Corrigé TD 3
Langages Formels

Luc Lapointe
luc.lapointe@ens-paris-saclay.fr
home.lmf.cnrs.fr/LucLapointe/

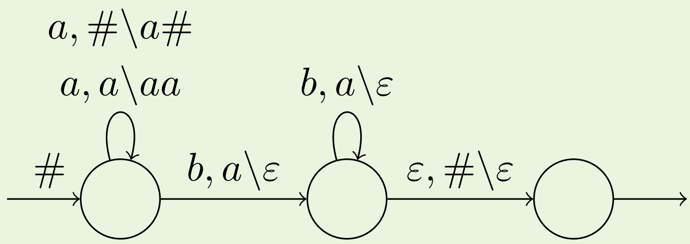
Exercice 1 – Construction d'AAP déterministes

Pour les langages suivants, construire des automates à piles déterministes les reconnaissant:

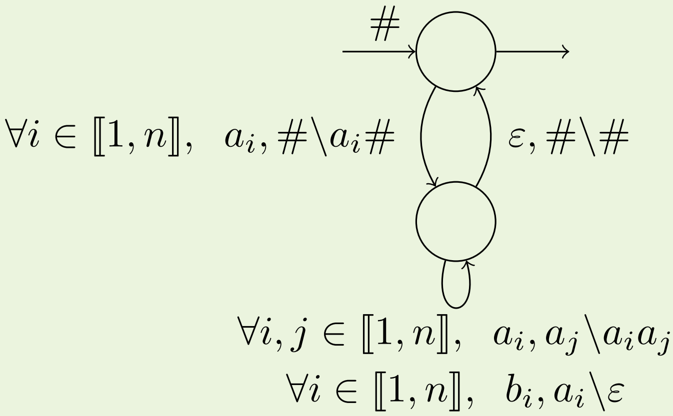
1. $\{a^n b^n \mid n \geq 1\}$
2. Le langage de Dyck D_n à n paires de parenthèses et sans mot vide.
3. $\{a^n b^m a^m b^n \mid n > 0 \land m > 0\}$
4. $\{a^n b^m a^m b^n \mid n > 0 \land m > 0\}^*$

Solutions

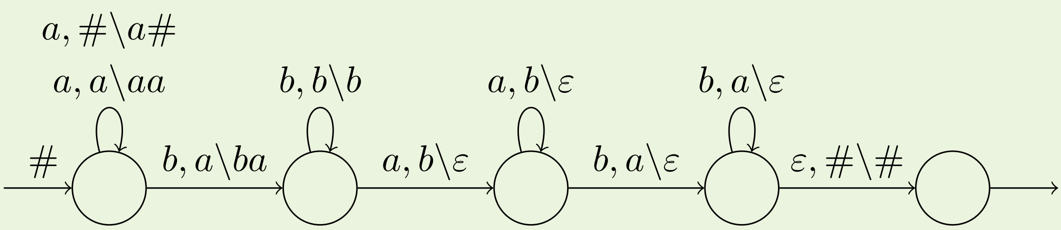
1.



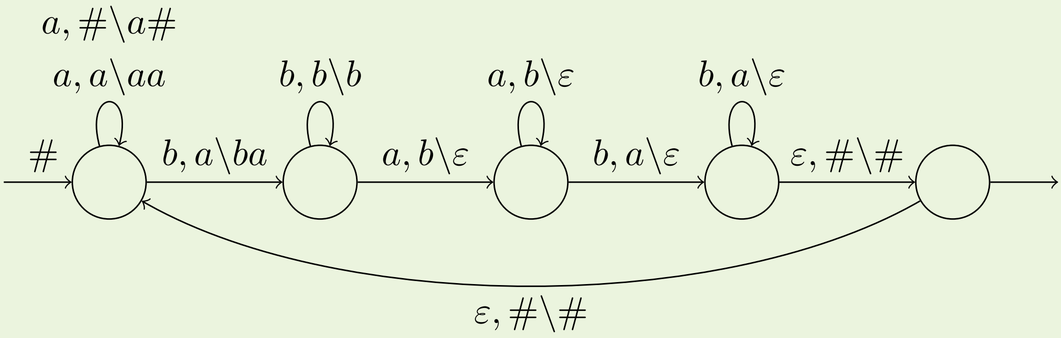
2.



3.



4.



Exercice 2 – ε-transition effaçantes

Soit A un automate à pile déterministe. Montrer qu'on peut effectivement construire un automate à pile qui reconnaît le même langage et dont les ε-transitions sont uniquement effaçantes : $(p, x) \rightarrow_\varepsilon (q, \varepsilon)$.

Idée de solution

Depuis chaque état p, calculer tous les chemins composés uniquement de ε-transitions. Pour chaque tel chemin non-effaçant $p \xrightarrow{\varepsilon}^* q$, pour chaque transition partant de q en lisant une lettre l et arrivant en r, rajouter une transition de p vers r lisant l et rajoutant sur la pile ce qui est rajouté par le chemin de ε-transitions en plus de ce qui est rajouté par la transition partant de q.

