TD10: Propriétés de fermeture

Feuille 4 – exercices 8, 9, 11, 12, 13

18 décembre 2020

Rappels de cours (1)

Pour montrer qu'un langage L n'est pas régulier, vous avez vu deux méthodes :

1. Propriétés de clôture
2. Lemme de l'étoile

Rappels de cours (1)

Pour montrer qu'un langage L n'est pas régulier, vous avez vu deux méthodes :

1. Propriétés de clôture
2. Lemme de l'étoile

1. On raisonne par l'absurde et on montre que si L est régulier, alors un autre langage M devrait être régulier et on sait que ce n'est pas le cas.

$$L \text{ régulier} \quad \overbrace{\Rightarrow \ldots \Rightarrow \cdots \Rightarrow}^{\text{clôture}} \quad M \text{ régulier}$$

Les langages réguliers sont clos par :

- union, concaténation, concaténation itérée (cf. cours 3)
- intersection, complémentaire
- substitution régulière (donc homomorphisme)

Rappels de cours (1) : exemple

On admet que $M = \{0^p 1^p \mid p \geq 0\}$ n'est pas régulier.
On montre que $L = \{wcw' \mid w, w' \in \{a, b\}^*, \ |w|_a = |w'|_b\}$ n'est pas régulier.

Rappels de cours (1) : exemple

On admet que $M = \{0^p 1^p \mid p \geq 0\}$ n'est pas régulier.
On montre que $L = \{wcw' \mid w, w' \in \{a,b\}^*,\ |w|_a = |w'|_b\}$ n'est pas régulier.

1. Supposons par l'absurde que L est régulier.
2. Alors $L' \stackrel{\text{def}}{=} L \cap a^*cb^* = \{a^pcb^p \mid p \geq 0\}$ est nécessairement régulier.
3. Soit l'homomorphisme h défini par :

$$h : \begin{cases} a & \mapsto & 0 \\ b & \mapsto & 1 \\ c & \mapsto & \varepsilon \end{cases}$$

   Le langage $h(L')$ est nécessairement régulier.
4. $$L \xrightarrow{\_\cap\, a^*cb^*} L' \xrightarrow{h} h(L') = M$$

   Contradiction

Feuille 4 – exercice 8

On admet que $M = \{a^n b^n \mid n \geq 0\}$ n'est pas régulier. Montrer que les langages suivants ne sont pas réguliers non plus **sans se servir du lemme de l'étoile** :

1. $L_1 = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$
2. $L_2 = \{a^i b^j c^k \mid i + j = k \geq 0\}$

Feuille 4 – exercice 8

On admet que $M = \{a^n b^n \mid n \geq 0\}$ n'est pas régulier. Montrer que les langages suivants ne sont pas réguliers non plus **sans se servir du lemme de l'étoile** :

1. $L_1 = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$
2. $L_2 = \{a^i b^j c^k \mid i + j = k \geq 0\}$

Correction :

1. On a $M = L_1 \cap a^* b^*$, donc L_1 n'est pas régulier.

Feuille 4 – exercice 8

On admet que $M = \{a^n b^n \mid n \geq 0\}$ n'est pas régulier. Montrer que les langages suivants ne sont pas réguliers non plus **sans se servir du lemme de l'étoile** :

1. $L_1 = \{w \in \{a, b\}^* \mid |w|_a = |w|_b\}$
2. $L_2 = \{a^i b^j c^k \mid i + j = k \geq 0\}$

Correction :

1. On a $M = L_1 \cap a^* b^*$, donc L_1 n'est pas régulier.
2. Soit h la fonction définie par :

$$h : \begin{cases} a & \mapsto & a \\ b & \mapsto & a \\ c & \mapsto & b \end{cases}$$

C'est un homomorphisme, et on a $h(L_2) = M$ donc L_2 n'est pas régulier.

Rappels de cours (2)

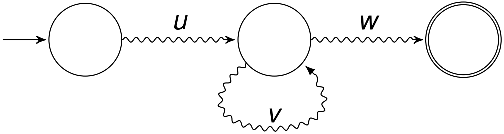
Théorème (Lemme de l'étoile)

Si L est un langage régulier, alors il existe un entier n tel que si z est dans L et de longueur au moins n, alors z est de la forme uvw, où $|uv| \leq n$, $|v| \geq 1$ et pour tout $i \geq 0$, $uv^iw \in L$.

Rappels de cours (2)

Théorème (Lemme de l'étoile)

Si L est un langage régulier, alors il existe un entier n tel que si z est dans L et de longueur au moins n, alors z est de la forme uvw, où $|uv| \leq n$, $|v| \geq 1$ et pour tout $i \geq 0$, $uv^iw \in L$.



Rappels de cours (2)

Théorème (Lemme de l'étoile)

Si L est un langage régulier, alors il existe un entier n tel que si z est dans L et de longueur au moins n, alors z est de la forme uvw, où $|uv| \leq n$, $|v| \geq 1$ et pour tout $i \geq 0$, $uv^iw \in L$.

