Exercice 1: Graphes parfaits

Dans tout le sujet, on considère des graphes non orientés $G = (S, A)$.

Définition

Un **coloriage** d'un graphe $G = (S, A)$ est une application $S \to \mathbb{N}$. Dans ce contexte, on appelle les entiers des **couleurs**. Un coloriage est **valide** si toute paire de sommets adjacents a des couleurs différentes. Le **nombre chromatique** de G, noté $\chi(G)$, est le nombre minimal de couleurs nécessaires pour créer un coloriage valide de G.

Une **clique** d'un graphe $G = (S, A)$ est un sous-ensemble de sommets $C \subseteq S$ tel que le sous-graphe $G[C]$ induit par C est un graphe complet, c'est-à-dire $G[C] = (C, \mathcal{P}_2(C))$. On note $\omega(G)$ la cardinalité de la plus grande clique de G.

Une **anticlique** d'un graphe $G = (S, A)$ est un sous-ensemble de sommets $B \subseteq S$ tel que le sous-graphe $G[B]$ induit par B ne contient pas d'arêtes, c'est-à-dire $G[B] = (B, \emptyset)$. On note $\alpha(G)$ la cardinalité de la plus grande anticlique de G.

1. Soit G un graphe quelconque. Montrer $\chi(G) \geqslant \omega(G)$.
2. Soit $G = (S, A)$ un graphe quelconque.
   (a) Montrer qu'un coloriage valide de G avec c couleurs existe si et seulement s'il existe une partition $\{B_1, \ldots, B_c\}$ de S en c anticliques.
   (b) Montrer $\chi(G)\alpha(G) \geqslant |S|$.

Définition

Un graphe G est dit **parfait** si tous ses sous-graphes induits $G[T]$ satisfont : $\chi(G[T]) = \omega(G[T])$.

Un graphe G est dit **imparfait minimal** s'il n'est pas parfait et que tous ses sous-graphes induits propres sont parfaits.

3. Donner un exemple de graphe parfait et de graphe imparfait minimal.

Pour simplifier les notations, dans les questions 4 à 8, on fixe G un graphe imparfait minimal quelconque. Sans perte de généralité, on pose $S = \{1, \ldots, n\}$. On note $\alpha = \alpha(G)$, $\omega = \omega(G)$ et $\chi = \chi(G)$.

4. Soit B une anticlique de G. Montrer que $\omega(G[S \setminus B]) = \omega$.
5. Soit B_0 une anticlique de G de cardinalité α. Montrer qu'il existe $\alpha\omega$ anticliques $B_1, \ldots, B_{\alpha\omega}$ telles que pour chaque sommet $s \in S$, s fait partie d'exactement α anticliques parmi $B_0, \ldots, B_{\alpha\omega}$.
6. On considère une suite d'anticliques $B_0, \ldots, B_{\alpha\omega}$ définie comme dans la question précédente. Montrer que pour tout i dans $\{0, \ldots, \alpha\omega\}$, il existe une clique C_i telle que $C_i \cap B_i = \emptyset$ et $\forall j \neq i$, $|C_i \cap B_j| = 1$.

   Indication : utiliser la question 4.

Définition

Étant donné une suite $S_0, \ldots, S_{\alpha\omega}$ de sous-ensembles de $S = \{1, \ldots, n\}$, on définit la **matrice d'incidence** de la suite (S_i) comme la matrice $M = (M_{i,j})_{1 \leqslant i \leqslant n, 0 \leqslant j \leqslant \alpha\omega} \in \{0, 1\}^{n \times (\alpha\omega+1)}$ définie par $M_{i,j} = 1$ si $i \in S_j$, 0 sinon.

7. Soit M_B la matrice d'incidence de la suite $B_0, \ldots, B_{\alpha\omega}$ de la question 5, et M_C la matrice d'incidence de la suite $C_0, \ldots, C_{\alpha\omega}$ de la question 6. On note $M^\top$ la transposée de M.

   Montrer que $M_B^\top M_C$ est de rang $\alpha\omega + 1$, où les matrices sont vues comme à coefficients dans $\mathbb{Q}$.
8. Montrer que $n \geqslant \alpha\omega + 1$.

Dans les questions suivantes, $G = (S, A)$ est un graphe quelconque.

