Problème d'informatique

Y. Lemaire

Bissections du monoïde libre

I Question de programmation

A tout automate déterministe complet $\mathcal{A} = (Q, \{d\}, F, \bullet)$ sur l'alphabet A, on associe le graphe (Q, G) dont les sommets sont les états de $\mathcal{A}$ et l'ensemble G des arcs est défini récursivement par les règles suivantes :

- si $q \in Q \setminus \{d\}$ alors $(q, d) \in G$;
- soient $(p, q) \in G$ et $a \in A$: si $p' = p \bullet a \neq q' = q \bullet a$ alors $(p', q') \in G$.

1. *Donner la représentation sagittale du graphe associé à l'automate de la figure 1.*

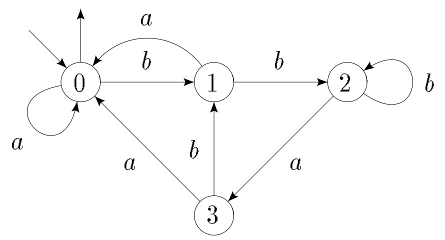

FIG. 1: *Un AFDC*

2. On suppose que $Q = \{0, 1, \dots, n-1\}$, que $d = 0$ et que les lettres de l'alphabet A sont numérotées de 0 à $p-1$: $A = \{a_0, \dots, a_{p-1}\}$. Alors la fonction de transition $\bullet$ peut être représentée par la matrice $(i \bullet a_j)_{i=0,\dots,n-1; j=0,\dots,p-1}$ de dimension $n \times p$. De plus on représente le graphe (Q, G) par sa matrice d'adjacence $(g_{i,j})_{i,j=0,\dots,n-1}$ définie par $g_{i,j} = 1$ si $(i, j) \in G$ et 0 sinon.

Ecrire une fonction CAML `graphe_associé : int vect vect -> int vect vect` *telle que, si Δ est la matrice représentant la fonction de transition d'un automate déterministe complet,* `graphe_associé` Δ *renvoie la matrice d'adjacence du graphe associé.*

Dans tout la suite du problème on fixe un alphabet fini A. On note μ le mot vide de A^* et a et b deux lettres distinctes de A (pour les exemples). On rappelle que tout automate est équivalent à un automate déterministe biaccessible (c.-à-d. dans lequel tous les états sont accessibles et coaccessibles).

II Codes préfixes et sous monoïdes unitaires de A^*

3. *Soit $L \subset A^*$. Montrer l'équivalence des trois propriétés suivantes:*

(i) $\mu + L^2 \subset L$ (ii) $L = L^*$ (iii) $\exists M \subset A^* : L = M^*$

Si les conditions (i), (ii), (iii) sont vérifiées, on dit que L est un *sous monoïde* de A^*. Si $L = M^*$, on dit que le langage M *engendre* le sous monoïde L. $B \subset A^*$ est appelé un *code* si l'écriture d'un mot de B^* comme produit d'éléments de B est unique; autrement dit, si la condition suivante est vérifiée :

$$\forall r, s \in \mathbf{N}, \forall x_1, \dots, x_r, y_1, \dots, y_s \in B : (x_1 \dots x_r = y_1 \dots y_s \Rightarrow r = s \text{ et } \forall i,\ x_i = y_i)$$

Un sous monoïde $L = B^*$ engendré par un code B est dit *libre*. On notera qu'un code ne peut pas contenir le mot vide μ, que $\emptyset$ est un code (donc $\mu = \emptyset^*$ est libre) et que A est un code (A^* est donc un sous monoïde libre de lui-même).

4. *Montrer que si $L = B^*$ où B est un code, alors $B = (L \setminus \mu) \setminus (L \setminus \mu)^2$.*

Si L est un sous monoïde libre, il est donc engendré par un unique code B ($L = B^*$) appelé la *base* de L.

5. *Parmi les langages suivants, quels sont ceux qui sont des codes?*

a. $B = a + ab + ba$; b. $C = ab + ba$; c. $D = aa + ba + baa + bb + bba$.

On dit qu'un langage $B \subset A^*$ est un langage *préfixe* si aucun préfixe strict d'un mot de B n'est dans B, autrement dit si : $\forall u, v \in A^*$, $(uv \in B$ et $v \neq \mu) \Rightarrow u \notin B$. On notera que si un langage préfixe contient μ, il est réduit à μ.

