueur à qui appartient s le déplace sur un voisin de s, et ainsi de suite. Une partie est un chemin traversé par le jeton.

Formellement, une **arène** est un triplet (G, S_1, S_2) tel que $G = (S, A)$ est un graphe et $S = S_1 \cup S_2$, $S_1 \cap S_2 = \emptyset$. Un **jeu** est un quadruplet (G, S_1, S_2, W) tel que (G, S_1, S_2) est une arène et $W \subseteq \mathcal{C}(G)$. Une **partie depuis un sommet initial** s est un chemin $\sigma = (s_i)_{0 \leqslant i < m}$ tel que $s_0 = s$. On dit que la partie σ est **gagnée** par le joueur 1 si $\sigma \in W$. Sinon, σ est dite gagnée par le joueur 2.

Pour $j \in \{1, 2\}$, une **stratégie pour le joueur** j est une application $f : S_j \to S$ telle que pour tout $s \in S_j$, $f(s) \in V(s)$. Une partie $\sigma = (s_i)_{0 \leqslant i < m}$ est dite une **f-partie** si pour tout $s_i \in S_j$, tel que $i + 1 < m$, $s_{i+1} = f(s_i)$. Une stratégie f pour le joueur j est dite **gagnante depuis** s si toute f-partie infinie depuis s est gagnée par le joueur j. Un sommet $s \in S$ est dit gagnant pour j s'il existe une stratégie gagnante pour j depuis s.

Un jeu est dit **positionnel** si tout sommet est gagnant pour 1 ou 2, c'est-à-dire s'il existe $R_1, R_2 \subseteq S$ tels que $S = R_1 \cup R_2$ et deux stratégie f_1 et f_2 telles que f_j est gagnante pour j depuis tout sommet de R_j, pour $j \in \{1, 2\}$. On remarquera que R_1 (resp. R_2) peut contenir à la fois des sommets de S_1 et des sommets de S_2.

On considèrera dans l'ensemble de cette partie que les graphes considérés sont sans puits, c'est-à-dire que pour tout $s \in S$, $V(s) \neq \emptyset$.

1.1 Préliminaires

On considère l'arène de la figure 1, où $S_1 = \{1\}$ et $S_2 = \{0, 2\}$ (les sommets de S_1 sont représentés par des cercles, ceux de S_2 par des carrés).

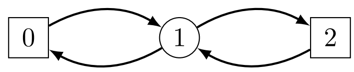

FIGURE 1 – Une arène (G, S_1, S_2).

Question 1 On suppose que W est l'ensemble des chemins ne visitant pas le sommet 0 : $W = \{\sigma \in \mathcal{C}(G) \mid |\sigma|_0 = 0\}$. Le jeu (G, S_1, S_2, W) est-il positionnel ? Si oui, donner les ensembles R_1 et R_2 et les stratégies f_1 et f_2 correspondants. Sinon, justifier.

Question 2 Même question si $W = \{\sigma \in \mathcal{C}(G) \mid |\sigma|_0 = \infty \text{ et } |\sigma|_2 = \infty\}$.

Question 3 Montrer que si un jeu est positionnel, alors les ensembles R_1 et R_2 sont disjoints.

1.2 Jeux d'accessibilité

On suppose dans cette partie que $T \subseteq S$ et que $W = \{\sigma \in \mathcal{C}(G) \mid |\sigma|_T > 0\}$. On représente un graphe $G = (S, A)$ en OCaml par un tableau de listes d'adjacence.

```ocaml
type graphe = int list array;;
```

Si `g` est une variable de type `graphe` correspondant à un graphe $G = (S, A)$, alors :
- $S = [\![0, n-1]\!]$, où $n =$ `Array.length g`;
- pour $s \in S$, `g.(s)` est une liste contenant les éléments de $V(s)$ (sans doublons, dans un ordre arbitraire).

Question 4 Écrire une fonction `transpose (g: graphe) : graphe` qui prend en argument un graphe $G = (S, A)$ et renvoie son graphe transposé $G^T = (S, A')$ où $A' = \{(s, t) \mid (t, s) \in A\}$. On garantira une complexité en $\mathcal{O}(|S| + |A|)$ mais on ne demande pas de le justifier.

