Feuille d'exercices n°1 - Expressions régulières et automates

Notions abordées

- expressions régulières
- opérations régulières sur les langages
- préfixes, suffixes et facteurs
- relation d'équivalence sur des mots
- automates, avec ou sans ε-transition
- completion, déterminisation, suppression des ε-transitions
- automates pour des calculs, preuve de correction d'automates

Exercice 1 : Habitants d'expressions régulières

Q. 1 Donner tous les mots de taille 1, 2, 3 et 4 des langages réguliers suivants.

1. $\Big((ab)^\star|a\Big)^\star$
2. $\Big(a\Big((bb)^\star|(a\emptyset)\Big)b\Big)|\varepsilon$

Q. 2 Donner tous les mots de taille 1, 2 et 3 des langages réguliers suivants.

1. $(a|b)^\star\,(a|c)^\star$
2. $(ab)^\star|(ac)^\star$

Exercice 2 : Propriétés sur les opérations régulières

Soient A, B et C trois langages sur le même alphabet Σ.

Q. 1 Que valent les langages suivants ? On attend une écriture la plus explicite possible.

1. $\emptyset^\star$
2. $\emptyset\cdot A$
3. $\{\varepsilon\}\cdot A$

Q. 2 Démontrer les propriétés suivantes.

1. $(A\cdot B)\cdot C = A\cdot(B\cdot C)$
2. $A\subseteq B \Rightarrow A^\star \subseteq B^\star$
3. $(A^\star)^\star = A^\star$
4. $A^\star\cdot A^\star = A^\star$
5. $(A\cup B)^\star = (A^\star\cdot B^\star)^\star$
6. $A(B\cup C) = (AB)\cup(AC)$

Q. 3 Pour les couples de langages (L_1, L_2) définis ci-dessous donner la relation la plus précise possible entre L_1 et L_2 ($\subseteq, \supseteq, =$ ou aucune).

1. $L_1 = (A\cdot B)^\star$ et $L_2 = (A^\star B^\star)$
2. $L_1 = (A\cdot B)^\star$ et $L_2 = (A^\star B^\star)^\star$
3. $L_1 = (A\cap B)^\star$ et $L_2 = (A^\star \cap B^\star)$
4. $L_1 = (A\cup B)^\star$ et $L_2 = (A^\star \cup B^\star)$
5. $L_1 = A\cdot(B\cap C)$ et $L_2 = (A\cdot B)\cap(A\cdot C)$
6. $L_1 = (A^\star \cup B)^\star$ et $L_2 = (A\cup B)^\star$

Exercice 3 : Langages

Soit A un alphabet contenant la lettre b. Soit $X = \{b\}$ et $Y = (A \setminus \{b\}) \cdot \{b\}^*$.

Q. 1 Décrire informellement les éléments de X^*, Y et Y^*.

Q. 2 Montrer que tout mot de A^* commençant par une lettre distincte de b appartient à Y^*.

Q. 3 Montrer que tout mot u de A^* s'écrit de façon unique sous la forme $u = vw$, où $v \in X^*$ et $w \in Y^*$.

Exercice 4 : Propriétés sur les mots

Dans tout cet exercice Σ désigne un alphabet.

Q. 1 Soient u, v et w trois mots de $\Sigma^\star$ tels que u et v soient préfixes de w.
Montrer que u est préfixe de v ou v est préfixe de u.

Q. 2 Soient a et b deux lettres de Σ et $u \in \Sigma^\star$ tel que $au = ub$. Montrer que $a = b$ et $u \in a^\star$.

Q. 3 Soit $(x, y) \in \Sigma^\star \times \Sigma^\star$. Montrer que $xy = yx$ ssi il existe $z \in \Sigma^\star$ tel que $x \in z^\star$ et $y \in z^\star$.

Q. 4 Soit x et y deux mots de $\Sigma^\star$. Montrer qu'il existe deux entiers non tous les deux nuls m et n tels que $x^m = y^n$ si et seulement s'il existe $z \in \Sigma^\star$ tel que $x \in z^\star$ et $y \in z^\star$.
Indication : montrer que xy et yx sont préfixes de $x^n\, y^m$.

