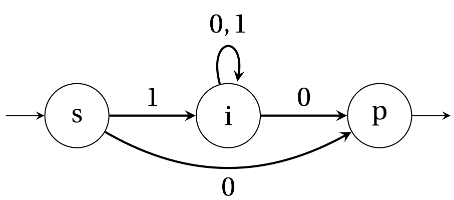
Langages et automates

OPTION INFORMATIQUE - Devoir n° 4 - Olivier Reynet

A Automates

A1. En exhibant un automate et sans justification particulière, montrer que le langage constitué par les entiers binaires pairs est reconnaissable.

Solution : Soit l'alphabet $\Sigma = \{0, 1\}$. En effet, les entiers binaires pairs commencent par un 1, se poursuivent par des 0 ou des 1 et s'achève par un 0. C'est pourquoi l'automate fini non déterministe suivant reconnaît les entiers binaires pairs :

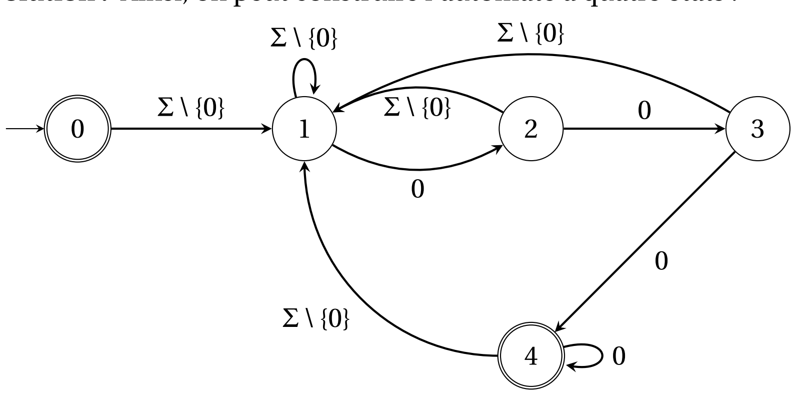


NB : cet automate n'est pas déterministe.

NB : le langage $\mathcal{L}_{ER}((\varepsilon|1(0|1)^*)0)$ est le langage des entiers binaires pairs.

A2. Soit l'alphabet des chiffres en base dix : $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. En exhibant un automate et sans justification particulière, montrer que le langage des multiples de 1000 en base 10 est reconnaissable.

Solution : Ainsi, on peut construire l'automate à quatre états :



Ce langage est donc reconnaissable.

■ **Définition 1 — Fonction de transition étendue aux mots (rappel).** La fonction de transition peut être étendue aux mots par passages successifs d'un état à un autre en lisant les lettres d'un

mot.
On définit inductivement cette fonction étendue noté δ^* :

$$\forall q \in Q, \delta^*(q,\varepsilon) = q \qquad (1)$$
$$\forall q \in Q, \forall w \in \Sigma^*, \forall a \in \Sigma, \delta^*(q, w.a) = \delta(\delta^*(q,w),a) \qquad (2)$$

A3. Montrer que l'opération de complétion d'un automate est correcte, c'est-à-dire que $\mathcal{L}_{rec}(C(\mathcal{A})) = \mathcal{L}_{rec}(\mathcal{A})$, si $C(\mathcal{A})$ est l'automate complété de $\mathcal{A}$.

Solution :

Démonstration. ($\supset$) Soit w un mot reconnu par l'automate $\mathcal{A}$. Alors, comme w est reconnu, cela signifie qu'il existe une transition définie pour chaque lettre du mot w. Ces transistions sont les mêmes dans l'automate $C(\mathcal{A})$, w ne conduit donc jamais à l'état puits. Comme l'ensemble des états accepteurs est le même pour les deux automates, w est reconnu par $C(\mathcal{A})$.

($\subset$) On procède de même symétriquement. Soit w un mot reconnu par $C(\mathcal{A})$. Alors il existe un chemin dans $C(\mathcal{A})$ qui, d'après le mot w, mène à un état accepteur de F, c'est-à-dire toutes les transitions sont définies pas δ et aucune lettre de w ne conduit au puits. Ces transitions seraient donc parcourues de la même manière dans $\mathcal{A}$. Comme l'ensemble accepteur est le même, w est reconnu par $\mathcal{A}$.

