Théorie des Langages
Épisode 2 — Automates finis

Thomas Pietrzak

Université Paul Verlaine — Metz

Reconnaisseur

Une tâche principale dans l'analyse lexicale et syntaxique est de décider soit pour une unité lexicale soit pour un programme si cette entrée est « un mot du langage » décrit par soit une expression régulière r soit une grammaire hors contexte G.
Il faut alors un reconnaisseur pour un langage L. C'est un programme qui prend en entrée un mot m et répond « oui » si $m \in L$ et « non » autrement.

Automate fini

On compile une expression régulière en un reconnaîsseur en construisant un diagramme de transition généralisé. Le modèle de machine correspondant est appelé automate fini.
Les automates finis, aussi bien déterministes que non déterministes sont capables de reconnaître précisément les langages réguliers.

Automate fini Déterministe

Un automate fini déterministe (AFD) est défini par $A = (E, \Sigma, \delta, e_0, F)$ où

- E est un ensemble fini d'états
- Σ est un ensemble fini de symboles d'entrée
- $\delta : E \times \Sigma \rightarrow E$ est une fonction de transition
- e_0 est un état de départ
- $F \subseteq E$ est l'ensemble des états d'acceptation

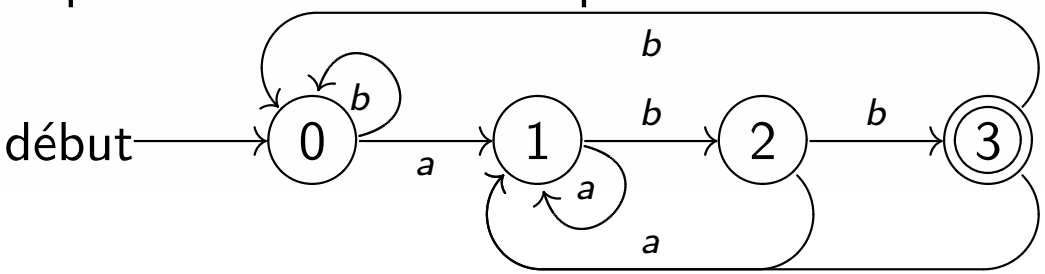
Exemple

Voici un automate pour l'expression régulière $(a|b)^*abb$

$A = (\{e_0, e_1, e_2, e_3\}, \{a, b\}, \delta, e_0, \{e_3\})$ où δ est défini par :

État	Symbole d'entrée	
	a	b
0	1	0
1	1	2
2	1	3
3	1	0

Représentation schématique :



Exemples de chaînes reconnues : *abb*, *aaaaabb*, *abbabb*, *abaababb*

Configuration

Une configuration d'un automate est une description complète de la situation de l'automate. On note (m, e) avec m le mot qui reste sur la bande et e l'état courant.
(m, e_0) est une configuration de départ, avec m l'entrée.
(ϵ, e), avec $e \in F$ est une configuration finale.
$(au, e) \mapsto_A (u, e') \Leftrightarrow \delta(e, a) = e'$ où $a \in \Sigma$, $u \in \Sigma^*$ et $e, e' \in E$.

Configuration suivant

Une configuration K' d'un automate A est une configuration suivant de K, noté $K \mapsto_A^* K'$, s'il y a une suite $K = K_0, K_1, \ldots, K_r = K'$ telle que $K_i \mapsto_A K_{i+1}$ pou chaque $i \in \{0, 1, \ldots, r-2\}$.

Acceptation

Un automate fini lit l'entrée une seule fois en passant la tête de lecture de gauche à droite sans se retourner. L'automate accepte un mot s'il arrive dans un état d'acceptation après avoir lu le mot au total.

Définition

Le langage L est accepté par un AFD A si L est défini par
$L = L(A) = \{m \in \Sigma^* | (m, e_0) \mapsto_A^* (\epsilon, e), e \in F\}$

Exemple

$A = (\{e_0, e_1\}, \{a, b\}, \delta, e_0, \{e_0\})$

δ	a	b
e_0	e_0	e_1
e_1	e_1	e_1

$L(A) = \{a^n | n \in \mathbb{N}\}$

Automate fini Non Déterministe

Un automate fini non déterministe (AFN) est défini par $A = (E, \Sigma, \delta, e_0, F)$ où

- E est un ensemble fini d'états
- Σ est un ensemble fini de symboles d'entrée
- $\delta \subseteq (E \times \Sigma) \times E$ est une relation de transition
- e_0 est un état de départ
- $F \subseteq E$ est l'ensemble des états d'acceptation

Au contraire d'un AFD, une transition d'un AFN $\delta(e, a)$ peut avoir plusieurs valeurs : $\delta(e, a) \subseteq Z$.
En conséquence, pour un AFN A une configuration suivant n'est plus uniquement déterminée : $(au, e) \mapsto_A (u, e') \Leftrightarrow e' \in \delta(e, a)$ où $a \in \Sigma$, $u \in \Sigma^*$ et $e, e' \in E$.

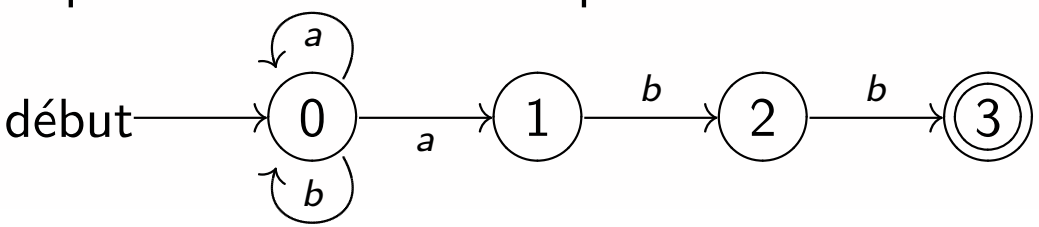
Exemple

En reprenant l'expression régulière $(a|b)^*abb$

$A = (\{e_0, e_1, e_2, e_3\}, \{a, b\}, \delta, e_0, \{e_3\})$ où δ est défini par :

État	Symbole d'entrée	
	a	b
0	$\{0, 1\}$	$\{0\}$
1	—	$\{2\}$
2	—	$\{3\}$

Représentation schématique :



Exemples de chaînes reconnues : *abb*, *aaaaabb*, *abbabb*, *abaababb*

Acceptation

Le langage L est accepté par un AFN A si L est défini par
$L = L(A) = \{m \in \Sigma^* | (m, e_0) \mapsto_A^* (\epsilon, e), e \in F\}$

On voit facilement que par définition chaque AFD est aussi un AFN.
D'autre part, généralement un AFN n'est pas toujours un AFD.