6. *Montrer qu'un langage préfixe différent de μ est un code.*

Un sous monoïde libre de la forme B^* où B est un code préfixe (c.-à-d. un langage préfixe $\neq \mu$) sera appelé un sous monoïde *unitaire*.

7. On pose, dans cette question, $L = \mu + (a+b)^*a$.

a. *Montrer que L est un sous monoïde unitaire et déterminer sa base B.*

b. *Pour chacun des deux langages B et L, donner un automate déterministe à deux états le reconnaissant.*

8. Soit $B \subset A^*$ un langage reconnaissable et soit $\mathcal{A} = (Q, \{d\}, F, \bullet)$ un automate déterministe biaccessible reconnaissant B.

a. *Montrer que le langage B est préfixe si et seulement si aucune transition n'admet un état final pour origine.*

b. On suppose que B est un code préfixe non vide ($\mathcal{A}$ vérifie donc la condition de la question précédente et, en outre, $d \notin F$ car $B \neq \mu$).

Expliquer pourquoi on peut supposer que F est un singleton: $F = \{f\}$. Montrer comment définir une fonction partielle $\bullet'$ sur $Q \times A$ pour que l'automate $\mathcal{A}' = (Q, \{f\}, \{f\}, \bullet')$ reconnaisse le langage unitaire $L = B^*$ (on fera en sorte que les transitions de $\mathcal{A}'$ d'origine $\neq f$ soient les mêmes que celles de $\mathcal{A}$).

c. On pose par exemple $B = (ab)^*(aa + b)$.

Donner un automate déterministe à trois états reconnaissant B (et vérifier sur cet automate que B est un code préfixe). Appliquer ensuite la construction de la question précédente pour trouver un automate déterministe reconnaissant B^*.

On appelle *automate unitaire* un automate déterministe admettant un unique état final qui est aussi l'état initial. La question précédente montre que tout sous monoïde unitaire reconnaissable est reconnu par un automate unitaire.

9. *Réciproquement, soit $\mathcal{A} = (Q, \{q_0\}, \{q_0\}, \bullet)$ un automate unitaire. Montrer que le langage L reconnu par $\mathcal{A}$ est un sous monoïde unitaire et donner un procédé permettant de construire un automate déterministe reconnaissant la base B de L.*

III Bissections

On appelle *bissection* de A^* un couple (X, Y) de codes tel que tout mot $u \in A^*$ s'écrit de manière unique $u = vw$ avec $v \in X^*$ et $w \in Y^*$. De manière équivalente, (X, Y) est une bissection si X et Y sont des langages tels que tout $u \in A^*$ s'écrit de manière unique sous la forme $u = x_1x_2 \ldots x_py_1y_2 \ldots y_q$ avec $p, q \in \mathbf{N}$, $x_1, x_2, \ldots, x_p \in X$ et $y_1, y_2, \ldots, y_q \in Y$.

10. *Montrer que si $A = \{a, b\}$ alors (b^*a, b) est une bissection.*

11. On suppose dans cette question que X et Y sont deux langages disjoints contenus dans A^+ (c.-à-d. X et Y ne contiennent pas μ) vérifiant la propriété $YX + A = X + Y$ et on se propose de démontrer que (X, Y) est une bissection.

a. *En utilisant le fait que $(X+Y) \setminus A \subset YX$, montrer que $X^* \setminus \mu \subset AX^*$. En déduire que tout suffixe d'un mot de X^* est encore dans X^*; autrement dit:*

$$\forall u, v \in A^*,\ uv \in X^* \Rightarrow v \in X^*.$$

On pourrait démontrer de même, mais on l'admettra, que

$$\forall u, v \in A^*,\ uv \in Y^* \Rightarrow u \in Y^*.$$

b. *Déduire de la question précédente que* $X^* \cap Y^* = \mu$ *puis que*

$$\forall v, v' \in X^*, w, w' \in Y^*,\ vw = v'w' \Rightarrow v = v' \text{ et } w = w'.$$

c. *Montrer que* X *est un code préfixe.*
On montrerait de même, mais on l'admettra, que Y est un code.

d. *Conclure.*

12. Soit (P, Q) une partition de $A^+ = A^* \setminus \mu$.

a. *Montrer qu'il existe une unique bissection* (X, Y) *de* A^* *telle que* $X \subset P$ *et* $Y \subset Q$. *On pourra chercher à caractériser les mots de* X *et de* Y *de longueur* 1, *puis ceux de longueur* 2 *et ainsi de suite.*

b. On prend ici $P = a + baa$ et $Q = A^+ \setminus P$. *Calculer* X *et l'ensemble des mots de* Y *de longueur* $\leqslant 3$. *Si* Z *est l'ensemble des mots de* Y *de longueur* > 3, *donner deux langages finis* B *et* C *tels que* $Z = B + ZC$ *et* $\mu \notin C$.