Q. 5 Un mot non vide $u \in \Sigma^+$ est dit **primitif** s'il n'est puissance d'aucun mot autre que lui même, *i.e.* $\forall v \in \Sigma^\star, \forall n \in \mathbb{N},\ u = v^n \Rightarrow n = 1$. Établir que tout mot non vide est puissance d'un mot primitif unique.

Exercice 5 : Une équivalence sur les mots

Soit Σ un alphabet. On définit sur $\Sigma^\star$ la relation $\equiv$ suivante :

$$u \equiv v \overset{\text{déf}}{\Leftrightarrow} \exists (x, y) \in \Sigma^\star \times \Sigma^\star,\ u = x \cdot y \text{ et } v = y \cdot x$$

Q. 1 Montrer que $\equiv$ est une relation d'équivalence.

Q. 2 Montrer que $u \equiv v$ si et seulement s'il existe un mot w tel que $u \cdot w = w \cdot v$.

Q. 3 Soit $n \in \mathbb{N}^\star$. Montrer que $u \equiv v$ si et seulement si $u^n \equiv v^n$.

Automates et langages

Exercice 6 : Vocabulaire des automates

Q. 1 Représenter l'automate $\mathcal{A}$ sur l'alphabet $\{a, b, c\}$ d'états $0, 1, 2, 3$, d'état initial 0, d'état terminal 3 et de transitions $(0, a, 0)$, $(0, a, 1)$, $(0, b, 0)$, $(0, c, 0)$, $(1, a, 2)$, $(1, b, 2)$, $(1, c, 2)$, $(2, a, 3)$, $(2, b, 3)$, $(2, c, 3)$.

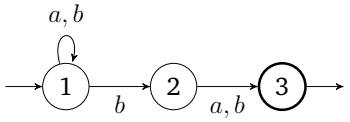
Q. 2 Cet automate est-il complet ? déterministe ? Justifier.

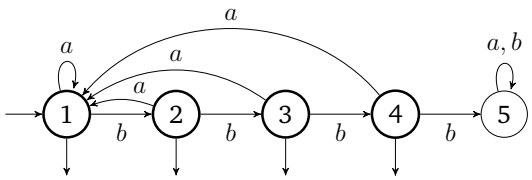
Q. 3 Les mots $baba$ et $cabcb$ sont-ils reconnus par $\mathcal{A}$?

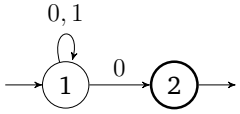
Q. 4 Décrire $\mathcal{L}(\mathcal{A})$ en langage ordinaire.

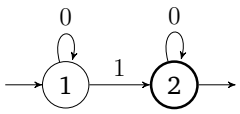
Exercice 7 : Langages reconnus

Q. 1 Décrire en français le langage reconnu par chacun des automates suivants. L'alphabet considéré est $\Sigma = \{a, b\}$ pour les deux premiers automates, $\Sigma = \{0, 1\}$ pour les deux derniers.

1. 

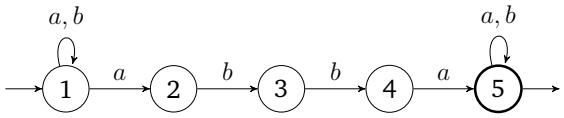
2. 

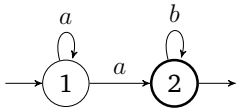
3. 

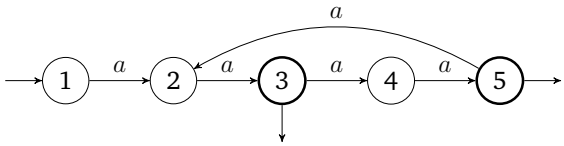
4. 

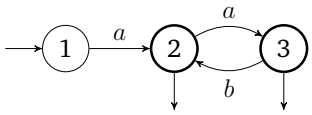
Exercice 8 : Expressions régulières et automates

Q. 1 Pour chacun des automates suivants, décrire le langage reconnu par une expression régulière.

1. 

