

Théorie des Langages Formels

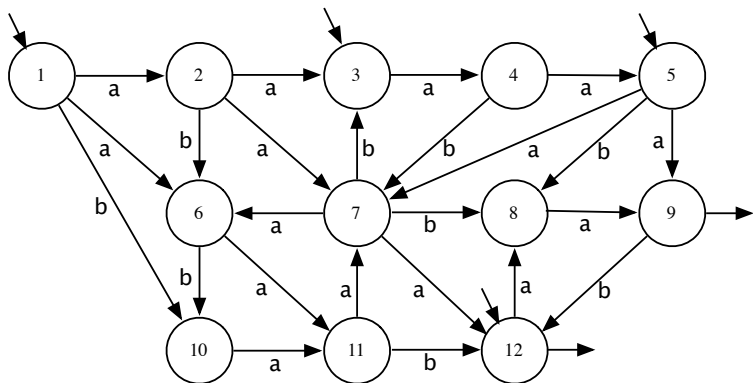
Chapitre 4 : Automates complets déterministes

Florence Levé

`Florence.Leve@u-picardie.fr`

Année 2017-2018

Introduction

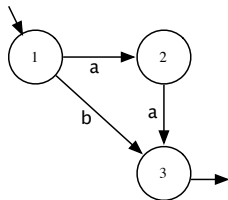
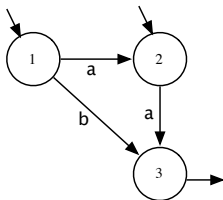
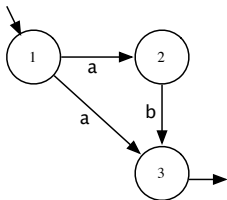


Recherche de : *abaab*, *ababba*

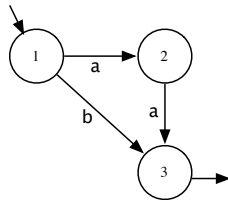
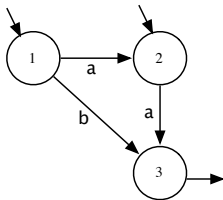
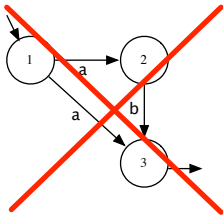
Automate déterministe

- **Automate déterministe.** Un automate est déterministe si et seulement si les deux conditions suivantes sont vérifiées :
 1. L'automate possède **un et un seul état initial** ;
 2. Pour chaque état q et pour chaque lettre α , il existe **au plus** une transition issue de q d'étiquette α .
- Quand un automate est déterministe, l'ensemble des transitions est souvent vu comme une fonction de $Q \times A$ dans Q . L'ensemble des transitions est alors présenté sous forme d'un tableau à deux dimensions. On note alors $\delta(q, a)$ l'état q' (s'il existe) tel que (q, a, q') est une transition.

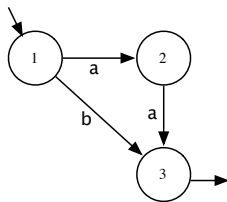
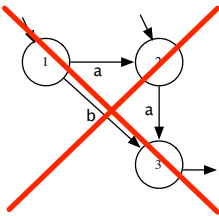
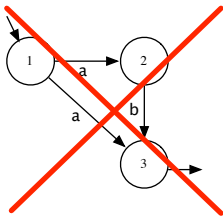
Examples



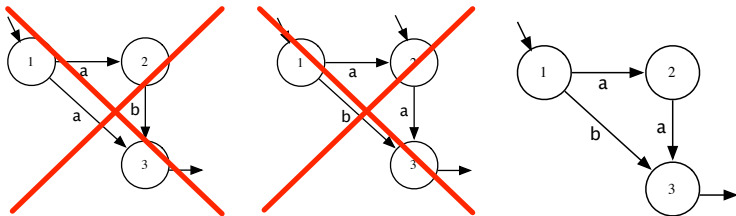
Examples



Examples



Exemples

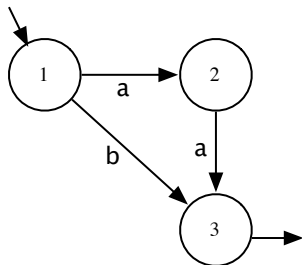


Le troisième automate est déterministe, mais pas complet.

