UE AL - Automates et Langages

Correction d'exercices de TD

Propriétés de clôture des langages reconnaissables

Définition 1 *Si Σ est un alphabet, on note $Rat(\Sigma^*)$ la famille des langages rationnels sur Σ, et $Rec(\Sigma^*)$ la famille des langages reconnaissables sur Σ.*

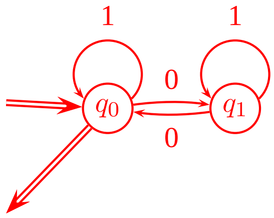
Dans cette série d'exercices, on cherche à définir ("expérimentalement") les propriétés de clôture de $Rec(\Sigma^*)$.

1 Clôture de $Rect(\Sigma^*)$ par les opérations booléennes

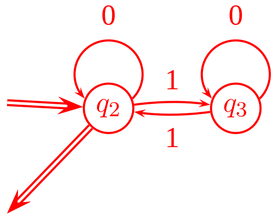
Proposition 1 *$Rec(\Sigma^*)$ est fermée par union.*

Exercice 1.1 :
Question 1.1.1 . Définir les automates A_0, resp. A_1, reconnaissant les mots sur $\Sigma = \{0, 1\}$ avec un nombre pair de 0, resp. de 1.
Correction:

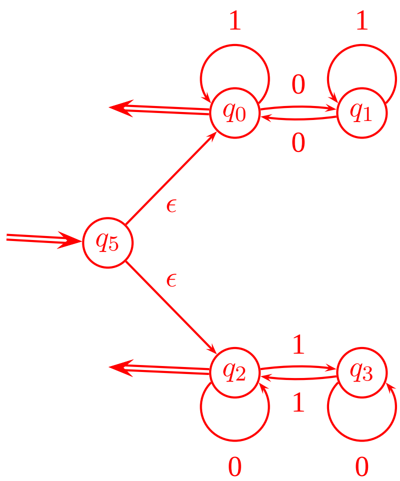
Automate A_0



Automate A_1



Question 1.1.2 . Définir un ϵ-AFN[1] $A_{0\cup 1}$ reconnaissant le langage $\mathcal{L}(A_0) \cup \mathcal{L}(A_1)$.
Correction:



Question 1.1.3 . Étant donnés les automates $M_1 = (\Sigma, Q_1, q_0^1, F_1, \delta_1)$ et $M_2 = (\Sigma, Q_2, q_0^2, F_2, \delta_2)$, définir un ϵ-AFN $M_{0\cup 1}$ tel que $\mathcal{L}(M_{0\cup 1}) = \mathcal{L}(M_1) \cup \mathcal{L}(M_2)$. On pourra supposer que Q_1 et Q_2 sont disjoints.
Correction: Supposons que Q_1 et Q_2 sont disjoints.
L'ϵ-AFN reconnaissant l'union des deux langages $\mathcal{L}(M_1)$ et $\mathcal{L}(M_2)$ est caractérisé par l'ajout d'un nouvel état q_0 qui sera l'état initial de $M_{0\cup 1}$, et l'ajout d' ϵ-transitions entre cet état et les états initiaux de M_1 et M_2.
Plus formellement, on a donc :

$$M_{0\cup 1} = (\Sigma, Q_1 \cup Q_2 \cup \{q_0\}, q_0, F_1 \cup F_2, \delta_1 \cup \delta_2 \cup \{(q_0, \epsilon, q_0^1), (q_0, \epsilon, q_0^2)\})$$

---

[1] automate fini non-déterministe utilisant des epsilon-transitions

Question 1.1.4 . Déduire de $A_{0\cup 1}$ un AFD[2] équivalent.

Correction: Déterminiser un ϵ-AFN repose sur un algorithme similaire à la déterminisation d'un AFN. La différence majeure entre ces derniers est liée à l'utilisation des ensembles nommés $\epsilon\text{-}close(q)$, qui contiennent l'ensemble des états pouvant être atteints à partir d'un état q en suivant 0, 1 ou plusieurs ϵ-transitions.