Les deux langages sont donc les mêmes. ■

A4. Soit Σ un alphabet et $\mathcal{A}$ l'automate fini déterministe défini par $\mathcal{A} = \left(Q, \Sigma, q_i, \delta, \{q_f\}\right)$. Pour cet automate, on suppose que pour chaque symbole s de Σ, on a $\delta(q_i, s) = \delta(q_f, s)$.

(a) En procédant par induction structurelle sur les mots, démontrer la propriété $\mathcal{P} : \forall u \in \Sigma^* \setminus \{\varepsilon\}, \delta^*(q_i, u) = \delta^*(q_f, u)$.

Solution : On procède par induction structurelle sur les mots définis par la gauche.

(Initialisation) soit s une lettre de Σ. Par hypothèse, $\delta(q_i, s) = \delta(q_f, s)$. s est un mot consituté d'une seulle lettre. Donc :

$$\delta^*(q_i, s) = \delta(q_i, s) = \delta(q_f, s) = \delta^*(q_f, s)$$

La propriété est vérifiée pour un mot d'une lettre.

(Pas d'induction) Soit w un mot de $\mathcal{L}_{rec}(A)$ et s une lettre de Σ. Un mot se construit inductivement pas la gauche. Soit u le mot défini par $u = s.w$.

$$\delta^*(q_i, u) = \delta^*(q_i, sw) = \delta^*(\delta(q_i, s), w) = \delta^*(\delta(q_f, s), w) = \delta^*(q_f, sw) = \delta^*(q_f, u)$$

La propriété est conservée par la règle de construction des mots.

(Conclusion) Comme la propriété est vérifiée pour un mot d'une lettre et que le pas d'induction garantit que tout mot construit vérifie la propriété, $\mathcal{P}$ est vraie pour tout mot non vide sur Σ.

(b) En procédant par récurrence sur k, démontrer la propriété

$$\mathcal{P}_k : \forall u \in \Sigma^* \setminus \{\varepsilon\}, \left(u \in \mathcal{L}_{rec}(A) \Rightarrow \forall k \in \mathbb{N}^\star, u^k \in \mathcal{L}_{rec}(A)\right)$$

Solution :

(Initialisation) Soit $u \in \mathcal{L}_{rec}(A)$. Comme $u^1 = u$, $\mathcal{P}_1$ est vérifiée.

(Hérédité) Soit k un entier naturel non nul. Supposons que $\mathcal{P}_k$ soit vérifiée. Soit $u \in \mathcal{L}_{rec}(A)$. On a $u^{k+1} = u^k u$. Alors :

$$\delta^*(q_i, u^{k+1}) = \delta^*(\delta^*(q_i, u^k), u) = \delta^*(q_f, u)$$

car par hypothèse de récurrence, u^k est reconnu par A et donc l'automate aboutit à l'état accepteur q_f (le seul). De part la propriété de la question précédente, on peut écrire que :

$$\delta^*(q_i, u^{k+1}) = \delta^*(q_f, u) = \delta^*(q_i, u) = q_f$$

car, par hypothèse, u est reconnu par A et $\delta^*(q_i, u)$ aboutit donc à l'état accepteur q_f. On en déduit que u^{k+1} est reconnnu par A.

(Conclusion) Comme $\mathcal{P}_1$ est vérifiée et que l'hérédité a été démontrée, la propriété est vraie pour tout entier naturel k non nul.

A5. Un automate fini déterministe $\mathcal{A}$ est défini par $\mathcal{A} = \left(Q, \Sigma, q_i, \delta, F\right)$.

(a) Proposer un algorithme COMPLEMENTAIRE($\mathcal{A}$) qui construit l'automate complémentaire de $\mathcal{A}$. Ne pas implémenter l'algorithme, le décrire.

