

# TP4 JFLAP

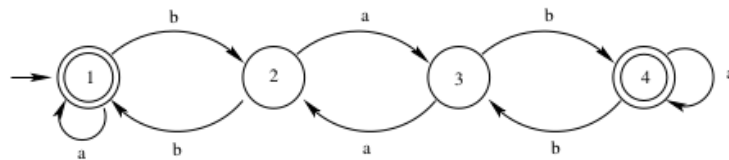
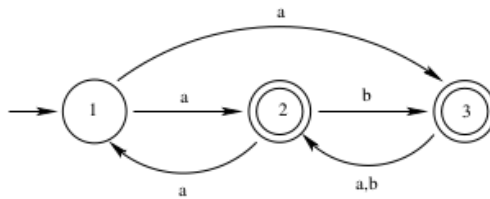
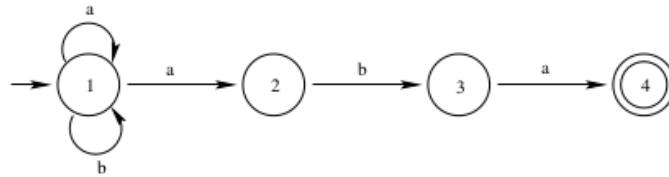
## Informatique théorique et automates

### L3 CDA

### Langages rationnels

### Exercice 1 : automate vers expression régulière

Avec JFLAP, à partir d'un automate, il est facile de générer l'expression régulière associée. Pour cela il suffit de choisir **Convert FA to RE** dans l'onglet **Convert**. Trouver l'expression régulière associée à chacun des automates finis suivants.



## Exercice 2

En construisant les automates correspondants, trouver les expressions régulières sur l'alphabet  $\{a, b\}$  permettant de décrire les langages suivants :

1. les mots comportant un nombre pair de  $a$  et un nombre pair de  $b$ .
2. les mots comportant un nombre impair de  $a$  et un nombre impair de  $b$ .
3. les mots comportant un nombre pair de  $a$  et un nombre impair de  $b$ .
4. Toutes les chaînes de  $a$  et  $b$  qui ne contiennent pas la sous-chaîne  $abb$ .
5.  $L = \{u \in \{a, b\}^* / |u|_a = 3n, n \in \mathbb{N} \text{ et ni } a^2 \text{ ni } b^3 \text{ ne sont facteurs de } u\}$ . Penser à minimiser l'automate (en passant par **Minimize DFA** dans **Convert**) avant de lancer le calcul de l'expression régulière associée !

## Exercice 3

On veut exprimer sous la forme d'une expression régulière le langage  $L$  tel que si  $L$  contient le facteur 0110 alors  $L$  ne contient pas 0101. Pour cela nous allons passer par la construction de l'automate reconnaissant  $L$ .

1. Construire d'abord l'automate fini déterministe complet reconnaissant le langage  $L_1$  des mots contenant le facteur 0110 et le sauvegarder.
2. Construire ensuite l'automate fini déterministe complet reconnaissant le langage  $L_2$  des mots contenant le facteur 0101 et le sauvegarder.
3. L'automate que l'on veut construire reconnaît en fait les mots ne contenant pas les deux facteurs 0110 et 0101 simultanément. Il s'agit donc du langage  $\overline{L_1 \cap L_2}$  qui peut s'écrire également  $\overline{L_1} \cup \overline{L_2}$  par De Morgan. Complémenter les deux automates précédents et les sauvegarder.
4. Faire l'union des deux automates précédents et déterminer l'automate.
5. Minimiser l'automate obtenu en cliquant sur **Minimize DFA** dans l'onglet **Convert**.
6. Générer l'expression régulière associée en cliquant sur **Convert FA to RE** dans l'onglet **Convert**.

## Exercice 4

On veut maintenant faire un ou exclusif entre les langages  $L_1$  et  $L_2$  précédents, c'est-à-dire trouver une expression régulière qui représente l'ensemble des mots contenant le facteur 0110 ou bien 0101. Ces mots ne peuvent donc contenir les deux facteurs simultanément mais contrairement au cas précédent ils doivent contenir au moins un des deux (alors que dans l'exercice 3 ils pouvaient ne contenir ni l'un ni l'autre). Le langage correspondant est donc  $L_1 \cap \overline{L_2} \cup \overline{L_1} \cap L_2$  que l'on doit écrire  $\overline{L_1 \cap L_2} \cup \overline{L_1 \cap L_2}$  pour pouvoir construire l'automate

Construire l'automate qui reconnaît  $\overline{L_1 \cup L_2} \cup \overline{L_1 \cup L_2}$ , le minimiser et en déduire l'expression régulière associée.

On peut également à partir d'une expression régulière construire l'automate associé avec **JFLAP**. Pour cela, dans la fenêtre d'accueil de **JFLAP**, choisir **Regular Expression**. Construire un automate qui reconnaît le langage décrit par l'expression régulière :

puis le minimiser.

Lors d'une addition de deux nombres  $a$  et  $b$  en base deux donnant le résultat  $c$  on peut toujours s'arranger pour que les trois nombres aient le même nombre de chiffres en rajoutant des 0 à gauche de certains d'entre-eux si nécessaire. Par exemple

peut s'écrire

1. Construire un automate reconnaissant l'ensemble des mots  $a_0b_0c_0a_1b_1c_1\dots a_nb_nc_n$  tels que la somme

soit correcte.

2. Le minimiser puis calculer l'expression régulière associée.
3. Faire la même chose (automate puis expression régulière) pour l'ensemble des mots  $a_n b_n c_n \dots a_0 b_0 c_0$ .