



INF 302 : LANGAGES & AUTOMATES

Chapitre 3 : Automates à états finis déterministes

— opérations sur les automates et fermeture des langages à états

Yliès Falcone

`ylies.falcone@univ-grenoble-alpes.fr` — `www.ylies.fr`

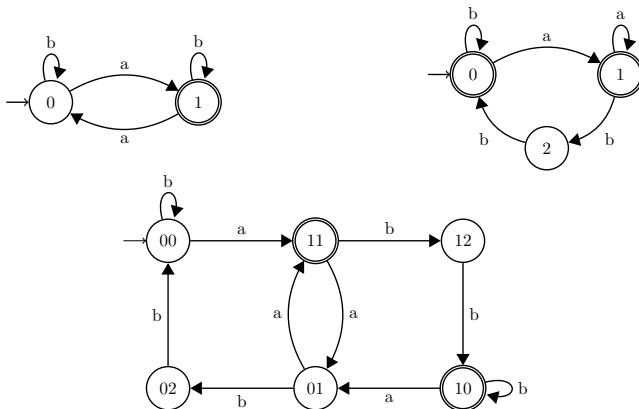
Univ. Grenoble-Alpes, Inria

Laboratoire d'Informatique de Grenoble - `www.liglab.fr`

Équipe de recherche LIG-Inria, CORSE - `team.inria.fr/corse/`

Année Académique 2018 - 2019

Intuition et objectifs



- Opérations sur automate/opérations sur langage :
 - négation/complémentation,
 - produit/intersection.
- Fermeture de l'ensemble des langages à états.

Fermeture de EF par complémentation et intersection

Fermeture de EF par **complémentation**

Soit A un AEFD.

- 1 Le langage $\Sigma^* \setminus L(A)$ est-il reconnaissable par un AEFD ?
- 2 Si oui, peut-on construire de manière effective un automate qui reconnaît $\Sigma^* \setminus L(A)$?

Fermeture de EF par **intersection**

Soient A et B deux AEFDs.

- 1 Le langage $L(A) \cap L(B)$ est-il reconnaissable par un AEFD ?
- 2 Si oui, peut-on construire de manière effective un automate qui reconnaît $L(A) \cap L(B)$?

Nous pourrions répondre de manière affirmative à toutes ces questions.

Plan Chap. 3 - Automates à états finis déterministes - opérations et fermeture des langages à états

- 1 Complétion d'un automate
- 2 Négation d'un automate / Complémentation du langage reconnu
- 3 Produit d'automates / Intersection des langages reconnus
- 4 Résumé

Plan Chap. 3 - Automates à états finis déterministes - opérations et fermeture des langages à états

- 1 Complétion d'un automate
- 2 Négation d'un automate / Complémentation du langage reconnu
- 3 Produit d'automates / Intersection des langages reconnus
- 4 Résumé

Complétion d'automates – idée



Soit $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$ un AEFD qui reconnaît un langage (noté $L(A)$).

Objectifs de la complétion :

- construire un AEFD **complet** qui reconnaît $L(A)$,
- travailler avec des automates complets pour certaines transformations,
- raisonner sur des automates complets est parfois plus simple.

Idée de la complétion :

- 1 Ajouter un nouvel état puits à Q .
- 2 Diriger toutes les transitions indéfinies dans A vers l'état puits.

◀ Complétion d'un automate

Complétion d'automates

Définition

Soit $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$ un AEFD qui reconnaît un langage (noté $L(A)$).

Définition (Complétion d'automates)

L'automate complété de A est $C(A) = (Q \cup \{q_p\}, \Sigma, q_{\text{init}}, C(\delta), F)$ tel que

- $q_p \notin Q$ et
- $C(\delta) : Q \cup \{q_p\} \times \Sigma \rightarrow Q \cup \{q_p\}$ est une *application* définie par :

$$C(\delta)(q, a) \stackrel{\text{def}}{=} \begin{cases} \delta(q, a) & \text{pour tout } (q, a) \in \text{dom}(\delta) \\ q_p & \text{sinon} \end{cases}$$

\Leftrightarrow Ajouter un état "puis"

Correction de la procédure de complétion

$$L(A) = L(C(A))$$

Démonstration.

En TD.



Plan Chap. 3 - Automates à états finis déterministes - opérations et fermeture des langages à états

- 1 Complétion d'un automate
- 2 Négation d'un automate / Complémentation du langage reconnu
- 3 Produit d'automates / Intersection des langages reconnus
- 4 Résumé

Négation d'un automate

Soit $A = (Q, \Sigma, q_{\text{init}}, \delta, F)$ un AEFD *complet*.

Définition (Complémentation d'un AEFD complet)



Le complémentaire de A est l'automate $A^c = (Q, \Sigma, q_{\text{init}}, \delta, Q \setminus F)$.

Procédure de complémentation d'un AEFD (quelconque) A :

- ① **Construire** $C(A)$. *Automate complète*
- ② Inverser les états accepteurs et non-accepteurs dans $C(A)$.

Propriété : correction de la procédure de complémentation

$$L(A^c) = \Sigma^* \setminus L(A).$$

Démonstration.

En TD.

