

Théorie des Langages 1

Cours 3 : ϵ -transitions + déterminisation

L. Rieg

Grenoble INP - Ensimag, 1^{re} année

Année 2023-2024

Notes

Une propriété

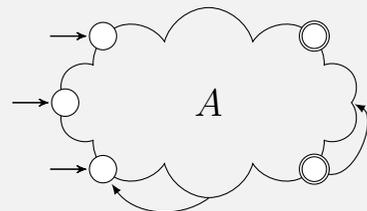
Proposition

Pour tout automate fini A , il existe un automate fini B avec :

- un unique état initial sans transition entrante,
- un unique état final sans transition sortante,

équivalent à A .

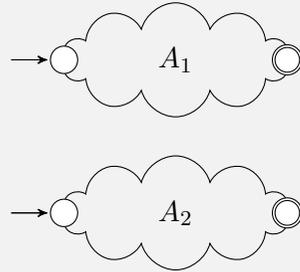
Construction :



Notes

Union

Etant donnés deux automates A_1 et A_2 , construire un automate reconnaissant $\mathcal{L}(A_1) \cup \mathcal{L}(A_2)$.



Formellement :

Si $A_1 = (Q_1, V, \delta_1, I_1, F_1)$ et $A_2 = (Q_2, V, \delta_2, I_2, F_2)$, on a :

$$A_1 \cup A_2 \stackrel{\text{def}}{=}$$

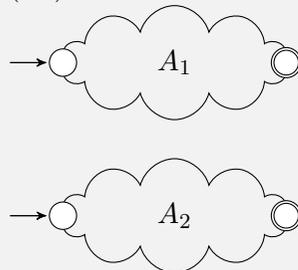
avec

$$\delta \stackrel{\text{def}}{=}$$

Notes

Concaténation

Etant donnés deux automates A_1 et A_2 , construire un automate reconnaissant $\mathcal{L}(A_1).\mathcal{L}(A_2)$.



Exercice : écrire formellement cette transformation

Si $A_1 = (Q_1, V, \delta_1, I_1, F_1)$ et $A_2 = (Q_2, V, \delta_2, I_2, F_2)$, on a :

Notes

Résumé

Théorème

La classe des langages réguliers est fermée :

- par union
- par concaténation
- par concaténation itérée

Nous en verrons d'autres au cours 7.

Question

Peut-on toujours effectuer les transformations

?

Dans la suite : **techniques de transformation**, preuves plus tard

Notes

Algorithme de suppression des ε -transitions

Deux étapes :

1. Déterminer les états qu'on peut atteindre en ne se servant que d' ε -transitions
2. Se servir de cette information pour construire un automate équivalent sans ε -transition

Etape 1

Définition (États accessibles)

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit par induction pour tout $p \in Q$ l'ensemble $\text{Acc}_\varepsilon(p)$ des états accessibles depuis p par ε -transitions :

Calcul de $\text{Acc}_\varepsilon(p)$ par

(cf cours 1)

Notes

Rappel sur le calcul par itération pour $\text{Acc}_\varepsilon(p)$

Idée : Calculer les états de $\text{Acc}_\varepsilon(p)$ accessibles en au plus n pas et faire croître n .

$\text{Acc}_\varepsilon(p) = \bigcup_{n \geq 0} A_n$, où la suite (A_n) est définie par :

algorithme $\text{Acc}_\varepsilon(p) =$
 $n \leftarrow 0, A_0 \leftarrow \{p\}$
répéter
 $A_{n+1} \leftarrow A_n \cup \{\kappa_i(e_1, \dots, e_{k_i}) \mid \kappa_i \in K, e_1, \dots, e_{k_i} \in A_n\}$
 $n \leftarrow n + 1$
jusqu'à $A_{n+1} = A_n$
renvoyer A_n

Question : Est-ce que ça termine toujours ?

Au besoin, on fait un tableau des A_n jusqu'à stabiliser.

Notes

Algorithme de suppression des ε -transitions (suite)

Etape 2.

Définition

Etant donné un automate $A = \langle Q, V, \delta, I, F \rangle$, on définit l'automate $B = \langle Q, V, \delta', I, F' \rangle$ de la façon suivante :

Propositions

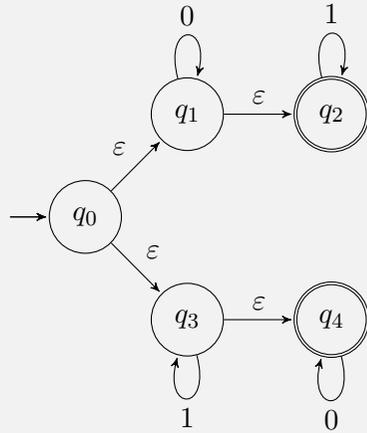
- B est un automate sans ε -transition.
- B est équivalent à A .

démo plus tard

Notes

Exercice

Construire B pour l'automate A suivant :



Notes

Exercice (solution)

Notes

Rappels sur les AFD complets

Définition (Automate déterministe complet)

Un AF $\langle Q, V, \delta, I, F \rangle$ est dit **déterministe complet** si

1. $\text{Card}(I) = 1$ (exactement un état initial)
2. $\nexists (q, \varepsilon, p) \in \delta$
3. $\forall (q, a) \in Q \times V, \exists! p \in Q, (q, a, p) \in \delta$.

Conséquences

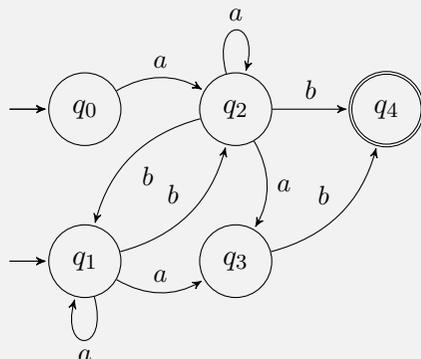
-
-
-
-

Notes

Principe de la déterminisation

Idée : suivre tous les chemins en parallèle

Exemple : L'automate suivant reconnaît-il *aab* ?



Notes

Définition

Définition (Automate des parties)

Etant donné un automate $A = \langle Q, V, \delta_A, I, F_A \rangle$ sans ϵ -transition, on construit l'automate $B = \langle \mathcal{P}(Q), V, \delta_B, \{I\}, F_B \rangle$, où :

- δ_B est défini par
$$\forall P \subseteq Q, \forall a \in V, \delta_B(P, a) = \{q \in Q \mid \exists p \in P : (p, a, q) \in \delta_A\}$$
- $F_B = \{P \subseteq Q \mid P \cap F_A \neq \emptyset\}$

Remarques

- $P \subseteq Q \iff P \in \mathcal{P}(Q)$ et $\emptyset \subseteq Q$: un état puits de B
- Certains $P \subseteq Q$ peuvent ne pas être accessibles depuis I donc on construit B de proche en proche à partir de I .

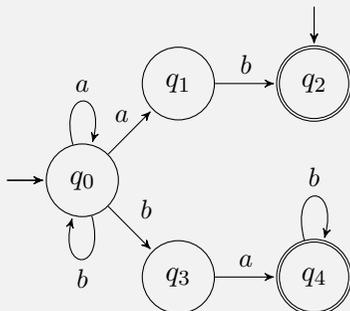
Proposition

L'automate B est un automate fini **déterministe complet** équivalent à A .

Notes

Exemple

Construire un AFD (complet) équivalent à :



7 états accessibles (sur 32 potentiels)
5 états acceptants accessibles (sur 24 potentiels)

Notes