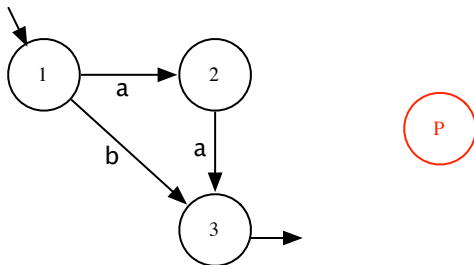
Automate complet

- **Définition.** Un automate est complet si pour chaque état q et pour chaque lettre α , il existe au moins une transition issue de q étiquetée par α .
- **Etat puits.** Pour un automate complet, on appelle *état puits*, tout état e non terminal tel que pour toute transition (e, α, f) , $e = f$.
- **Algorithme de complétion.** Pour rendre un automate complet :
 - ▶ Ajouter un état puits P ;
 - ▶ Ajouter pour chaque état q , et chaque lettre α , une transition (q, α, P) s'il n'existe pas déjà une transition partant de q par la lettre α .
 - ▶ Pour chaque lettre α , ajouter la transition (P, α, P) .
- **Remarque :** La complétion ne change ni le langage reconnu, ni le déterminisme éventuel de l'automate.

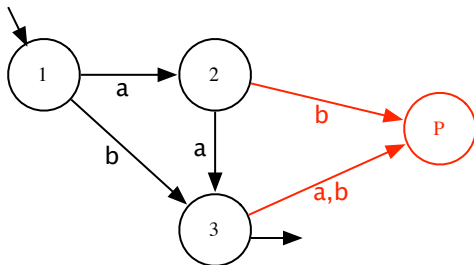
Exemple



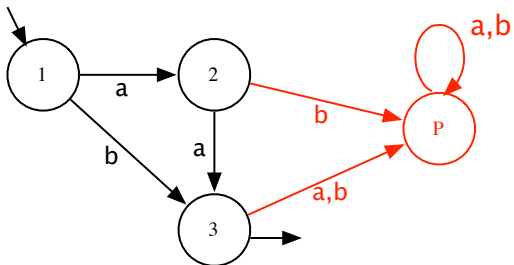
Exemple



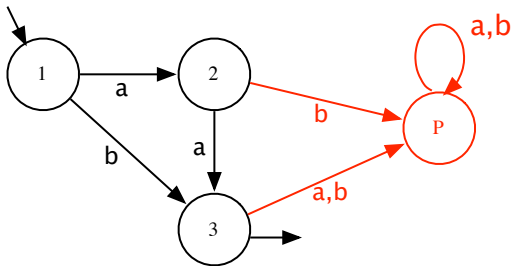
Exemple



Exemple

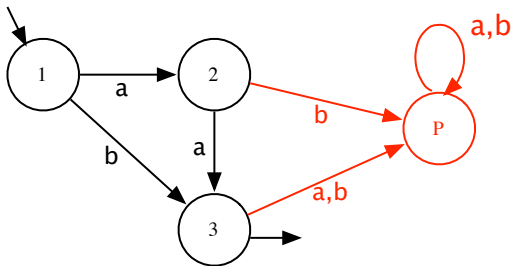


Exemple



- Attention : P n'est pas un état d'acceptation !!

Exemple



- Attention : P n'est pas un état d'acceptation !!
- Remarque : un automate peut être non déterministe mais complet.

Déterminisation (méthode des sous-ensembles)

- Pour tout automate fini $Aut = \langle A, Q, D, F, \delta \rangle$, l'automate Aut_d suivant est déterministe complet et reconnaît le même langage que Aut : $Aut_d = \langle A, Q_d, D_d, F_d, \delta_d \rangle$ où
 - ▶ $Q_d = 2^Q$
i.e. les états de Aut_d sont les parties (sous-ensembles) de Q ;
 - ▶ $D_d = \{D\}$
i.e. D_d ne contient qu'un état qui est l'ensemble des états initiaux de Aut ;
 - ▶ $F_d = \{P \in Q_d \mid P \cap F \neq \emptyset\}$
i.e. est état d'acceptation toute partie de Q qui contient au moins un état d'acceptation de Aut ;
 - ▶ $\delta_d = \{(P, \alpha, P') \mid P, P' \in Q_d, \alpha \in A \text{ tels que } P' = \{q' \mid (q, \alpha, q') \in \delta, q \in P\}\}$
i.e. l'unique transition dans Aut_d qui part de $P \in Q_d$ étiquetée par la lettre α mène dans l'ensemble P' qui est l'ensemble des états de Aut qu'on peut atteindre par une transition étiquetée par α en partant d'un état de P .