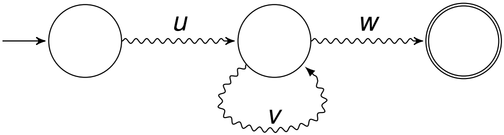
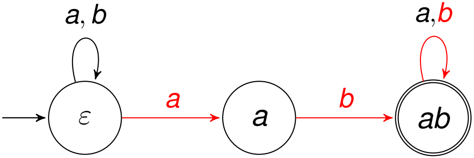


Illustration pour le langage V^*abV^* et $z = abb$:



$n = 3$ (nombre d'états)

$u = ab$, $v = b$, $w = \varepsilon$

Rappels de cours (2)

Théorème (Lemme de l'étoile)

Si L est un langage régulier, alors il existe un entier n tel que si z est dans L et de longueur au moins n, alors z est de la forme uvw, où $|uv| \leq n$, $|v| \geq 1$ et pour tout $i \geq 0$, $uv^i w \in L$.

Comment s'en servir ?

Rappels de cours (2)

Théorème (Lemme de l'étoile)

Si L est un langage régulier, alors il existe un entier n tel que si z est dans L et de longueur au moins n, alors z est de la forme uvw, où $|uv| \leq n$, $|v| \geq 1$ et pour tout $i \geq 0$, $uv^iw \in L$.

Comment s'en servir ?

On procède par l'absurde : on suppose que L est régulier et satisfait donc le lemme de l'étoile.

- On considère l'entier n du lemme. (on ne le choisit pas)
- On choisit un mot $z \in L$ de longueur au moins n.
- Le mot z est décomposé en uvw, où $|uv| \leq n$ et $|v| \geq 1$.
  (on ne contrôle pas la façon dont z est décomposé)
- On choisit une valeur de i telle que $uv^iw \notin L$.

On obtient une contradiction : L ne peut pas être régulier.

Feuille 4 – exercice 9

Montrer que les langages suivants ne sont pas réguliers en se servant du lemme de l'étoile :

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a,b\}^n\}$
2. $L_2 = \{w \in \{a,b\}^* \mid w \text{ est un palindrome}\}$

Feuille 4 – exercice 9

Montrer que les langages suivants ne sont pas réguliers en se servant du lemme de l'étoile :

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a, b\}^n\}$
2. $L_2 = \{w \in \{a, b\}^* \mid w \text{ est un palindrome}\}$

Correction :

1. Soit n l'entier du lemme de l'étoile. Prenons le mot $z = a^n b^n$.

Feuille 4 – exercice 9

Montrer que les langages suivants ne sont pas réguliers en se servant du lemme de l'étoile :

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a,b\}^n\}$
2. $L_2 = \{w \in \{a,b\}^* \mid w \text{ est un palindrome}\}$

Correction :

1. Soit n l'entier du lemme de l'étoile. Prenons le mot $z = a^n b^n$. Comme $|z| \geq n$, z est de la forme uvw, et comme $1 \leq |uv| \leq n$, on a $uv \in a^+$. Donc $v \in a^+$, et on devrait avoir $uv^2w = a^{n+|v|}b^n \in L_1$, ce qui est impossible.

Feuille 4 – exercice 9

Montrer que les langages suivants ne sont pas réguliers en se servant du lemme de l'étoile :

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a, b\}^n\}$
2. $L_2 = \{w \in \{a, b\}^* \mid w \text{ est un palindrome}\}$

Correction :

1. Soit n l'entier du lemme de l'étoile. Prenons le mot $z = a^n b^n$. Comme $|z| \geq n$, z est de la forme uvw, et comme $1 \leq |uv| \leq n$, on a $uv \in a^+$. Donc $v \in a^+$, et on devrait avoir $uv^2w = a^{n+|v|}b^n \in L_1$, ce qui est impossible.
2. Soit $z = a^n ba^n$, avec n l'entier du lemme de l'étoile.

Feuille 4 – exercice 9

Montrer que les langages suivants ne sont pas réguliers en se servant du lemme de l'étoile :

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a, b\}^n\}$
2. $L_2 = \{w \in \{a, b\}^* \mid w \text{ est un palindrome}\}$

Correction :

1. Soit n l'entier du lemme de l'étoile. Prenons le mot $z = a^n b^n$. Comme $|z| \geq n$, z est de la forme uvw, et comme $1 \leq |uv| \leq n$, on a $uv \in a^+$. Donc $v \in a^+$, et on devrait avoir $uv^2w = a^{n+|v|}b^n \in L_1$, ce qui est impossible.
2. Soit $z = a^n ba^n$, avec n l'entier du lemme de l'étoile. Ce mot est un palindrome de longueur au moins n, il est donc de la forme uvw, et nécessairement, $v \in a^+$. Mais on a $uv^0w = a^{n-|v|}ba^n$ qui n'est pas un palindrome, une contradiction.

Feuille 4 – exercice 9

Question bonus : Montrer que ces langages sont hors-contexte.

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a,b\}^n\}$
2. $L_2 = \{w \in \{a,b\}^* \mid w \text{ est un palindrome}\}$

Feuille 4 – exercice 9

Question bonus : Montrer que ces langages sont hors-contexte.

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a,b\}^n\}$
2. $L_2 = \{w \in \{a,b\}^* \mid w \text{ est un palindrome}\}$

Correction :
Voici des grammaires hors-contexte qui les engendrent.