2. 

3. 

4. 

Exercice 9 : Construction d'automates

Soit $A = \{a, b, c\}$. Donner des automates finis reconnaissant les langages suivants, on s'efforcera de donner des automates déterministes complets.

Q. 1 L'ensemble des mots de longueur paire.

Q. 2 L'ensemble des mots où le nombre d'occurrences de "b" est divisible par 3.

Q. 3 L'ensemble des mots se terminant par "b".

Q. 4 L'ensemble des mots non vides ne se terminant pas par "b".

Q. 5 L'ensemble des mots contenant au moins un "b".

Q. 6 L'ensemble des mots contenant au plus un "b".

Q. 7 L'ensemble des mots contenant exactement un "b".

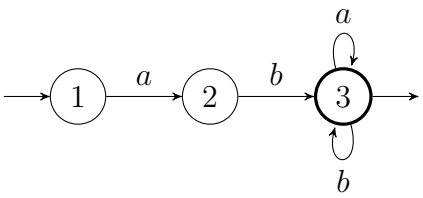
Q. 8 L'ensemble des mots ne contenant aucun "b".

Q. 9 L'ensemble des mots contenant au moins un "a" et dont la première occurrence de "a" n'est pas suivie par un "c".

Transformations en automates équivalents

Exercice 10 : Complétion d'automate

Q. 1 Expliquer pourquoi l'automate ci-contre sur $\{a, b\}$ n'est pas complet.



Q. 2 Quel langage reconnaît-il ?

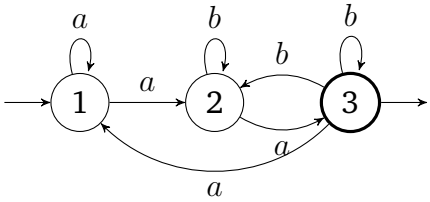
Q. 3 Donner un automate complet équivalent.

Exercice 11 : Déterminisations

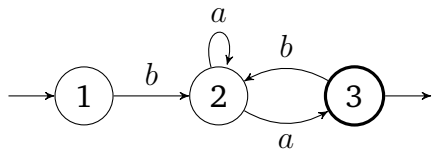
Q. 1 Déterminiser les automates représentés dans l'exercice 7.

Q. 2 Déterminiser les automates suivants :

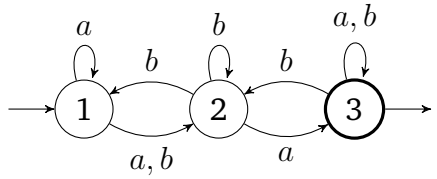
1.



2.



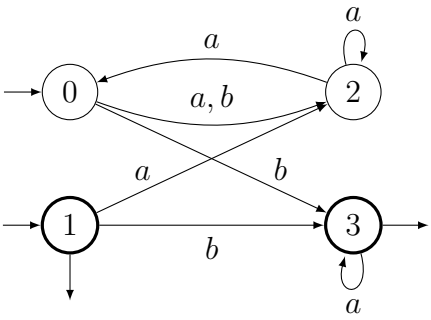
3.



Q. 3 Donner un automate non déterministe reconnaissant L le langage des mots sur l'alphabet $A = \{a, b\}$ ayant trois occurrences successives de la lettre a.

Q. 4 Déterminiser l'automate obtenu à la question précédente.

Q. 5 **En appliquant exactement l'algorithme de déterminisation du cours,** donner un automate déterministe et complet équivalent à l'automate ci-dessous. On attend à la fois l'automate résultat et la table qui a permis de le construire.