9. Montrer que G est parfait si et seulement si $\omega(G[T])\alpha(G[T]) \geqslant |T|$ pour tout $T \subseteq S$.
10. Soit $\overline{G} = (S, \mathcal{P}_2(S) \setminus A)$ le **graphe complémentaire** de G. Montrer que G est parfait si et seulement si $\overline{G}$ est parfait.

Corrigé

1. Deux sommets d'une même clique sont adjacents, donc doivent être de couleur différente dans un coloriage valide. Par ailleurs, un coloriage valide pour G doit être valide pour tout sous-graphe induit de G. On en déduit le résultat.
2. (a) On considère φ le coloriage. Sans perte de généralité, supposons que $\varphi(S) = \{1, \ldots, c\}$. On pose $B_i = \{s \in S \mid \varphi(s) = i\}$. Par définition, deux sommets de B_i ne sont pas adjacents (car φ est valide), donc S_i forme bien une anticlique. Par ailleurs, l'ensemble des B_i forme bien une partition de S (ils sont non vides et disjoints, et chaque sommet de S a bien une couleur entre 1 et c).

   (b) Soit φ un coloriage utilisant $c = \chi(G)$ couleurs. Par la question précédente :

$$|S| = \sum_{i=1}^{c} |B_i| \leqslant \sum_{i=1}^{c} \alpha(G) = \chi(G)\alpha(G)$$

3. Notons C_k le graphe-cycle de longueur k. Alors C_3 et les C_{2k}, pour $k \geqslant 2$, sont des graphes parfaits. De plus, les C_{2k+1} pour $k \geqslant 2$ sont des graphes imparfaits minimaux. En effet :
   - $\omega(C_3) = 3$ et $\omega(C_k) = 2$ pour $k > 3$;
   - $\chi(C_{2k}) = 2$ et $\chi(3) = \chi(C_{2k+1}) = 3$ pour $k \geqslant 2$;
   - tout sous-graphe induit propre de C_k est une forêt F, qui est bien un graphe parfait (car $\omega(F) = \chi(F) = 2$ si F possède au moins une arête, et $\omega(F) = \chi(F) = 1$ sinon).
4. Si $B = \emptyset$, alors $\omega(G[S \setminus B]) = \omega(G) = \omega$. Sinon, $G' = G[S \setminus B]$ est un sous-graphe induit propre de G, qui est donc parfait par hypothèse. On a donc $\omega(G') = \chi(G')$.

   Or, G étant imparfait minimal, $\chi(G) > \omega(G)$. De plus, $\omega(G') \leqslant \omega(G)$, car G' est un sous-graphe de G. Enfin, $\chi(G') \geqslant \chi(G) - 1$. En effet, à partir d'un coloriage de G', on peut colorier G avec une couleur de plus, en coloriant les sommets de B avec une nouvelle couleur (car ils forment une anticlique). Finalement :
$$\chi(G') = \omega(G') \leqslant \omega(G) \leqslant \chi(G) - 1 \leqslant \chi(G')$$
   Ces inégalités sont donc des égalités, et on a bien $\omega(G') = \omega(G)$.
5. Soit $B_0 = \{s_1, \ldots, s_\alpha\}$ une anticlique de cardinal α. Pour $s_i \in B_0$, $G[-s_i]$ est donc un graphe parfait, et par la question précédente, $\chi(G[-s_i]) = \omega(G[-s_i]) = \omega$. On peut donc partitionner les sommets de $G[-s_i]$ en ω anticliques $B_{i,1}$, $B_{i,2}$, $\ldots$, $B_{i,\omega}$.

   Dès lors, si $s_i \in B_0$, alors s_i appartient à B_0, et pour $j \in [\![1, \alpha]\!]$, $j \neq i$, s_i appartient à exactement l'un des $B_{j,k}$. Si $s \notin B_0$, alors pour $i \in [\![1, \alpha]\!]$, s appartient à exactement l'un des $B_{i,k}$.
6. Par la question 4, $\omega(G[S \setminus B_i]) = \omega(G)$. Il existe donc une clique C_i de taille ω qui est disjointe de B_i.