Solution : Pour constuire le complémentaire d'un AFD $\mathcal{A}$, on procède comme suit :

1. On constuit le complété de $\mathcal{A} = \left(Q, \Sigma, q_i, \delta, F\right)$, $C(\mathcal{A}) = \left(Q \cup \{p\}, \Sigma, q_i, C(\delta), F\right)$, ajout de l'état puits p et ajout des toutes les transitions non définies à destination de p.
2. On construit le complémentaire de $\mathcal{A}$ à partir de $C(\mathcal{A})$: $\bar{\mathcal{A}} = \left(Q \cup \{p\}, \Sigma, q_i, C(\delta), (Q \cup \{p\}) \cap \bar{F}\right)$, tous les états non accpeteurs de $\mathcal{A}$ deviennent accepteurs et les états accepteurs de $\mathcal{A}$ ne le sont plus.

(b) Proposer un algorithme ACCESSIBLES($\mathcal{A}$) qui renvoie l'ensemble des états accessibles d'un automate $\mathcal{A}$. Ne pas implémenter l'algorithme, le décrire.

Solution : Pour cela, il suffit de parcourir en largeur le graphe de l'automate $\mathcal{A}$ à partir de l'état de départ q_i. On obtient ainsi la liste de tous les sommets accessibles depuis le départ.

(c) Proposer un algorithme EST_VIDE($\mathcal{A}$) qui permet de savoir si le langage reconnu par un automate est vide. La donnée d'entrée est de type automate, la valeur renvoyée est booléenne. Ne pas implémenter l'algorithme, le décrire. Il est possible de se servir des algorithmes précédents.

Solution : On procède comme suit :

1. $E \leftarrow$ ACCESSIBLES($\mathcal{A}$)
2. on renvoie vrai si $E \cap F = \emptyset$ car cela signifie qu'aucun état accepteur n'est accessible et donc qu'aucun mot ne pourra être reconnu par l'automate : le langage reconnu est vide.

3. on renvoie faux sinon, car il existe un état accessible et accepteur : un mot peut être reconnu par $\mathcal{A}$ et donc le langage reconnu n'est pas vide.

On considère deux langages réguliers $\mathcal{L}$ et $\mathcal{L}'$ et leurs automates finis déterministes associés $\mathcal{A}$ et $\mathcal{A}'$.

(d) Proposer une relation équivalente à $\mathcal{L} \subseteq \mathcal{L}'$, en utilisant l'intersection et la complétentation.

Solution : Une relation équivalente est $\mathcal{L} \cap \bar{\mathcal{L}}' = \emptyset$.

(e) En déduire un algorithme INCLUS_DANS($\mathcal{A}$,$\mathcal{A}'$)permettant de déterminer si $\mathcal{L} \subseteq \mathcal{L}'$.

Solution : On procède comme suit :

1. $C \leftarrow$ COMPLEMENTAIRE($\mathcal{A}'$)
2. $P \leftarrow \mathcal{A} \times C$ l'automate produit de $\mathcal{A}$ et C (reconnaît l'intersection des langages)
3. Renvoyer EST_VIDE(P)

(f) En déduire un algorithme EGAL($\mathcal{A}$,$\mathcal{A}'$) permettant de déterminer si $\mathcal{L} = \mathcal{L}'$.

Solution : On procède comme suit, par double inclusion :

1. renvoyer INCLUS_DANS($\mathcal{A}$,$\mathcal{A}'$) ET INCLUS_DANS($\mathcal{A}'$,$\mathcal{A}$)

A6. Soit l'expression régulière $e = aba^*(ba|ca)$. En utilisant l'algorithme de Berry-Sethi, construire l'automate de Glushkov associé à e.

Solution : Expliciter la méthode du cours(linéarisation, automate local, relabellisation des états)