1. $$S \rightarrow aSb \mid bSb \mid \varepsilon$$

Feuille 4 – exercice 9

Question bonus : Montrer que ces langages sont hors-contexte.

1. $L_1 = \{wb^n \mid n \in \mathbb{N}, w \in \{a, b\}^n\}$
2. $L_2 = \{w \in \{a, b\}^* \mid w \text{ est un palindrome}\}$

Correction :
Voici des grammaires hors-contexte qui les engendrent.

1. $$S \rightarrow aSb \mid bSb \mid \varepsilon$$
2. $$S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon$$

Feuille 4 – exercice 11

Etant donné un vocabulaire V, on considère une famille $(L_i)_{i \in \mathbb{N}}$ de langages sur V^* telle que pour tout $i \in \mathbb{N}$, L_i est un langage régulier.

1. Soit $n \geq 0$. Montrer que $M_n = \bigcup_{0 \leq i \leq n} L_i$ est régulier.
2. Peut-on en déduire que $\bigcup_{0 \leq i} L_i$ est régulier ? Justifier.

Feuille 4 – exercice 11

Etant donné un vocabulaire V, on considère une famille $(L_i)_{i \in \mathbb{N}}$ de langages sur V^* telle que pour tout $i \in \mathbb{N}$, L_i est un langage régulier.

1. Soit $n \geq 0$. Montrer que $M_n = \bigcup_{0 \leq i \leq n} L_i$ est régulier.
2. Peut-on en déduire que $\bigcup_{0 \leq i} L_i$ est régulier ? Justifier.

Correction :

1. Le résultat se prouve par récurrence sur n.

Feuille 4 – exercice 11

Etant donné un vocabulaire V, on considère une famille $(L_i)_{i\in\mathbb{N}}$ de langages sur V^* telle que pour tout $i \in \mathbb{N}$, L_i est un langage régulier.

1. Soit $n \geq 0$. Montrer que $M_n = \bigcup_{0\leq i\leq n} L_i$ est régulier.
2. Peut-on en déduire que $\bigcup_{0\leq i} L_i$ est régulier ? Justifier.

Correction :

1. Le résultat se prouve par récurrence sur n.
   - Pour $n = 0$ on a $M_0 = L_0$ qui est régulier d'après l'énoncé.

Feuille 4 – exercice 11

Etant donné un vocabulaire V, on considère une famille $(L_i)_{i \in \mathbb{N}}$ de langages sur V^* telle que pour tout $i \in \mathbb{N}$, L_i est un langage régulier.

1. Soit $n \geq 0$. Montrer que $M_n = \bigcup_{0 \leq i \leq n} L_i$ est régulier.
2. Peut-on en déduire que $\bigcup_{0 \leq i} L_i$ est régulier ? Justifier.

Correction :

1. Le résultat se prouve par récurrence sur n.
   - Pour $n = 0$ on a $M_0 = L_0$ qui est régulier d'après l'énoncé.
   - Pour $n > 0$ on a $M_n = \bigcup_{0 \leq i \leq n} L_i = \left(\bigcup_{0 \leq i \leq n-1} L_i \right) \cup L_n$.

     Par HR, $\bigcup_{0 \leq i \leq n-1} L_i$ est régulier, et L_n est régulier d'après l'énoncé, or l'union de deux langages réguliers est un langage régulier (vu en cours) donc M_n est un langage régulier.

Feuille 4 – exercice 11

Etant donné un vocabulaire V, on considère une famille $(L_i)_{i\in\mathbb{N}}$ de langages sur V^* telle que pour tout $i \in \mathbb{N}$, L_i est un langage régulier.

1. Soit $n \geq 0$. Montrer que $M_n = \bigcup_{0\leq i\leq n} L_i$ est régulier.
2. Peut-on en déduire que $\bigcup_{0\leq i} L_i$ est régulier ? Justifier.

Correction :

1. Le résultat se prouve par récurrence sur n.
   - Pour $n = 0$ on a $M_0 = L_0$ qui est régulier d'après l'énoncé.
   - Pour $n > 0$ on a $M_n = \bigcup_{0\leq i\leq n} L_i = \left(\bigcup_{0\leq i\leq n-1} L_i\right) \cup L_n$.

     Par HR, $\bigcup_{0\leq i\leq n-1} L_i$ est régulier, et L_n est régulier d'après l'énoncé, or l'union de deux langages réguliers est un langage régulier (vu en cours) donc M_n est un langage régulier.
2. La réponse est non : prenons la famille $(\{a^i b^i\})_{i\in\mathbb{N}}$. Chacun de ces langages est un singleton donc est régulier, mais l'union de tous ces langages est $\{a^n b^n \mid n \geq 0\}$, non régulier.