Exercice 12 : Fonctions de transition et de transition étendue

Soit $\mathcal{A}$ un automate déterministe. Grâce à son caractère déterministe, on peut définir comme suit sa **fonction de transition**, notée ici δ^1, et par suite sa **fonction de transition étendue**, notée ici δ^*,

$$\delta^1 = \left(\begin{array}{ll} Q\times\Sigma \longrightarrow Q \\ (q,l) \quad \mapsto \text{ l'unique } q'\in Q \text{ tel que } (q,l,q')\in\delta \end{array} \right) \delta^* = \left(\begin{array}{lll} Q\times\Sigma^\star & \longrightarrow & Q \\ (q,\varepsilon) & \mapsto & q \\ (q,l\cdot u) & \mapsto & \delta^*\Big(\delta^1(q,l),u\Big) \end{array} \right)$$

Q. 1 Montrer que $\forall w\in\Sigma^\star, \forall q\in Q,\ \delta^\star(q,w) = \delta\Big(\delta^\star(q,w_1\dots w_{|w|-1}), w_{|w|}\Big)$.

Q. 2 Montrer que $\forall (u,v)\in\Sigma^\star\times\Sigma^\star, \forall q\in Q,\ \delta^\star(q,u\cdot v) = \delta^\star\Big(\delta^\star(q,u),v\Big)$.

Exercice 13 : Déterminisation de taille exponentielle

On s'intéresse dans cet exercice à la taille des automates (en nombre d'états) obtenus par l'algorithme de déterminisation.

Q. 1 Étant donné un automate $\mathcal{A} = (\Sigma, Q, \delta, I, F)$ ayant n états, donner un majorant (en fonction de n) sur le nombre d'états de l'automate $\mathsf{det}(\mathcal{A})$ obtenu par l'algorithme de déterminisation vu en classe.

Q. 2 Donner un exemple de deux automates équivalents $\mathcal{A}$ et $\mathcal{B}$ tels que $\mathcal{B}$ est déterministe et que le nombre d'états de $\mathsf{det}(\mathcal{A})$ est strictement plus grand que le nombre d'états de $\mathcal{B}$.

La question précédente peut laisser penser que l'explosion du nombre d'états lors de la déterminisation est due à l'algorithme utilisé. Dans le reste de cet exercice on démontre que ce n'est pas le cas. Plus précisément on démontre que pour tout $n\in\mathbb{N}$, il existe un automate $\mathcal{A}_n$ ayant $n+1$ états qu'on ne peut déterminiser avec moins de 2^n états, c'est-à-dire tel que tout automate déterministe équivalent a au moins 2^n états.

Q. 3 Étant donné un entier $n \geqslant 1$, on note L_n le langage des mots sur $\Sigma = \{a,b\}$ de longueur supérieure à n, et dont la n-ième lettre en partant de la fin est un a. Construire un automate non déterministe $\mathcal{A}_n$ sur Σ ayant au plus $n+1$ états et reconnaissant L_n.

Q. 4 Représenter $\mathcal{A}_3$, le déterminiser.

Dans la suite, on suppose donné un automate complet déterministe $\mathcal{D}_n$ reconnaissant L_n. On note alors i son unique état initial et $\delta^\star$ sa fonction de transition étendue.

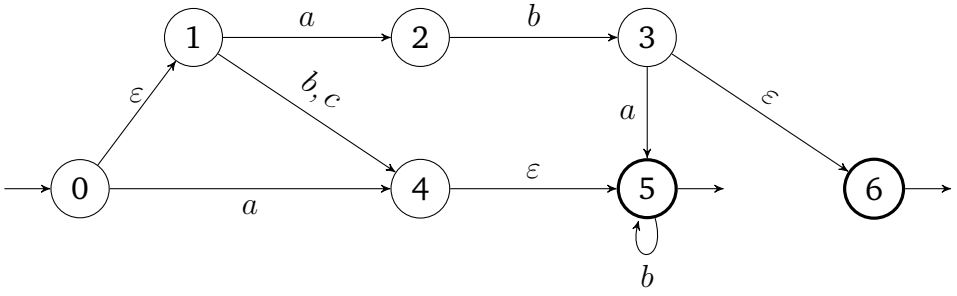
Q. 5 Soient deux mots distincts u et v de Σ^n, montrer que $\delta^*(i,u) \neq \delta^*(i,v)$.

