

TP3 JFLAP

Informatique théorique et automates

L3 CDA

Automates finis

Pour ce TP et les deux suivants nous allons utiliser JFLAP.

Sous Windows il suffit d'aller sur `J:\enseignants\guillou\L3CDA\LogiqueEtAutomates\TP` et de double-cliquer sur `JFLAP_THIN.jar`.

Sous Linux il suffit de lancer la commande : `java -jar /home/commun_depinfo/enseignants/guillou/L3CDA/LogiqueEtAutomates/TP/JFLAP_THIN.jar`.

Vous pouvez également le récupérer pour le copier sur votre machine personnelle.

Configurer JFLAP

Dans la fenêtre d'accueil de JFLAP, cliquer sur **Preferences** puis sur **Set the Empty String Character** et choisir **Epsilon**.

Ensuite choisir **Finite Automaton** dans le menu principal.

Exercice 1 : définition d'automates

Construire et sauvegarder les automates déterministes complets reconnaissant les langages suivants :

1. Sur l'alphabet $A = \{a\}$, l'ensemble des mots de longueur paire.
2. Sur l'alphabet $A = \{a\}$, l'ensemble des mots de longueur impaire.
3. Sur l'alphabet $A = \{a, b\}$, l'ensemble des mots contenant un nombre pair de a et au moins trois b .
4. Sur l'alphabet $A = \{a, b\}$, l'ensemble des mots contenant un nombre pair de a et exactement deux b .
5. Sur l'alphabet $A = \{a\}$, l'ensemble des mots contenant un nombre de a divisible par deux ou par trois.
6. Sur l'alphabet $A = \{a, b\}$, l'ensemble des mots contenant au moins trois a ou au moins trois b .
7. Sur l'alphabet $A = \{a, b\}$, l'ensemble des mots contenant au moins deux a ou au moins trois b .

Exercice 2 : mots acceptés ou refusés

Pour le troisième automate de l'exercice précédent, c'est-à-dire l'automate reconnaissant les mots contenant un nombre pair de a et au moins trois b , tester les mots suivants en sélectionnant **Multiple Run** dans le menu **Input** : $aabbb$, $babab$, $baaaa$, $aaaaaa$, bbb . Quels sont ceux qui sont refusés et ceux qui sont acceptés ? Pour $babab$ comment s'interprète le résultat affiché par **View Trace** ?

Exercice 3 : détermination et complémentaire

On veut construire l'automate fini déterministe qui reconnaît les mots sur l'alphabet $A = \{0, 1\}$ ne contenant pas le facteur 110. Pour cela nous allons passer par son complémentaire.

1. Construire un automate indéterministe reconnaissant les mots contenant le facteur 110 (4 états suffisent).
2. Aller dans le menu **Convert** et choisir **Convert to DFA** afin de le déterminer.
3. Si besoin ajouter un état puits non-accepteur afin de rendre complet l'automate précédent.
4. Complémenter l'automate précédent en changeant la nature de tous les états : les états accepteurs deviennent non-accepteurs et les états non-accepteurs deviennent accepteurs. Sauver l'automate sous le nom **aNon110**.
5. Tester l'automate sur des mots bien choisis.

Exercice 4 : intersection

Toujours sur l'alphabet $A = \{0, 1\}$ on veut maintenant construire un automate qui reconnaît les mots de A^* contenant le facteur 110 et ne contenant pas le facteur 101. Il s'agit clairement d'une intersection c'est-à-dire d'un produit d'automates que l'on ne peut pas faire directement avec **JFLAP**. Pour contourner ce problème on va utiliser De Morgan : si L_1 et L_2 sont deux langages reconnaissables alors $L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$.

1. Construire l'automate contenant le facteur 101 et le nommer **a101**.
2. Construire l'automate reconnaissant le langage des mots ne contenant pas le facteur 110 ou contenant le facteur 101. Pour cela aller dans la fenêtre de l'automate **aNon110** sélectionner **Convert** puis **Combine Automata** et choisir l'automate précédent **a101**. Une nouvelle fenêtre apparaît avec maintenant les deux automates. A partir de ces deux automates construire l'automate voulu.
3. Déterminer l'automate précédent puis le compléter si nécessaire.
4. Pour finir le compléter.
5. Le tester sur les mots suivants : 110, 101, 1001100011, 01101111 et 110110.