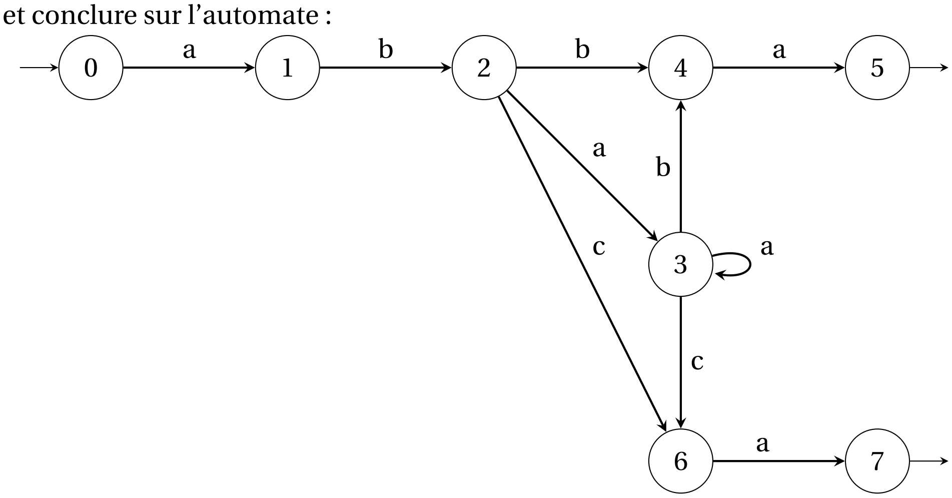
$$E = a_1 b_2 a_3^* (b_4 a_5 | c_6 a_7)$$

$$P = \{a_1\}$$

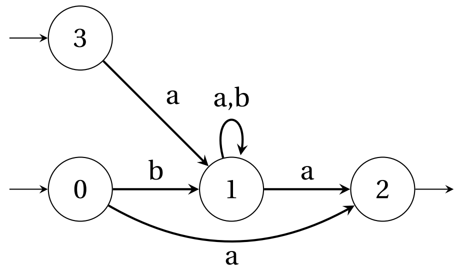
$$S = \{a_7, a_5\}$$

$$F = \{a_1 b_2, b_2 a_3, b_2 b_4, b_2 c_6, a_3 a_3, a_3 b_4, a_3 c_6, b_4 a_5, c_6 a_7\}$$

et conclure sur l'automate :



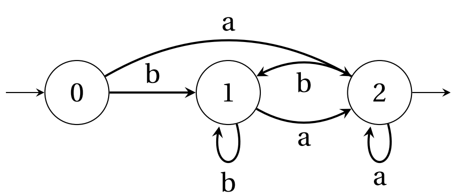
A7. Déterminiser l'automate suivant :



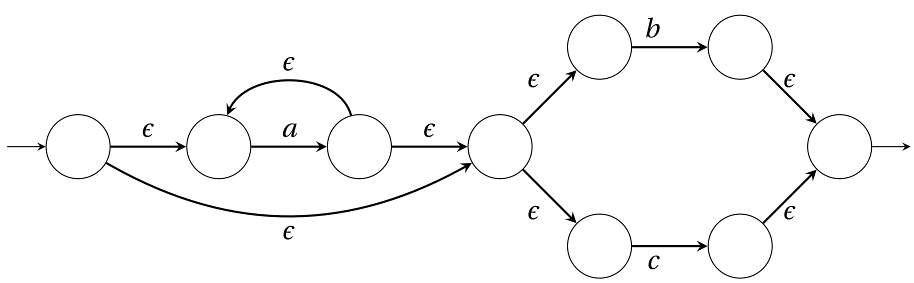
Solution : L'automate n'est pas déterministe (cf. états 2 et 4). On construit donc l'automate des parties.

La table des transitions l'automate des parties est :

	↓{0,3}	{1}	{1,2}↑
a	{1,2}	{1,2}	{1,2}
b	{1}	{1}	{1}

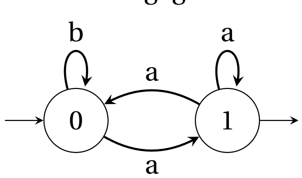


A8. En utilisant la méthode compositionnelle (automates de Thompson), construire un automate non déterministe avec transitions spontanées qui reconnaît le langage dénoté par $a^*(b|c)$.

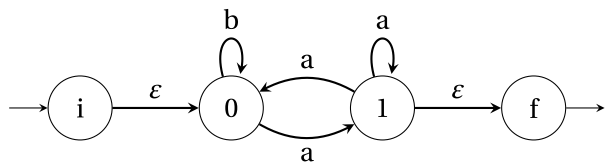


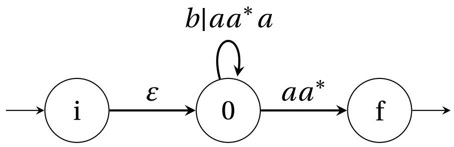
Solution :

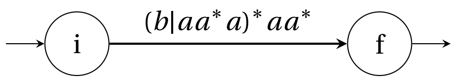
A9. En utilisant la méthode de l'élimination des états, calculer une expression régulière dont le langage dénote le langage reconnu par l'automate suivant.



