

Automates Avancés

Travaux Dirigés n°3

► Exercice 1. *Algorithme L^**

Appliquez l'algorithme L^* pour apprendre l'automate *minimal* pour le langage de l'automate de la figure 1. Les contre-exemples qui sont renvoyés successivement après les tests d'équivalence sont 01,10 et 0001. Est-ce qu'un autre premier contre-exemple permettrait de faire moins de tests d'équivalence ?

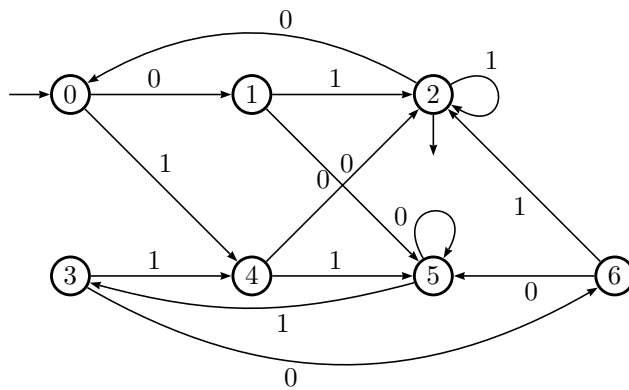


Figure 1: Automate à apprendre.

► Exercice 2. *Algorithme L^**

La table construite dans l'algorithme L^* peut être non-consistante, c.à-d. on peut avoir deux lignes $T(u)$ et $T(v)$ en haut de la table avec $T(u) = T(v)$ mais $T(ua) \neq T(va)$ pour un $a \in \Sigma$.

- Modifiez l'algorithme de sorte que la table reste toujours consistante. Indication: Changez le traitement du contre-exemple renvoyé par le test d'équivalence.
- Refaites l'exemple de l'exercice précédent avec le nouvel algorithme.

► Exercice 3.

Est-ce qu'on peut apprendre un automate pour un langage régulier si on peut demander au professeur

- uniquement des requêtes de membre ?
- uniquement des requêtes d'équivalence ?

► **Exercice 4.**

Soit Σ un alphabet fini. Un *échantillon* est une paire (Pos, Neg) de deux ensembles *finis* de mots $Pos, Neg \subseteq \Sigma^*$. Un automate A est appelé compatible avec l'échantillon (Pos, Neg) , si $Pos \subseteq L(A)$ et $L(A) \cap Neg = \emptyset$.

- En général combien y a-t-il d'automates compatibles ?
- Considérez $Pos = \{acb, aacb, acbbe, aaace\}$ et $Neg = \{c, bca, bb, aaa\}$.
 - Quel est l'automate déterministe minimal compatible qui reconnaît le moins de mots possible ?
 - Quel est l'automate déterministe minimal compatible qui reconnaît le plus de mots possible ?
 - Donnez un automate déterministe compatible plus petit que les deux automates précédents. Donnez les grandes lignes d'un algorithme pour l'obtenir.
 - Si $Neg = \emptyset$ quel problème se pose-t-il ?