Graphe de transition

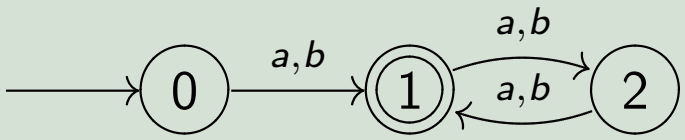
La fonction de transition d'un AFD ou la relation de transition d'un AFN peuvent être représentées par un graphe orienté appelé graphe de transition.
Les noeuds du graphe sont les états de l'automate et les arcs sont étiquetés comme suit

- il y a un arc (e, e') étiqueté $a \in \Sigma$ si et seulement si $\delta(e, a) = e'$ dans un AFD ou $e' \in \delta(e, a)$ dans un AFN.
- il y a des arcs multiples (e, e') dans le graphe de transition si $e' \in \delta(e, a)$ pour plusieurs caractères $a \in \Sigma$.

Exemple

- $\delta(e_0, a) = e_1$
- $\delta(e_0, b) = e_1$
- $\delta(e_1, a) = e_2$
- $\delta(e_1, b) = e_2$
- $\delta(e_2, a) = e_1$
- $\delta(e_2, b) = e_1$

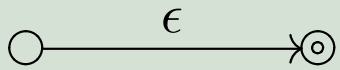
$F = \{e_1\}$



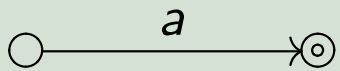
Construire un AFN à partir d'une expression régulière

Pour toute expression régulière r on peut construire un AFN qui reconnaît le même langage. On construit l'automate par induction sur la syntaxe :

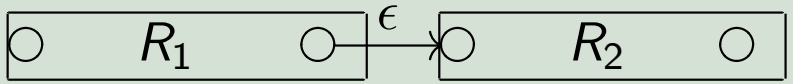
ϵ



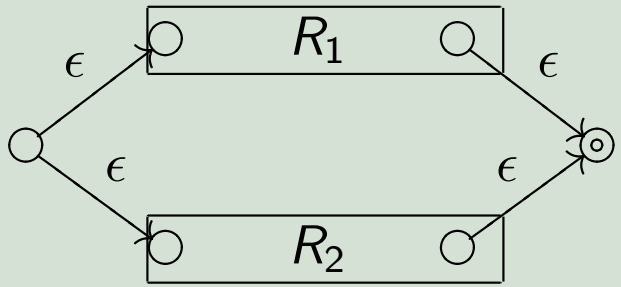
a



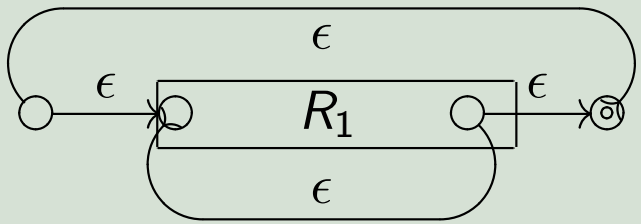
$R_1 R_2$



$R_1 \mid R_2$

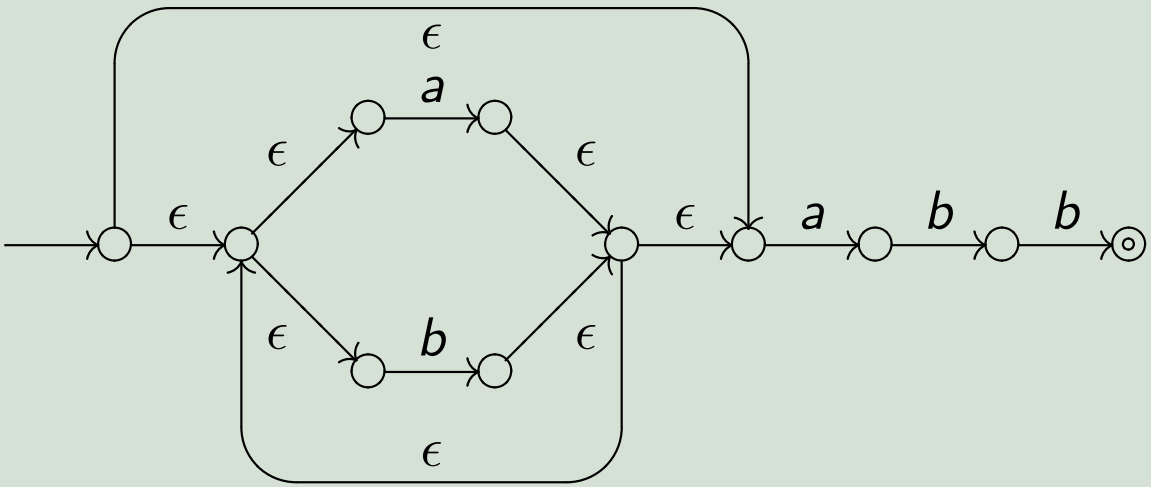


R_1^*



Exemple

L'expression régulière $(a \mid b)^* abb$ peut s'écrire sous la forme d'un AFN :



Théorème de Rabin et Scott

Pour tout AFN A il existe un AFD A' tel que $L(A) = L(A')$.

Démonstration

Soit $A = (E, \Sigma, \delta, e_0, F)$ un AFN. On construit un AFD $A' = (E', \Sigma, \delta', e_0', F')$ comme suit :

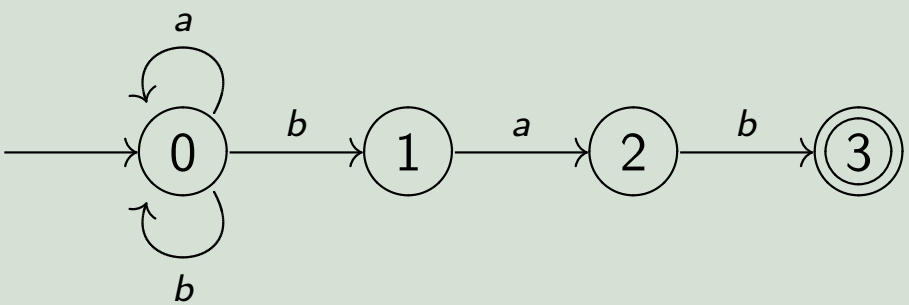
- $E' := \mathcal{P}(E)$
- $e_0' := \{e_0\}$
- $F' := \{X \subseteq E \mid X \cap F \neq \emptyset\}$
- Pour tout $X \subseteq E$ et tout $a \in \Sigma$: $\delta'(X, a) := \bigcup_{x \in X} \delta(x, a)$

Pour démontrer que la construction est correcte il faut montrer que $L(A) = L(A')$.