   Sachant que deux sommets de C_i ne peuvent pas appartenir à la même anticlique, pour $j \in [\![0, \alpha\omega]\!]$, $|C_i \cap B_j| \leqslant 1$.

   Sachant que chaque sommet de C_i appartient à exactement α anticliques, $\sum_{j=0}^{\alpha\omega} |C_i \cap B_j| = \alpha\omega$.

   En combinant les deux informations, on en déduit bien que pour $i \neq j$, $|C_i \cap B_j| = 1$.
7. Par la question précédente, $M_B^\top M_C = (\mathbb{1}_{i \neq j}) = J - I$. Cette matrice est inversible, d'inverse $\frac{1}{\alpha\omega} J - I$. Elle est donc de rang égal à sa dimension, soit $1 + \alpha\omega$.
8. On en déduit que M_T et M_C sont également de rang $1 + \alpha\omega$ (le rang d'un produit est inférieur au maximum des rangs), qui est donc inférieure à l'autre dimension des matrices, c'est-à-dire n.
9. Si G est parfait, alors pour tout T, $G[T]$ est parfait, donc $\omega(G[T]) = \chi(G[T])$ et la question 2.b permet de conclure.

   Réciproquement, montrons que si G n'est pas parfait, alors l'un de ses sous-graphes induits ne vérifie pas l'inégalité. En particulier, l'un des sous-graphes induits $G[T]$ est minimalement imparfait, donc d'après la question précédente, $|T| > \alpha(G[T])\omega(G[T])$.
10. On remarque que $\alpha(\overline{G}) = \omega(G)$ et que $\omega(\overline{G}) = \alpha(G)$. La question 9 permet donc de conclure.

Exercice 2: Ensembles inévitables

Définition

Dans tout le sujet, on fixe $\Sigma = \{a, b\}$ un alphabet. Pour $u \in \Sigma^*$, on écrit $u = u_1 \ldots u_n$ où $n = |u|$ est la **longueur** de u. On dit qu'un mot $u \in \Sigma^*$ **évite** un mot $v \in \Sigma^*$ si v n'apparait **pas** comme facteur de u. On dit que u **évite** un ensemble de mots $V \subseteq \Sigma^*$ s'il évite chaque mot de V.

On dit qu'un ensemble $V \subseteq \Sigma^*$ est **inévitable** s'il n'existe qu'un nombre fini de mots $u \in \Sigma^*$ qui évitent V. Sinon, il est dit **évitable**.

1. Déterminer un mot de longueur au moins 12 qui évite $V = \{aaaa, aaab, aba, baaa, bab, bbbb\}$.
2. Donner le pseudo-code d'un algorithme naïf qui détermine, étant donné un mot $u \in \Sigma^*$ et un ensemble fini $V \subseteq \Sigma^*$, si u évite V. Déterminer sa complexité en temps et en espace quand tous les mots de V font la même longueur.
3. L'ensemble de la question 1 est-il inévitable ? L'ensemble $\{aaa, abb, baa, abab\}$ est-il inévitable ?
4. Montrer que $\{aaa, bab, baab, bbb\}$ est inévitable.
5. Montrer le lemme suivant : pour tout $k \in \mathbb{N}$, il existe $n \in \mathbb{N}$ tel que pour tout mot $u \in \Sigma^*$ tel que $|w| > n$, il existe deux entiers $p < q$ tels que pour tout entier $\ell \in [\![0, k-1]\!]$, $u_{p+\ell} = u_{q+\ell}$.
6. En déduire un algorithme naïf qui, étant donné un ensemble fini $V \subseteq \Sigma^*$, détermine si V est inévitable. Déterminer sa complexité en temps et en espace.
7. Déterminer une méthode permettant de décider si un langage rationnel est inévitable.

Définition

Soit L un langage rationnel et $V \subseteq \Sigma^*$. On dit que V est **inévitable pour** L s'il n'existe qu'un nombre fini de mots de L qui évitent V.

8. Soit L un langage rationnel et $V \subseteq \Sigma^*$ un ensemble fini. Déterminer une méthode permettant de décider si V est inévitable pour L.
9. Montrer que pour tout ensemble inévitable $V \subseteq \Sigma^*$, il existe $V' \subseteq V$ tel que V' est fini et inévitable.
10. Soit L un langage rationnel inévitable. Déterminer une méthode pour calculer un sous-ensemble de L fini et inévitable.

