Minimisation des Automates Finis

Informatique Théorique 2
Licence 3 informatique

Sébastien Verel
verel@univ-littoral.fr
http://www-lisic.univ-littoral.fr/~verel

Université du Littoral Côte d'Opale
Laboratoire LISIC
Equipe OSMOSE

Plan

Objectifs de la séance 02

- Savoir minimiser un automate fini
- Savoir comparer deux automates

Source, bibliographie

Sandrine Julia, Université de Nice.

Voir aussi :
`https://fr.wikipedia.org/wiki/Minimisation_d%27un_automate_fini_d%C3%A9terministe`

Comment savoir que 2 automates finis sont équivalents ?

Comment savoir que 2 automates finis reconnaissent le même langage ?

Automate minimal

Définition automate minimal

Un automate minimal qui reconnait le langage L est un automate dont le nombre d'état est minimal et qui reconnait le langage L.

Théorème (admis)

Un langage rationnel est reconnu par un **unique** automate fini déterministe minimal (complet).

Au renomage près, on parle donc de l'automate minimal fini déterministe reconnaissant un langage rationnel

Construction de l'automate minimal

Principe de Minimisation

Fusionner les états équivalents, principe de séparation des états :

1. Séparation en 2 classes : états finaux et non finaux
2. Dans chaque classe, on sépare les états non équivalents
3. On renouvelle (2) jusqu'à stabilisation

Etats équivalents

Définition

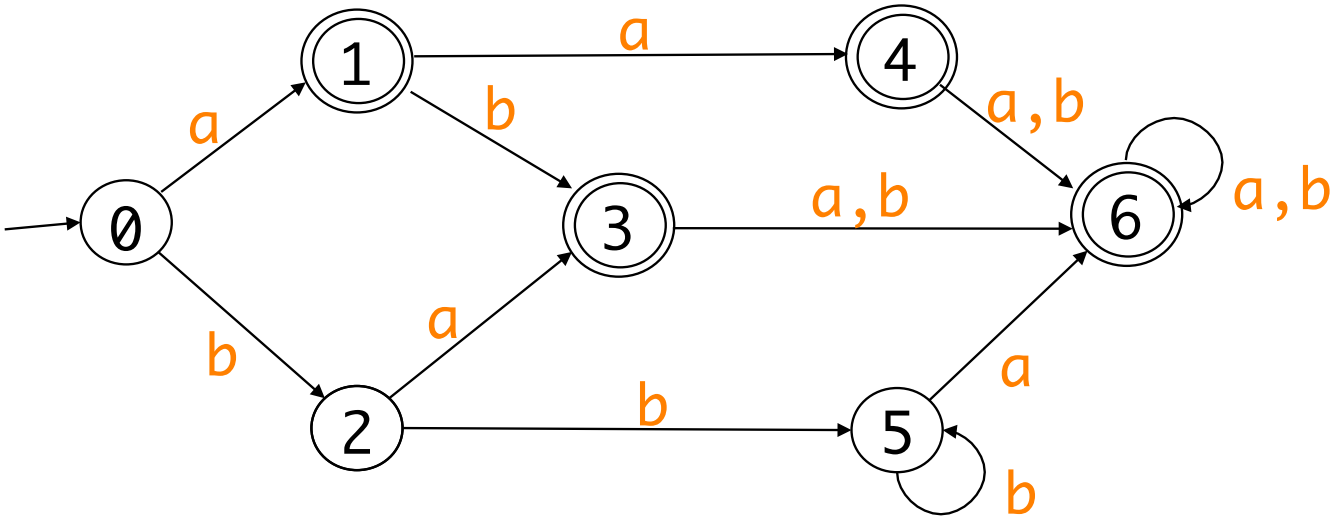
Soit A un AFD, deux états p et q sont équivalents si et seulement si leurs langages respectifs associés sont identiques :

$p \approx q$ ssi $L_p(A) = L_q(A)$

$p \approx q$ ssi $\forall w \in \Sigma^*$, $\delta^*(p, w) \in F$ et $\delta^*(q, w) \in F$
ou bien $\delta^*(p, w) \notin F$ et $\delta^*(q, w) \notin F$

$\approx$ est une relation d'équivalence,
et l'on note $[q]$ l'ensemble des états équivalents à q.

Exemple (de S. Julia)



Les états 0 et 4 sont-ils équivalents ?
Les états 3 et 6 sont-ils équivalents ?

Automate minimal

Soit $A = (\Sigma, Q, \delta, q_0, F)$ un automate fini déterministe complet.