IV Première composante d'une bissection

On dit qu'un automate unitaire $\mathcal{A} = (Q, \{d\}, \{d\}, \bullet)$ est *bissecteur* s'il est complet ($\bullet$ est une application) et s'il existe une relation d'ordre $\preccurlyeq$ sur l'ensemble Q des états qui vérifie les deux propriétés suivantes :

(1) $\preccurlyeq$ est compatible avec $\bullet$: $\forall q, q' \in Q,\ \forall a \in A,\ q \preccurlyeq q' \Rightarrow q \bullet a \preccurlyeq q' \bullet a$;

(2) l'état de départ est le plus grand état: $\forall q \in Q,\ q \preccurlyeq d$.

13. *Quelle propriété doit vérifier le graphe* G *associé (au sens de la partie* I*) à un automate unitaire complet* $\mathcal{A}$ *pour que cet automate soit bissecteur? En déduire que l'automate de la figure 1 est bissecteur.*

Soit L un langage reconnaissable. On se propose de démontrer l'équivalence des quatre propriétés suivantes:

(i) il existe un sous monoïde P tel que $L = \{w \in A^* \mid \forall u, v \in A^*,\ w = uv \Rightarrow v \in P\}$;

(ii) L est un sous monoïde et $\forall u, v \in A^*,\ uv \in L \Rightarrow v \in L$;

(iii) L est reconnu par un automate unitaire bissecteur;

(iv) il existe une bissection (X, Y) telle que $L = X^*$.

On adopte le schéma de démonstration suivant:

(i) $\Leftrightarrow$ (ii) question 14,
(ii) $\Rightarrow$ (iii) question 15,
(iii) $\Rightarrow$ (iv) question 16,
(iv) $\Rightarrow$ (ii) a déja été démontré (question 11.a).

14. *Montrer que* (i) $\Leftrightarrow$ (ii).

15. On suppose (ii). On rappelle (il est inutile de le redémontrer) que L est reconnu par son automate des résiduels $\mathcal{A}_0 = (Q_0, \{d_0\}, F_0, \bullet_0)$ défini de la manière suivante: $Q_0 = \{u^{-1}L \mid u \in A^*\}$ où $u^{-1}L = \{v \in A^* \mid uv \in L\}$, $d_0 = L$, $F_0 = \{R \in Q_0 \mid \mu \in R\}$ et $R \bullet_0 a = a^{-1}R$.

Montrer que $\mathcal{A}_0$ *est un automate unitaire bissecteur (on prendra comme ordre sur* Q_0 *la relation d'inclusion).*

16. On suppose (iii). Soit donc $\mathcal{A} = (Q, \{d\}, \{d\}, \bullet)$ un automate unitaire bissecteur reconnaissant L et soit $\preccurlyeq$ une relation d'ordre sur Q vérifiant les propriétés (1) et (2). On définit $M = \{u \in A^* \mid \forall v \neq \mu, (v \text{ est préfixe de } u \Rightarrow d \bullet v \neq d)\}$.

a. *Montrer que tout mot* $u \in A^*$ *s'écrit de manière unique* $u = vw$ *avec* $(v, w) \in L \times M$.

b. *Montrer que* M *est un sous monoïde.*

c. *Montrer que tout préfixe d'un élément de* M *est un élément de* M *et en déduire que* M *est libre.*

d. *Montrer* (iv).

Corrigé

1. Figure 2.

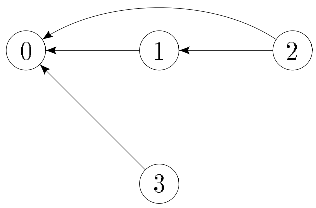

FIG. 2: *Graphe associé à l'automate de la figure 1*

2. Le programme 1 construit au fur et à mesure les arcs de G en appliquant les deux règles:

- on applique d'abord la première règle, ce qui fournit les arcs $(q, 0)$ pour $q \neq 0$, et on stocke ces arcs dans une pile.
- tant que la pile n'est pas vide, on applique la deuxième règle à l'arc situé au sommet de la pile (que l'on supprime de la pile). Chaque nouvel arc est ajouté à la pile.