Corrigé

1. Le mot $aabbaabbaabb$ convient.
2. On propose l'algorithme suivant :

```
Entrée : mot u, tableau V de longueur m de mots,
         tableau nV des longueurs des mots de V
Pour i = 1 à m Faire
   Pour j = 1 à n - nV[i] Faire
      k <- 1
      Tant que k <= nV[i] et u[j + k] = V[i][k] Faire
         k += 1
      Si k > nV[i]
         Renvoyer Faux
Renvoyer Vrai
```

   Si les m mots de V sont de taille au plus ℓ, alors la complexité totale est en $\mathcal{O}(n \times m \times \ell)$.
3. On voit que pour tout $k \in \mathbb{N}$, $(aabb)^k$ évite V qui est donc évitable. Pour le deuxième ensemble, tous les mots de b^+ évitent V.
4. On procède par l'absurde. Comme V contient aaa et bbb, un mot suffisamment long u qui évite V doit contenir un facteur ba. Ce facteur doit alors être suivi d'un a ou d'un b. Comme $bab \in V$, on en déduit que u contient un facteur baa. Mais alors ce facteur ne peut être suivi ni de a, ni de b, sinon aaa ou $baab$ seraient des facteurs de u.

5. On pose $n = |\Sigma|^k + k$. Un mot de taille $> n$ mot a strictement plus de $|\Sigma|^k$ positions où commence un facteur de taille k. Comme il existe au plus $|\Sigma|^k$ facteurs de taille k, par le principe des tiroirs, on en déduit l'existence de deux positions où commence un même facteur de taille k.
6. Soit k la longueur maximale d'un mot de V. Montrons que V est évitable si et seulement s'il existe un mot évitant V avec deux occurrences distinctes d'un même facteur de longueur k. Pour le sens direct, il existe une infinité de mots évitant V, en particulier un mot de taille $> n$ du lemme précédent qui permet de conclure. Pour le sens réciproque, soit u un mot évitant V avec deux occurrences d'un mot w de taille k. Sans perte de généralité, on peut supposer que u se termine à la fin de la deuxième occurrence de w. Soit $d > 0$ le décalage entre les deux occurrences de w. On peut alors construire des mots arbitrairement grand en rajoutant une occurrence de w commençant d positions après le début de la dernière occurrence de w, donc V est évitable.

   Ainsi, V est inévitable si et seulement si tous les mots de longueur $> n$ ont un facteur dans V, n étant défini à la question précédente. On en déduit que V est inévitable si et seulement si tous les mots de longueur $n + 1$ ont un facteur dans V. Le sens direct utilise l'équivalence précédente. Pour le sens réciproque, si tout mot de longueur $n + 1$ a un facteur dans V, alors c'est également le cas pour tout mot de longueur $> n + 1$ (on rajoute des lettres).

   Ainsi, il suffit d'énumérer tous les mots de longueur $n+1$ et de tester s'il en existe un qui évite V pour répondre à la question. Comme $n = |\Sigma|^k + k$, et qu'il existe $|\Sigma|^{n+1}$ mots de taille $n+1$, on en déduit, en notant k la taille maximale d'un mot de V et m le nombre de mots de V que la complexité temporelle est $\mathcal{O}(mk4^k2^{2^k})$. La complexité spatiale nécessite le stockage d'un mot de longueur $n + 1$, soit $\mathcal{O}(2^k)$.
7. La question revient à savoir si l'ensemble des mots qui évitent L est fini ou non, c'est-à-dire si $\Sigma^* \setminus \Sigma^* L \Sigma^*$ est fini ou non. À partir d'un automate reconnaissant L, on peut construire un automate reconnaissant $\Sigma^* L \Sigma^*$ (en rajoutant 2 états et des transitions). On déterminise ensuite cet automate. On peut alors construire un automate reconnaissant $\Sigma^* \setminus \Sigma^* L \Sigma^*$ (on complète l'automate et on inverse état final et non final). Enfin, on émonde (suppression des états non accessibles et non co-accessibles) l'automate ainsi obtenu et il suffit de vérifier l'existence de cycles. En termes de complexité, les étapes sont coûteuses à partir de la déterminisation qui peut former un automate de taille exponentielle en la taille de l'automate initial.
8. Comme précédemment, cela revient à décider si $L \cap (\Sigma^* \setminus \Sigma^* V \Sigma^*)$ est fini ou non. Il est possible de construire un automate pour ce langage, en rajoutant l'étape d'intersection (automate produit).
9. Soit V inévitable. Soit n la longueur maximale d'un mot qui évite V. Ainsi, tout mot de longueur $n + 1$ a un facteur dans V. Soit V' l'ensemble de ces facteurs. C'est un ensemble fini et inévitable, d'après une propriété prouvée précédemment.
10. Il suffit de suivre la méthode précédente : si $L \cap (\Sigma^* \setminus \Sigma^* V \Sigma^*)$ est fini, on peut déterminer la longueur maximale n d'un de ses mots. On peut alors générer tous les mots de longueur $n + 1$ et ne garder que les facteurs qui sont dans V.