Dans un premier temps, dressons le tableau des $\epsilon\text{-}close(q)$ de l'automate $A_{0\cup 1}$:

q	$\epsilon\text{-}close(q)$
q_0	$\{q_0\}$
q_1	$\{q_1\}$
q_2	$\{q_2\}$
q_3	$\{q_3\}$
q_5	$\{q_0, q_2, q_5\}$

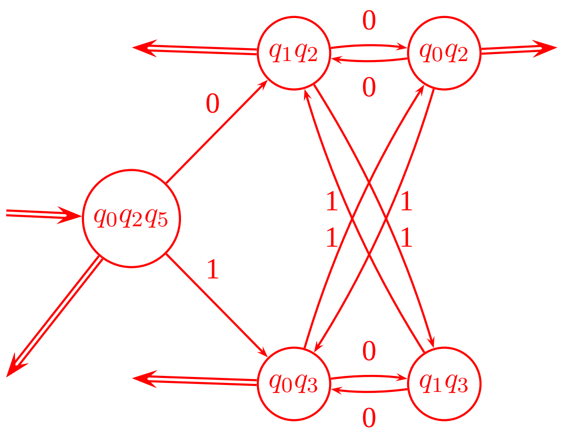
L'algorithme de déterminisation d'un ϵ-AFN se fait alors de la même manière que la déterminisation d'un AFN, en remplaçant chaque apparition d'un état q par $\epsilon\text{-}close(q)$.

On sait que l'état initial de $A_{0\cup 1}$ est q_5. Par conséquent, l'état initial de l'AFD équivalent à $A_{0\cup 1}$ est $\epsilon\text{-}close(q_5) = \{q_0, q_2, q_5\}$.

Le tableau permettant de construire l'AFD équivalent à $A_{0\cup 1}$ est alors :

	0	1
$\{q_0, q_2, q_5\}$	$\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_2) = \{q_1, q_2\}$	$\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_3) = \{q_0, q_3\}$
$\{q_1, q_2\}$	$\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_2) = \{q_0, q_2\}$	$\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_3) = \{q_1, q_3\}$
$\{q_0, q_3\}$	$\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_3) = \{q_1, q_3\}$	$\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_2) = \{q_0, q_2\}$
$\{q_0, q_2\}$	$\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_2) = \{q_1, q_2\}$	$\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_3) = \{q_0, q_3\}$
$\{q_1, q_3\}$	$\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_3) = \{q_0, q_3\}$	$\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_2) = \{q_1, q_2\}$

Un AFD équivalent à $A_{1\cup 2}$ est donc :



Proposition 2 *$Rec(\Sigma^*)$ est fermée par intersection.*

Exercice 1.2 :

Question 1.2.1 . Pour les automates A_0 et A_1 définis précédemment, définissez un AFD $A_{0\cap 1}$ reconnaissant le langage $\mathcal{L}(A_0) \cap \mathcal{L}(A_1)$.

Correction: Intuitivement, le moyen le plus simple de vérifier si un mot est dans le langage $\mathcal{L}(A_0) \cap \mathcal{L}(A_1)$ est de lire le mot en parallèle dans les automates A_0 et A_1. Si la lecture du mot dans les deux automates aboutit à un état final, alors le mot appartient à l'intersection des langages reconnus par ces deux automates.

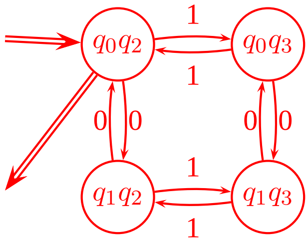
On peut donc caractériser l'automate que l'on va construire par les propriétés suivantes :

- Les états de $A_{0\cap 1}$ sont le produit cartésien des états de A_0 et des états de A_1;
- Les états initiaux de $A_{0\cap 1}$ sont les états de $A_{0\cap 1}$ ne contenant que des états initiaux de A_0 et A_1;
- Les états finaux de $A_{0\cap 1}$ sont les états de $A_{0\cap 1}$ ne contenant que des états finaux de A_0 et A_1.

