MITRO210: Automates et données structurées

Feuille d'exercices 6

Antoine Amarilli

1 Exemples d'automates d'arbres

Dans cet exercice, on construit des automates d'arbres pour quelques langages explicites afin de se familiariser avec le concept.

Question 0. Dans cette question et la suivante, on prend $\Sigma = \{a, b, c\}$. Soit L_0 le langage des Σ-arbres où il y a un nœud étiqueté a. Montrer que L_0 est reconnaissable.

Question 1. Soit L_1 le langage des Σ-arbres où il y a un nœud étiqueté a dont l'enfant gauche a un descendant étiqueté b et dont l'enfant droit a un descendant étiqueté c. Montrer que L_1 est reconnaissable. (Ici, un descendant d'un nœud n est soit le nœud n lui-même soit un descendant strict de n.)

Question 2. Dans cette question et la suivante, on prend $\Sigma' = \{0, 1, +, \times\}$. On considère le langage L_2 formé des Σ'-arbres qui représentent une opération arithmétique bien formée qui s'évalue à un entier non-nul. Montrer que ce langage est reconnaissable.

Question 3. Que penser du langage L_3 formé des Σ'-arbres qui sont bien formés et s'évaluent à l'entier 42 ?

Question 4. Montrer que, pour tout Σ-arbre T, le langage $\{T\}$ formé du seul arbre T est un langage reconnaissable. On pourra utiliser le nondéterminisme, ou la notion de *sérialisation* d'un arbre.

Question 5. Sur un Σ-arbre T, on appelle *première feuille* la feuille obtenue en partant de la racine et en prenant l'enfant gauche à chaque fois jusqu'à atteindre une feuille, et *dernière feuille* la feuille obtenue en partant de la racine et en prenant l'enfant droit à chaque fois jusqu'à atteindre une feuille.

Montrer que le langage des Σ-arbres où la première feuille et la dernière feuille ont la même étiquette est un langage reconnaissable.

2 Propriétés des automates d'arbres

Dans cet exercice, on se familiarise avec la notion d'automates d'arbres de bas en haut, en démontrant certaines propriétés simples à leur sujet.

Question 0. Montrer que les langages reconnaissables d'arbres sont clos par l'opération de complémentation : si un certain langage d'arbres L est reconnaissable, alors le langage des arbres qui ne sont pas dans L est également reconnaissable.

Question 1. Montrer que les langages reconnaissables d'arbres sont clos par union et par intersection : pour tous langages d'arbres L_1 et L_2 reconnaissables, les langages $L_1 \cup L_2$ et $L_1 \cap L_2$ sont également reconnaissables.

Question 2. Montrer que tous les langages finis d'arbres sont reconnaissables.

Question 3. Montrer que les automates d'arbres déterministes et nondéterministes de bas en haut définissent effectivement la même notion de langage reconnaissable.

Question 4 (bonus). Énoncer et démontrer un lemme de pompage pour les langages réguliers d'arbres. En déduire que le langage des arbres où les sous-arbres gauche et droit de la racine ont le même nombre de feuilles n'est pas un langage d'arbres reconnaissable.

3 3-coloriage avec contraintes

Dans cet exercice, on s'intéresse à un langage spécifique dit de 3-coloriage avec contraintes *qui sera étudié de manière plus générale la semaine prochaine.*

On considère l'alphabet $\Sigma = \{R, G, B, \perp\}$. Étant donné un Σ-arbre T, un *coloriage* de T est une fonction c des nœuds de T dans $\{R, G, B\}$. Le coloriage est *valide* s'il satisfait les deux conditions suivantes :

- Pour tout nœud n portant une étiquette $a \in \{R, G, B\}$, on a $c(n) = a$.
- Pour toute paire n et n' de nœuds distincts adjacents dans T (c'est-à-dire que n est parent de n' ou vice-versa), on a $c(n) \neq c(n')$.

On dit qu'un Σ-arbre T est *coloriable* s'il admet un coloriage valide.

Question 0. On considère le langage L_0 des Σ-arbres coloriables où l'étiquette $\perp$ n'apparaît pas. Montrer que L_0 est un langage reconnaissable.

Question 1. En déduire que le langage L_1 des Σ-arbres coloriables est un langage reconnaissable. On pourra utiliser le nondéterminisme.

Question 2. En conclure que l'on peut tester en temps linéaire, étant donné un Σ-arbre, s'il est coloriable ou non.

Question 3. On souhaite à présent résoudre le problème suivant : étant donné un Σ-arbre T, compter le nombre de coloriages valides de T. Proposer un algorithme efficace pour résoudre ce problème.