Exercice 3: Nombre prescrit de chemins

Définition

Un **graphe orienté** est une paire $G = (S, A)$ où $A \subseteq S \times S$ est l'ensemble des arêtes. Un **chemin** de $s \in S$ à $t \in S$ est une séquence $s_1, \ldots, s_n$ de sommets telle que $s_1 = s$, $s_n = t$ et pour tout $i \in [\![1, n-1]\!]$, $(s_i, s_{i+1}) \in A$. Un **graphe pointé** est un triplet (G, s, t) où $G = (S, A)$ est un graphe orienté et $s, t \in S$. On dit qu'un graphe pointé **réalise** un entier $n \in \mathbb{N}$ s'il existe exactement n chemins de s à t dans G.

1. Construire un graphe pointé qui réalise 3.
2. Pour tout $n \in \mathbb{N}$, proposer une construction naïve d'un graphe pointé qui réalise n et détailler son nombre de sommets.
3. Quels sommets peuvent être ignorés dans l'étude d'un graphe pointé réalisant un entier $n \in \mathbb{N}^*$? Quelles hypothèses peut-on alors faire sur les arêtes ?
4. Pour $k \in \mathbb{N}$, donner une preuve constructive de l'existence d'un graphe pointé à $k+2$ sommets réalisant 2^k.
5. Pour $n \in \mathbb{N}^*$, construire un graphe pointé à $\lceil \log_2 n \rceil + 2$ sommets réalisant n.
6. Cette construction est-elle optimale ?

Définition

Pour la suite de l'exercice, on appelle **automate** un automate fini déterministe $A = (Q, \Sigma, \delta, q_0, F)$ sur l'alphabet $\Sigma = \{a, b\}$ où $|F| = 1$. On dit qu'un automate **réalise** un entier $n \in \mathbb{N}$ si le langage reconnu par A contient exactement n mots.

7. Montrer que si un automate A réalise un entier $n \in \mathbb{N}$ avec un nombre minimal d'états, on peut supposer que l'automate est standard, que l'état final n'a pas de transitions sortantes et que tout autre état a exactement 2 transitions sortantes.
8. Montrer que pour tout $n > 0$, il existe un automate qui réalise n dont le nombre d'états est $\lfloor \log_2 n \rfloor$ plus un état par bit égal à 1 dans l'écriture en base 2 de n.

Corrigé

1. De nombreuses possibilités, on peut se contenter de la version $n = 3$ de la question suivante.
2. On considère le graphe à $n+2$ sommets : $V = \{s, t\} \cup \{u_i | i \in [\![1, n]\!]\}$ et $E = \{(s, u_i) | i \in [\![1, n]\!]\} \cup \{(u_i, t) | i \in [\![1, n]\!]\}$.
3. En utilisant le vocabulaire des automates, on peut considérer le graphe « émondé », c'est-à-dire où on a supprimé les sommets non accessibles (depuis s) ou non co-accessibles (depuis t). En effet, ces sommets ne peuvent pas appartenir à un chemin de s à t. De plus, on peut considérer que le graphe ne possède pas de cycle, car un chemin doit être constitué de sommets distincts.
4. Considérons le graphe $G = (V, E)$ où $V = \{u_i | i \in [\![0, k+1]\!]\}$ et $E = \{(u_i, u_j) | 0 \leqslant i < j \leqslant k+1\}$ et montrons par récurrence décroissante que (G, u_i, u_{k+1}) réalise 2^{k-i} pour $i \in [\![0, k]\!]$.
   - si $i = k$: il n'existe qu'un seul chemin de u_k à u_{k+1}, à savoir l'arête (u_k, u_{k+1}) ;
   - supposons le résultat vrai pour $i \in [\![1, k]\!]$ et montrons qu'il reste vrai pour $i - 1$: un chemin de u_{i-1} à u_{k+1} est constitué d'une arête de u_{i-1} à u_j où $j \in [\![i, k+1]\!]$, suivi d'un chemin de u_j à u_{k+1}. En utilisant l'hypothèse de récurrence, le nombre total de tels chemin est :