Pourquoi ça marche ?

- On vérifie aisément que l'automate construit est déterministe et complet.
- *Propriété* : si P et P' sont deux ensembles d'états de Q (deux états de Q_d), il existe un chemin partant de P aboutissant en P' étiqueté par un mot u si et seulement s'il existe un état p dans P , un état p' dans P' tels qu'il existe un chemin dans Aut partant de p arrivant dans p' d'étiquette u .
Se démontre par récurrence sur u .
- D'où l'égalité des langages.
- Remarque : l'automate construit n'est en général pas émondé.

Construction de la table de transitions

Donnée : un automate $Aut = \langle A, Q, D, F, \delta \rangle$

Résultat : la table de transitions de l'automate accessible de Aut_d .

1. Définir un tableau à deux dimensions ayant une colonne de plus que de lettres dans l'alphabet.
2. Étiqueter la première colonne "états" (les valeurs dans cette colonne qui seront les états de l'automate déterministe, étiquetteront les lignes).
3. Étiqueter les autres colonnes par les lettres de l'alphabet.
4. Placer sur une première ligne en première colonne, l'ensemble des états de départ de Aut .
5. Tant qu'au moins une case du tableau n'est pas remplie :
 - ▶ choisir une case à remplir : notons P l'ensemble d'états de Aut figurant en première colonne de la ligne de cette case, et notons α la lettre étiquetant la colonne de la case ;
 - ▶ calculer la valeur de la case : $\delta(P, \alpha)$;
 - ▶ si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

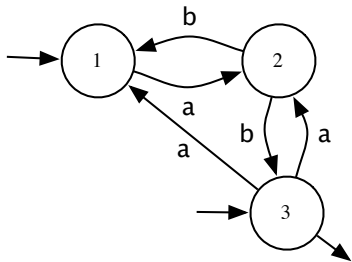
Calcul de l'automate déterministe complet

Donnée : un automate $Aut = \langle A, Q, D, F, \delta \rangle$,

Résultat : construit l'automate accessible Aut' de l'automate Aut_d .

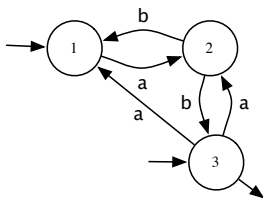
- Émonder l'automate Aut .
- Appliquer l'algorithme précédent de construction de la table de transitions.
 - ▶ à la fin de cet algorithme, sont connus l'alphabet de Aut' (c'est A), les états de Aut' (ce sont les états qui apparaissent dans la première colonne de la table), les transitions de Aut' (elles se lisent dans la table).
- Marquer l'état initial de Aut' : il s'agit de l'état sur la première ligne de la table ; le marquer visuellement par une flèche \rightarrow juste devant l'état de la table.
- Déterminer et marquer les états terminaux de Aut' : pour le marquage mettre une flèche \leftarrow devant chaque état concerné (si l'état initial est aussi terminal, on pourra remplacer la flèche \rightarrow par une flèche \leftrightarrow)

Exemple

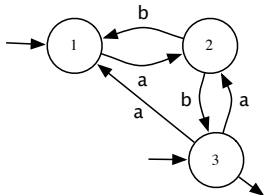


L'automate est émondé.

Exemple : calcul de la table de transition



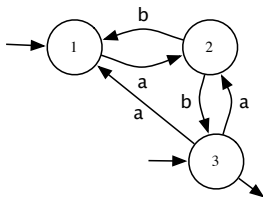
Exemple : calcul de la table de transition



1. Définir un tableau à deux dimensions ayant une colonne de plus que de lettres dans l'alphabet.

--	--	--

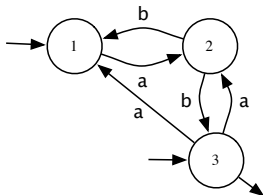
Exemple : calcul de la table de transition



2. Étiqueter la première colonne "états".