Une partie $T \subseteq S$ sera représentée par un tableau de booléens `tab` de taille n tel que `tab.(s)` vaut `true` si et seulement si $s \in T$.

Question 5 Dans cette question, on suppose que le jeu est à un seul joueur, c'est-à-dire que $S_2 = \emptyset$. Écrire une fonction `strategie (g: graphe) (tab: bool array) : int array` qui prend en argument un graphe G et une partie $T \subseteq S$ et renvoie un tableau `f` de taille n tel que pour tout $s \in S$:

- si $s \in T$, alors `f.(s)` $= n$;
- sinon, s'il existe une stratégie gagnante f_1 depuis s, alors `f.(s)` $= f_1(s)$;
- sinon, `f.(s)` $= -1$.

On garantira une complexité en $\mathcal{O}(|S| + |A|)$ et on demande de justifier cette complexité.

On suppose pour la suite que $S_1 \neq \emptyset$ et $S_2 \neq \emptyset$. Pour $X \subseteq S$, on définit par induction l'ensemble $\text{Attr}_i(X)$, pour $i \in \mathbb{N}$, par :

- $\text{Attr}_0(X) = X$;
- $\text{Attr}_{i+1}(X) = \text{Attr}_i(X) \cup \{s \in S_1 \mid V(s) \cap \text{Attr}_i(X) \neq \emptyset\} \cup \{s \in S_2 \mid V(s) \subseteq \text{Attr}_i(X)\}$.

Question 6 Donner, sans justifier, une description en français de $\text{Attr}_i(X)$ en termes de stratégie pour le joueur 1.

Question 7 Montrer qu'il existe $k \in \mathbb{N}$ tel que $\bigcup_{i \in \mathbb{N}} \text{Attr}_i(X) = \text{Attr}_k(X)$. On notera $\text{Attr}(X)$ cet ensemble pour la suite et on l'appellera **attracteur de** X.

Question 8 Montrer que le jeu est positionnel. Plus précisément :

1. Montrer que le joueur 1 a une stratégie gagnante depuis $R_1 = \text{Attr}(T)$.
2. Montrer que le joueur 2 a une stratégie gagnante depuis $R_2 = S \setminus \text{Attr}(T)$.

Question 9 Écrire une fonction `attracteur (g: graphe) (s1: bool array) (tab: bool array) : bool array` qui prend en argument un graphe G, un tableau représentant $S_1 \subseteq S$ et un tableau représentant une partie $T \subseteq S$ et renvoie un tableau `attr` de taille n représentant $\text{Attr}(T)$. Donner la complexité de la fonction.

1.3 Jeux de Büchi

On suppose dans cette partie que $T \subseteq S$ et que $W = \{\sigma \in \mathcal{C}(G) \mid |\sigma|_T = \infty\}$, c'est-à-dire qu'une partie est gagnée par le joueur 1 si elle visite infiniment souvent un sommet de T.

Question 10 Dans cette question, on suppose que le jeu est à un seul joueur, c'est-à-dire que $S_2 = \emptyset$. On considère $s \in S$. Donner une condition nécessaire et suffisante en termes d'existence de chemin et de cycle pour qu'il existe une stratégie gagnante depuis s. Avec quelle complexité temporelle peut-on calculer tous les sommets s tels qu'il existe une stratégie gagnante depuis s ? Détailler.

On suppose pour la suite que $S_1 \neq \emptyset$ et $S_2 \neq \emptyset$. Pour $X \subseteq S$, on définit par induction l'ensemble $\text{Attr}_i^+(X)$, pour $i \in \mathbb{N}$, par :

- $\text{Attr}_0^+(X) = \emptyset$;
- $\text{Attr}_{i+1}^+(X) = \text{Attr}_i^+(X) \cup \{s \in S_1 \mid V(s) \cap (\text{Attr}_i^+(X) \cup X) \neq \emptyset\} \cup \{s \in S_2 \mid V(s) \subseteq \text{Attr}_i^+(X) \cup X\}$.