Algorithme

- Initialisation : $E' := \{\{e_0\}\}$, $T := \{\{e_0\}\}$
- Tant que $T \neq \emptyset$
  - Choisir un élément $S \in T$
  - Pour tout symbole $a \in \Sigma$
    - Calculer l'état $S' = \bigcup_{e \in S} \delta(e, a)$
    - Si S' c'est pas déjà dans E', l'ajouter dans T et dans E'
    - Ajouter un arc sur l'automate entre S et S' et le valuer par a
  - $T := T \setminus S$

Exemple

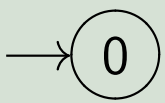


Exemple

$E' = \{\{e_0\}\}$ $T = \{\{e_0\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	—	—

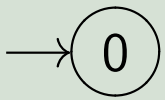


Exemple

$E' = \{\{e_0\}\}$ $T = \{\{e_0\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	—	—

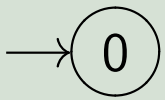


Exemple

$E' = \{\{e_0\}\}$ $T = \{\{e_0\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	—	—

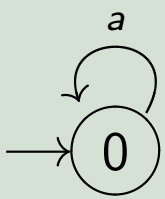


Exemple

$E' = \{\{e_0\}\}$ $T = \{\{e_0\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	—

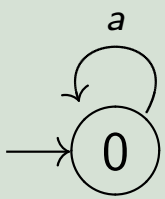


Exemple

$E' = \{\{e_0\}\}$ $T = \{\{e_0\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	—

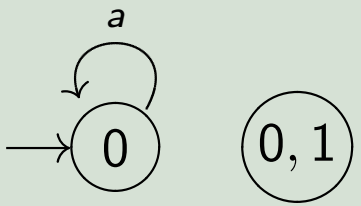


Exemple

$E = \{\{e_0\}, \{e_0, e_1\}\}$ $T = \{\{e_0\}, \{e_0, e_1\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	—
$\{e_0, e_1\}$	—	—



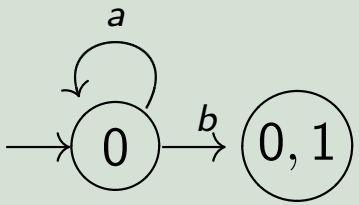
Exemple

$E = \{\{e_0\}, \{e_0, e_1\}\}$

$T = \{\{e_0\}, \{e_0, e_1\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	—	—

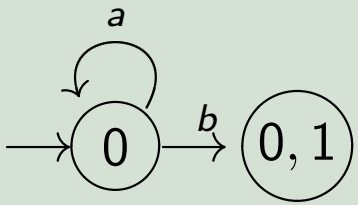


Exemple

$E = \{\{e_0\}, \{e_0, e_1\}\}$ $\quad T = \{\{e_0, e_1\}\}$

- $S := \{e_0\}$
- $S' := \delta(e_0, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) = \{e_0, e_1\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	—	—

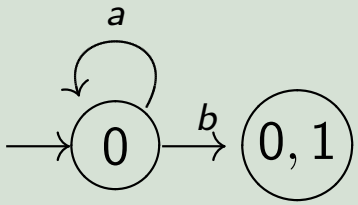


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}\}$ $T = \{\{e_0, e_1\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	—	—

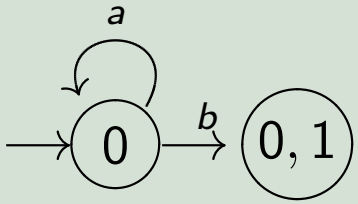


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}\}$ $T = \{\{e_0, e_1\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	—	—

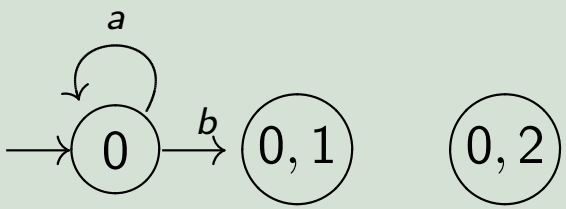


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_1\}, \{e_0, e_2\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	—	—
$\{e_0, e_2\}$	—	—

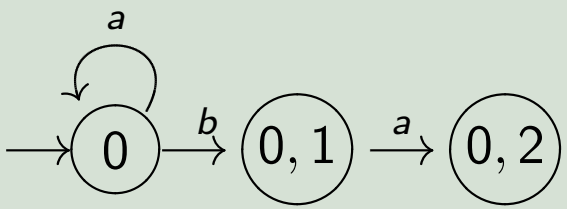


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_1\}, \{e_0, e_2\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	—
$\{e_0, e_2\}$	—	—

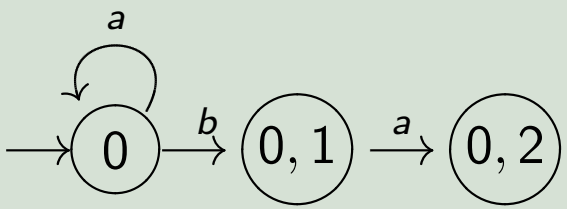


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_1\}, \{e_0, e_2\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	—
$\{e_0, e_2\}$	—	—



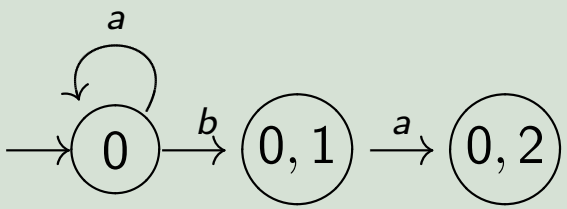
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$

$T = \{\{e_0, e_1\}, \{e_0, e_2\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	—
$\{e_0, e_2\}$	—	—

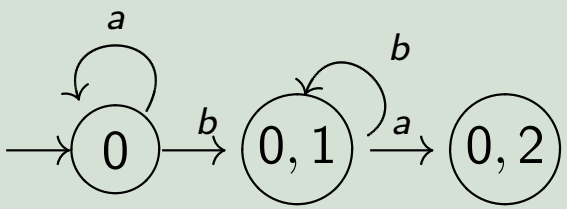


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_1\}, \{e_0, e_2\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	—	—

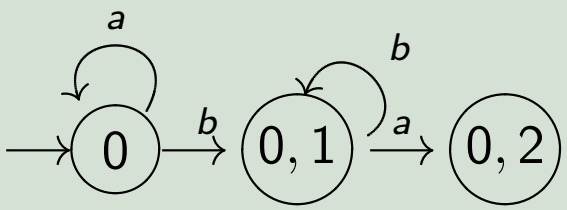


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $\quad T = \{\{e_0, e_2\}\}$

- $S := \{e_0, e_1\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) = \{e_0, e_2\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_1\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1\}, b) = \{e_0, e_1\}$
- On supprime $\{e_0, e_1\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	—	—

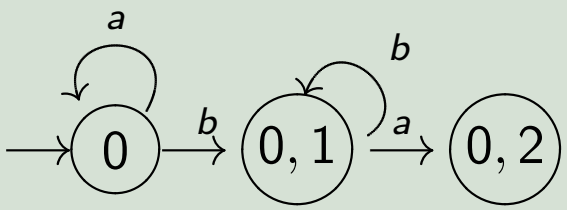


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_2\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	—	—