Feuille 4 – exercice 12

Le produit de deux automates est défini de la façon suivante : soient $A_1 = (Q_1, V, \delta_1, \{i_1\}, F_1)$ et $A_2 = (Q_2, V, \delta_2, \{i_2\}, F_2)$ deux automates qu'on suppose complets et sans ε-transition. On considère l'automate

$$A_1 \times A_2 \;=\; (Q_1 \times Q_2,\; V,\; \delta,\; \{\langle i_1, i_2\rangle\},\; F_1 \times F_2),$$

où δ est défini par : pour tous $\langle q_1, q_2\rangle, \langle q_1', q_2'\rangle \in Q_1 \times Q_2$ et pour tout $a \in V$,

$$(\langle q_1, q_2\rangle, a, \langle q_1', q_2'\rangle) \in \delta \quad \text{ssi} \quad (q_1, a, q_1') \in \delta_1 \wedge (q_2, a, q_2') \in \delta_2.$$

On souhaite prouver que $\mathcal{L}(A_1 \times A_2) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

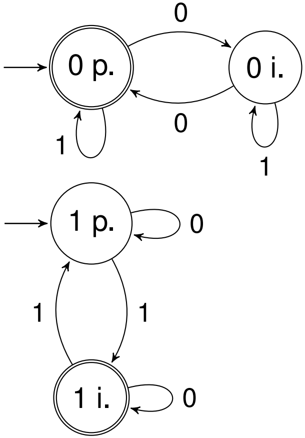
Exercice 12 – question 1

Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0, 1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

Exercice 12 – question 1

Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0, 1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

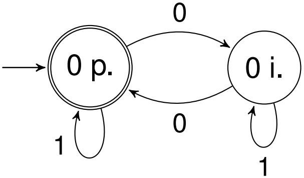
Correction :

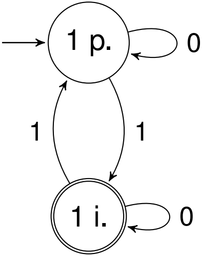


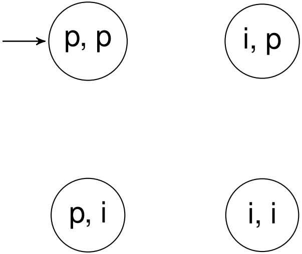
Exercice 12 – question 1

Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0, 1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

Correction :



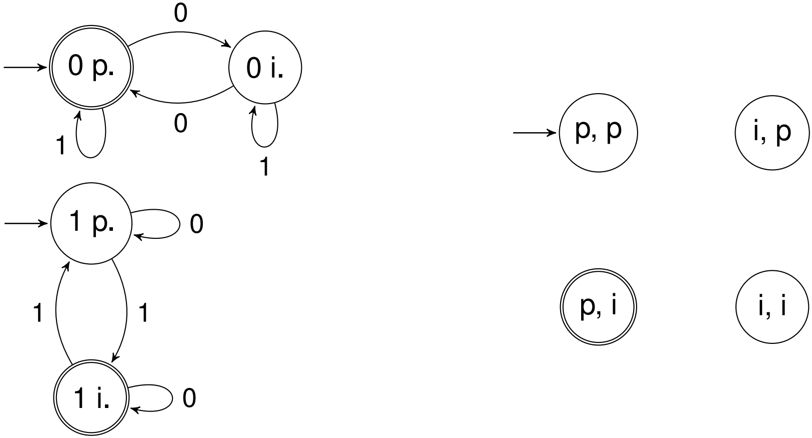




Exercice 12 – question 1

Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0, 1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

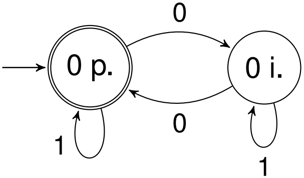
Correction :

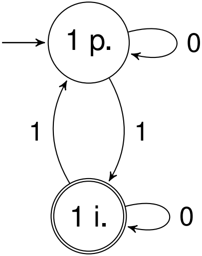


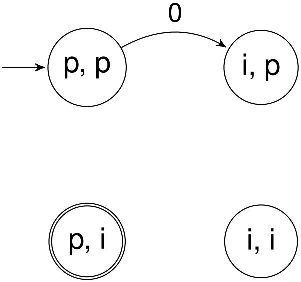
Exercice 12 – question 1

Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0, 1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

Correction :



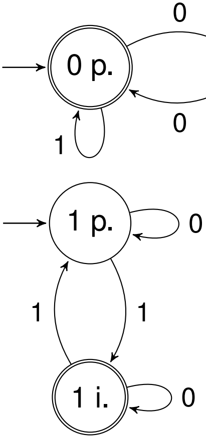


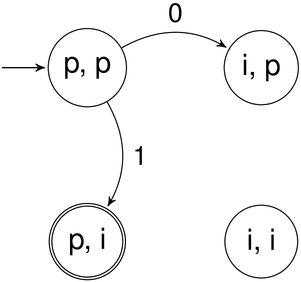


Exercice 12 – question 1

Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0, 1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

Correction :

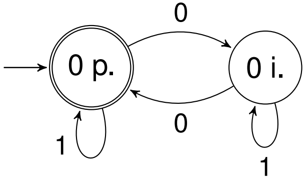


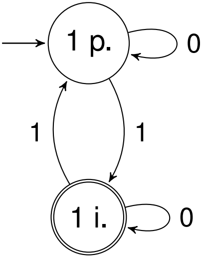


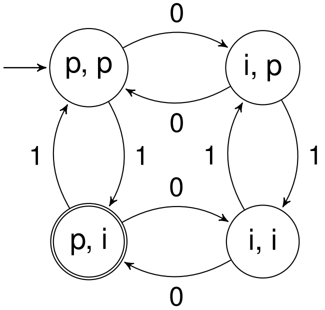
Exercice 12 – question 1

Se servir du produit de deux automates pour déterminer un automate reconnaissant les mots sur $\{0, 1\}^*$ contenant un nombre pair de 0 **et** un nombre impair de 1.

