

Analyse Syntaxique – TP6

Automates à pile

Nous utilisons dans ce TP le logiciel JFLAP conçu pour expérimenter les concepts de la théorie des langages formels et de la théorie des automates. Le logiciel consiste en une archive Java JFLAP7.1.jar (disponible sur inscription) fournie dans le dépôt gitlab. Vous lancez son exécution avec la commande `java -jar JFLAP7.1.jar`.

Tutoriel

Réalisez le tutoriel sur la création et la simulation des automates à pile dans JFLAP.

Question 1. Vous allez utiliser JFLAP pour répondre aux questions suivantes.

1. Recréez dans JFLAP l'automate à pile de la figure 1.
2. Ecrivez les transitions correspondantes.
3. Cet automate est-il déterministe ?
4. Donnez la suite de transitions effectuées par l'automate pour le mot `aabbbb`.
5. Quel est le langage reconnu par cet automate ?

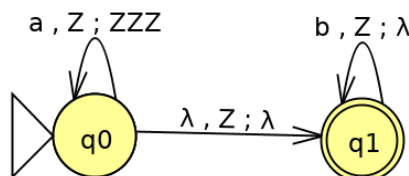


FIGURE 1 – Automate de la question 1.

JFLAP représente la pile de droite (bas) à gauche (sommet), à l'inverse du cours.

Palindromes

Pour un mot w , on note \bar{w} le miroir de w .

Question 2. Construire avec JFLAP un automate à pile déterministe acceptant $L_{pc} = \{wc\bar{w} \mid w \in \{a,b\}^*\}$, le langage des palindromes sur $\{a,b\}$ avec marqueur central c .

Question 3. Construire avec JFLAP un automate à pile acceptant $L_p = \{w\bar{w} \mid w \in \{a,b\}^*\}$, le langage des palindromes de longueur paire sur $\{a,b\}$.

Expressions arithmétiques

Nous reprenons la grammaire des expressions arithmétiques vue dans un précédent TP, sous une forme modifiée de sorte que les noms de variables soient constitués d'un seul caractère (c'est une contrainte de JFLAP) :

$$\begin{aligned}
 E &\rightarrow T \ G \\
 G &\rightarrow + \ T \ G \\
 G &\rightarrow - \ T \ G \\
 G &\rightarrow \epsilon \\
 T &\rightarrow F \ U \\
 U &\rightarrow * \ F \ U \\
 U &\rightarrow / \ F \ U \\
 U &\rightarrow \epsilon \\
 F &\rightarrow (\ E \) \\
 F &\rightarrow A \\
 A &\rightarrow i \ | \ n \ | \ - \ i \ | \ - \ n
 \end{aligned}$$

Question 4. Construire avec JFLAP un automate à pile acceptant par pile vide le langage associé à cette grammaire en utilisant la méthode descendante. Il n'est pas nécessaire ici de créer une variable par terminal, puisqu'il est possible d'empiler/dépiler des terminaux dans JFLAP. Ajouter par contre une transition initiale $(q_0, \epsilon, Z) \rightarrow (q_1, E)$ pour empiler l'axiome E . Toutes les transitions seront ensuite de q_1 vers q_1 .

Question 5. Simuler la reconnaissance du mot $i + i$ (Input \rightarrow Step with closure) et observer à chaque pas l'ensemble des configurations possibles affichées par JFLAP. Cela illustre l'explosion combinatoire qui peut survenir si on utilise un algorithme naïf pour construire un analyseur syntaxique à partir de l'automate à pile associé à une grammaire.

JSON

Nous reprenons la grammaire de JSON vue dans un précédent TP, sous une forme modifiée de sorte que les noms de variables soient constitués d'un seul caractère

$$\begin{aligned}
 J &\rightarrow O \ | \ A \\
 O &\rightarrow \{ \ M \ N \ } \ | \ \{ \ } \\
 M &\rightarrow s \ : \ V \\
 N &\rightarrow , \ M \ N \ | \ \epsilon \\
 A &\rightarrow [\ V \ B \] \ | \ [\] \\
 B &\rightarrow , \ V \ B \ | \ \epsilon \\
 V &\rightarrow s \ | \ n \ | \ f \ | \ t \ | \ z \ | \ O \ | \ A
 \end{aligned}$$

Question 6. Construire avec JFLAP un automate à pile acceptant par pile vide le langage associé à cette grammaire en utilisant la méthode ascendante. Empilez directement les terminaux comme précédemment, et faites attention à l'ordre des éléments de la pile. La transition vidant la pile quand l'axiome est en sommet de pile s'écrit en JFLAP $(q, \epsilon, JZ) \rightarrow (q, \epsilon)$.

Question 7. *Simuler la reconnaissance du mot $\{s: [n, n], s: f\}$ sans pas-à-pas (Input -> Fast Run) et observer l'explosion combinatoire des configurations possibles signalées par JFLAP. Cette explosion provient des règles $B \rightarrow \epsilon$ et $N \rightarrow \epsilon$ créant des transitions possibles à chaque pas. Observons précisément quand B doit vraiment donner le mot vide : quand le terminal est $]$ et que le sommet de pile contient V précédé d'une virgule $,$. On va donc remplacer la transition $(q, \epsilon, \epsilon) \rightarrow (q, B)$ de l'automate par la transition $(q,], V,) \rightarrow (q,]B)$. Faites de même avec N . La simulation doit maintenant fonctionner.*