[2] Automate Fini Déterministe

- Les transitions de $A_{0\cap 1}$ sont obtenus en effectuant une opération proche d'un produit cartésien : si $x \in \{0,1\}$, si q_0 et q_0' sont des états de A_0, si (q_0, x, q_0') est une transition de A_0, si q_1 et q_1' sont des états de A_1, et si (q_1, x, q_1') est une transition de A_1, alors $\big(\{q_0, q_1\}, x, \{q_0', q_1'\}\big)$ est une transition de $A_{0\cap 1}$.

Un automate reconnaissant le langage $\mathcal{L}(A_0) \cap \mathcal{L}(A_1)$ est alors :



Cet automate est déterministe, nous avons donc trouvé $A_{0\cap 1}$.

Question 1.2.2 . Étant donnés les automates $M_1 = (\Sigma, Q_1, q_0^1, F_1, \delta_1)$ et $M_2 = (\Sigma, Q_2, q_0^2, F_2, \delta_2)$, définir un AFD $M_{1\cap 2}$ tel que $\mathcal{L}(M_{1\cap 2}) = \mathcal{L}(M_1) \cap \mathcal{L}(M_2)$. On pourra supposer que Q_1 et Q_2 sont disjoints.

Correction: Supposons que Q_1 et Q_2 sont disjoints. D'après les informations fournies dans la correction de la question précédente, on a :

$$M_{1\cap 2} = \Big(\Sigma, \{(q,q') | q \in Q_1, q' \in Q_2\}, (q_0^1, q_0^2)\Big), \{(f,f') | f \in F_1, f' \in F_2\},$$
$$\{\big((q,q'), \sigma, (r,r')\big) | (q,\sigma,q') \in \delta_1, (r,\sigma,r') \in \delta_2\}\Big)$$

Qui peut s'écrire en abrégé :

$$M_{1\cap 2} = \Big(\Sigma, Q_1 \times Q_2, (q_0^1, q_0^2), F_1 \times F_2, \{\big((q,q'), \sigma, (r,r')\big) | (q,\sigma,q') \in \delta_1, (r,\sigma,r') \in \delta_2\}\Big)$$

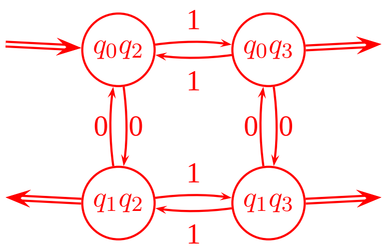
Proposition 3 *$Rec(\Sigma^*)$ est fermée par complémentation.*

Exercice 1.3 :

Question 1.3.1 . On considère l'automate $A_{0\cap 1}$ défini précédemment, définir un AFD $A_{\overline{0\cap 1}}$ qui reconnait le langage $\overline{\mathcal{L}(A_0) \cap \mathcal{L}(A_1)}$.

Correction: Le complémentaire d'un langage est l'ensemble des mots qui ne sont pas contenus dans le langage.

Le langage $\mathcal{L}(A_0) \cap \mathcal{L}(A_1)$ reconnait les mots contenant un nombre pair de 0 et un nombre pair de 1. Par conséquent, le complémentaire de ce langage est reconnait les mots contenant un nombre impair de 0 ou un nombre impair de 1. Ainsi, l'automate suivant reconnait le langage $\overline{\mathcal{L}(A_0) \cap \mathcal{L}(A_1)}$:



Cet automate est déterministe. Il correspond donc à l'automate $A_{\overline{0\cap 1}}$

Question 1.3.2 . Étant donné un automate $M = (\Sigma, Q, q_0, F, \delta)$ définir un automate $\overline{M}$ tel que $\mathcal{L}(\overline{M}) = \overline{\mathcal{L}(M)}$. On supposera que M est déterministe et complet, car pour tout automate, il existe un automate déterministe et complet équivalent.