Correction :







Exercice 12 – question 2

On considère maintenant deux automates A_1 et A_2 quelconques, complets et sans ε-transition. Soient, pour $n \geq 1$, les ensembles d'états $\{p_0, \ldots, p_n\} \subseteq Q_1$ et $\{q_0, \ldots, q_n\} \subseteq Q_2$. Soit $w \in V^*$. Démontrer par induction que les assertions suivantes sont équivalentes :

- $(p_0, a_1, p_1) \cdots (p_{n-1}, a_n, p_n)$ est un chemin de trace $w = a_1 \cdots a_n$ dans A_1 et $(q_0, a_1, q_1) \cdots (q_{n-1}, a_n, q_n)$ est un chemin de trace w dans A_2 ;
- $(\langle p_0, q_0 \rangle, a_1, \langle p_1, q_1 \rangle) \cdots (\langle p_{n-1}, q_{n-1} \rangle, a_n, \langle p_n, q_n \rangle)$ est un chemin de trace w dans $A_1 \times A_2$.

Exercice 12 – question 2

On considère maintenant deux automates A_1 et A_2 quelconques, complets et sans ε-transition. Soient, pour $n \geq 1$, les ensembles d'états $\{p_0, \ldots, p_n\} \subseteq Q_1$ et $\{q_0, \ldots, q_n\} \subseteq Q_2$. Soit $w \in V^*$. Démontrer par induction que les assertions suivantes sont équivalentes :

- $(p_0, a_1, p_1) \cdots (p_{n-1}, a_n, p_n)$ est un chemin de trace $w = a_1 \cdots a_n$ dans A_1 et $(q_0, a_1, q_1) \cdots (q_{n-1}, a_n, q_n)$ est un chemin de trace w dans A_2 ;
- $(\langle p_0, q_0 \rangle, a_1, \langle p_1, q_1 \rangle) \cdots (\langle p_{n-1}, q_{n-1} \rangle, a_n, \langle p_n, q_n \rangle)$ est un chemin de trace w dans $A_1 \times A_2$.

Correction : Preuve par induction sur les chemins.

- Si (p_0, a_1, p_1) et (q_0, a_1, q_1) sont des chemins de trace a_1, ceci signifie que $(p_0, a_1, p_1) \in \delta_1$ et $(q_0, a_1, q_1) \in \delta_2$. Par construction, ceci est équivalent à $(\langle p_0, q_0 \rangle, a_1, \langle p_1, q_1 \rangle) \in \delta$, d'où le résultat.

Exercice 12 – question 2 (suite)

- Supposons le résultat vrai pour les chemins

$$\begin{array}{lll} \chi_1 & = (p_0, a_1, p_1) \cdots (p_{n-1}, a_n, p_n) & \text{dans } A_1 \\ \chi_2 & = (q_0, a_1, q_1) \cdots (q_{n-1}, a_n, q_n) & \text{dans } A_2 \\ \chi & = (\langle p_0, q_0 \rangle, a_1, \langle p_1, q_1 \rangle) \cdots (\langle p_{n-1}, q_{n-1} \rangle, a_n, \langle p_n, q_n \rangle) & \text{dans } A_1 \times A_2 \end{array}$$

  tous ces chemins étant de trace $w = a_1 \ldots a_n$.
  Considérons maintenant les chemins $(p, a, p_0)\chi_1$ et $(q, a, q_0)\chi_2$ de trace aw dans A_1 et A_2 respectivement.
  Alors par définition, on a $(p, a, p_0) \in \delta_1$ et $(q, a, q_0) \in \delta_2$.
  Par construction, on a également $(\langle p, q \rangle, a, \langle p_0, q_0 \rangle) \in \delta$, ce qui prouve que $(\langle p, q \rangle, a, \langle p_0, q_0 \rangle)\chi$ est un chemin de trace aw dans $A_1 \times A_2$.

La réciproque (passage de χ à χ_1 et χ_2) se montre de la même manière, par induction sur χ.