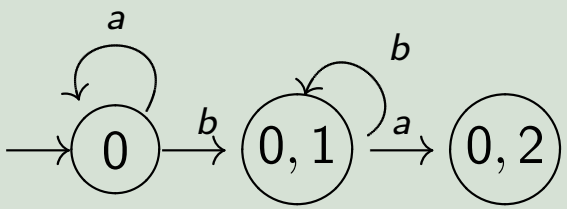


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_2\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	—	—

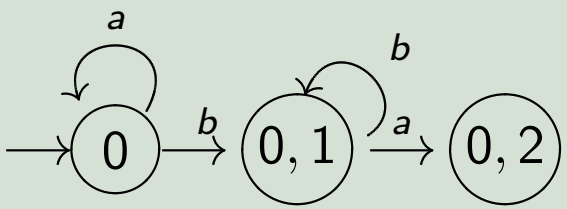


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_2\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	—	—

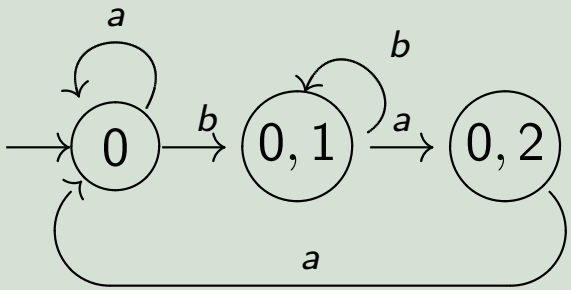


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_2\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	—

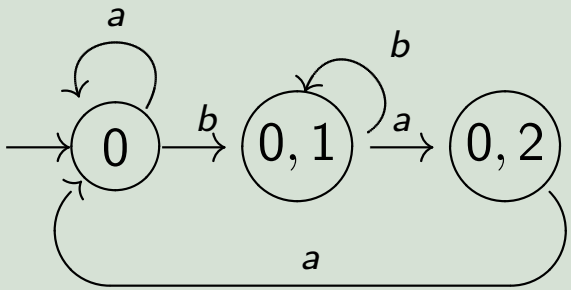


Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}\}$ $T = \{\{e_0, e_2\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	—



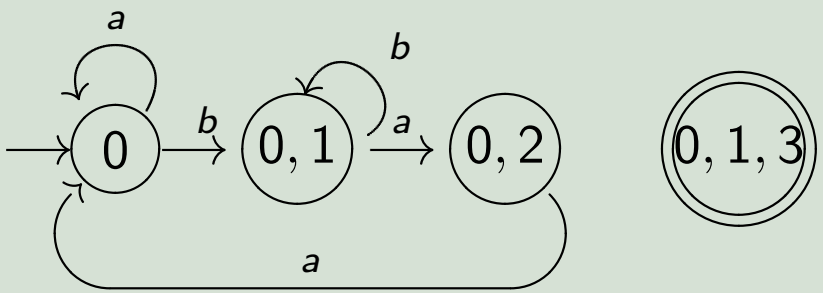
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	—
$\{e_0, e_1, e_3\}$	—	—



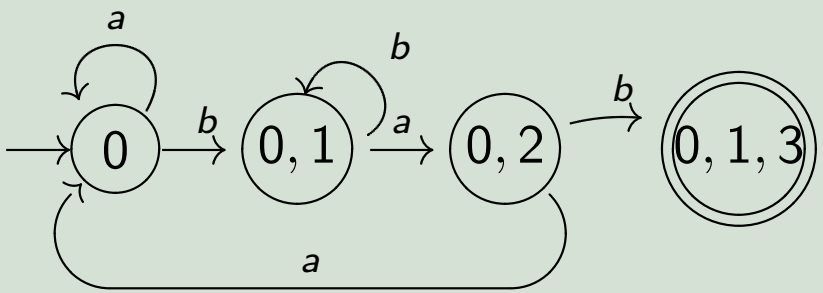
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	—	—



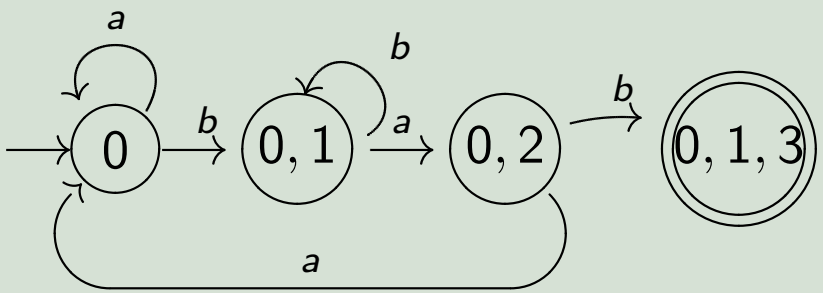
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_2\}$
- $S' := \delta(e_0, a) \cup \delta(e_2, a) = \{e_0\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, a) = \{e_0\}$
- $S' := \delta(e_0, b) \cup \delta(e_2, b) = \{e_0, e_1, e_3\}$
- S' n'est pas encore dans E' donc on l'ajoute
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	—	—



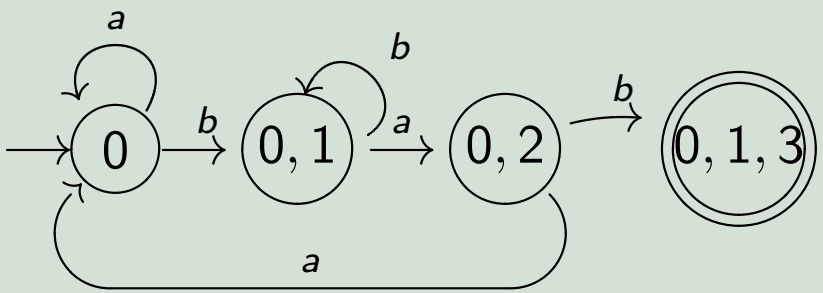
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	—	—



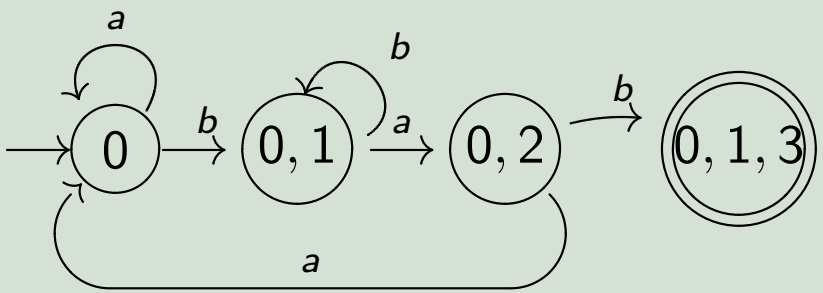
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	—	—