On pose de plus $\text{Attr}^+(X) = \bigcup_{i \in \mathbb{N}} \text{Attr}_i^+(X)$.

Question 11 Si $X \subseteq S$, montrer que :

1. tout sommet de $\text{Attr}^+(X) \cap S_1$ a un voisin dans $\text{Attr}(X)$;
2. tout sommet de $\text{Attr}^+(X) \cap S_2$ a tous ses voisins dans $\text{Attr}(X)$;

où $\text{Attr}(X)$ a été défini à la partie 1.2.

On définit $E_0(T) = T$ et pour $i \in \mathbb{N}$, $E_{i+1}(T) = \text{Attr}^+(E_i(T)) \cap T$. On pose $E(T) = \bigcap_{i \in \mathbb{N}} E_i(T)$.

Question 12 Déduire de la question précédente une stratégie pour le joueur 1 gagnante depuis $\text{Attr}(E(T))$.

Indication : on pourra commencer par montrer que la suite $(E_i)_{i\in\mathbb{N}}$ est décroissante puis définir la stratégie par ses restrictions sur les ensembles $E(T)$ et $Attr(E(T)) \setminus E(T)$.

Question 13 Montrer que le joueur 2 a une stratégie gagnante depuis $S \setminus \text{Attr}(E(T))$.

2 Automates de Büchi

2.1 Mots infinis

Soit Σ un alphabet. On appelle **mot** sur Σ une suite u, finie ou infinie, de lettres de Σ, $(a_i)_{1\leqslant i<m}$, avec $m \in \mathbb{N}^* \cup \{\infty\}$. La **taille** de u, notée $|u|$ est $m-1$ si $m < \infty$. L'unique mot de taille 0 sera noté ε. On note Σ^* l'ensemble des mots finis et Σ^ω l'ensemble des mots infinis. L'ensemble de tous les mots est noté $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$.

Question 14 Montrer qu'un mot non vide (fini ou infini) peut être considéré comme un chemin dans un graphe à définir en fonction de Σ.

Si $u = (a_i)_{1\leqslant i<m} \in \Sigma^*$ est un mot fini et $v = (b_j)_{1\leqslant j<n} \in \Sigma^\infty$ est un mot fini ou infini, on définit la **concaténation** de u et v, comme le mot $uv = (c_k)_{1\leqslant k<n+m-1}$ défini par :

- $c_k = a_k$ si $1 \leqslant k < m$;
- $c_k = b_{k-m+1}$ sinon.

On remarque que si v est un mot infini, alors uv est également un mot infini.

Question 15 Soit $u \in \Sigma^*$ un mot fini non vide. Montrer qu'il existe un unique mot infini $u^\omega \in \Sigma^\omega$ tel que $u^\omega = uu^\omega$.

Question 16 On considère $u = (a_i)_{i\in\mathbb{N}^*}$ tel que $a_i = a$ si i est un carré et $a_i = b$ sinon. Montrer qu'il n'existe pas de mots $v, w \in \Sigma^*$ tels que $u = vw^\omega$.

On appelle **langage** une partie de Σ^∞. Si $L \subseteq \Sigma^*$ et $L' \subseteq \Sigma^\infty$, on note $LL' = \{uv \mid u \in L, v \in L'\}$ la concaténation des deux langages. Si $\varepsilon \notin L$, on note L^ω l'unique partie de Σ^ω telle que $L^\omega = LL^\omega$.

On appelle **expression ω-régulière** une expression de la forme $e = e_1 f_1^\omega | e_2 f_2^\omega | \ldots | e_k f_k^\omega$ où $k \in \mathbb{N}^*$ et pour $i \in [\![1, k]\!]$, e_i est une expression régulière et f_i est une expression régulière dont l'interprétation ne contient pas le mot vide. L'interprétation d'une telle expression ω-régulière est le langage de mots infinis $\mathcal{L}(e) = \mathcal{L}(e_1)\mathcal{L}(f_1)^\omega \cup \mathcal{L}(e_2)\mathcal{L}(f_2)^\omega \cup \ldots \cup \mathcal{L}(e_k)\mathcal{L}(f_k)^\omega$.