Exercice 12 – question 3

En déduire que $\mathcal{L}(A_1 \times A_2) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

Exercice 12 – question 3

En déduire que $\mathcal{L}(A_1 \times A_2) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

Correction :

$w \in \mathcal{L}(A_1 \times A_2)$

Exercice 12 – question 3

En déduire que $\mathcal{L}(A_1 \times A_2) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

Correction :

$$\begin{aligned} & w \in \mathcal{L}(A_1 \times A_2) \\ \iff\ & \exists \text{ un chemin } \chi \text{ dans } A_1 \times A_2 \text{ de trace } w, \\ & \text{d'origine } \langle i_1, i_2 \rangle \text{ et d'extrémité } \langle q_1, q_2 \rangle \in F_1 \times F_2 \end{aligned}$$

Exercice 12 – question 3

En déduire que $\mathcal{L}(A_1 \times A_2) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

Correction :

$$\begin{aligned}
& w \in \mathcal{L}(A_1 \times A_2) \\
\iff\ & \exists \text{ un chemin } \chi \text{ dans } A_1 \times A_2 \text{ de trace } w, \\
& \text{d'origine } \langle i_1, i_2 \rangle \text{ et d'extrémité } \langle q_1, q_2 \rangle \in F_1 \times F_2 \\
\iff\ & \exists \text{ des chemins } \chi_1 \text{ et } \chi_2 \text{ dans } A_1 \text{ et } A_2 \text{ respectivement,} \\
& \text{de trace } w\text{, d'origines } i_1 \text{ et } i_2\text{, et d'extrémités } q_1 \text{ et } q_2
\end{aligned}$$

Exercice 12 – question 3

En déduire que $\mathcal{L}(A_1 \times A_2) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

Correction :

$w \in \mathcal{L}(A_1 \times A_2)$

$\iff$ $\exists$ un chemin χ dans $A_1 \times A_2$ de trace w,
d'origine $\langle i_1, i_2 \rangle$ et d'extrémité $\langle q_1, q_2 \rangle \in F_1 \times F_2$

$\iff$ $\exists$ des chemins χ_1 et χ_2 dans A_1 et A_2 respectivement,
de trace w, d'origines i_1 et i_2, et d'extrémités q_1 et q_2

$\iff$ $w \in \mathcal{L}(A_1)$ et $w \in \mathcal{L}(A_2)$

Exercice 12 – question 3

En déduire que $\mathcal{L}(A_1 \times A_2) = \mathcal{L}(A_1) \cap \mathcal{L}(A_2)$.

Correction :

$$
\begin{aligned}
& w \in \mathcal{L}(A_1 \times A_2) \\
\iff & \exists \text{ un chemin } \chi \text{ dans } A_1 \times A_2 \text{ de trace } w, \\
& \text{d'origine } \langle i_1, i_2 \rangle \text{ et d'extrémité } \langle q_1, q_2 \rangle \in F_1 \times F_2 \\
\iff & \exists \text{ des chemins } \chi_1 \text{ et } \chi_2 \text{ dans } A_1 \text{ et } A_2 \text{ respectivement,} \\
& \text{de trace } w\text{, d'origines } i_1 \text{ et } i_2\text{, et d'extrémités } q_1 \text{ et } q_2 \\
\iff & w \in \mathcal{L}(A_1) \text{ et } w \in \mathcal{L}(A_2) \\
\iff & w \in \mathcal{L}(A_1) \cap \mathcal{L}(A_2)
\end{aligned}
$$

Exercice 12 – question 4

Quel langage aurait été reconnu si l'ensemble des états acceptants de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, et non pas $F_1 \times F_2$? Justifier.

Exercice 12 – question 4

Quel langage aurait été reconnu si l'ensemble des états acceptants de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, et non pas $F_1 \times F_2$? Justifier.

Correction :

Si l'ensemble des états finaux de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, l'automate aurait reconnu $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

Exercice 12 – question 4

Quel langage aurait été reconnu si l'ensemble des états acceptants de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, et non pas $F_1 \times F_2$? Justifier.

Correction :

Si l'ensemble des états finaux de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, l'automate aurait reconnu $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

En effet, on aurait alors

$$\begin{aligned} & w \in \mathcal{L}(A_1 \times A_2) \\ \iff \ & \exists \text{ un chemin } \chi \text{ dans } A_1 \times A_2 \text{ de trace } w, \\ & \text{d'origine } \langle i_1, i_2 \rangle \text{ et d'extrémité } \langle q_1, q_2 \rangle \in F_1 \times Q_2 \cup Q_1 \times F_2 \end{aligned}$$

Exercice 12 – question 4

Quel langage aurait été reconnu si l'ensemble des états acceptants de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, et non pas $F_1 \times F_2$? Justifier.

Correction :

Si l'ensemble des états finaux de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, l'automate aurait reconnu $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

En effet, on aurait alors

$$
\begin{aligned}
& w \in \mathcal{L}(A_1 \times A_2) \\
\iff\ & \exists \text{ un chemin } \chi \text{ dans } A_1 \times A_2 \text{ de trace } w, \\
& \text{d'origine } \langle i_1, i_2 \rangle \text{ et d'extrémité } \langle q_1, q_2 \rangle \in F_1 \times Q_2 \cup Q_1 \times F_2 \\
\iff\ & \exists \text{ des chemins } \chi_1 \text{ et } \chi_2 \text{ dans } A_1 \text{ et } A_2 \text{ respectivement,} \\
& \text{de trace } w \text{, d'origines } i_1 \text{ et } i_2 \text{, et d'extrémités } q_1 \text{ et } q_2 \\
& \text{avec } q_1 \in F_1 \text{ (et } q_2 \in Q_2\text{) ou } q_2 \in F_2 \text{ (et } q_1 \in Q_1\text{)}
\end{aligned}
$$

Exercice 12 – question 4

Quel langage aurait été reconnu si l'ensemble des états acceptants de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, et non pas $F_1 \times F_2$? Justifier.

