Université de Bordeaux **4 mars 2021**
4TIN604U **Licence 3 Informatique**
Modèles de la Programmation et du Calcul

Corrigé du devoir surveillé

Exercice 1. Déterminisation d'automates (5 points).
Soient $\mathcal{A}_1$ et $\mathcal{A}_2$ deux automates sur l'alphabet $A = \{a, b\}$:

$$\mathcal{A}_1 = \Big(Q = \{0,1,2,3,4\},\ I = \{0\},\ F = \{4\},\ \delta = \{(0,a,1),(1,a,2),(1,b,3),(2,a,2),(2,b,4),(3,a,4)\}\Big),$$

$$\mathcal{A}_2 = \Big(Q = \{0,1,2,3\},\ I = \{0\},\ F = \{3\},$$

$$\delta = \{(0,a,1),(0,b,1),(0,b,2),(1,a,3),(1,b,1),(2,a,2),(2,b,1),(2,b,3),(3,a,3),(3,b,3)\}\Big).$$

Dessinez les automates.
Solution :

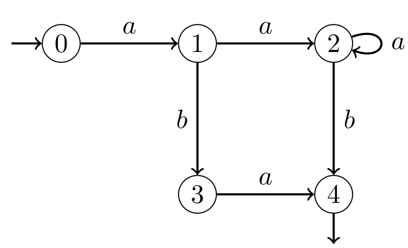

Figure 1 – Automate $\mathcal{A}_1$.

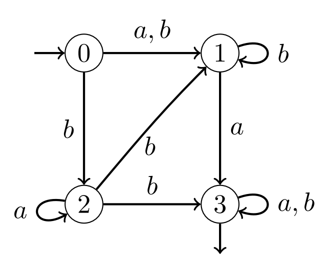

Figure 2 – Automate $\mathcal{A}_2$.

Pour **chaque automate** répondez aux questions suivantes :

1. Est-il complet ? Si non, complétez-le.
   Solution :
   L'automate $\mathcal{A}_1$ n'est pas complet (par exemple aucune transition ayant origine dans l'état

4 n'est définie). Pour le compléter on ajoute un état "puits" appelé 5 et les transitions $\{(0,b,5),(3,b,5),(5,a,5),(5,b,5)\}$.
Ce qui donne :

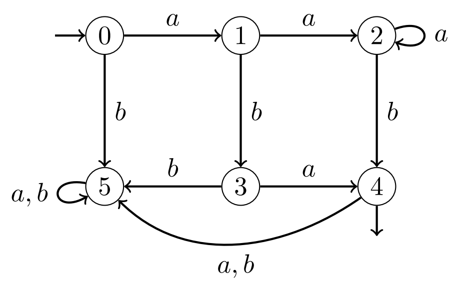

FIGURE 3 – Automate $\mathcal{A}_1$ complété.

L'automate $\mathcal{A}_2$ est complet car pour chaque état $q \in Q$ et chaque lettre $x \in A$ il existe un état $r \in Q$ tel que $(q,x,r) \in \delta$.

2. Est-il déterministe ? Si non, déterminisez-le en utilisant la méthode de sous-ensembles (il suffit de dessiner la partie accessible de l'automate).
   Solution :
   L'automate $\mathcal{A}_1$ est déterministe car il a un unique état initial et pour chaque état $q \in Q$ et chaque lettre $x \in A$ il existe au plus une seule transition issue de q et étiquetée par x dans δ.

   L'automate $\mathcal{A}_2$ n'est pas déterministe car il existe deux transitions issues de l'état 0 étiquetées par la même lettre b.

   On déterminise $\mathcal{A}_2$ en calculant la fonction de transition pour les sous-ensembles de Q accessibles depuis le sous-ensemble $\{0\}$ contenant l'état initial de $\mathcal{A}_2$.
   Table donnant la fonction de transition de l'automate déterministe :

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

   L'ensemble des états de l'automate déterministe obtenu est $\{\{0\},\{1\},\{1,2\},\{3\},\{2,3\},\{1,3\}\}$, $\{0\}$ est état initial, l'ensemble des états terminaux est $\{\{3\},\{1,3\},\{2,3\}\}$. Si on numérote les sous-ensembles de 0 à 5 (dans l'ordre d'apparition dans la table), on a l'automate :

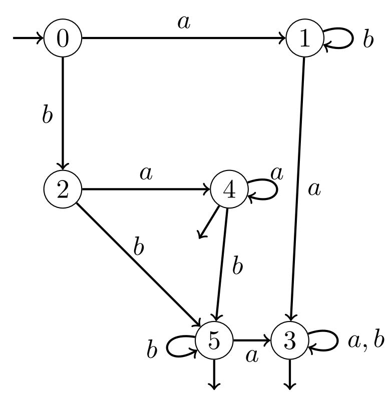

FIGURE 4 – Automate $\mathcal{A}_2$ déterminisé.

Exercice 2. Algorithme de Glushkov (4 points).
À l'aide de l'algorithme de Glushkov, donnez un automate fini qui reconnaît le langage

$$a(ab+bb)^*b.$$

Solution :

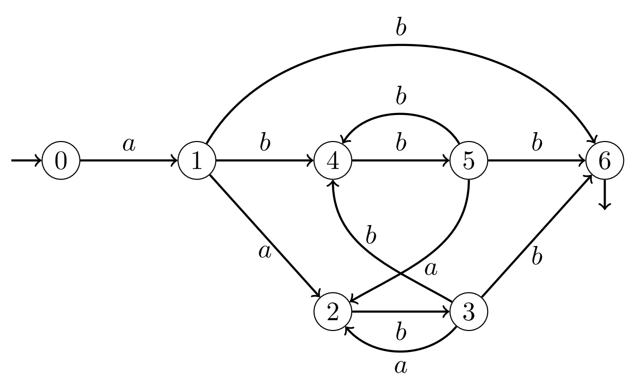

FIGURE 5 – Automate reconnaissant l'expression rationnelle $a(ab+bb)^*b$.