Question 17 Donner, sans justifier, des expressions ω-régulière dont l'interprétation correspond aux langages suivants sur $\Sigma = \{a, b\}$:

1. Les mots infinis qui ne contiennent qu'un nombre fini de b.
2. Les mots infinis où toute position paire contient un a.
3. Les mots infinis tel qu'un a est toujours suivi d'un b.
4. Les mots infinis contenant aucun a **ou** contenant une infinité de a.

2.2 Langage de Büchi d'un automate

On appelle **automate fini sur Σ** un quadruplet $A = (Q, \delta, q_0, F)$ tel que :

- Q est un ensemble fini non vide dit d'**états** ;
- $q_0 \in Q$ est un état dit **initial** ;
- $F \subseteq Q$ est un ensemble d'états dits **finaux** ;
- $\delta : Q \times \Sigma \to \mathcal{P}(Q)$ est une fonction dite de **transition**.

L'automate est dit **déterministe** si pour tout couple $(q, a) \in Q \times \Sigma$, $|\delta(q, a)| \leqslant 1$. Il est dit **non déterministe** sinon.

Attention, que l'automate soit déterministe ou non, il ne possède qu'un seul état initial. De plus, on suppose que l'automate ne contient pas d'ε-transitions.

On étend naturellement la définition de δ à $\delta : \begin{array}{ccc} \mathcal{P}(Q) \times \Sigma & \to & \mathcal{P}(Q) \\ (X, a) & \mapsto & \bigcup\limits_{q \in X} \delta(q, a) \end{array}$. On définit alors la **fonction de transition étendue** $\delta^* : Q \times \Sigma^* \to \mathcal{P}(Q)$ par induction par :

- $\delta^*(q, \varepsilon) = \{q\}$;
- si $u \in \Sigma^*$ et $a \in \Sigma$, $\delta^*(q, ua) = \delta(\delta^*(q, u), a)$.

Si $u = (a_i)_{1 \leqslant i < m} \in \Sigma^\infty$, on appelle **calcul de** u **dans** A toute suite de sommets $c = (q_i)_{0 \leqslant i < m}$ telle que $q_{i+1} \in \delta(q_i, a_{i+1})$ dès lors que $0 \leqslant i$ et $i + 1 < m$. Si $q \in Q$, on note $|c|_q$ le nombre d'occurrences de q dans c (potentiellement infini), c'est-à-dire $|c|_q = |\{i \in [\![0, m[\![\mid q_i = q\}|$. De même, si $X \subseteq Q$, on note $|c|_X = |\{i \in [\![0, m[\![\mid q_i \in X\}|$. On note $\mathcal{C}_u(A)$ l'ensemble des calculs de u dans A.

Enfin, on appelle **langage reconnu par** A l'ensemble $L(A) = \{u \in \Sigma^* \mid \delta^*(q_0, u) \cap F \neq \emptyset\}$ et **langage Büchi-reconnu par** A l'ensemble des mots dont il existe un calcul qui passe une infinité de fois par un état final, c'est-à-dire $L_B(A) = \{u \in \Sigma^\omega \mid \exists c \in \mathcal{C}_u(A), |c|_F = \infty\}$. Par exemple, le langage Büchi-reconnu par l'automate A_0 de la figure 2 est le langage des mots sur $\Sigma = \{a, b\}$ contenant une infinité de a.

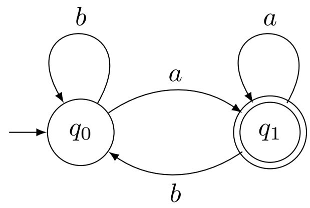

FIGURE 2 – L'automate A_0.

Question 18 Déterminer, en justifiant, le langage Büchi-reconnu par l'automate A_1 de la figure 3. On donnera une expression ω-régulière dont l'interprétation est $L_B(A_1)$.