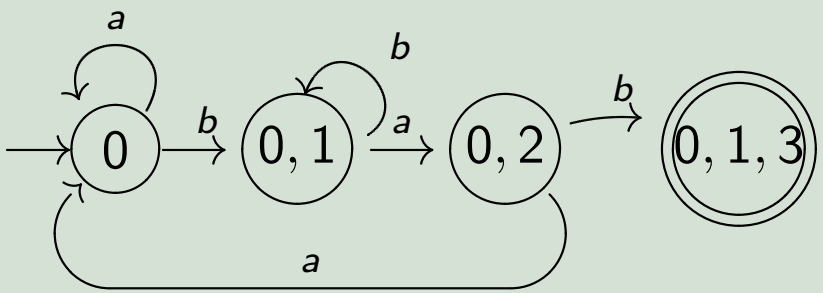
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	—	—



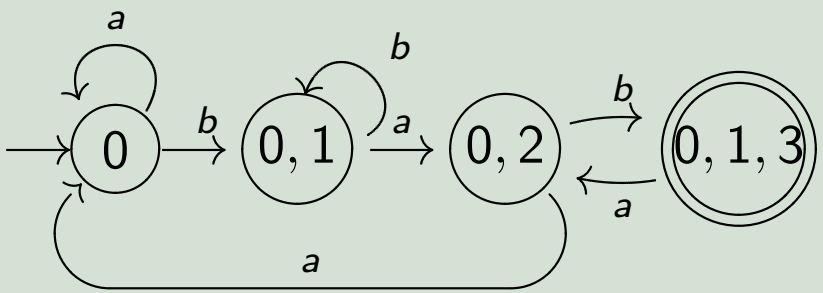
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	$\{e_0, e_2\}$	—



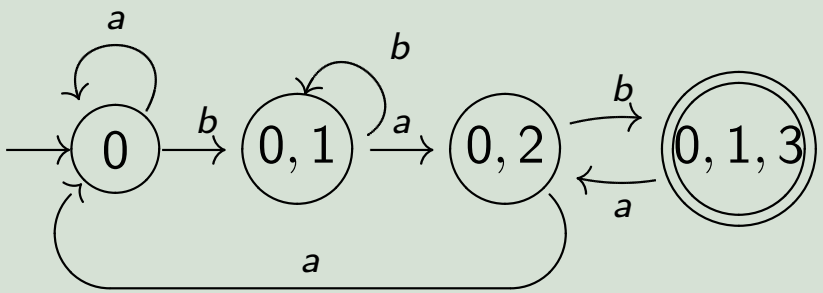
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	$\{e_0, e_2\}$	—



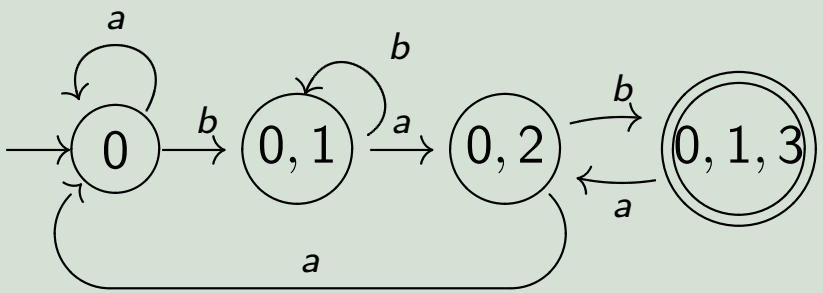
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	$\{e_0, e_2\}$	—



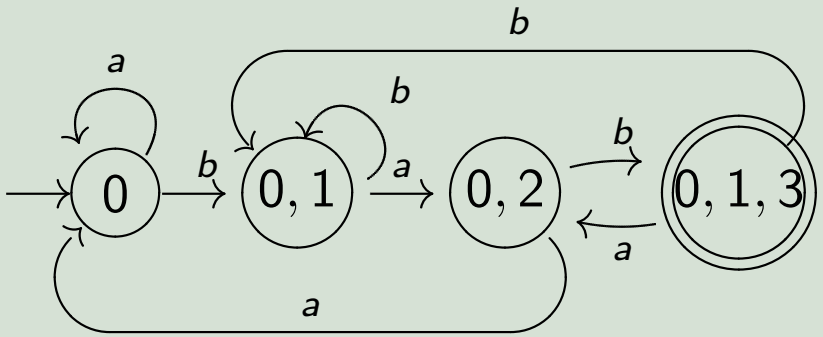
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\{e_0, e_1, e_3\}\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$



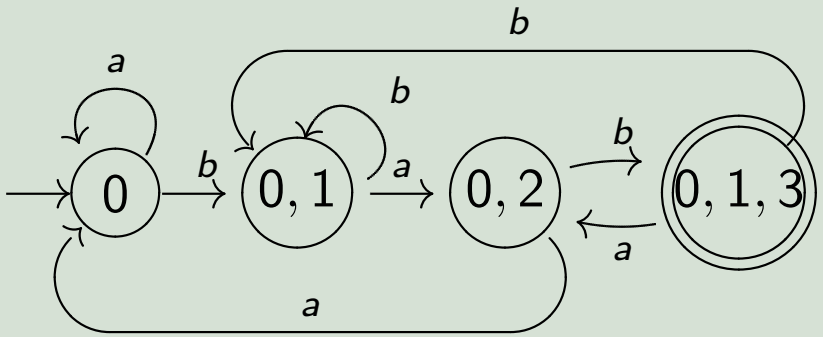
Exemple

$E' = \{\{e_0\}, \{e_0, e_1\}, \{e_0, e_2\}, \{e_0, e_1, e_3\}\}$

$T = \{\}$

- $S := \{e_0, e_1, e_3\}$
- $S' := \delta(e_0, a) \cup \delta(e_1, a) \cup \delta(e_3, a) = \{e_0, e_2\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_1, e_3\}, a) = \{e_0, e_2\}$
- $S' := \delta(e_0, b) \cup \delta(e_1, b) \cup \delta(e_3, b) = \{e_0, e_1\}$
- S' est déjà dans E' donc on ne l'ajoute pas
- $\delta'(\{e_0, e_2\}, b) = \{e_0, e_1, e_3\}$
- On supprime $\{e_0, e_2\}$ de T

e	a	b
$\{e_0\}$	$\{e_0\}$	$\{e_0, e_1\}$
$\{e_0, e_1\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$
$\{e_0, e_2\}$	$\{e_0\}$	$\{e_0, e_1, e_3\}$
$\{e_0, e_1, e_3\}$	$\{e_0, e_2\}$	$\{e_0, e_1\}$



Suppression des ϵ-transitions

Si on génère l'AFN avec l'algorithme de la section précédente, il faut enlever les ϵ-transitions lors de la création d'un AFD. Pour cela on définit quelques fonctions.

