TD 4
Lemme de pompage et déterminisation

Version du 26 septembre 2016

Exercice 1 – Listes de listes de listes...

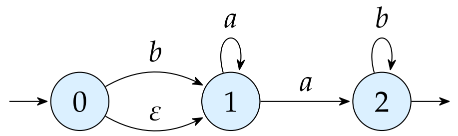
Reprenons une question du TD précédent :

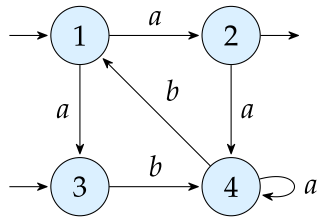
> La notion de liste est étendue récursivement pour inclure les listes de listes, les listes de liste de liste... comme ((1 : 3) : 3 : (2 : 1) : ((1 : 2))). Est-il possible de reconnaître les listes avec un automate fini ?

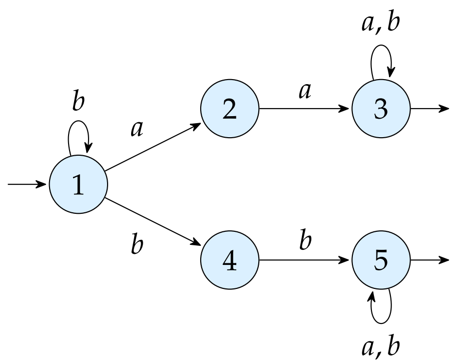
1. Utilisez le lemme de pompage pour les langages rationnels afin de démontrer que le langage $L_p = \{({}^n1)^n \mid n \geq 0\}$ n'est pas rationnel.
2. Déduisez-en qu'il n'est pas possible de reconnaître le langage L_l, composé de listes, listes de listes, listes de listes de listes... avec un automate fini.

Exercice 2

On suppose $\Sigma = \{a, b\}$. En utilisant les méthodes du cours, construisez un automate déterministe équivalent à chacun des automates suivants :







Exercice 3 – Recherche de motifs

Dans cet exercice on considère $\Sigma = \{a, b, c\}$.

1. Soit le mot $m = abab$ et L le langage des mots qui ont m comme suffixe. L contient les mots de la forme $v = um$, avec $u \in \Sigma^\star$; soit, par exemple, les mots *aaabab* et *babab*. En revanche, *caabc* n'appartient pas à L. Prouvez que L est rationnel. Proposez un automate fini *non-déterministe* A_n pour L. Vous justifierez votre construction.
2. Transformez A_n en un automate déterministe complet A_d équivalent, en utilisant une construction étudiée en cours. Vous expliciterez les étapes de l'application de l'algorithme.
3. Pourquoi est-il évident que A_d soit complet et émondé ?

4. On modifie l'alphabet avec $\Sigma = \{a, b, c, d, e\}$ pour cette question uniquement. Comment répercuter cette modification sur A_d ?
5. On considère l'algorithme suivant :

   // $u = u_1 \ldots u_n$ est le mot dans lequel on cherche.
   // $A_d = (\Sigma, Q, \{q_0\}, F, \delta)$ est un automate déterministe pour L.
   $q \leftarrow q_0$
   $i \leftarrow 1$
   $c \leftarrow 0$
   tant que ($i \leq n$) **faire**
       $q \leftarrow \delta(q, u_i)$
       $i \leftarrow i + 1$
       si ($q \in F$) **alors** $c \leftarrow c + 1$ **fin si**
   fin tant que

   a. Illustrez le fonctionnement de cet algorithme lorsque $u = bcababcabbababac$ et l'automate A_d calculé à la question 2. Vous donnerez pour chaque passage dans la boucle principale la valeur de c.
   b. Que calcule cet algorithme en général ? Justifiez votre affirmation.
   c. Quelle est la complexité de cet algorithme ?
   d. Que vaut c à la fin de l'exécution de l'algorithme pour $u = cabbababababc$? Que remarquez vous ?
   e. Comment faudrait-il modifier l'automate A_d pour compter que le nombre maximal d'occurrences disjointes du motif au sein de la chaîne d'entrée ? Par exemple la réponse devrait être 2 dans le dernier exemple.