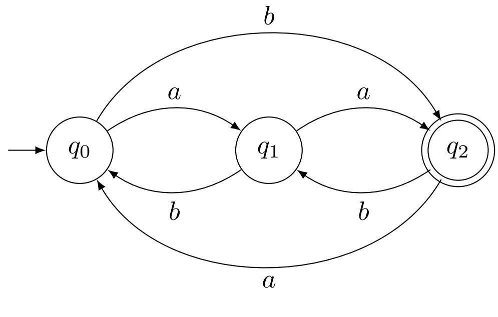

FIGURE 3 – L'automate A_1.

On définit l'expression ω-régulière $e_0 = (a|b)^* a^\omega$ sur $\Sigma = \{a, b\}$.

Question 19 Déterminer, sans justifier, un automate non déterministe A tel que $L_B(A) = \mathcal{L}(e_0)$.

Question 20 Montrer qu'il n'existe pas d'automate déterministe A tel que $L_B(A) = \mathcal{L}(e_0)$.

Question 21 On considère le problème de décision `LANGUAGE NON VIDE` :

* **Instance** : un automate fini A.
* **Question** : est-ce que $L_B(A) \neq \emptyset$?

Montrer que le problème `LANGUAGE NON VIDE` se réduit à un problème d'existence d'une stratégie gagnante dans un jeu de Büchi à un joueur. En déduire qu'il est décidable.

2.3 Propriétés de clôture et équivalence

On cherche à montrer le théorème suivant.

Théorème

Soit $L \subseteq \Sigma^\omega$. Alors L est Büchi-reconnu par un automate fini si et seulement si L est l'interprétation d'une expression ω-régulière.

Question 22 Montrer le sens direct du théorème.

Indication : on pourra considérer les langages $L_{p,q} = \{u \in \Sigma^* \mid q \in \delta^*(p,u)\}$ *pour* $(p,q) \in Q^2$.

Pour les questions 23, 24 et 25, on donnera la construction de l'automate et une explication succincte de sa correction. On n'attend pas de preuve détaillée de l'égalité des langages.

Question 23 Soit $A = (Q, \delta, q_0, F)$ un automate fini tel que $q_0 \notin F$. Montrer qu'il existe un automate fini A^ω tel que $L_B(A^\omega) = L(A)^\omega$.

Question 24 Soient A et B deux automates finis. Montrer qu'il existe un automate fini $A_\cup$ tel que $L_B(A_\cup) = L_B(A) \cup L_B(B)$.

Question 25 Soient A et B deux automates finis. Montrer qu'il existe un automate fini C tel que $L_B(C) = L(A)L_B(B)$.

Question 26 En déduire le sens réciproque du théorème.

On suppose que $A = (Q_A, \delta_A, q_A, F_A)$ et $B = (Q_B, \delta_B, q_B, F_B)$ sont deux automates finis.

On définit $A_\cap = (Q, \delta, q_0, f)$ par :

- $Q = Q_A \times Q_B \times \{1,2\}$;
- $q_0 = (q_A, q_B, 1)$;
- $F = Q_A \times F_B \times \{2\}$;
- pour $(p_1, q_1) \in Q_A^2$, $(p_2, q_2) \in Q_B^2$, $(i,j) \in \{1,2\}^2$ et $a \in \Sigma$, $(q_1, q_2, j) \in \delta((p_1, p_2, i), a)$ si et seulement si $q_1 \in \delta_A(p_1, a)$ et $q_2 \in \delta_B(p_2, a)$ et l'une des conditions suivantes est vérifiée :
  * $i = 1$ et $j = \begin{cases} 1 & \text{si } p_1 \notin F_A \\ 2 & \text{sinon} \end{cases}$
  * $i = 2$ et $j = \begin{cases} 1 & \text{si } p_2 \in F_B \\ 2 & \text{sinon} \end{cases}$

Question 27 Montrer que $L_B(A_\cap) = L_B(A) \cap L_B(B)$.

Question 28 On suppose A déterministe. Montrer que $\Sigma^\omega \setminus L_B(A)$ n'est pas nécessairement Büchi-reconnu par un automate fini déterministe.

⋆⋆⋆