Définitions

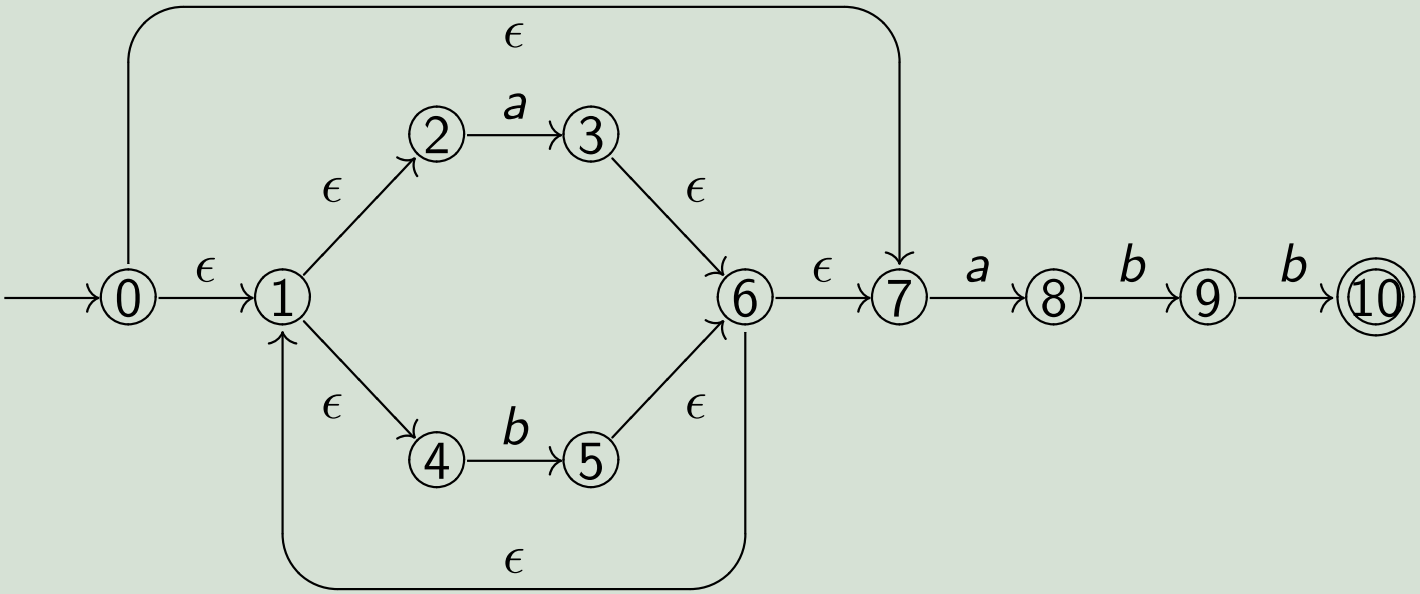
- $\epsilon - fermeture(e)$: ensemble des états de l'AFN accessibles depuis l'état e de l'AFN par des ϵ-transitions uniquement.
- $\epsilon - fermeture(T)$: ensemble des états de l'AFN accessibles depuis un état e appartenant à T par des ϵ-transitions uniquement.
- $Transiter(T, a)$: Ensemble des états de l'AFN vers lesquels il existe une transition sur le symbole d'entrée a à partir d'un état e appartenant à T.

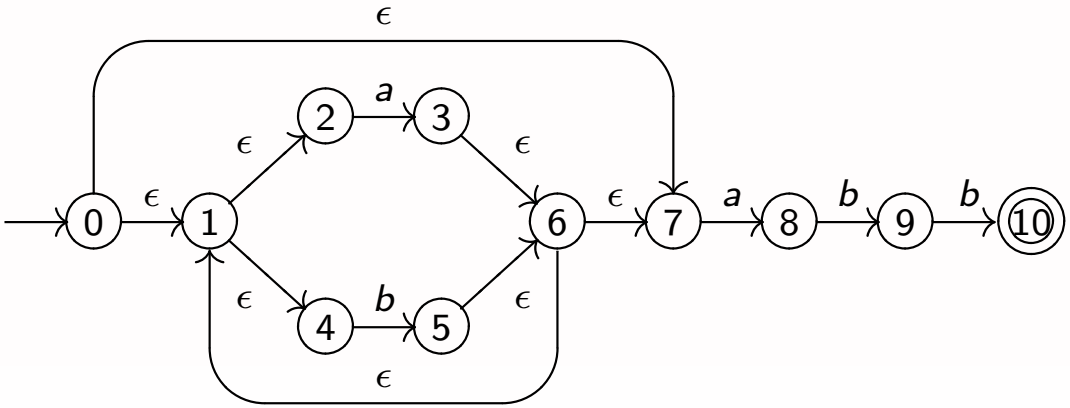
Algorithme

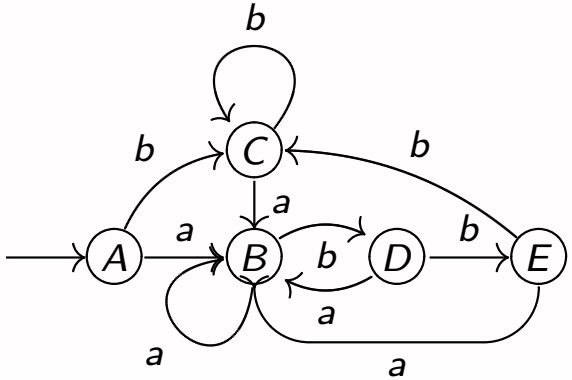
Il suffit de remplacer dans l'algorithme précédent les $\bigcup_{e \in S} \delta(e, a)$ par $\epsilon - fermeture(Transiter(S, a))$ et initialiser E' et T par $\epsilon - fermeture(e_0)$.

Exemple

Prenons l'exemple précédent : $(a \mid b)^* abb$ et l'AFN qu'on en a déduit :







Exemple

$A = \epsilon - fermeture(e_0) = \{0, 1, 2, 4, 7\}$
$\epsilon - f(Transiter(A, a)) = \epsilon - f(\{3, 8\}) = \{1, 2, 3, 4, 6, 7, 8\} = B$
$\epsilon - f(Transiter(A, b)) = \epsilon - f(\{5\}) = \{1, 2, 4, 5, 6, 7\} = C$
$\epsilon - f(Transiter(B, a)) = \epsilon - f(\{3, 8\}) = B$
$\epsilon - f(Transiter(B, b)) = \epsilon - f(\{5, 9\}) = \{1, 2, 4, 5, 6, 7, 9\} = D$
$\epsilon - f(Transiter(C, a)) = \epsilon - f(\{3, 8\}) = B\ \epsilon - f(Transiter(C, b)) = \epsilon - f(\{5\}) = C$
$\epsilon - f(Transiter(D, a)) = \epsilon - f(\{3, 8\}) = B$
$\epsilon - f(Transiter(D, b)) = \epsilon - f(\{5, 10\}) = \{1, 2, 4, 5, 6, 7, 10\} = E$
$\epsilon - f(Transiter(E, a)) = \epsilon - f(\{3, 8\}) = B\ \epsilon - f(Transiter(E, b)) = \epsilon - f(\{5\}) = C$

Description des langages réguliers

Théorème : Soit Σ un alphabet et L un langage sur Σ. Alors les propriétés suivantes sont équivalentes :

- L est un langage régulier, c'est à dire qu'il y a une grammaire régulière G telle que $L = L(G)$.
- Il existe une expression régulière r dénotant L.
- Il existe un AFN A tel que $L = L(A)$.
- Il existe un AFD A' tel que $L = L(A')$.