   $$\sum_{j=i}^{k} 2^{k-j} + 1 = 2^{k-i+1}$$

   le $+1$ dans l'expression ci-dessus étant le chemin qui est l'arête (u_{i-1}, u_{k+1}).

   On conclut par récurrence.
5. Si n est une puissance de 2, on peut utiliser le graphe précédent. Si n n'est pas une puissance de 2, l'idée est de repartir du graphe précédent pour $k = \lfloor \log_2 n \rfloor$, de rajouter un sommet et des arêtes. En notant $G = (V, E)$ le graphe précédent avec les mêmes notations, on pose $V' = \{s\} \cup V$ et en notant $n = \sum_{i=0}^{k} b_i 2^i$ sa décomposition binaire, on pose $E' = \{(s, u_{k-i}) | b_i = 1\} \cup E$. En effet, le nombre de chemins de s à

u_{k+1} est alors :

$$\sum_{i=0}^{k} b_i 2^{k-(k-i)} = n$$

Avec le sommet rajouté, ce graphe a $k+3 = \lceil \log_2 n \rceil + 2$ sommets.

6. Pour $n = 1$, ce graphe n'est pas optimal, car il existe un graphe pointé à un seul sommet qui réalise 1. Pour $n > 1$, montrons que ce graphe est optimal. Montrons qu'un graphe à $k+2$ sommets a au plus 2^k chemins d'un sommet u à un sommet v. Ce résultat est vrai pour $k = 0$. Supposons le vrai pour $k \in \mathbb{N}$ et soit $G = (V, E)$ un graphe à $k+3$ sommets. Posons $V = \{u_i | i \in [\![0, k+2]\!]\}$, les sommets étant numérotés par tri topologique, c'est-à-dire tel qu'il n'existe pas de chemin de u_i à u_j si $i > j$ (possible car le graphe est supposé acyclique). Alors u_i peut atteindre $k+2-i$ sommets au plus par un chemin, et en sommant en utilisant l'hypothèse de récurrence, on obtient bien le résultat escompté.

   Dès lors, un graphe réalisant $n > 0$ doit avoir au moins $\lceil \log_2 n \rceil + 2$ sommets, ce qui montre l'optimalité de la borne.

7. En considérant l'automate comme émondé, s'il existe une transition sortante depuis l'état final, alors elle est vers un état co-accessible. Cela signifie qu'il existe un cycle dans l'automate et donc une infinité de mots reconnus. Par le même raisonnement, il n'existe aucune transition vers l'état initial, l'automate est donc standard.

   L'automate étant déterministe, chaque état a au plus 2 transitions sortantes (l'alphabet est de cardinal 2). Chaque état non final étant co-accessible, il admet au moins une transition sortante.

   Supposons qu'un état non final n'ait qu'une seul transition sortante, et montrons qu'il existe un automate réalisant n avec un état de moins :

   - si cet état est l'état initial, on le supprime et on considère comme nouvel état initial l'état d'arrivée de cette transition ;
   - si cet état q n'est pas l'état initial, notons q' l'arrivée de la transition sortante. On remplace alors chaque $\delta(p, \alpha) = q$ par $\delta(p, \alpha) = q'$.

   L'automate ainsi construit vérifie toujours les conditions, avec un état de moins.