Correction :

Si l'ensemble des états finaux de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, l'automate aurait reconnu $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

En effet, on aurait alors

$$
\begin{aligned}
& w \in \mathcal{L}(A_1 \times A_2) \\
\iff\ & \exists \text{ un chemin } \chi \text{ dans } A_1 \times A_2 \text{ de trace } w, \\
& \text{d'origine } \langle i_1, i_2 \rangle \text{ et d'extrémité } \langle q_1, q_2 \rangle \in F_1 \times Q_2 \cup Q_1 \times F_2 \\
\iff\ & \exists \text{ des chemins } \chi_1 \text{ et } \chi_2 \text{ dans } A_1 \text{ et } A_2 \text{ respectivement,} \\
& \text{de trace } w \text{, d'origines } i_1 \text{ et } i_2 \text{, et d'extrémités } q_1 \text{ et } q_2 \\
& \text{avec } q_1 \in F_1 \text{ (et } q_2 \in Q_2\text{) ou } q_2 \in F_2 \text{ (et } q_1 \in Q_1\text{)} \\
\iff\ & w \in \mathcal{L}(A_1) \text{ ou } w \in \mathcal{L}(A_2)
\end{aligned}
$$

Exercice 12 – question 4

Quel langage aurait été reconnu si l'ensemble des états acceptants de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, et non pas $F_1 \times F_2$? Justifier.

Correction :

Si l'ensemble des états finaux de $A_1 \times A_2$ avait été $F_1 \times Q_2 \cup Q_1 \times F_2$, l'automate aurait reconnu $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.

En effet, on aurait alors

$$
\begin{aligned}
& w \in \mathcal{L}(A_1 \times A_2) \\
\iff\ & \exists \text{ un chemin } \chi \text{ dans } A_1 \times A_2 \text{ de trace } w, \\
& \text{d'origine } \langle i_1, i_2 \rangle \text{ et d'extrémité } \langle q_1, q_2 \rangle \in F_1 \times Q_2 \cup Q_1 \times F_2 \\
\iff\ & \exists \text{ des chemins } \chi_1 \text{ et } \chi_2 \text{ dans } A_1 \text{ et } A_2 \text{ respectivement,} \\
& \text{de trace } w \text{, d'origines } i_1 \text{ et } i_2 \text{, et d'extrémités } q_1 \text{ et } q_2 \\
& \text{avec } q_1 \in F_1 \text{ (et } q_2 \in Q_2 \text{) ou } q_2 \in F_2 \text{ (et } q_1 \in Q_1 \text{)} \\
\iff\ & w \in \mathcal{L}(A_1) \text{ ou } w \in \mathcal{L}(A_2) \\
\iff\ & w \in \mathcal{L}(A_1) \cup \mathcal{L}(A_2)
\end{aligned}
$$

Feuille 4 – exercice 13

Lors d'un TD de TL1, un élève propose un automate qui ne ressemble aucunement à celui proposé par le chargé de TD. La classe est partagée : personne ne peut trouver un contre-exemple mais personne n'est convaincu qu'il est correct.

1. Donner une méthode pour déterminer si deux automates sont équivalents.
2. Si deux automates ne sont pas équivalents, comment faire pour exhiber un contre-exemple, c.-à-d. un mot accepté par l'un mais pas par l'autre ?
3. En se basant sur la construction de l'automate produit (exercice 12), comment faire cette construction plus directement ?

Correction exercice 13

1. Pour tester l'équivalence entre automates, on peut les déterminiser (au besoin) puis les minimiser. Par unicité de l'automate minimal (au renommage des états près), on peut alors facilement déterminer s'ils sont équivalents.

Correction exercice 13

1. Pour tester l'équivalence entre automates, on peut les déterminiser (au besoin) puis les minimiser. Par unicité de l'automate minimal (au renommage des états près), on peut alors facilement déterminer s'ils sont équivalents.
2. Notons A_1 et A_2 les deux automates. Pour trouver un mot accepté par l'un et non par l'autre, on peut s'intéresser aux langages $\mathcal{L}(A_1) \setminus \mathcal{L}(A_2) = \mathcal{L}(A_1) \cap \overline{\mathcal{L}(A_2)}$ et $\mathcal{L}(A_2) \setminus \mathcal{L}(A_1)$.

Correction exercice 13

1. Pour tester l'équivalence entre automates, on peut les déterminiser (au besoin) puis les minimiser. Par unicité de l'automate minimal (au renommage des états près), on peut alors facilement déterminer s'ils sont équivalents.
2. Notons A_1 et A_2 les deux automates. Pour trouver un mot accepté par l'un et non par l'autre, on peut s'intéresser aux langages $\mathcal{L}(A_1) \setminus \mathcal{L}(A_2) = \mathcal{L}(A_1) \cap \overline{\mathcal{L}(A_2)}$ et $\mathcal{L}(A_2) \setminus \mathcal{L}(A_1)$.
   Comme les langages réguliers sont clos par complémentaire, intersection et union, on peut déterminer un automate qui reconnaît $(\mathcal{L}(A_1) \setminus \mathcal{L}(A_2)) \cup (\mathcal{L}(A_2) \setminus \mathcal{L}(A_1))$.