L'automate minimal associé à A est $A_{min} = (\Sigma, Q', \delta', [q_0], F')$ avec :

- $Q' = \{[q], q \in Q\}$
- $\delta'([p], \sigma) = [q]$ s'il existe $p \in [p]$ et $q \in [q]$ tels que $\delta(p, \sigma) = q$
- $F' = \{[f], f \in F\}$

Algorithme de minimisation

Algorithme de Moore (1956)

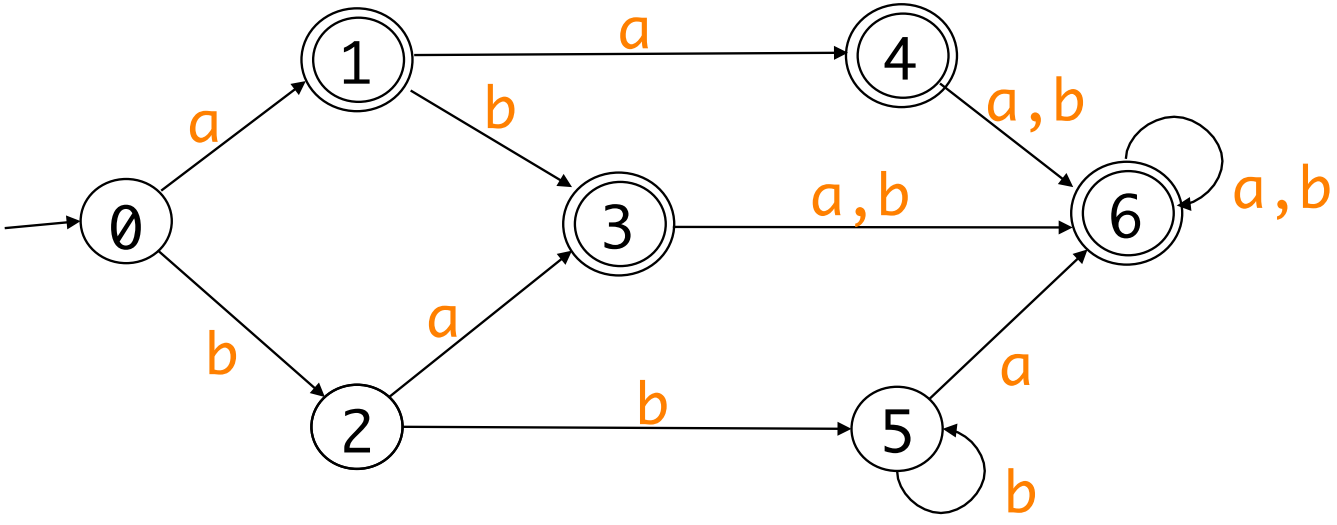
Initialisation :

$p \approx_0 q$ ssi $(p \in F$ et $q \in F)$ ou $(p \notin F$ et $q \notin F)$

Récurrence :

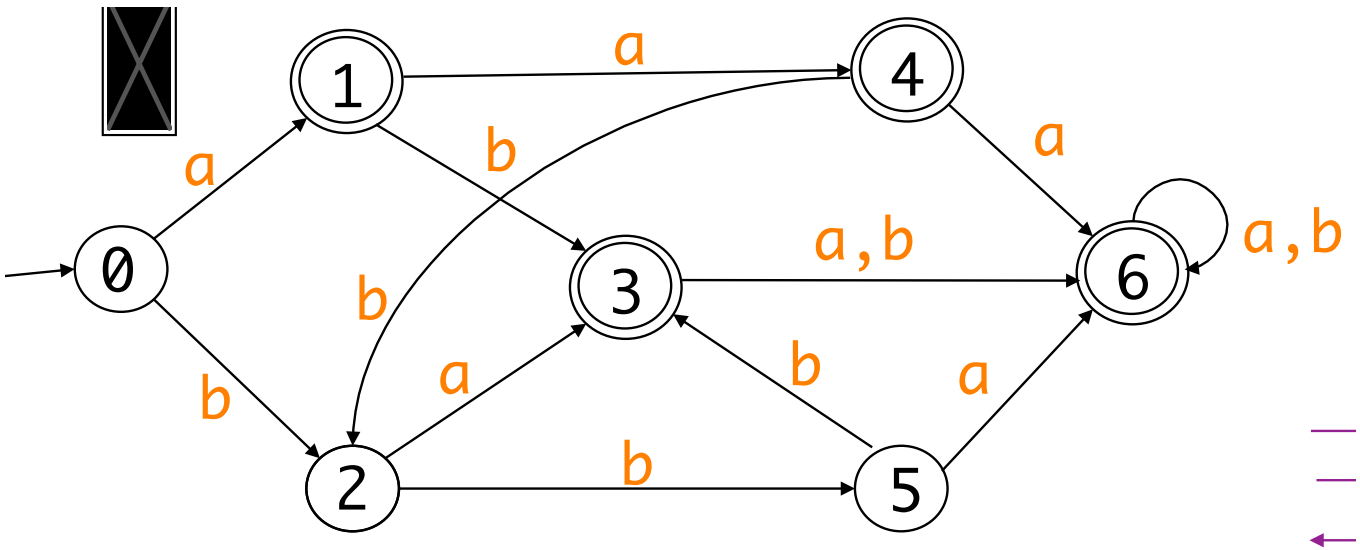
$p \approx_{i+1} q$ ssi $(p \approx_i q)$ et $(\forall \sigma \in \Sigma,\ \delta(p, \sigma) \approx_i \delta(q, \sigma))$

Exemple 1 de minimisation



Ecrire la première partition, puis la seconde en examinant tous les états de chaque classe.

Exemple 2 de minimisation (S. Julia)



Conclusion

L'automate minimal occupe moins de place mémoire.

La minimisation d'un automate permet de mettre celui-ci sous forme canonique (unique)
et ainsi de pouvoir comparer 2 automates.