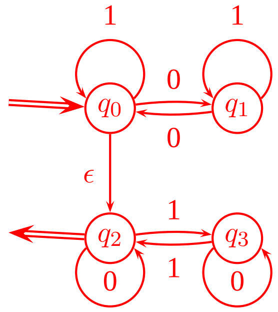
Correction: Supposons que M soit déterministe et complet. Dans ce cas, on sait qu'il existe un unique chemin menant de l'état initial de l'automate M à un état quelconque de M permettant de lire n'importe quel mot de l'alphabet Σ. L'ensemble des mots de Σ reconnus par M sont ceux dont la lecture aboutit à un état final. Par conséquent, les mots reconnus par le complémentaire de ce langage sont ceux dont la lecture dans M aboutit à un état n'étant pas un état final. On peut donc en déduire que :

$$\overline{M} = (\Sigma, Q, q_0, Q \backslash F, \delta)$$

2 Clôture de $Rec(\Sigma^*)$

Proposition 4 $Rec(\Sigma^*)$ *est fermée par concaténation.*

Exercice 2.1 :
Question 2.1.1 . Pour les automates A_0 et A_1 définis précedemment, définir un ϵ-AFN $A_{0.1}$ qui reconnait le langage $\mathcal{L}(A_0).\mathcal{L}(A_1)$.
Correction: Pour construire un tel automate, nous lions les états finaux de A_0 aux états initiaux de A_1 à l'aide d'epsilon-transitions. $A_{0.1}$ devient alors :



Question 2.1.2 . Étant donnés les automates $M_1 = (\Sigma, Q_1, q_0^1, F_1, \delta_1)$ et $M_2 = (\Sigma, Q_2, q_0^2, F_2, \delta_2)$, définir un ϵ-AFN $M_{1.2}$ tel que $\mathcal{L}(M_{1.2}) = \mathcal{L}(M_1).\mathcal{L}(M_2)$. On pourra supposer que Q_1 et Q_2 sont disjoints.
Correction: D'après les informations fournies dans la correction de la question précédente, on a :

$$M_{1.2} = \Big(\Sigma, Q_1 \cup Q_2, q_0^1, F_2, \delta_1 \cup \delta_2 \cup \{(f, \epsilon, q_0^2) | f \in F_1\}\Big)$$

Question 2.1.3 . Déduire de $A_{0.1}$ un AFD équivalent.
Correction:

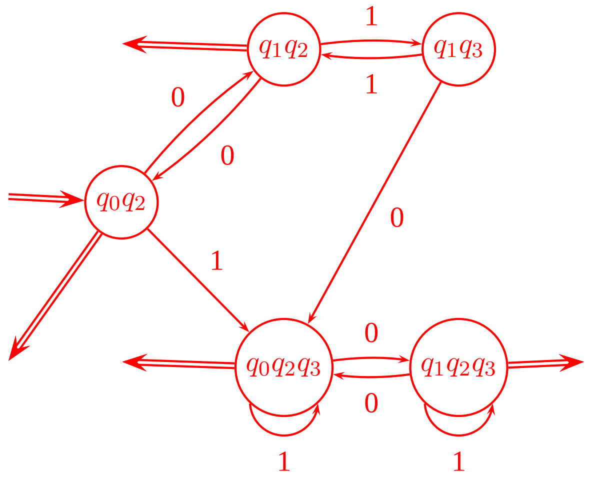
En reprenant les principes énoncés dans le début de la correction, on a :

q	ϵ-close(q)
q_0	$\{q_0, q_2\}$
q_1	$\{q_1\}$
q_2	$\{q_2\}$
q_3	$\{q_3\}$

Par conséquent, le tableau permettant de construire un AFD équivalent à $A_{0.1}$ est :