On a donc l'invariant suivant: les arcs de G sont ceux déja trouvés et ceux qui s'obtiennent en appliquant la deuxième règle aux arcs de la pile.

Comme chaque arc de G est empilé une et une seule fois, le corps de la boucle `while` est éxécuté $\mathrm{Card}(G)$ fois et la complexité est $\mathrm{O}(p \times \mathrm{Card}(G))$.

Programme 1 Calcul du graphe associé à un AFDC

```
#open "stack";;

let graphe_associé delta =
  let n = vect_length delta
  and p = vect_length delta.(0)
  and pile = new() in
  let g = make_matrix n n 0 in
  let ajouter q q' =
    if q <> q' & g.(q).(q') = 0 then (
      g.(q).(q') <- 1;
      push (q,q') pile
      ) in
  for q = 1 to n - 1 do ajouter q 0 done;
  while pile <> new() do
    let (q,q') = pop pile in
    for j = 0 to p - 1 do ajouter delta.(q).(j) delta.(q').(j) done
  done;
  g;;

graphe_associé [|[|0;1|];[|0;2|];[|3;2|];[|0;1|]|];;
(* renvoie [|[|0;0;0;0|];[|1;0;0;0|];[|1;1;0;0|]; [|1;0;0;0|]|]*)
```

3. On démontre (i) $\Rightarrow$ (ii) $\Rightarrow$ (iii) $\Rightarrow$ (i):
(i) $\Rightarrow$ (ii): si (i), alors, par récurrence, $\forall n$, $L^n \subset L$, d'où $L^* \subset L$ et (ii);
(ii) $\Rightarrow$ (iii): évident;
(iii) $\Rightarrow$ (i): facile, sachant que $(M^*)^2 = M^*$.

4. On note $B' = (L \setminus \mu) \setminus (L \setminus \mu)^2$.

$B \subset B'$. En effet : Soit $u \in B$. Si $u = vw$ où $v, w \in L \setminus \mu$ alors les factorisations de v et w comme produits d'éléments de B fournissent une factorisation de $u \in B$ comme produit d'au moins deux éléments de B, ce qui contredit le fait que B est un code. Ainsi, $u \notin (L\backslash\mu)^2$. De plus $u \neq \mu$ donc $u \in L \setminus \mu$ et $u \in B'$.

$B' \subset B$ car si $u \in B'$, comme $u \in L \setminus \mu$, $u = x_1 \dots x_r$ avec $r \geqslant 1$ et $x_1, \dots, x_r \in B$. On a $r = 1$ et $u = x_1 \in B$ car sinon $u = (x_1)(x_2 \dots x_r) \in (L \setminus \mu)^2$.

5.a $a(ba) = (ab)a$ donc B n'est pas un code.

5.b C est un code car tout mot u de C^* est de longueur paire et admet une unique factorisation par des mots de C qui s'obtiennent en découpant u par tranches de deux lettres (plus généralement, si tous les mots d'un langage ont une même longueur $p > 0$, ce langage est un code).

5.c Soit $u \in D^* \setminus \mu$. Si $u = vw$ où $v \in D$ et$w \in D^*$, montrons que (v, w) est unique (ce qui démontrera que D est un code car si $u = x_1 \dots x_r$ où les $x_i \in D$, alors x_1 sera déterminé de manière unique, puis x_2, et ainsi de suite):

Si la première lettre de u est un a, $v = aa$ (ce qui prouve par récurrence que si $N(u)$ est le plus grand entier n tel que a^n soit préfixe de u, alors $N(u)$ est pair).

Si la première lettre de u est b alors, en notant c la deuxième lettre de u, on a $v = bc$ ou $v = bca$. Notons p le plus grand entier tel que bca^n soit préfixe de u. Si p est pair (resp. impair), alors $v = bc$ (resp. $v = bca$) car sinon $N(w)$ serait impair.

6. Soit $u \in B^* \setminus \mu$. Si $u = vw = v'w'$ où $v, v' \in B$ et $w, w' \in A*$, alors $(v, w) = (v', w')$ car si, par exemple, $|v'| > |v|$ alors $v \in B$ serait un préfixe strict de $v' \in B$ ce qui est faux. B est donc bien un code (même raisonnement qu'à la question 5.c).