Les démonstrations d'équivalence sont basées sur des transformations constructives.

Théorème de Myhill-Nerode

Le théorème de Myhill-Nerode donne une condition nécessaire et suffisante pour qu'un langage soit régulier.
Typiquement on utilise ce théorème pour vérifier si un langage est régulier ou non.

Définition

Soit $L \subseteq \Sigma^*$ un langage. La relation $\mathcal{R}_L$ sur Σ^* est définie par $x\mathcal{R}_L y$ si pour tout $w \in \Sigma^*$, soit $xw \in L$ et $yw \in L$, soit $xw \notin L$ et $yw \notin L$.
On peut montrer que $\mathcal{R}_L$ est une relation d'équivalence sur Σ^*.
Le nombre de classes d'équivalences d'une relation $\mathcal{R}$ est appelée indice de $\mathcal{R}$.

Théorème

Un langage $L \in \Sigma^*$ est régulier si et seulement si l'indice de $\mathcal{R}_L$ est fini.

Théorème

Un résultat théorique important est une conséquence du théorème Myhill-Nerode.

Corollaire : chaque langage régulier est reconnu par un AFD à un nombre d'états minimal, qui est unique à un renommage près des états.

On appelle cet AFD l'automate minimal de L. Le nombre d'états de l'automate minimal est égal à l'indice de $\mathcal{R}_L$. En effet, si les classes d'équivalence de L sont connues, on peut construire l'automate minimal de L en utilisant les classes de $\mathcal{R}_L$ comme ensemble des états.

Le meilleur algorithme connu s'exécute en temps $\mathcal{O}(n \log n)$ où n est le nombre d'états de l'automate A.

Pour éviter des problèmes techniques nous supposons que dans un AFD $A = (E, \Sigma, \delta, e_0, F)$ quelconque, chaque état e a une transition pour symbole de Σ.

Si δ n'est pas défini pour chaque couple $(e, a) \in E \times \Sigma$, alors nous pourrions introduire un nouvel « état mort » $E_{mort} \notin F$ avec $\delta(e_{mort}, a) = e_{mort}$ pour chaque $a \in \Sigma$ et $\delta(e, a)$ n'est pas défini initialement.

La fonction généralisée de transition d'un AFD $A = (E, \Sigma, \delta, e_0, F)$, notée $\hat{\delta}$, est définie par $\hat{\delta}(m, e) = e' \Leftrightarrow (m, e) \mapsto_A^* (\epsilon, e')$, où $e, e' \in E$ et $m \in \Sigma^*$.

C'est à dire $\hat{\delta}(m, e) = e'$ si et seulement si partant de l'AFD A dans l'état e et le faisant fonctionner avec le mot d'entrée m on termine dans l'état e'.

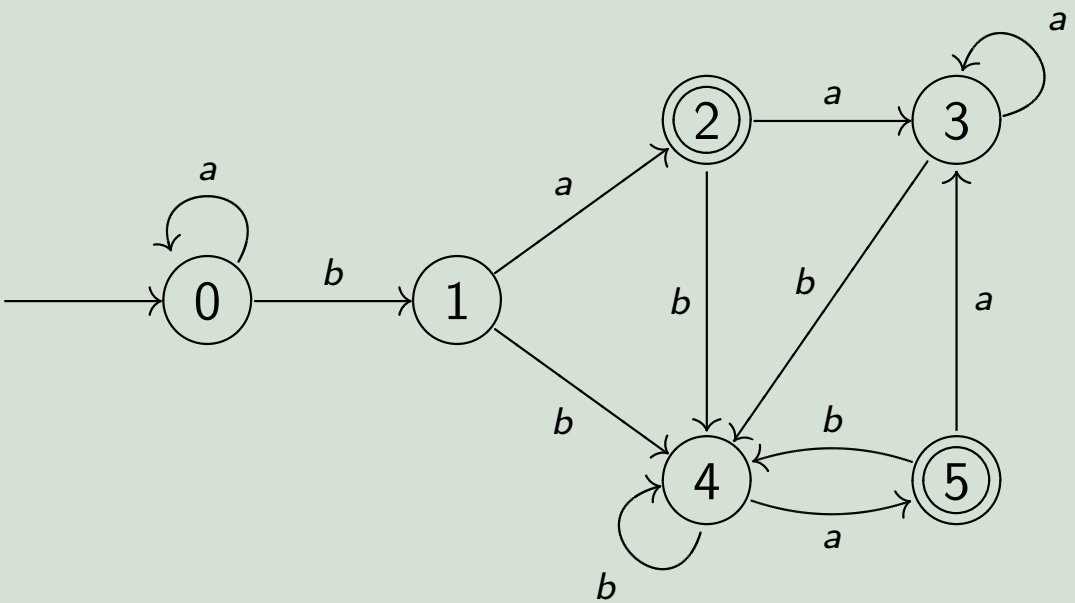
L'état e' est accessible depuis l'état de départ e_0.

On dit qu'un mot $m \in \Sigma^*$ distingue l'état e et l'état e' de l'AFD A si $\hat{\delta}(m, e) \in F \Leftrightarrow \hat{\delta}(m, e') \notin F$

Étape 1

Construire une partition initiale π de l'ensemble des états avec deux groupes : les états d'acceptation F et les états de non-acceptation $E \setminus F$.

Exemple



$\pi = \{2,5\},\{0,1,3,4\}$

Étape 2

Appliquer la procédure ci-dessous afin de construire une nouvelle partition π_n.
Pour chaque groupe G de π faire :

- Partitionner G en sous-groupes de manière que deux états e et f de G soient dans le même sous-groupe si et seulement si pour tout symbole $a \in \Sigma$, les états e et f ont des transitions sur a vers des états du même groupe de π. Au pire un état formera un groupe à lui seul.
- remplacer G dans π_n par tous les sous-groupes ainsi formés.

Étape 3

Si $\pi_n = \pi$, alors soit $\pi_f = \pi$ et continuer à l'étape 4, sinon répéter l'étape 2 avec $\pi = \pi_n$.

Exemple

Notre partition était : $\pi = \{2,5\},\{0,1,3,4\}$.
Soit $G_1 = \{2,5\}$ et $G_2 = \{0,1,3,4\}$.

symbole	2	5	0	1	3	4
a	G_2	G_2	G_2	G_1	G_2	G_1
b	G_2	G_2	G_2	G_2	G_2	G_2

- Les transitions de tous les états de G_1 vont vers les mêmes groupes, donc G_1 n'est pas divisé en sous groupes.
- Par contre dans G_2 il y a des différences entre $\{0,3\}$ et $\{1,4\}$. On obtient donc une nouvelle partition : $\pi_n = \{2,5\},\{0,3\},\{1,4\}$.
- On recommence l'étape 2 sur cette partition.

