

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année



INF 302 : LANGAGES & AUTOMATES

Chapitre 4 : Opérations sur les automates déterministes et fermeture des langages à états

Yliès Falcone

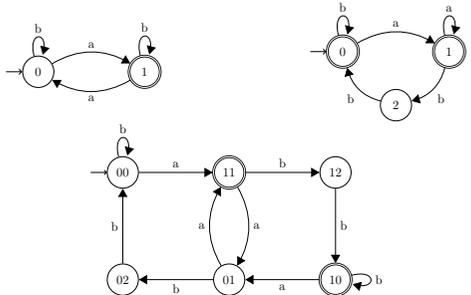
yliès.falcone@univ-grenoble-alpes.fr — www.ylies.fr

Univ. Grenoble-Alpes, Inria
Laboratoire d'Informatique de Grenoble - www.liglab.fr
Équipe de recherche LIG-Inria, COISE - team.inria.fr/corse/

Y. Falcone (UGA - Inria) INF 302 : Langages & Automates 1 / 17

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année

Intuition et objectifs



- Correspondance entre opérations sur les langages et opérations associées sur les automates.

opérations sur automate	opérations sur langage
négation	complémentation
produit	intersection

- Fermeture de l'ensemble des langages à états.

Y. Falcone (UGA - Inria) INF 302 : Langages & Automates 2 / 17

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année

Fermeture de EF par complémentation et intersection

Fermeture de EF par **complémentation**

Soit A un AD.

- Le langage $\Sigma^* \setminus L(A)$ est-il reconnaissable par un AD ?
- 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 ADs.

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

Nous pourrons répondre de manière *affirmative* à toutes ces questions.

Y. Falcone (UGA - Inria) INF 302 : Langages & Automates 3 / 17

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année

Plan Chap. 4 - Opérations sur les automates déterministes et fermeture des langages à états

- Complétion d'un automate
- Négation d'un automate / Complémentation du langage reconnu
- Produit de deux automates / Intersection des langages reconnus
- Résumé

Y. Falcone (UGA - Inria) INF 302 : Langages & Automates 4 / 17

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année

Plan Chap. 4 - Opérations sur les automates déterministes et fermeture des langages à états

- Complétion d'un automate
- Négation d'un automate / Complémentation du langage reconnu
- Produit de deux automates / Intersection des langages reconnus
- Résumé

Y. Falcone (UGA - Inria) INF 302 : Langages & Automates 5 / 17

Univ. Grenoble Alpes, Département Licence Sciences et Technologies, Licence deuxième année

Complétion d'automates

Intuition

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

Effet de la complétion : un AD **complet** qui reconnaît $L(A)$.

Objectifs de la complétion :

- travailler avec des automates complets pour certaines transformations,
- raisonner sur des automates complets est parfois plus simple.

Idée de la complétion :

- Ajouter un *nouvel état* à Q . Cet état est appelé *état puits*.
- Diriger toutes les transitions non définies dans A vers l'état puits.

< Complétion d'un automate

Y. Falcone (UGA - Inria) INF 302 : Langages & Automates 6 / 17

Complétion d'automates

Définition

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

Définition (Complétion d'un automate)

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 :

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

Correction de l'opération de complétion

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

Idée de la preuve

Montrer que A et $C(A)$ acceptent les mêmes mots en considérant les exécutions des automates. Voir exercices.

Plan Chap. 4 - Opérations sur les automates déterministes et fermeture des langages à états

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

Négation d'un automate

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

Définition (Négation d'un AD complet)

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

L'opération de négation est aussi appelée opération de complémentation.

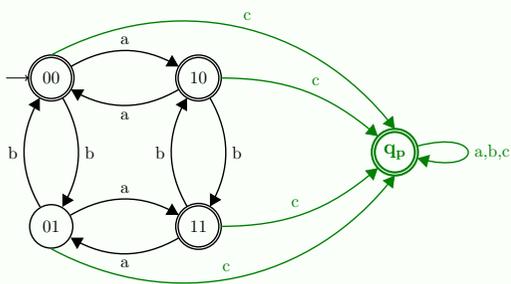
Procédure de complémentation d'un AD (quelconque) A :

- 1 **Construire** $C(A)$ — la version complète de A .
- 2 Inverser les états accepteurs et non-accepteurs dans $C(A)$.

Négation d'un automate : exemple

Exemple (Négation d'un automate)

Sur $\Sigma = \{a, b, c\}$:
un nombre **impair** de a ou un nombre **pair** de b
ou un c



Correction de l'opération de négation et fermeture par complémentation

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

Correction de la procédure de complémentation

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

Idée de la preuve

Montrer qu'un mot accepté par A n'est pas accepté par A^c et vice-versa en utilisant l'exécution de ces mots. Voir exercices.

Corollaire

La classe EF des langages à états est **fermée par complémentation**.

Plan Chap. 4 - Opérations sur les automates déterministes et fermeture des langages à états

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

Produit d'automates

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

Objectifs du produit d'automate :

- 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;
- réaliser cette construction de manière *compositionnelle*.

< Produit de deux automates

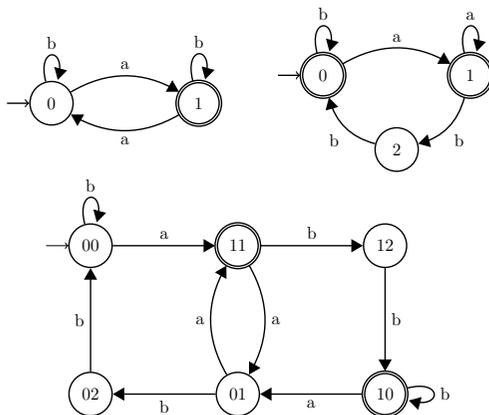
Définition (Produit d'automates)

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

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



Correction de l'opération de produit et fermeture par intersection

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 ADs.

Correction de l'opération de produit

$$L(A \times B) = L(A) \cap L(B).$$

Idée de la preuve

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

- $L(A \times B) \subseteq L(A) \cap L(B)$, cad :
 - $L(A \times B) \subseteq L(A)$,
 - $L(A \times B) \subseteq L(B)$ et
- $L(A) \cap L(B) \subseteq L(A \times B)$

Pour montrer 1), à partir de l'exécution d'un mot accepté par $A \times B$, déduire l'exécution sur A et B .

Pour montrer 2), construire l'exécution sur $A \times B$ d'un mot accepté par A et par B , puis utiliser les critères d'acceptation.

Voir exercices.

Corollaire

La classe EF des langages à états est **fermée par intersection**.

Plan Chap. 4 - Opérations sur les automates déterministes et fermeture des langages à états

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

Résumé du chapitre 4 : opérations sur les automates et fermeture des langages à états

Opérations sur les automates déterministes et fermeture des langages à états

- 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 à états par *complémentation* et *intersection*.

En TD

- Donner les algorithmes pour procédures de complétion et complémentation (après le chapitre 5).
- Définir des procédures permettant de calculer des automate reconnaissant l'union et le ou exclusif des langages d'automates passés en paramètres.
- ...