Q. 6 En déduire un minorant sur le nombre d'états de $\mathcal{D}_n$ en fonction du nombre de mots de Σ^n de longueur n.

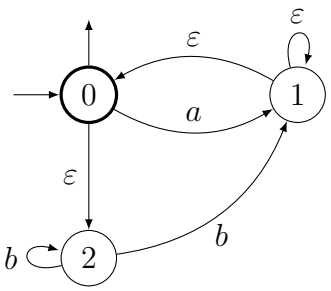
Q. 7 Conclure.

Exercice 14 : Suppression des ε-transitions

Q. 1 Donner l'automate obtenu, **en appliquant exactement l'algorithme de suppression des ε-transitions du cours,** sur l'automate suivant. On s'autorisera, après chaque suppression, à émonder l'automate obtenu.



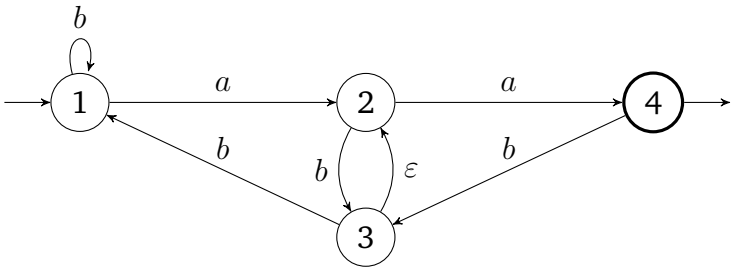
Q. 2 Donner l'automate obtenu, **en appliquant exactement l'algorithme de suppression des ε-transitions du cours,** sur l'automate suivant, en traitant les états dans l'ordre $0, 1, 2$. On attend une représentation de l'automate obtenu après le traitement de chaque état.



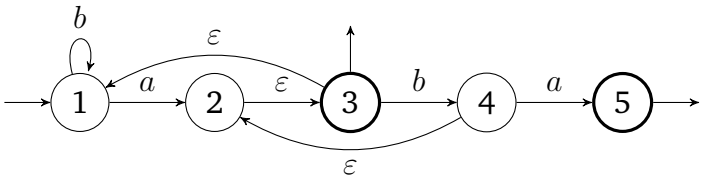
Exercice 15 : Déterminisation d'automates avec ε-transitions

Q. 1 Donner des automates déterministes reconnaissant les mêmes langages que les automates non déterministes avec ε-transitions représentés ci-dessous.

1.



2.



Automates et calculs

Exercice 16 : Automates pour le calcul de modulo

Dans cet exercice on s'intéresse à la définition d'un automate permettant de tester si deux nombres, représentés par leur écriture en base 2, sont congrus modulo 3.

Q. 1 Donner un automate sur l'alphabet $\Sigma = \{0\}$ acceptant le langage des $\{0^p \mid 2^p \equiv 1[3]\}$, à savoir l'ensemble des nombres $n \in \mathbb{N}$, écrits en unaire, tels que $2^n \equiv 1[3]$.

Q. 2 Grâce à la question précédente, donner un automate sur l'alphabet $\Sigma = \{0, 1\}$ acceptant le langage des $\{b_0 b_1 \dots b_{n-1} \mid \overline{b_{n-1} b_{n-2} \dots b_0}^2 \equiv 1[3]\}$. ⚠ **Attention :** les bits de poids faibles sont donnés en premier.

Q. 3 Définir un automate sur l'alphabet $\Sigma = \{0, 1\}^2$ reconnaissant le langage suivant.

$$\{(b_0, c_0)(b_1, c_1) \dots (b_{n-1}, c_{n-1}) \mid \overline{b_{n-1} b_{n-2} \dots b_0}^2 \equiv \overline{c_{n-1} c_{n-2} \dots c_0}^2 [3]\}$$

Q. 4 Grâce à la question précédente, donner un automate sur l'alphabet $\Sigma = \{0, 1\}$ acceptant le langage des $\{b_0 b_1 \dots b_{n-1} \mid \overline{b_0 b_1 \dots b_{n-1}}^2 \equiv 1[3]\}$. ⚠ **Attention :** les bits de poids forts sont donnés en premier.