Exemple

Notre nouvelle partition est : $\pi_n = \{2,5\},\{0,3\},\{1,4\}$.
Soit $G'_1 = \{2,5\}$, $G'_2 = \{0,3\}$ et $G'_3 = \{1,4\}$.

symbole	2	5	0	3	1	4
a	G'_2	G'_2	G'_2	G'_2	G'_1	G'_1
b	G'_3	G'_3	G'_3	G'_3	G'_3	G'_3

- Les transitions de tous les états de tous les groupes vont vers les mêmes groupes, donc ils ne sont pas divisés en sous groupes.
- On peut passer à l'étape 4 avec $\pi_f = \{2,5\},\{0,3\},\{1,4\}$.

Étape 4

Choisir un état dans chaque groupe de la partition π_f en tant que représentant de ce groupe. Les représentants seront les états de l'AFD réduit A'.

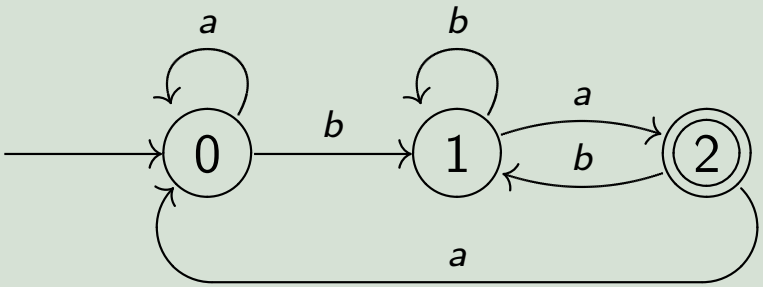
- Soit e un état représentatif, et supposons que dans l'automate original A il y ait une transition de e vers f sur a. Soit r le représentant du groupe de f (r peut être égal à f). Alors A a une transition de e vers r sur a.
- L'état de départ de A' est le représentant du groupe qui contient l'état de départ e_0 de A.
- Les états d'acceptation de A' sont les représentants des états de F. Notons que, pour tout groupe G de π_f, soit G est entièrement composé d'états de F, soit G n'en contient aucun.

Exemple

La partition finale était : $\pi_f = \{2,5\},\{0,3\},\{1,4\}$.

- Prenons 2, 0 et 1 comme représentants.
- 2 est un état final.
- 0 est l'état de départ.

On obtient l'AFD minimal :



Étape 5

Si A' a un état mort, c'est à dire un état e qui n'est pas un état d'acceptation et qui a des transitions vers lui même sur tous les symboles de Σ, alors supprimer e de A'. Supprimer aussi tous les états non accessibles depuis l'état de départ. Toutes les transitions vers e depuisd'autres états deviennent indéfinis.

Validité de l'algorithme

La validité de l'algorithme n'est pas évidente. La démonstration se fait par récurrence sur le nombre d'étapes.

Lemme de pompage

Le lemme de pompage est un outil pour montrer qu'un langage n'est pas régulier. Un langage non régulier ne peut être ni spécifié par une expression régulière, ni pas un AFN ou un AFD.

Lemme

Soit $L \subseteq \Sigma^*$ un langage régulier. Alors il existe une constante $n \in \mathbb{N}$ telle que pour tout $m \in L$ de longueur au moins n, il existe $u, v, w \in \Sigma^*$ satisfaisant :

- $m = uvw$
- $|uv| \leq n$
- $|v| \geq 1$
- $uv^i w \in L$ pour chaque entier $i \geq 0$

Utilisation

Utilisé pour montrer qu'un langage n'est pas régulier.

Démonstration

Soit L un langage régulier, A un AFD tel que $L = L(A)$, et n le nombre d'états de A.

- Prenons un mot $m \in L$ tel que $|m| \geq n$.
- Il existe donc au moins un état de A par lequel on passe plusieurs fois lors de la reconnaissance de m.
- Soit q le premier état par lequel on passe pour la deuxième fois.
- Soit u le sous-mot tel que $(u\gamma, e_0) \mapsto_A (\gamma, q)$
- Soit v et w les deux sous-mots tels que $(vw, q) \mapsto_A (w, q)$ avec $|v| \geq 1$ (possible car on passe plusieurs fois par q).
- Comme q est le seul état visité plus d'une fois après la reconnaissance de uv, on en déduit que $|uv| \leq n$.
- On a $(u\gamma, e_0) \mapsto_A (\gamma, q)$ et $(vw, q) \mapsto_A (w, q)$, donc par transitivité on peut avoir $(v^i w, q) \mapsto_A (w, q)$ et $(uv^i w, q) \mapsto_A (w, q)$ $\forall i \in \mathbb{N}$, donc $uv^i w \in L$ $\forall i \in \mathbb{N}$.

Lemme de pompage

On va montrer que le langage $L = \{0^k 1^k | k \in \mathbb{N}\}$ n'est pas régulier. Supposons le langage régulier et aboutissons à une contradiction. Pour cela on va construire un mot en utilisant le Lemme et montrer qu'il n'est pas dans le langage.

- En utilisant le fameux entier n du lemme, on utilise le mot $0^n 1^n$ qui appartient bien à L.
- D'après la première propriété, $uvw = 0^n 1^n$.
- D'après la seconde propriété $|uv| \leq n$, donc $uv = 0^m$, avec $m \leq n$.
- D'après la troisième propriété $|v| \geq 1$. Donc $v = 0^{|v|}$ et $|v| \geq 1$.
- D'après la quatrième propriété $uv^2w \in L$.
- Or $uv^2w = 0^p 0^n$ avec $p > q$ car le second v a ajouté des 0, mais pas de 1. Et ce mot ne peut pas appartenir à L par la définition de L.
- L n'est donc pas régulier.

Ensembles non réguliers

On ne peut pas décrire certains langages par des expressions régulières.

On ne peut pas utiliser les expressions régulières pour décrire des constructions équilibrées ou imbriquées.

Par exemple, l'ensemble des chaînes de parenthèses équilibrées ne peut pas être décrit par des expressions régulières. D'un autre côté, ce langage peut être spécifié par une grammaire hors contexte.

Des chaînes répétées ne peuvent pas être décrites par des expressions régulières. L'ensemble $\{wcw \mid w \in (a \mid b)^*\}$ ne peut être dénoté ni par une expression régulière, ni par une grammaire hors contexte.

On peut utiliser des expressions régulières pour dénoter seulement un nombre fixe ou infini de répétitions d'une construction donnée. Deux nombres arbitraires ne peuvent pas être comparés pour voir si ce sont les mêmes.