Correction exercice 13

1. Pour tester l'équivalence entre automates, on peut les déterminiser (au besoin) puis les minimiser. Par unicité de l'automate minimal (au renommage des états près), on peut alors facilement déterminer s'ils sont équivalents.
2. Notons A_1 et A_2 les deux automates. Pour trouver un mot accepté par l'un et non par l'autre, on peut s'intéresser aux langages $\mathcal{L}(A_1) \setminus \mathcal{L}(A_2) = \mathcal{L}(A_1) \cap \overline{\mathcal{L}(A_2)}$ et $\mathcal{L}(A_2) \setminus \mathcal{L}(A_1)$.
   Comme les langages réguliers sont clos par complémentaire, intersection et union, on peut déterminer un automate qui reconnaît $(\mathcal{L}(A_1) \setminus \mathcal{L}(A_2)) \cup (\mathcal{L}(A_2) \setminus \mathcal{L}(A_1))$.
   Pour trouver un mot reconnu par cet automate, il suffit alors de faire un parcours de graphe.

Correction exercice 13

1. Pour tester l'équivalence entre automates, on peut les déterminiser (au besoin) puis les minimiser. Par unicité de l'automate minimal (au renommage des états près), on peut alors facilement déterminer s'ils sont équivalents.
2. Notons A_1 et A_2 les deux automates. Pour trouver un mot accepté par l'un et non par l'autre, on peut s'intéresser aux langages $\mathcal{L}(A_1) \setminus \mathcal{L}(A_2) = \mathcal{L}(A_1) \cap \overline{\mathcal{L}(A_2)}$ et $\mathcal{L}(A_2) \setminus \mathcal{L}(A_1)$.
   Comme les langages réguliers sont clos par complémentaire, intersection et union, on peut déterminer un automate qui reconnaît $(\mathcal{L}(A_1) \setminus \mathcal{L}(A_2)) \cup (\mathcal{L}(A_2) \setminus \mathcal{L}(A_1))$.
   Pour trouver un mot reconnu par cet automate, il suffit alors de faire un parcours de graphe.
3. Il suffit de choisir $F_1 \times (Q_2 \setminus F_2) \cup F_2 \times (Q_1 \setminus F_1)$ comme ensemble pour reconnaître la différence symétrique.

Correction exercice 13

1. Pour tester l'équivalence entre automates, on peut les déterminiser (au besoin) puis les minimiser. Par unicité de l'automate minimal (au renommage des états près), on peut alors facilement déterminer s'ils sont équivalents.
2. Notons A_1 et A_2 les deux automates. Pour trouver un mot accepté par l'un et non par l'autre, on peut s'intéresser aux langages $\mathcal{L}(A_1) \setminus \mathcal{L}(A_2) = \mathcal{L}(A_1) \cap \overline{\mathcal{L}(A_2)}$ et $\mathcal{L}(A_2) \setminus \mathcal{L}(A_1)$.
   Comme les langages réguliers sont clos par complémentaire, intersection et union, on peut déterminer un automate qui reconnaît $(\mathcal{L}(A_1) \setminus \mathcal{L}(A_2)) \cup (\mathcal{L}(A_2) \setminus \mathcal{L}(A_1))$.
   Pour trouver un mot reconnu par cet automate, il suffit alors de faire un parcours de graphe.
3. Il suffit de choisir $F_1 \times (Q_2 \setminus F_2) \cup F_2 \times (Q_1 \setminus F_1)$ comme ensemble pour reconnaître la différence symétrique.

Question : Est-ce que ceci donne une méthode alternative sans déterminisation pour montrer l'équivalence entre automates ?

Correction exercice 13

1. Pour tester l'équivalence entre automates, on peut les déterminiser (au besoin) puis les minimiser. Par unicité de l'automate minimal (au renommage des états près), on peut alors facilement déterminer s'ils sont équivalents.
2. Notons A_1 et A_2 les deux automates. Pour trouver un mot accepté par l'un et non par l'autre, on peut s'intéresser aux langages $\mathcal{L}(A_1) \setminus \mathcal{L}(A_2) = \mathcal{L}(A_1) \cap \overline{\mathcal{L}(A_2)}$ et $\mathcal{L}(A_2) \setminus \mathcal{L}(A_1)$.
   Comme les langages réguliers sont clos par complémentaire, intersection et union, on peut déterminer un automate qui reconnaît $(\mathcal{L}(A_1) \setminus \mathcal{L}(A_2)) \cup (\mathcal{L}(A_2) \setminus \mathcal{L}(A_1))$.
   Pour trouver un mot reconnu par cet automate, il suffit alors de faire un parcours de graphe.
3. Il suffit de choisir $F_1 \times (Q_2 \setminus F_2) \cup F_2 \times (Q_1 \setminus F_1)$ comme ensemble pour reconnaître la différence symétrique.
   Attention : les automates doivent être déterministes !

Question : Est-ce que ceci donne une méthode alternative sans déterminisation pour montrer l'équivalence entre automates ? Non