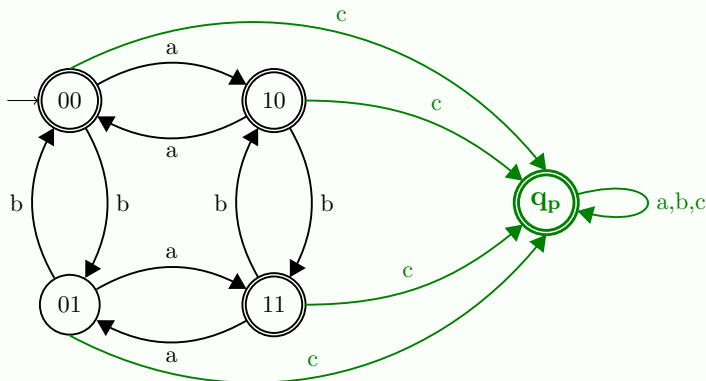


Négation d'un automate : exemple

Exemple (Complémentation d'un automate)

Sur $\Sigma = \{a, b, c\}$:

un nombre **impair** de a ou un nombre **pair** de b
ou un c



Plan Chap. 3 - Automates à états finis déterministes - opérations et fermeture des langages à états

- 1 Complétion d'un automate
- 2 Négation d'un automate / Complémentation du langage reconnu
- 3 Produit d'automates / Intersection des langages reconnus
- 4 Résumé

Produit d'automates

Considérons deux AEFDs : $A = (Q^A, \Sigma, q_{\text{init}}^A, \delta^A, F^A)$ et $B = (Q^B, \Sigma, q_{\text{init}}^B, \delta^B, F^B)$.

Objectif : construire un automate qui accepte les mots reconnus par les deux automates (à la fois). Le langage reconnu par l'automate produit est donc l'*intersection* des langages des automates passés en paramètres.

◁ Produit de deux automates

Définition (Produit d'automates)

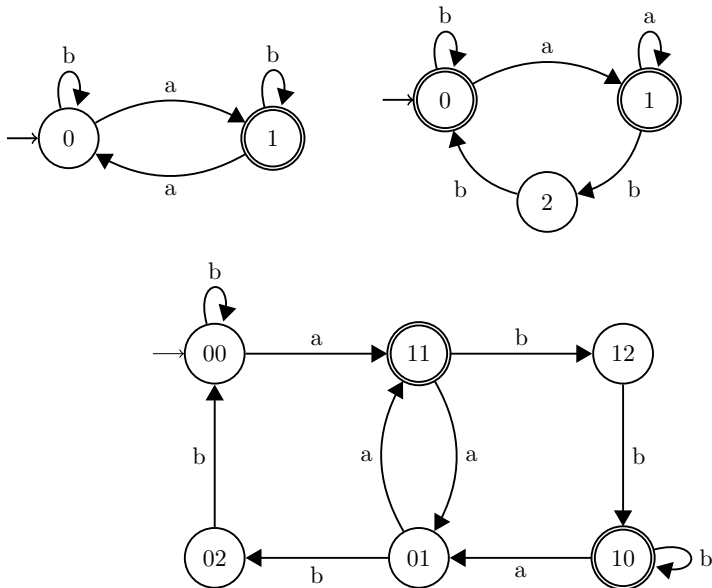
L'automate produit de A et de B est $A \times B = (Q, \Sigma, q_{\text{init}}, \delta, F)$ où :

- $Q = Q^A \times Q^B$
- $q_{\text{init}} = (q_{\text{init}}^A, q_{\text{init}}^B)$
- $\delta : (Q^A \times Q^B) \times \Sigma \rightarrow (Q^A \times Q^B)$ est telle que

$$\delta((q^A, q^B), a) = (\delta^A(q^A, a), \delta^B(q^B, a))$$

- $F = F^A \times F^B$.

Produit d'automates : exemple



Fermeture par intersection

Théorème

Soient $A = (Q^A, \Sigma, q_0^A, \delta^A, F^A)$ et $B = (Q^B, \Sigma, q_0^B, \delta^B, F^B)$ deux AEFDs.

- $L(A \times B) = L(A) \cap L(B)$.
- La classe EF des langages à états est **fermée par intersection**.

Pour montrer $L(A \times B) = L(A) \cap L(B)$, on doit montrer :

1. $L(A \times B) \subseteq L(A)$,
2. $L(A \times B) \subseteq L(B)$ et
3. $L(A) \cap L(B) \subseteq L(A \times B)$

Pour 1.) et 2.) nous attendrons un prochain cours pour avoir un moyen élégant pour le faire

3) sera fait sous forme d'exercices en TD.

Plan Chap. 3 - Automates à états finis déterministes - opérations et fermeture des langages à états

- 1 Complétion d'un automate
- 2 Négation d'un automate / Complémentation du langage reconnu
- 3 Produit d'automates / Intersection des langages reconnus
- 4 Résumé**

Résumé du chapitre 3 : opérations sur les automates et fermeture des langages à états

Automate à États Fini Déterministes

- Calcul de l'automate *complété* (même langage).
- Calcul de l'automate *complémentaire* (complémentaire d'un langage).
- Calcul de l'automate *produit* de deux automates (intersection de langages).
- *Fermeture* des langages réguliers par *complémentation* et *intersection*.

En TD

- Donner les algorithmes pour procédures de complétion et complémentation.
- Définir des procédures permettant de calculer des automate reconnaissant l'union et le xor des langages d'automates passés en paramètres.
- ...