TD : Automates finis non déterministes

Exercice 1

Soit $\Sigma = \{a, b\}$. Dans chaque cas, donner un afnd reconnaissant le langage proposé.

1. $L_1 = \varnothing$.
2. $L_2 = \{\varepsilon\}$.
3. $L_3 = \{bab\}$.
4. $L_4 = \{\text{mots commençant par } baba\}$.
5. $L_5 = \{\text{mots finissant par } baba\}$.
6. $L_6 = \{\text{mots contenant le facteur } baba\}$.

Exercice 2

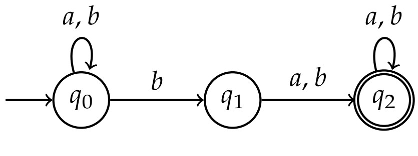
1. Donner un afnd A_1 reconnaissant les mots sur $\Sigma = \{a, b\}$ contenant le facteur *abba*.
2. Déterminiser A_1.
3. En déduire une fonction en langage *C* :

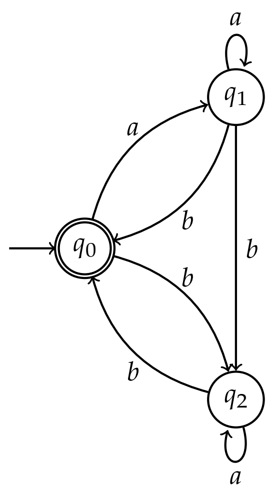
   ```
   bool facteur_abba(char s[], int n)
   ```

   qui détermine si le facteur baba existe dans une chaîne de caractères s ne contenant que des caractères a ou b.
4. Soit un texte T de longueur n et un motif P de longueur k. Justifier que l'on peut construire une fonction qui détecte la présence du motif P dans T en temps $O(n)$ en fonction de n et k.

Exercice 3

Déterminiser, les automates suivants :

1. 

2. 

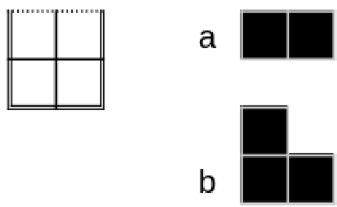
Exercice 4

Soit $\Sigma = \{a, b\}$. Soit L le langage des mots sur Σ tels que leur dernière lettre apparaît au moins deux fois dans le mot.

1. Construire un automate non déterministe reconnaissant L.
2. Déterminiser cet automate.

Exercice 5

Partie de Tétris parfaite



On considère un jeu de mini Tétris où la grille de jeu est carrée de dimension 2×2. On dispose de deux types de pièces a et b représentées ci-dessus. Lorsqu'une pièce arrive, le joueur peut la faire pivoter d'un quart de tour autant de fois qu'il le souhaite et la placer où il veut. Au moment où la pièce est placée, elle ne doit pas dépasser la ligne en pointillés. Une fois cette vérification effectuée, les lignes complètes se détruisent et les pièces au-dessus chutent. Une partie est représentée par un mot sur $\Sigma = \{a, b\}$ qui correspond à la liste de pièces proposées au joueur pendant la partie. Une partie est parfaite s'il est possible pour le joueur de placer toutes les pièces et de terminer avec la grille vide.

1. Montrer qu'on peut se ramener à cinq configurations possibles.
2. Donner un automate fini non déterministe reconnaissant les parties parfaites.
3. Déterminiser cet automate.
4. Décrire en français le langage des parties parfaites.

⋆Exercice 6

Le barman aveugle

Un barman joue avec un client au jeu suivant. Le client place 4 pièces sur leur côté pile ou face, en formant un carré, sur un plateau circulaire. Le but pour le barman est de réussir à retourner les pièces pour qu'elles soient toutes du côté face, ou toutes du côté pile. Pour cela, à chaque tour, il peut :

- retourner une des pièces,
- retourner deux pièces voisines,
- retourner deux pièces opposées.

Entre chaque tour, le client cherche à déstabiliser le barman en tournant le plateau d'un nombre quelconque de quart de tours. De plus, le client se doit de prévenir le barman s'il est arrivé dans une position gagnante.

1. En utilisant les symétries, montrer que l'on peut se restreindre à 4 configurations possibles pour le placement des pièces.
2. Représenter les actions possibles du barman à l'aide d'un automate fini non déterministe A. Dans cet automate, on fera boucler les transitions des états finaux pour modéliser le fait que le client annonce lorsque la partie est gagnée.
3. Déterminiser A.
4. En déduire une stratégie gagnante pour le barman.

⋆Exercice 7

Justifier que les langages suivants ne sont pas reconnaissables :

1. $L_1 = \{u \in \{a, b\}^* \mid |u|_a = |u|_b\}$
2. $L_2 = \{a^i b^j \mid j > i\}$
3. $L_3 = \{a^p \mid p \text{ est premier}\}$

⋆Exercice 8

Soit $\Sigma = \{a, b\}$ et $n \geq 2$ un entier. On considère $L = \Sigma^*.\{a\}.\Sigma^{n-2}$.

1. Donner un automate non déterministe à n états reconnaissant L.
2. Soit A un automate déterministe reconnaissant L, justifier que A possède au moins 2^{n-1} états. **Indication** : étudier l'application qui a tout mot $u \in \Sigma^{n-1}$ associe $\delta^*(q_0, u)$.
3. Que dire de l'algorithme de déterminisation d'un automate non détermininiste ?