7.a $B = b^*a$ est un code préfixe car un préfixe strict de $b^n a$ est de la forme b^p et n'appartient donc pas à B. Par ailleurs $B^+ = (a + b)^*a$ (et donc $B^* = \mu + B^+ = L$) car si $u \in (a + b)^*a$, en considérant toutes les occurences de a dans u, on peut écrire $u = b^{k_1}ab^{k_2}a \dots b^{k_p}a$ où les $k_i \in \mathbf{N}$.

7.b Figure 3.

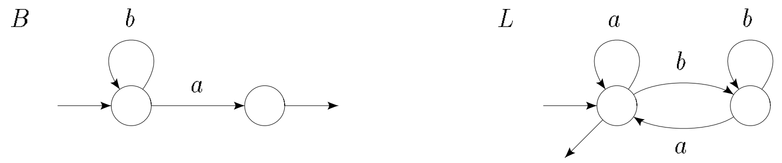

FIG. 3: *Automates reconnaissant respectivement $B = b^*a$ et $L = B^*$*

8.a Si B n'est pas préfixe, il existe $u \in B$ et $v \neq \mu$ tels que $w = uv \in B$. Alors $d \bullet u$ est un état final origine d'au moins une transition puisque $(d \bullet u) \bullet v = d \bullet w$ est défini (et $\in F$).

Réciproquement, s'il existe un état final f origine d'une transition, soit $a \in A$ tel que $f \bullet a$ soit défini; comme f est accessible, il existe $u \in A^*$ tel que $d \bullet u = f$ et comme $f \bullet a$ est coaccessible, il existe $v \in A^*$ tel que $(f \bullet a) \bullet v = f \bullet av \in F$. Alors $d \bullet uav = (d \bullet u) \bullet av \in F$ donc $uav \in B$ et, comme $u \in B$, B n'est pas préfixe.

8.b Pour trouver un automate équivalent à $\mathcal{A}$ ayant un seul état final, on supprime les états finaux et on les remplace par un seul état final sur lequel on fait arriver toutes les transitions qui arrivaient dans $\mathcal{A}$ sur un état de F. Plus précisément, soit $f \notin Q$, on pose $\mathcal{A}_1 = (Q \cup \{f\} \setminus F, \{d\}, \{f\}, \bullet_1)$ où

$$q \bullet_1 a = \begin{cases} q \bullet a & \text{si } q \neq f \text{ et } q \bullet a \notin F \\ f & \text{si } q \neq f \text{ et } q \bullet a \in F \\ \text{non défini} & \text{si } q = f. \end{cases}$$

Montrons que $\mathcal{A}_1$ est équivalent à $\mathcal{A}$:

Si $d \xrightarrow[\mathcal{A}]{a_1} q_1 \cdots \xrightarrow[\mathcal{A}]{a_n} q_n \in F$ est un calcul réussi de $\mathcal{A}$, alors, comme le seul état de ce calcul qui soit dans F est q_n (aucune transition de $\mathcal{A}$ n'ayant pour origine un état final), on obtient un calcul réussi de $\mathcal{A}_1$ en remplaçant q_n par f: $d \xrightarrow[\mathcal{A}_1]{a_1} q_1 \cdots \xrightarrow[\mathcal{A}_1]{a_n} f$

Réciproquement, si $d \xrightarrow[\mathcal{A}_1]{a_1} q_1 \cdots \xrightarrow[\mathcal{A}_1]{a_n} f$ est un calcul réussi de $\mathcal{A}_1$, les états $d, q_1, \dots, q_{n-1}$ sont distincts de f (aucune transition de $\mathcal{A}_1$ n'admettant f comme origine). Par définition de $\bullet_1$, l'état $q_n = q_{n-1} \bullet a_n$ est dans F. On a un calcul réussi de $\mathcal{A}$ en remplaçant f par q_n: $d \xrightarrow[\mathcal{A}]{a_1} q_1 \cdots \xrightarrow[\mathcal{A}]{a_n} q_n$.

Enfin, il est clair que $\mathcal{A}_1$ est biaccessible.

On suppose donc que $F = \{f\}$ et on pose $\mathcal{A}' = (Q, \{f\}, \{f\}, \bullet')$ où

$$q \bullet' a = \begin{cases} q \bullet a & \text{si } q \neq f \\ d \bullet a & \text{si } q = f. \end{cases}$$

Montrons que $\mathcal{A}'$ reconnait B^*: On remarque d'abord que le mot vide appartient à la fois à $|\mathcal{A}'|$ et à B^*; on ne s'occupe donc que des mots $\neq \mu$.