états		
-------	--	--

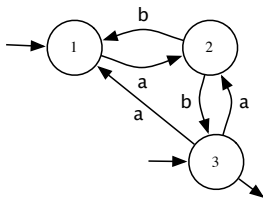
Exemple : calcul de la table de transition



3. Étiqueter les autres colonnes par les lettres de l'alphabet.

états	a	b

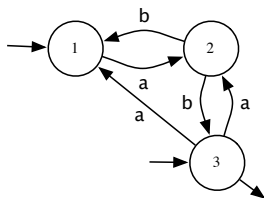
Exemple : calcul de la table de transition



4. Placer sur une première ligne en première colonne, l'ensemble des états de départ de *Aut*.

états	a	b
{1,3}		

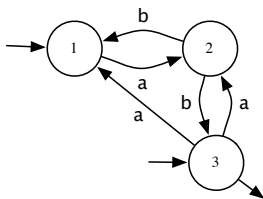
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
{1, 3}		

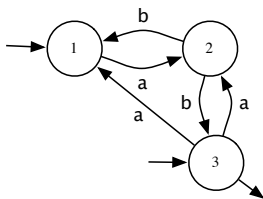
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
{1, 3}	{1, 2}	

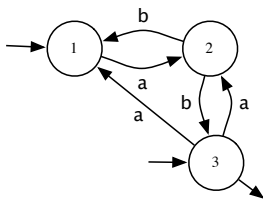
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
{1, 3}	{1, 2}	
{1, 2}		

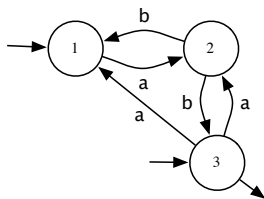
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
{1, 3}	{1, 2}	\emptyset
{1, 2}		

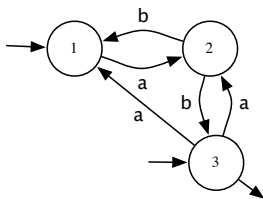
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
{1, 3}	{1, 2}	\emptyset
{1, 2}		
\emptyset		

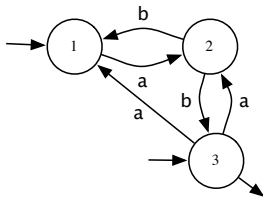
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
{1, 3}	{1, 2}	\emptyset
{1, 2}	{2}	
\emptyset		

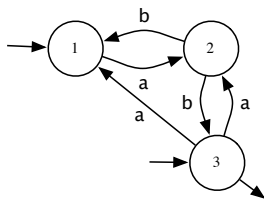
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
{1, 3}	{1, 2}	\emptyset
{1, 2}	{2}	
\emptyset		
{2}		

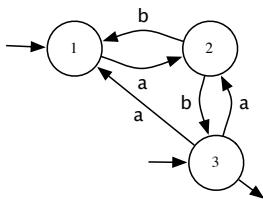
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
$\{1, 3\}$	$\{1, 2\}$	\emptyset
$\{1, 2\}$	$\{2\}$	$\{1, 3\}$
\emptyset		
$\{2\}$		

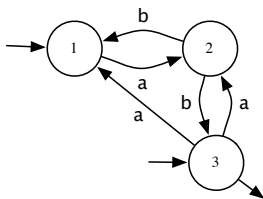
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
$\{1, 3\}$	$\{1, 2\}$	\emptyset
$\{1, 2\}$	$\{2\}$	$\{1, 3\}$
\emptyset		
$\{2\}$	\emptyset	

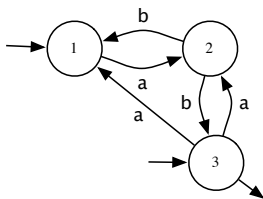
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
$\{1, 3\}$	$\{1, 2\}$	\emptyset
$\{1, 2\}$	$\{2\}$	$\{1, 3\}$
\emptyset		
$\{2\}$	\emptyset	$\{1, 3\}$

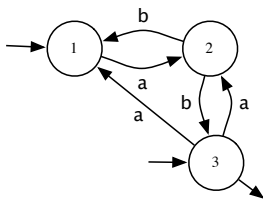
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