Solution :







L'expression régulière cherchée est donc $(b|aa^*a)^*aa^*$.

B Langages

B1. Soit L un langage. Montrer que les propositions suivantes sont équivalentes :

(i) $\varepsilon \in L$

(ii) $\forall i \geq 0, \varepsilon \in L^i$

(iii) $\forall i \geq 0, L^i \subseteq L^{i+1}$

Solution :

Démonstration. On procède par stratégie circulaire.

(i) ⇒ (ii) Supposons que $\varepsilon \in L$. Par récurrence sur i qu'on a alors pour tout $i \geq 0, \varepsilon \in L^i$.

Initialisation pour $i = 0$. $L^0 = \{\varepsilon\}$ donc $\varepsilon \in L^0$

Hérédité Soit un entier i et supposons que $\varepsilon \in L^i$. Montrons $\varepsilon \in L^{i+1}$. Comme $\varepsilon \in L$ et $\varepsilon \in L^i$, on a $\varepsilon.\varepsilon = \varepsilon \in L.L^i$. Par définition de la puissance d'un langage, cela signifie que $\varepsilon \in L^{i+1}$.

Conclusion Si (i) est vraie, alors (ii) est vraie pour tout entier i.

(ii)⇒ (iii) Supposons que $\forall i \geq 0, \varepsilon \in L^i$. Alors en particulier, $\varepsilon \in L$. Soit w un mot de L^n. Alors $\varepsilon w = w$ est un mot de L^{n+1} et donc on a $L^n \subseteq L^{n+1}$.

(iii) ⇒ (i) Supposons que $\forall i \geq 0, L^i \subseteq L^{i+1}$. Alors en particulier, $L^0 \subseteq L^1$, i.e., $\varepsilon \in L$.

■

B2. On considère un alphabet $\Sigma = \{a, b\}$. Pour les ensembles suivant, trouver l'expression régulière qui le dénote.

(a) L'ensemble des mots de longueur paire.

Solution : $(\Sigma\Sigma)^*$

(b) L'ensemble des mots contenant un nombre impair de a. Par exemple : ab ou baabbbab.

> **Solution :** $b^* a(b^* ab^* a)^* b^*$

(c) L'ensemble des mots de composés des motifs aa ou bb ou du mot vide. Par exemple : bbaaaabb.

> **Solution :** $(aa|bb)^*$

B3. Simplifier, en justifiant, les expressions régulières suivantes :

(a) $\varepsilon|ab|abab|(ab)^*$

> **Solution :** $(ab)^*$ inclusions de langages

(b) $a(ab^*|aa)|aa(b^*|a)$

> **Solution :** factorisation $aa(b^*|a)$ et idempotence de l'union.

B4. Le langage $\mathcal{L} = \{a^p b^q, (p,q) \in \mathbb{N}^2\}$ est-il régulier ?

> **Solution :** Oui, car c'est langage dénoté par l'expression rationnelle $a^* b^*$, l'ensemble des mots comportant un nombre quelconque de a suivi d'un nombre quelconque de b.

B5. Montrer que le langage L défini par $L = \{baba^2ba^3 \dots ba^n, n \in \mathbb{N}^\star\}$ n'est pas régulier.