Soit $u \in B^* \setminus \mu$: $u = x_1 \dots x_p$ où $p \geqslant 1$ et les $x_i \in B$. En écrivant $x_i = a_i y_i$ où a_i est la première lettre de x_i, il existe un calcul réussi de $\mathcal{A}$ de la forme $d \xrightarrow[\mathcal{A}]{a_i} q_i \xrightarrow[\mathcal{A}]{y_i} f$. Comme aucune transition de $\mathcal{A}$ n'est d'origine f, la seule occurrence de f dans ce calcul est la dernière. On peut alors trouver un calcul réussi de $\mathcal{A}'$ d'étiquette u: $f \xrightarrow[\mathcal{A}']{a_1} q_1 \xrightarrow[\mathcal{A}']{y_1} f \cdots f \xrightarrow[\mathcal{A}']{a_p} q_p \xrightarrow[\mathcal{A}']{y_p} f$. On a donc bien $u \in |\mathcal{A}'|$.

Réciproquement, si $u \in |\mathcal{A}'| \setminus \mu$, en considérant toutes les occurrences de f dans un calcul réussi de $\mathcal{A}'$ d'étiquette u, on peut écrire un tel calcul sous la forme $f \xrightarrow[\mathcal{A}']{a_1} q_1 \xrightarrow[\mathcal{A}']{y_1} f \cdots f \xrightarrow[\mathcal{A}']{a_p} q_p \xrightarrow[\mathcal{A}']{y_p} f$ où les a_i sont des lettres et où chaque calcul $q_i \xrightarrow[\mathcal{A}']{y_i} f$ ne contient qu'une occurrence de f, de sorte que l'on dispose du calcul $d \xrightarrow[\mathcal{A}]{a_i} q_i \xrightarrow[\mathcal{A}]{y_i} f$ de $\mathcal{A}$ qui prouve que $a_i y_i \in B$. Finalement, $u = (a_1 y_1) \dots (a_p y_p) \in B^*$.

8.c Figure 4.

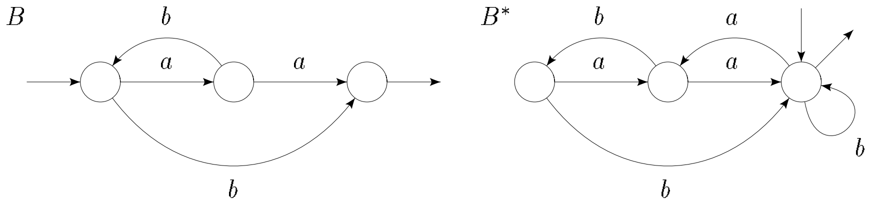

FIG. 4: *Automates reconnaissant respectivement $B = (ab)^*(aa + b)$ et B^**

9. Soit $f \notin Q$; on pose $\mathcal{A}' = (Q \cup \{f\}, \{q_0\}, \{f\}, \bullet')$ où

$$q \bullet' a = \begin{cases} q \bullet a & \text{si } q \neq f \text{ et } q \bullet a \neq q_0 \\ f & \text{si } q \neq f \text{ et } q \bullet a = q_0 \\ \text{non défini} & \text{si } q = f. \end{cases}$$

Le langage B reconnu par $\mathcal{A}'$ est un code préfixe (d'après 8.a) et $B^* = L$ car le procédé de la question 8.b appliqué à $\mathcal{A}'$ redonne l'automate $\mathcal{A}$.

10. b^*a et b sont des codes (voir 7.a) et tout mot u s'écrit de manière unique sous la forme $u = vb^n$ où $v \in \mu + A^*a = (b^*a)^*$ (b^n est le plus long suffixe de u appartenant à b^*).

11.a Soit $u \in X^* \setminus \mu$. Montrons que $u \in AX^*$. On peut écrire $u = xv$ où $x \in X$ et $v \in X^*$. Si $x \in A$, c'est fini. Sinon, $x \in (X + Y) \setminus A \subset YX$ (par hypothèse) donc $x = y_1 x_1$ où $y_1 \in Y$ et $x_1 \in X$. Si $y_1 \in A$, c'est fini (car $u = y_1 x_1 v \in y_1 X^*$). Sinon $y_1 = y_2 x_2$ où $y_2 \in Y$ et $x_2 \in X$. Si $y_2 \in A$, c'est fini, sinon, on continue. Ce processus s'arrête car la suite $|y_i|$ est strictement décroissante (les x_i sont $\neq \mu$).