★ **Q. 5** On peut en fait généraliser ce qui est fait dans cet exercice de la manière suivante : étant donné une base m et un modulo k, construire un automate $\mathcal{A}_{m,k}$ lisant un mot $b_0 b_1 \dots b_{n-1}$ et calculant $\sum_{i=0}^{n-1} b_i m^i [k]$. Définir une fonction OCaml : `eval_poly (m: int) (k: int) : automata` prenant en argument l'entier m et l'entier k et retournant $\mathcal{A}_{m,k}$.

Exercice 17 : Automates pour le calcul de l'addition en binaire

Dans cet exercice on s'intéresse au problème de la vérification, par un automate, de la somme de deux nombres écrits en binaire : l'automate reçoit en entrée trois mots u, v et w représentant trois entiers p, q et r et doit décider si $p + q = r$.
Dans un premier temps on suppose que les trois nombres font la même taille, dans un deuxième temps on enlève cette hypothèse.

1. Nombres de même tailles

On se donne pour alphabet $\Sigma = \{0, 1\}$. Étant donné un mot $w = w_0 w_1 \dots w_n$ on note $\overline{w}^2 = \sum_{i=0}^{n} w_i 2^i$, la valeur de w en base 2. On souhaite construire dans cette section un automate sur l'alphabet $\Sigma \times \Sigma \times \Sigma$ reconnaissant le langage suivant.

$$L = \{(u_0, v_0, w_0)(u_1, v_1, w_1) \dots (u_{n-1}, v_{n-1}, w_{n-1}) \mid \overline{u_0 u_1 \dots u_{n-1}}^2 + \overline{v_0 v_1 \dots v_{n-1}}^2 = \overline{w_0 w_1 \dots w_{n-1}}^2\}$$

Q. 1 Proposer un automate déterministe à 2 états reconnaissant L.

Q. 2 Énoncer une propriété invariante sur l'automate proposé. C'est une propriété de la forme : "la lecture d'un mot $(u_0, v_0, w_0)(u_1, v_1, w_1) \dots (u_{n-1}, v_{n-1}, w_{n-1})$ conduit à un état q si et seulement si …".

Q. 3 Prouver l'invariant proposé. En déduire la correction de l'automate proposé.

2. Nombres de tailles distinctes

On se donne un alphabet $\Sigma = \{0, 1, \#\}$. On appelle **nombre bien formé sur** Σ une suite finie $(b_i) \in \Sigma^p$ telle que $\forall i \in [\![0, p[\![, b_i = \# \Rightarrow \Big(\forall j \geqslant i, b_j = \#) \wedge (i \geqslant 1 \Rightarrow b_{i-1} \in \{\#, 1\})\Big)$, autrement dit

- si la suite contient un symbole $\#$, la suite est alors stationnaire sur cette valeur,
- s'il existe, le carcatère précédant le premier $\#$, est un 1.

Un nombre bien formé w, est donc de la forme $b \cdot \#^q$ où $b \in \varepsilon|(\{0, 1\}^\star \cdot 1)$ et $q \in \mathbb{N}$, on note alors : $\overline{w}^2 = \sum_{i=0}^{|b|-1} b_i 2^i$.

On souhaite construire dans cette section un automate sur l'alphabet $\Sigma \times \Sigma \times \Sigma$ reconnaissant le langage suivant.

$$\{(u_0, v_0, w_0)(u_1, v_1, w_1) \dots (u_{n-1}, v_{n-1}, w_{n-1}) \mid u, v, w \text{ sont bien formés et } \overline{u_0 u_1 \dots u_{n-1}}^2 + \overline{v_0 v_1 \dots v_{n-1}}^2 = \overline{w_0 w_1 \dots w_{n-1}}^2\}$$

Q. 4 Construire un tel automate. Justifier sa correction en exhibant un invariant (on ne demande pas de le prouver).