états	a	b
$\{1, 3\}$	$\{1, 2\}$	\emptyset
$\{1, 2\}$	$\{2\}$	$\{1, 3\}$
\emptyset	\emptyset	
$\{2\}$	\emptyset	$\{1, 3\}$

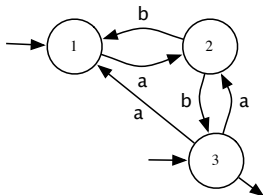
Exemple : calcul de la table de transition



5. Tant qu'au moins une case du tableau n'est pas remplie :
- choisir une case à remplir et calculer sa valeur $\delta(P, \alpha)$;
 - si aucune ligne n'est associée à cette valeur, commencer une nouvelle ligne étiquetée par cette valeur.

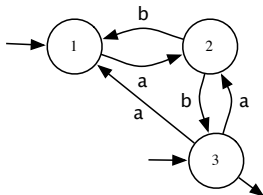
états	a	b
$\{1, 3\}$	$\{1, 2\}$	\emptyset
$\{1, 2\}$	$\{2\}$	$\{1, 3\}$
\emptyset	\emptyset	\emptyset
$\{2\}$	\emptyset	$\{1, 3\}$

Exemple : marquer l'état initial



	états	a	b
→	{1, 3}	{1, 2}	\emptyset
	{1, 2}	{2}	{1, 3}
	\emptyset	\emptyset	\emptyset
	{2}	\emptyset	{1, 3}

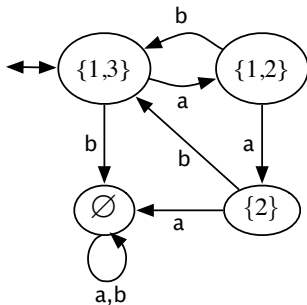
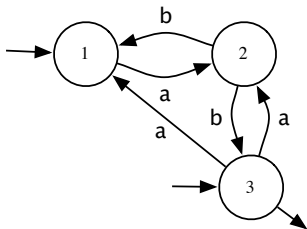
Exemple : marquer les états d'acceptation



	états	a	b
\leftrightarrow	{1, 3}	{1, 2}	\emptyset
	{1, 2}	{2}	{1, 3}
	\emptyset	\emptyset	\emptyset
	{2}	\emptyset	{1, 3}

Exemple : l'automate déterministe correspondant

	états	a	b
\leftrightarrow	$\{1, 3\}$	$\{1, 2\}$	\emptyset
	$\{1, 2\}$	$\{2\}$	$\{1, 3\}$
	\emptyset	\emptyset	\emptyset
	$\{2\}$	\emptyset	$\{1, 3\}$



Complexité

- La détermination est une opération fondamentalement exponentielle.
 - ▶ En effet, il existe des automates non déterministes, pour lesquels tout automate déterministe reconnaissant le même langage contient un nombre exponentiel d'état.
 - ▶ Par exemple, pour n entier, le langage $(a + b)^* a (a + b)^{n-1}$ des mots sur $\{a, b\}$ dont la n -ième lettre avant la fin est un a est reconnu par un automate non déterministe à $n + 1$ états ($Q = \{0, \dots, n\}$, $D = \{0\}$, $F = \{n\}$, $\delta = \{(0, a, 0), (0, b, 0), (0, a, 1)\} \cup \{(i - 1, a, i), (i - 1, b, i) \mid i \in \{1, \dots, n\}\}$), mais tout automate déterministe doit contenir au moins 2^n états (il faut un état pour le mot vide et un état pour chaque mot de la forme au avec $|au| \leq n$ pour chaque mot de longueur au plus $n - 1$).
- Tester si un automate est déterministe peut se faire en temps polynomial.

Utilisation d'automates déterministes : complémentation

- **Complémentation**

Soit $Aut = \langle A, Q, \{d\}, F, \delta \rangle$ un automate fini déterministe complet reconnaissant un langage L .

Le langage $A^* \setminus L$ (*complémentaire* de L dans A^*) est reconnu par l'automate

$$Aut' = \langle A, Q, \{d\}, Q \setminus F, \delta \rangle .$$

- Aucune transition n'ayant été enlevée ou ajoutée, cet automate est également déterministe complet.

