

Bruno Mery (Doctorant, LaBRI bureau 123, mery@labri.fr)  
Université de Bordeaux, année 2007-2008  
Informatique Théorique 1  
INF154T Groupe CSB5A2

## Travaux Dirigés

# Mots, langages et Automates

Addendum : Exemple de minimisation à la volée

Voici un exemple complet de minimisation d'un automate, en prenant pour base un exercice d'André Arnold (2003-2004). L'automate de départ est donné en Fig. 1. (La procédure est abrégée par rapport au cours et moins tabulatoire, mais reste – je l'espère – explicite.)

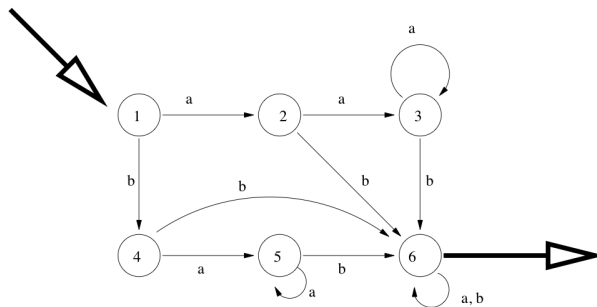


FIG. 1 – Automate à minimiser

- Au départ, on considère deux ensembles d'états : l'état 6 est terminal, les 1, 2, 3, 4, 5 sont non-terminaux.
- Considérons les transitions étiquetées par  $a$  :  $a$  envoie 1 dans un état de  $\{1, 2, 3, 4, 5\}$ , de même pour 2, 3, 4 et 5. Il n'y a donc pas lieu de partitionner les ensembles.
- Considérons maintenant les transitions étiquetées par  $b$  : alors que  $a$  envoie 1 sur un état de  $\{1, 2, 3, 4, 5\}$ , il envoie 2, 3, 4 et 5 sur 6. On distingue donc trois ensembles :  $\{1\}$ ,  $\{2, 3, 4, 5\}$ ,  $\{6\}$ .

- Remarque : il est impossible de raffiner les ensembles n'ayant qu'un état, on ne s'intéresse donc qu'à ceux qui en ont au moins deux. . .
- Reprenons avec  $a$  :  $a$  envoie 2 dans un état de  $\{2,3,4,5\}$ , de même pour 3, 4 et 5. Les comportements restent identiques, il n'y a pas lieu de partitioner les ensembles.
- Reprenons avec  $b$  :  $b$  envoie 2 dans 6, de même pour 3,4,5. Pas de distinction non plus.
- Les ensembles sont stables, la minimisation est donc finie.
- Pour construire l'automate minimal, on a donc trois états : 1 (initial), 6 (final) et  $K = \{2,3,4,5\}$ , qui correspond à la fusion de ces quatre états. On se base sur l'automate de départ pour avoir les transitions correctes :  $a$  et  $b$  envoient 1 vers  $K$ ,  $a$  boucle sur  $K$ ,  $b$  envoie  $K$  vers 6,  $a$  et  $b$  bouclent sur 6. Le résultat est donné en Fig. 2.

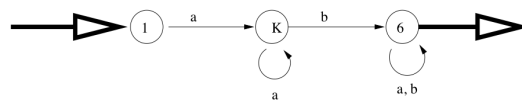


FIG. 2 – Automate minimal

Le point principal est : on partitione un ensemble d'états quand ils n'ont pas tous le même *comportement*, c'est-à-dire quand une transition ne les envoie pas tous sur une même classe (quelle qu'elle soit). On peut aussi utiliser la même méthode pour *vérifier qu'un automate est minimal* (l'algorithme donne les mêmes états qu'au début).