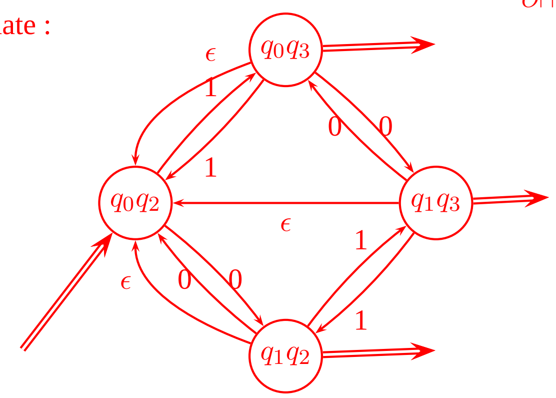
	0	1
$\{q_0, q_2\}$	$\{\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_2)\} = \{q_1, q_2\}$	$\{\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_3)\} = \{q_0, q_2, q_3\}$
$\{q_1, q_2\}$	$\{\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_2)\} = \{q_0, q_2\}$	$\{\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_3)\} = \{q_1, q_3\}$
$\{q_0, q_2, q_3\}$	$\{\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_2) \cup \epsilon\text{-}close(q_3)\} = \{q_1, q_2, q_3\}$	$\{\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_2) \cup \epsilon\text{-}close(q_3)\} = \{q_0, q_2, q_3\}$
$\{q_1, q_3\}$	$\{\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_3)\} = \{q_0, q_2, q_3\}$	$\{\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_2)\} = \{q_1, q_2\}$
$\{q_1, q_2, q_3\}$	$\{\epsilon\text{-}close(q_0) \cup \epsilon\text{-}close(q_2) \cup \epsilon\text{-}close(q_3)\} = \{q_0, q_2, q_3\}$	$\{\epsilon\text{-}close(q_1) \cup \epsilon\text{-}close(q_2) \cup \epsilon\text{-}close(q_3)\} = \{q_1, q_2, q_3\}$

Un AFD équivalent à $A_{0.1}$ est donc :

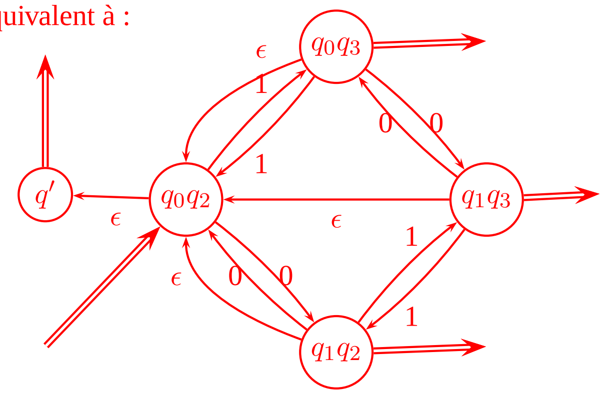


Proposition 5 *$Rec(\Sigma^*)$ est fermée par étoile.*

Exercice 2.2 :
Question 2.2.1 . Pour l'automate $A_{\overline{O\cap 1}}$ défini précédemment, définir un ϵ-AFN A' qui reconnait le langage $\mathcal{L}(A_{\overline{O\cap 1}})*$.
Correction: L'étoile de Kleene consiste à concaténer 0, 1 ou plusieurs fois des mots provenant d'un langage. En nous basant sur ce qui a été fait dans l'exercice précédent, on constate que construire A' revient à ajouter des epsilon-transitions entre les états finaux de $A_{\overline{O\cap 1}}$ et ses états initiaux. Nous aboutissons alors à l'automate :



Un problème subsiste dans cet automate : le mot vide ne peut pas être reconnu, car l'état initial de l'automate n'est pas un état final. Transformer l'état initial en un état final est proscrit, car cela va influer sur le langage reconnu. Par conséquent, pour remédier à ce problème, il faut ajouter un nouvel état final q', et ajouter une epsilon-transition entre l'état initial et q'.
Par conséquent, A' est équivalent à :



Question 2.2.2 . Étant donné un automate $M = (\Sigma, Q, q_0, F, \delta)$ définir un ϵ-AFN M^* tel que $\mathcal{L}(M^*) = \mathcal{L}(M)^*$.
Correction: D'après les éléments énoncés dans la correction de la question précédente, on peut écrire :

$$M^* = (\Sigma, Q \cup \{q'\}, q_0, F \cup \{q'\}, \delta \cup \{(f, \epsilon, q_0 | f \in F)\} \cup \{q_0, \epsilon, q'\})$$