8. On considère la construction suivante :

   - si $n = 1$, on considère l'automate à 1 état reconnaissant $\{\varepsilon\}$;
   - si n est pair, on construit A l'automate qui réalise $\frac{n}{2}$, on rajoute un nouvel état initial ayant deux transitions étiquetées par a et b vers l'état initial de A ;
   - si n est impair, on construit A l'automate qui réalise $\frac{n-1}{2}$, on rajoute un nouvel état initial et un nouvel état intermédiaire, l'état initial ayant une transition étiquetée par a vers l'état final de A et une transition étiquetée par b vers l'état intermédiaire, et l'état intermédiaire ayant deux transitions vers l'état initial de A.

   Cette construction convient bien pour la réalisation de tout entier n, avec le bon nombre d'états.

Exercice 4: Problème du mot

Soit Σ un alphabet et E un ensemble **d'équations orientées**, $E = \{x_1 \mapsto y_1, \dots, x_n \mapsto y_n\}$ entre mots de Σ^*. Soit G_E un graphe infini dont les sommets sont les mots de Σ^* et dans lequel on a une arête de m vers m' si et seulement s'il existe $u, v \in \Sigma^*$ et $i \in [\![1;n]\!]$ tels que $m = ux_iv$ et $m' = uy_iv$. On note alors $m \rightsquigarrow m'$. On note de plus $m \rightsquigarrow^* m'$ s'il existe un chemin (éventuellement vide) de m vers m' et on appelle m' un **descendant** de m.

Étant donnés deux mots x et y, on veut résoudre le **problème du mot** : déterminer si $x \sim_E y$ où $\sim_E$ est la plus petite relation d'équivalence contenant $\rightsquigarrow$.

1. On se donne $\Sigma = \{a, b, c\}$ et les équations $E = \{ba \mapsto ab, ac \mapsto b\}$. A-t-on $abac \sim_E abca$?
2. Pourquoi ne peut-on pas résoudre ce problème à l'aide des algorithmes habituels de recherche de chemins dans un graphe ? Énoncer une condition suffisante sur E pour qu'un algorithme de recherche de chemins fonctionne.

Définition

En supposant que l'alphabet Σ est totalement ordonné, on définit **l'ordre hiérarchique** $\prec$ sur les mots par : $m \prec m'$ si $|m| < |m'|$ ou $|m| = |m'|$ et, en notant $m = a_1 \dots a_n$ et $m' = b_1 \dots b_n$, il existe $i \in [\![1;n]\!]$ tel que $a_i < b_i$ et $\forall j < i, a_i = b_i$. Un ensemble d'équations est **bien orienté** si pour toute équation $x_i \mapsto y_i$ dans E, on a $x_i \succ y_i$.

3. Montrer qu'il existe une fonction strictement monotone de $(\Sigma^*, \prec)$ dans $(\mathbb{N}, <)$ et en déduire qu'il n'existe pas de séquence infinie strictement décroissante pour $\prec$ dans Σ^*.
   En supposant que E est bien orienté, en déduire que tout mot m n'admet qu'un nombre fini de descendants.

Définition

Un système d'équations orientées est dit **confluent** si $\forall m_0, m_1, m_2 \in \Sigma^*$ avec $m_0 \rightsquigarrow^* m_1$ et $m_0 \rightsquigarrow^* m_2$, il existe $m_3 \in \Sigma^*$ tel que $m_1 \rightsquigarrow^* m_3$ et $m_2 \rightsquigarrow^* m_3$.
Une **forme normale** est un mot m qui n'a d'autres descendants que lui-même.

4. Étant donné un système d'équations bien orienté et confluent, montrer que tout mot admet parmi ses descendants une unique forme normale. En déduire un algorithme qui réussit toujours et résout le problème du mot.

Définition

Un système d'équations orientées est dit **localement confluent** si $\forall m_0, m_1, m_2 \in \Sigma^*$ avec $m_0 \rightsquigarrow m_1$ et $m_0 \rightsquigarrow m_2$, il existe $m_3 \in \Sigma^*$ tel que $m_1 \rightsquigarrow^* m_3$ et $m_2 \rightsquigarrow^* m_3$

5. Le système d'équations $E = \{a \mapsto b, b \mapsto a, a \mapsto c, b \mapsto d\}$ est-il confluent ? Localement confluent ?
6. Montrer qu'un système bien orienté localement confluent est confluent.