Exercice 3 (6 points).
Un entier est divisible par 2 si et seulement si le bit le plus faible (donc la dernière lettre) de sa représentation binaire est 0. Par exemple, 14 est divisible par 2, et sa représentation binaire 1110 finit par 0. Par contre, 5 n'est pas divisible par 2, et sa représentation binaire 101 finit par 1.
Dans cet exercice, nous allons construire des automates qui traitent des entiers (y compris 0) encodés en binaire. Ces représentations seront toujours lues de gauche à droite (donc le bit le plus faible est à droite) et l'alphabet est $A=\{0,1\}$.

1. Donnez un automate $\mathcal{A}_1$ à deux états qui accepte les représentations binaires des entiers divisibles par 2.
   Solution :

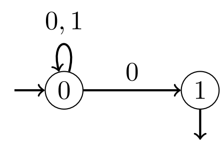

FIGURE 6 – Automate pour $\mathcal{A}_1$.

D'autres options sont possibles, par exemple :

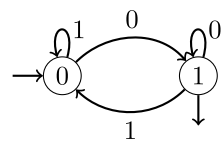

FIGURE 7 – Automate déterministe $\mathcal{A}_1'$.

2. Donnez un automate $\mathcal{A}_2$ à trois états qui accepte les représentations binaires comportant un zéro comme deuxième bit de poids faible (*ie* l'avant dernière lettre).
   Solution :

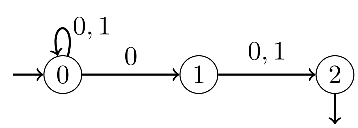

FIGURE 8 – Automate pour $\mathcal{A}_2$.

3. Donnez un automate $\mathcal{A}_3$ qui accepte l'intersection des langages acceptés par $\mathcal{A}_1$ et $\mathcal{A}_2$. Pour cela, utiliser le produit d'automates (rappel cours : la construction du produit des automates est correcte dans le cas de l'intersection, même si les automates $\mathcal{A}_1$ et $\mathcal{A}_2$ ne sont pas complets).

   Solution :
   On construit l'automate intersection par produit cartesien. On renomme respectivement Q_1, I_1, F_1 l'ensemble d'états, l'état initial et l'ensemble des étas terminaux de $\mathcal{A}_1$ et Q_2, I_2, F_2 les ensembles analogues de $\mathcal{A}_2$.

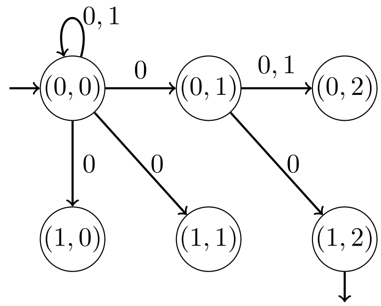

FIGURE 9 – Automate $\mathcal{A}_3$.

Les trois états $\{(1,0),(1,1),(0,2)\}$ n'appartiennent à aucun calcul de l'état initial vers l'état terminal, on peut donc les supprimer sans modifier le langage reconnu :

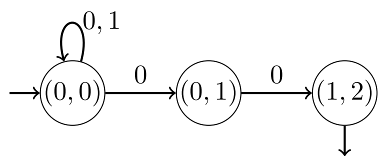

FIGURE 10 – Automate $\mathcal{A}_3$ après suppression des états inutiles.

L'automate obtenu n'est pas déterministe. Si on le déterminise on obtient :

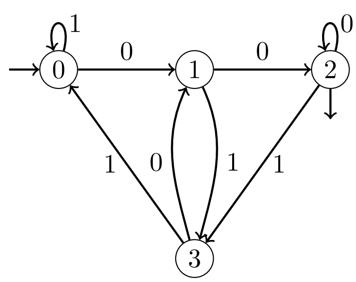

FIGURE 11 – Automate déterministe $\mathcal{A}_3$.

4. Donnez les cinq plus petits entiers dont les représentations binaires sont acceptés par $\mathcal{A}_3$. Déduisez-en le langage accepté par $\mathcal{A}_3$.

   Solution :
   0, 4, 8, 12, et 16. Le langage accepté est tout mot binaire représentant un entier divisible par 4.

Exercice 4. Langages et induction (5 points).
Soit un alphabet à deux lettres $A = \{a, b\}$ et soit le langage L, défini comme suit par induction :

- $a \in L$
- si le mot $u \in L$, alors les mots aua, aub, bua et $bub \in L$.

Prouver que les mots de L ont une longueur impaire.

Solution :
Pour prouver que tout mot $\in L$ est de taille impaire, prouvons la propriété suivante pour tout $n \in \mathbb{N}$:
P_n : "Tout mot dans L de taille n, est de taille impaire".
Nous prouvons maintenant P_n pour tout entier n par induction.

- *n=1* : Le seul mot concerné est a, qui est bien de longueur impaire.
- *n+1* : Supposons que P_n est vraie.
  - Si L ne contient pas de mot de longueur n+1, on n'a rien à démontrer.
  - Si $w \in L$ est de longueur $n + 1$: par définition de L, on a $w = xuy$, avec $x, y \in A$ et $u \in L$. Comme $|u| < |w|$, $|u|$ est impair. De plus, $|w| = 2 + |u|$, ce qui montre que $|w|$ est impair également.