> **Solution :**
>
> *Démonstration.* Par l'absurde.
>
> Supposons que $\mathcal{L}$ est régulier. D'après le théorème de Kleene, il existe une automate fini **à N états** qui reconnait le langage $\mathcal{L}$.
>
> Soit $w = baba^2 \dots ba^N$. w est un mot de le mot de $\mathcal{L}$ qui possède plus de N lettres.
>
> D'après le lemme de l'étoile, il existe donc une décomposition de w en xyz tels que $|xy| \leqslant N$, $y \neq \varepsilon$ et $xy^*z \subseteq \mathcal{L}$.
>
> Le préfixe de taille N de w est nécessairement dans le début de la séquence (les groupes ba^k sont courts au début). Si on itère y, par exemple avec le mot $u = xy^2z$, on va soit :
>
> — changer le nombre de b dans le mot. Or, la structure de L impose que le nombre de b dans w soit N.
>
> — changer la longueur d'un bloc de a sans changer les suivants. Or, la structure de L impose que le k-ième bloc de a ait exactement longueur k.
>
> Le mot itéré (ou pompé) xy^2z brisera donc cette progression arithmétique stricte et donc $u = xy^2z \notin L$, contradiction. ∎

■ **Définition 2 — Mot miroir.** Soit Σ un alphabet et w un mot sur Σ. Soient $a_1, \dots, a_n$ des lettres de Σ. Le mot miroir d'un mot $w = a_1 a_2 \dots a_n$ est $w^R = a_n a_{n-1} \dots a_1$.

B6. Soit $\Sigma = \{a, b\}$, un alphabet. En utilisant un langage proche du langage des puissances, montrer que $\mathcal{L} = \{w \in \Sigma^*, w = w^R\}$ n'est pas régulier.

Solution :
On a $\mathcal{L} \cap \mathcal{L}_{ER}(a^* ba^*) = \{a^n ba^n, n \in \mathbb{N}\}$. Si $\mathcal{L}$ était régulier, comme les langages réguliers sont stables par intersection et que $\mathcal{L}_{ER}(a^* ba^*)$ est un langage régulier, cela signifierait que $\{a^n ba^n, n \in \mathbb{N}\}$ est régulier.
Or, on peut montrer que $\{a^n ba^n, n \in \mathbb{N}\}$ n'est pas régulier, en procédant comme pour le langage des puissances, grâce au lemme de l'étoile. Donc, $\mathcal{L}$ n'est pas régulier.

B7. Soient x, y, z des mots sur un alphabet Σ. Montrer que :

$$x^2 = y^2 z^2 \Leftrightarrow \exists w \in \Sigma^*, x = yz \wedge (x, y, z) \in \{w\}^*$$

Solution :

Démonstration. **(⇐)** Soient des entiers i, j, k et w le mot tel que : $x = w^i$, $y = w^j$ et $z = w^k$ et $x = yz$. Nécessairement, $i = j + k$. Alors $x^2 = w^{2i} = w^{2(j+k)} = w^{2j} w^{2k} = w^{j^2} w^{k^2} = y^2 z^2$.

(⇒) Soient x, y, z des mots sur un alphabet Σ tels que $x^2 = y^2 z^2$.

1. Longueurs de mots (notée $|mot|$) : de l'équation hypothèse, on peut déduire que $|x| + |x| = 2|x| = 2|y| + 2|z|$. Soit $|x| = |y| + |z|$.
2. On peut appliquer le lemme de Lévi à l'expression $xx = y^2 z^2$. Soient t,u,v et s quatre mots de Σ^*. Si $tu = vs$ alors il existe un unique mot $z \in \Sigma^*$ tel que :
   - soit $t = vz$ et $zu = s$,
   - soit $v = tz$ et $zs = u$.

   Appliquons le lemme à $t = x$, $u = x$, $v = y$ et $s = yz^2$. Il existe donc un mot w tel que $x = yw$ et $wx = yz^2$.
3. De $x = yw$ on déduit que $|x| = |y| + |w|$. Soit $|w| = |x| - |y|$.
4. De $wx = yz^2$ on déduit que $yzz = wyw$. Or, $|z| = |x| - |y| = |w|$. Donc $z = w$ et donc $x = yz$.
5. De plus, si on considère de nouveau $yzz = wyw$, on peut simplifier à droite par $w = z$. On obtient que $yz = zy$, ce qui signifie que y commute avec z. Par conséquent, y et z sont des puissances d'un même mot. Comme z est une puissance de w on a donc $y \in \{w\}^*$ (cf. exercice A3 su TP d'introduction sur les langages).

On a donc bien $x = yz$ et les x,y,z sont des puissances de w.

■