Corrigé

1. Dans un premier temps, remarquons que $u \sim_E v$ si et seulement s'il existe un chemin du sommet u à v dans la version non orientée du graphe. De plus, remarquons que $abac \rightsquigarrow aabc$, que $abac \rightsquigarrow abb$, et que les mots $aabc$, abb et $abca$ n'ont aucun descendant.

   Cependant, on remarque que $baca$ est un ancêtre de $abca$, et que $baca \rightsquigarrow bba \rightsquigarrow bab \rightsquigarrow abb$. Enfin, on remarque que $abac \rightsquigarrow abb$. Ainsi, les deux mots sont bien équivalents pour cette relation.
2. Il s'agit d'un problème de connectivité résolu simplement par un parcours dans un graphe fini. Le problème ici est le côté infini du graphe, et arbitrairement long des chemins. Cependant, si pour chaque $i \in [\![1;n]\!]$ on a $|x_i| = |y_i|$, alors les descendants et parents sont tous de même taille, et on peut se limiter à une partie finie du graphe.
3. Si on suppose que $\Sigma = \{a_1, \dots, a_n\}$, alors posons la fonction $\mathsf{num} : a_i \mapsto i$ et $f : u = u_0 \dots u_k \mapsto$

$\sum_{i=0}^{k} n^{k-1-i}\mathsf{num}(u_i)$ (on compte en base n sur k chiffres). La fonction f vérifie bien les conditions requises (c'est même une bijection).

$\mathbb{N}$ étant bien fondé, on sait qu'il n'existe pas de suite infinie strictement décroissante dans $\mathbb{N}$, donc dans Σ^* non plus.

4. Dans un premier temps, montrons l'existence : si on suppose qu'un mot n'a pas de forme normale, alors il possède une suite infinie strictement décroissante de descendants. E étant bien orienté, c'est absurde.

   Supposons qu'un mot u admette deux formes normales v et w. Alors $u \rightsquigarrow^* v$ et $u \rightsquigarrow^* w$. Cependant, E étant confluent, il existe x tel que $v \rightsquigarrow^* x$ et $w \rightsquigarrow^* x$. Cependant, comme on a supposé que v et w étaient des formes normales, cela implique que $v = w = x$.

   On propose l'algorithme suivant pour résoudre le problème du mot pour x et y :

   (a) On cherche une forme normale pour x et pour y : pour chercher une forme normale, on parcourt le mot x jusqu'à trouver un facteur x_i, qu'on modifie en y_i. On recommence tant que c'est possible (un nombre fini de fois).

   (b) Si x et y ont la même forme normale, alors $x \sim_E y$. Sinon, ils ne sont pas équivalents.

5. Le système n'est pas confluent, car c et d sont des descendants de a, mais n'ont pas de descendant commun.

   Pour vérifier qu'il est localement confluent, il suffit de vérifier que deux descendants directs d'un même mot ont toujours un descendant commun. Il s'agit donc juste de vérifier que (b, c) ont un descendant commun (c convient) et que (a, d) ont un descendant commun (d convient).

6. On prouve ce résultat par induction. Pour un mot u donné, on appelle **hauteur** du mot la plus grande taille d'un chemin de u à une forme normale. Cette grandeur est bien définie dans l'hypothèse où E est bien orienté.

   Soient u, v, w tels que $u \rightsquigarrow^* v$ et $u \rightsquigarrow^* w$. Montrons qu'il existe x tel que $v \rightsquigarrow^* x$ et $w \rightsquigarrow^* x$. E étant bien orienté, les chemins sont de taille finie. Posons n la hauteur de u.

   - Si $n = 1$, le résultat est immédiat (par confluence locale).
   - Sinon, supposons le résultat vrai pour tout $k \leqslant n$, $n \in \mathbb{N}^*$ fixé. Soient v_1 et w_1 tels que $u \rightsquigarrow v_1 \rightsquigarrow^* v$ et $u \rightsquigarrow w_1 \rightsquigarrow^* w$. Par confluence locale, il existe x_1 un descendant commun à v_1 et w_1. Par hypothèse d'induction, il existe x_2 un descendant commun à v et x_1. Par hypothèse d'induction à nouveau, il existe x un descendant commun à x_2 et w. x est bien un descendant commun à v et w.