Soit maintenant $u = a_1 \dots a_n \in X^*$, les $a_i \in A$ étant les lettres de u. De la propriété précédente on tire $a_2 \dots a_n \in X^*$, puis $a_3 \dots a_n \in X^*$, et ainsi de suite jusqu'à $a_n \in X^*$. Tous les suffixes de u (y compris u et μ) sont donc dans X^*.

11.b Soit $u \in X^* \cap Y^* \setminus \mu$: $u = xv = wy$ où $x \in X$, $v \in X^*$, $y \in Y$ et $w \in Y^*$. Chacun des deux mots x et y appartient à $X^* \cap Y^* \setminus \mu$ (en effet y est un suffixe de $u \in X^*$ donc, d'après la question précédente, $y \in X^*$; de même, $x \in Y^*$) mais l'un de ces deux mots est de longueur $<$ à celle de u (car $X \cap Y = \emptyset$). Ce procédé permet donc de construire une suite infinie de mots de $X^* \cap Y^* \setminus \mu$ dont les longueurs décroissent strictement, ce qui est impossible.

Si $v, v' \in X^*$, $w, w' \in Y^*$ sont tels que $vw = v'w'$, supposons par exemple que $|v'| \geqslant |v|$. Alors le suffixe z de v' de longueur $|v'| - |v|$ est aussi le préfixe de w de longueur $|w| - |w'|$. De la question précédente, on tire $z \in X^* \cap Y^*$ donc $z = \mu$ et $v = v'$, d'où $w = w'$.

11.c Si X n'est pas un code préfixe, soit $x \in X$ de longueur minimale admettant un préfixe strict $x' \in X$. Comme $x \in (X + Y) \setminus A \subset YX$, $x = yx_1$ où $(y, x1) \in Y \times X$. x' ne peut pas être préfixe de y (si c'était le cas, on aurait $x' \in X^* \cap Y^* = \mu$) donc $x' = yu$ où $u \in X^* \setminus \mu$. Soit x'_1 un préfixe de u appartenant à X. Alors $x'_1 \in X$ est un préfixe strict de $x_1 \in X$ ce qui contredit la minimalité de $|x|$.

11.d Il reste à montrer que $A^* = X^*Y^*$: Soit $u \in A^*$. Comme $A \subset X + Y$, il existe $z_1, \ldots, z_p \in X + Y$ tels que $u = z_1 \ldots z_p$. Supposons que p est minimal. Alors, pour tout $i = 1, \ldots, p-1$, $z_i \in Y \Rightarrow z_{i+1} \in Y$ (car si $z_{i+1} \in X$, alors on pourrait remplacer les deux facteurs z_i et z_{i+1} par le seul facteur $z_i z_{i+1} \in YX \subset X + Y$, ce qui contredirait la minimalité de p). Donc $u \in X^*Y^*$.

Remarque Voici une indication pour démontrer le résultat admis[1] selon lequel si (X, Y) est une bissection, alors, $YX + A = X + Y$: montrer d'abord

$$(u \in X \text{ et } v \neq \mu) \text{ ou } (v \in Y \text{ et } u \neq \mu) \Rightarrow uv \notin X + Y.$$

12.a *Unicité*: Si (X, Y) convient, notons $X_n = A^n \cap X$ et $Y_n = A^n \cap Y$. Comme $A \subset X + Y$ on a:

$$X_1 = A \cap P \text{ et } Y_1 = A \cap Q \tag{1}$$

et pour $n \geqslant 2$,
$X_n = A^n \cap (X + Y) \cap P = A^n \cap (YX + A) \cap P = A^n \cap YX \cap P = \displaystyle\sum_{p+q=n} Y_p X_q \cap P$. Donc

$$\forall n \geqslant 2,\ X_n = \sum_{p+q=n} Y_p X_q \cap P \text{ et } Y_n = \sum_{p+q=n} Y_p X_q \cap Q \tag{2}$$

Comme dans les sommes intervenant dans (2), p et q sont $< n$, les relations (1) et (2) déterminent les X_n et les Y_n et donc X et Y de manière unique.
Existence: On définit deux suites de langages X_n et Y_n par les relations (1) et (2) et on pose $X = \displaystyle\sum_{n \in \mathbf{N}^*} X_n$ et $Y = \displaystyle\sum_{n \in \mathbf{N}^*} Y_n$. On démontre alors par récurrence que $X_n \subset A^n \cap P$, $Y_n \subset A^n \cap Q$ et $A^n \cap (YX + A) = A^n \cap (X + Y)$, ce qui montre que $X \subset P$, $Y \subset Q$ et $YX + A = X + Y$.