Exercice 3 – Clôtures

Montrer que la famille des langages algébriques est fermée par :

1. union
2. image miroir
3. intersection avec un langage rationnel

Idée de solution

1. Soient L_1 et L_2 deux langages algébriques, et $\mathcal{A}_1, \mathcal{A}_2$ des automates les reconnaissant. L'union des automates $\mathcal{A}_1$ et $\mathcal{A}_2$ reconnaît $L_1 \cup L_2$.
2. Soit L un langage algébrique, et G une grammaire le reconnaissant. Remplacer dans chaque règle de production de la grammaire le mot produit par son mot miroir donne une grammaire qui reconnaît le langage miroir de L.
3. Soit L_A algébrique et L_R rationnel, et $\mathcal{A}_A$ et $\mathcal{A}_R$ automates associés, où $\mathcal{A}_A$ accepte par état final. Considérer un « automate produit » de $\mathcal{A}_A$ et $\mathcal{A}_R$ où la partie correspondant à $\mathcal{A}_R$ n'interagit pas avec la pile.

Exercice 4 – Non ambiguïté

L'objectif de cet exercice est de montrer que tout langage reconnu par un automate à pile déterministe (AAPD) est non ambigu.

1. Montrer qu'un langage reconnu par un AAPD peut être reconnu par un AAPD dont aucune ε-transition ne part d'un état final.
2. La construction du cours pour passer d'un AAP à une grammaire nécessite un automate qui accepte par pile vide. Or, l'AAPD de la question précédente accepte par état final. Trouver une manière de résoudre cette difficulté pour pouvoir quand même exploiter le squelette de la démonstration du cours.
3. Imiter le passage d'automate à grammaire dans le cours mais avec une propriété à montrer par récurrence plus forte pour conclure.
4. Montrer que l'inclusion est stricte en considérant le langage

$$L = \{a^n b^n \ |n \geq 1\} \cup \{a^n b^{2n} \ |n \geq 1\}$$

Idée de solution

1. cf cours
2. On peut rajouter depuis l'état final une ε-transition vers un nouvel état, qui vide la pile grâce à des ε-transitions.
3. L'hypothèse mentionnée dans les transparents est :

$$\langle q, z, q' \rangle \to^* w \text{ dans } G \text{ ssi } qz \xrightarrow{w}^* q' \text{ dans } \mathcal{A}$$

   Essayer à la place : Si $\langle q_0, z_0, q_f \rangle = \alpha_0 \to \ldots \to \alpha_n = w$ est une dérivation gauche de G, alors α_i est de la forme

$$u_i \langle q_{i,1}, z_{i,1}, q_{i,2} \rangle \langle q_{i,2}, z_{i,2}, q_{i,3} \rangle \ldots$$

   tels que u_i est composé de terminaux, et que $q_0 z_0 \to q_{i,1} z_{i,1} \to \ldots$ est un calcul de $\mathcal{A}$.

   Cette propriété se prouve par récurrence sur n. On conclut en constatant que en considérant deux dérivations gauches différentes d'un même mot w, on obtient deux calculs différents sur w dans $\mathcal{A}$, d'où une contradiction.
4. Le langage L est non ambigu, et est par exemple produit par :
   $S \to S_1 \mid S_2$
   $S_1 \to aS_1b \mid ab$
   $S_2 \to aS_2bb \mid abb$
   Par contre, si un AAPD reconnaît L, alors soit $q_i z_i$ la configuration dans laquelle il se trouve après avoir lu $a^i b^i$. Par lemme des tiroirs et car L n'est pas régulier, il existe une sous-famille d'indices I telle que :

- Il existe q tel que pour tout $i \in I$, lire $a^i b^i$termine dans q,
- Pour tout $N \in \mathbb{N}$, il existe un $j \in I$ arbitrairement grand tel que lire $a^j b^j$ termine dans une configuration avec une pile de taille au moins N.

Soit $i_0 \in I$, et $j > i_0$ tel que lire $a^j b^j$ termine dans une configuration dont la pile est de taille au moins i_0. Alors le calcul à partir de $a^{i_0} b^{i_0}$ qui accepte en lisant i_0 lettres b peut se dérouler à l'identique depuis $a^j b^j$ et aboutir à une acceptation de $a^j b^{j+i_0}$, d'où une contradiction.

Contrôle continu – Propriétés d'AAP

1. Montrer qu'un langage L est déterministe et préfixe $(L \cap L\Sigma^+ = \emptyset)$ ssi il existe un automate déterministe qui accepte L par pile vide.
2. Montrer que pour les automates à pile déterministes, l'acceptation par pile vide est équivalente à l'acceptation par pile vide ET état final.