Remarques

- Le fait que l'automate soit ici déterministe est fondamental.
- Exemple : $A = \{a\}$, $Q = \{1, 2, 3\}$, $D = \{1\}$, $F = \{2\}$, $\delta = \{(1, a, 1), (1, a, 2), (2, a, 3)\}$. Cet automate reconnaît a^+ . Si on prend $F = \{1, 3\}$ on obtient un automate qui reconnaît a^* et non $\{\varepsilon\}$ le complémentaire attendu.
- Le fait que l'automate soit ici complet est également fondamental (ici l'alphabet de référence joue également un rôle capital).
- Exemple : $A = \{a, b\}$, $Q = \{1\} = D = F$, $\delta = \{(1, a, 1)\}$. La complémentation va reconnaître l'ensemble vide et non tous les mots contenant au moins un b .
- Tous les états de l'automate résultat ne seront pas nécessairement co-accessibles même s'ils l'étaient au départ.

Complexité

- Étant donné un automate de n états, la complexité de la complémentation est en $O(n)$ si on ne fait pas une copie de l'automate. Elle est en $O(n^2)$ sinon.

Utilisation pour l'intersection

- Étant donnés deux automates reconnaissant des langages L_1 et L_2 , existe-t-il un algorithme qui permet de construire un automate reconnaissant l'intersection $L_1 \cap L_2$?
 - ▶ Oui !
 - ▶ Cela peut se faire avec des automates déterministes, en obtenant un automate déterministe.
 - ▶ L'idée est de suivre simultanément des chemins dans les deux automates.
 - ▶ La même technique permet de construire un automate déterministe reconnaissant l'union de deux langages reconnus par des automates déterministes.

Union et intersection déterministe

- Soient $Aut_1 = \langle A, Q_1, \{d_1\}, F_1, \delta_1 \rangle$ et $Aut_2 = \langle A, Q_2, \{d_2\}, F_2, \delta_2 \rangle$ deux automates finis déterministes complets reconnaissant les langages L_1 et L_2 .
- Les langages $L_1 \cap L_2$ et $L_1 \cup L_2$ sont respectivement reconnus par les automates finis déterministes complets

$$Aut_{\cap} = \langle A, Q_1 \times Q_2, \{(d_1, d_2)\}, F, \delta \rangle$$

et

$$Aut_{\cup} = \langle A, Q_1 \times Q_2, \{(d_1, d_2)\}, F', \delta \rangle$$

où :

- ▶ $\delta = \{((p_1, p_2), a, (q_1, q_2)) \mid (p_1, a, q_1) \in \delta_1, (p_2, a, q_2) \in \delta_2\}$;
- ▶ $F = F_1 \times F_2$ (un mot est dans $L_1 \cap L_2$ s'il est accepté par Aut_1 **et** Aut_2) ;
- ▶ $F' = (F_1 \times Q_2) \cup (Q_1 \times F_2)$ (un mot est dans $L_1 \cup L_2$ s'il est accepté par Aut_1 **ou** Aut_2).

Remarques

- Là encore, le fait que les automates soient complets est capital pour l'union.
- Pour l'union, l'automate a plus d'état que pour l'union non déterministe vue précédemment.
- Tous les états de l'automate d'union (et a fortiori pour l'intersection) ne sont pas nécessairement accessibles et co-accessibles.
- Si n_1 et n_2 sont les nombres d'états respectifs du premier et du deuxième automate, alors les algorithmes précédents sont en $O(n_1^2 \times n_2^2)$.

Conséquences

- **Proposition :** La famille des langages rationnels (resp. reconnaissables) est close par intersection et complémentation.
- $L_1 \setminus L_2 = L_1 \cap (A^* \setminus L_2)$.
 $\Rightarrow L_1 \setminus L_2$ est calculable par les algorithmes précédents.
- La proposition précédente explique l'usage d'expressions régulières étendues dans les logiciels avec possibilité d'intersection et de complémentation
 - ▶ Cela permet une plus grande facilité d'expression
 - ▶ Par exemple, pour les identifiants de variables, il est plus facile d'exclure tous les mots clefs plutôt que de prévoir directement une expression rationnelle ou un automate représentant le langage