12.b Ici $X_1 = a$, $Y_1 = b$, $X_2 = \emptyset$, $Y_2 = ba$, $X_3 = baa$, $Y_3 = \emptyset$ donc $X = P$ et les mots de Y de longueur $\leqslant 3$ sont b et ba. On a $Z = (b + ba)baa + Z(a + baa)$; on peut en déduire que $Z = (b + ba)baa(a + baa)^*$ puis $Y = b + ba + (b + ba)baa(a + baa)^*$.

13. $\mathcal{A}$ est bissecteur si et seulement si G est sans circuit; ce qui est le cas du graphe de la figure 2.

14. Si (ii) est vérifiée alors on a (i) avec $P = L$.

Réciproquement, si (i) est vérifiée, L est l'ensemble des mots dont tous les suffixes sont dans P. L est alors un sous monoïde. En effet $\mu \in P$ donc $\mu \in L$ et si $u, v \in L$ alors $uv \in L$ car si w est un suffixe de uv, ou bien w est un suffixe de v auquel cas $w \in P$, ou bien $w = u'v$ où u' est un suffixe de u auquel cas $w \in P^2 \subset P$. Enfin, si $uv \in L$ alors $v \in L$ car tout suffixe de v est un suffixe de uv et il appartient donc à P.

15. $\mathcal{A}_0$ est unitaire. En effet $F_0 = \{d_0\}$ car si $R = u^{-1}L \in Q_0$, on a (en utilisant (ii)):

$$\begin{aligned} R \in F_0 &\Leftrightarrow \mu \in u^{-1}L \\ &\Leftrightarrow u \in L \\ &\Leftrightarrow (\forall v,\ uv \in L \Leftrightarrow v \in L) \\ &\Leftrightarrow (\forall v,\ v \in R \Leftrightarrow v \in L) \\ &\Leftrightarrow R = d_0 \end{aligned}$$

1. Une autre démonstration utilise la notion de série formelle.

$\subset$ est compatible avec $\bullet$ car

$$\begin{aligned} R \subset R' &\Rightarrow (\forall v,\ av \in R \Rightarrow av \in R') \\ &\Rightarrow a^{-1}R \subset a^{-1}R' \\ &\Rightarrow R \bullet_0 a \subset R' \bullet_0 a \end{aligned}$$

d_0 est maximum. En effet, pour tout u, $uv \in L \Rightarrow v \in L$ donc $u^{-1}L \subset L = d_0$.

16.a Soit $u \in A^*$ et soit v le plus long préfixe de u tel que $d \bullet v = d$. Alors $u = vw$ avec $v \in L$ et $w \in M$ et il est clair que c'est la seule factorisation de u de cette forme.

16.b Il est clair que $\mu \in M$. Soient $u, v \in M$. Pour montrer que $uv \in M$, il faut montrer que pour tout préfixe $w \neq \mu$ de uv, on a $d \bullet w \neq d$: Si w est un préfixe de u, c'est clair. Sinon $w = ut$ où $t \neq \mu$ est un préfixe de v; alors $d \bullet u \preccurlyeq d$ donc, par compatibilité de $\preccurlyeq$ avec $\bullet$, on a $d \bullet w = d \bullet u \bullet t \preccurlyeq d \bullet t \prec d$.

16.c Un préfixe d'un élément de M est clairement un élément de M. $Y = (M \setminus \mu) \setminus (M \setminus \mu)^2$ est un code. En effet si $uy = vz$ avec $u, v \in Y^*$ et $y, z \in Y$ alors $y = z$ car si, par exemple, $|y| > |z|$, soit z' tel que $z'z = y$; alors z' et z appartiennent à $M \setminus \mu$ donc $y \in (M \setminus \mu)^2$, ce qui est impossible. Finalement, $M = Y^*$ est libre.

16.d L et M sont des sous monoïdes libres (pour L, cela provient de ce qu'il est reconnu par un automate unitaire) donc si on note X et Y leurs bases respectives, X et Y sont des codes tels que $X^* = L$ et, avec a., (X, Y) est une bissection.

Remarque L'équivalence des propriétés (i), (ii) et (iv) reste vraie même si on ne fait pas l'hypothèse que L est reconnaissable. La démonstration est la même à condition d'autoriser des automates à un nombre infini d'états.

Référence

M.Lothaire. *Combinatorics on words*. Cambridge University Press, 1997.