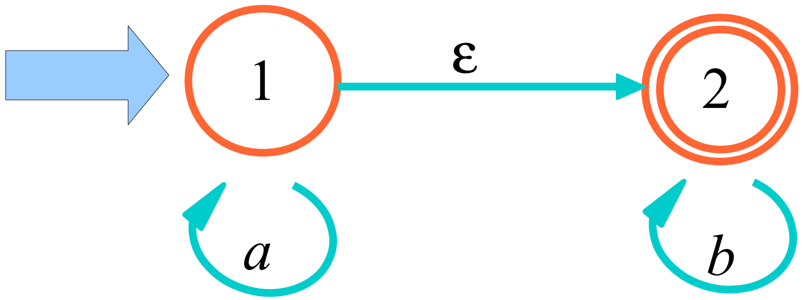
Automates finis

- Automate fini déterministe (AFD)
- Automates finis non déterministes AFN
- Construction d'automates
- Langage d'un AFD
- Théorème de Kleene
- Équivalence d'automates – Minimisation

Automates finis non déterministes à ε transition AFNε

- Dans un second temps on va autoriser des transitions spontanées entre états, sans aucune lecture de lettre. On peut voir ça comme une transition par lecture du mot vide, c'est pourquoi on les nomme ε-transitions.

Exemple :
Il est simple alors d'écrire un automate reconnaissant le langage *a*b**



Définition

Un <u>automate fini non déterministe à ε-transitions</u> (AFNε) est un quintuplet :

$$\mathcal{A} = (Q, \Sigma, T, I, A)$$

où un point est modifié par rapport à un AFN

- la fonction de transition est maintenant une application
  $$T : Q.(\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q)$$
  pouvant provoquer un changement d'état spontané appelé « ε-transition », c'est-à-dire sans lecture de lettre.

- Il faut ajouter une colonne pour les ε-transitions dans la table de transition de l'automate. La table de l'automate donné en exemple est alors :

T	*a*	*b*	*ε*
1	1		2
2		2	

- Nous montrerons au prochain paragraphe que cela ne changera pas théoriquement l'ensemble des langages automatiques.

- Les transitions spontanées pouvant avoir lieu (ou pas) à n'importe quel moment, le comportement de ces automates est hautement non déterministe.

- Comme pour un AFN, on accepte un mot si au moins une façon de le lire conduit à un état acceptant.

Lecture d'un mot

Soit $\mathcal{A} = (Q, \Sigma, T, I, A)$ un automate non déterministe à ε-transitions.

Si $w=\ell_1..\ell_n$ est un mot sur l'alphabet Σ , ce mot peut être ε–complété (rembourré par quelques ε bien placés) en un mot $w'=a_1a_2..a_p$ sur l'alphabet $\Sigma\cup\{\varepsilon\}$ tel que :

- $p \geq n$
- pour tout i, $a_i \in \{ \ell_1..\ell_n \}\cup\{\varepsilon\}$
- $\ell_1.\ell_2. \, . \,\ell_n = a_1.a_2...a_p$ (produits de concaténations entre mots)

UNE lecture du mot $w=\ell_1..\ell_n$ par $\mathcal{A}$ est alors UNE lecture par l'AFN $\mathcal{A}$ de n'importe quel mot w', ε – complété de w.

- La lecture d'un mot conduit là encore à une forêt de lecture et d'après la définition donnée, une lecture de ce mot est un chemin dans cette forêt, allant d'une des racines à une feuille du même arbre.

- La position et le nombre de transitions spontanées peut être telle qu'il est pénible de dessiner tout l'arbre de lecture d'un mot et qu'on y renonce...

On retrouve dans ce cadre, nos trois définitions devenues classiques...

Langage d'un AFN$_\varepsilon$

Définition

On dit que le mot w est accepté par l'automate non déterministe à ε-transitions $\mathcal{A} = (Q, \Sigma, T, I, A)$ s'il existe une lecture du mot w conduisant à un état acceptant $q \in A$.

Définition

On appelle langage de l'automate $\mathcal{A} = (Q, \Sigma, T, I, A)$ l'ensemble des mots acceptés par cet automate.

On le note $L(\mathcal{A})$.

On dit encore que $\mathcal{A}$ reconnaît ou accepte ce langage.

Définition

Deux automates sont dits équivalents ssi ils reconnaissent le même langage.

Déterminisation d'un AFN

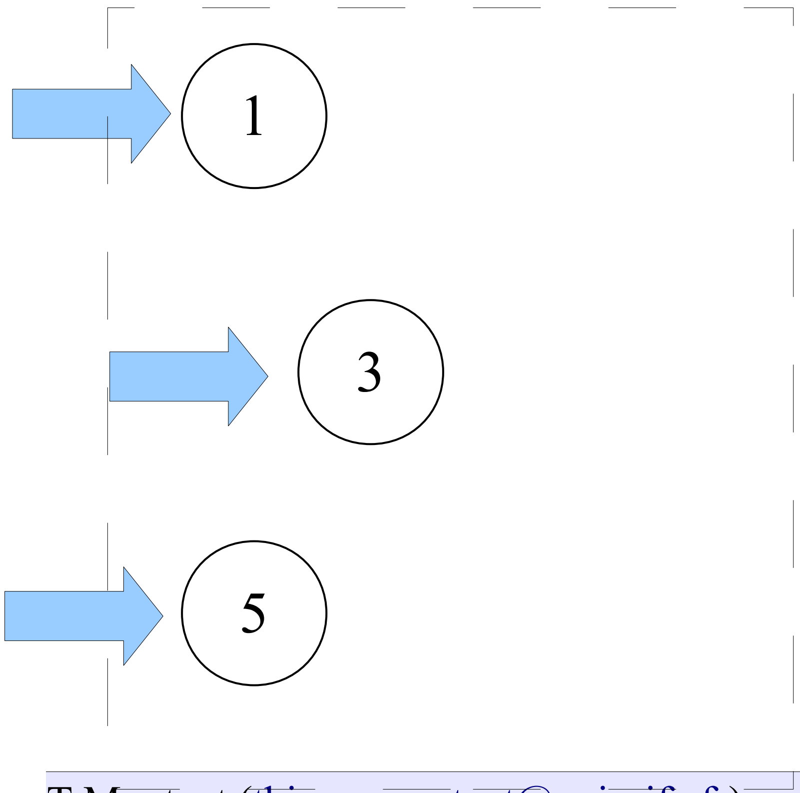
Théorème

Pour tout automate non déterministe $\mathcal{A} = (Q, \Sigma, T, I, A)$, il existe un automate fini déterministe $\mathcal{A}'$ équivalent à $\mathcal{A}$.

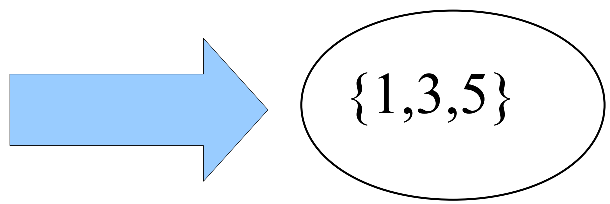
Ainsi la notion d'AFN, beaucoup plus souple d'emploi, ne change rien à l'ensemble des langages automatiques.

Nous donnons de ce résultat une preuve constructive fournissant également l'algorithme de déterminisation d'un AFN.
Traitons les deux sources de non déterminisme :

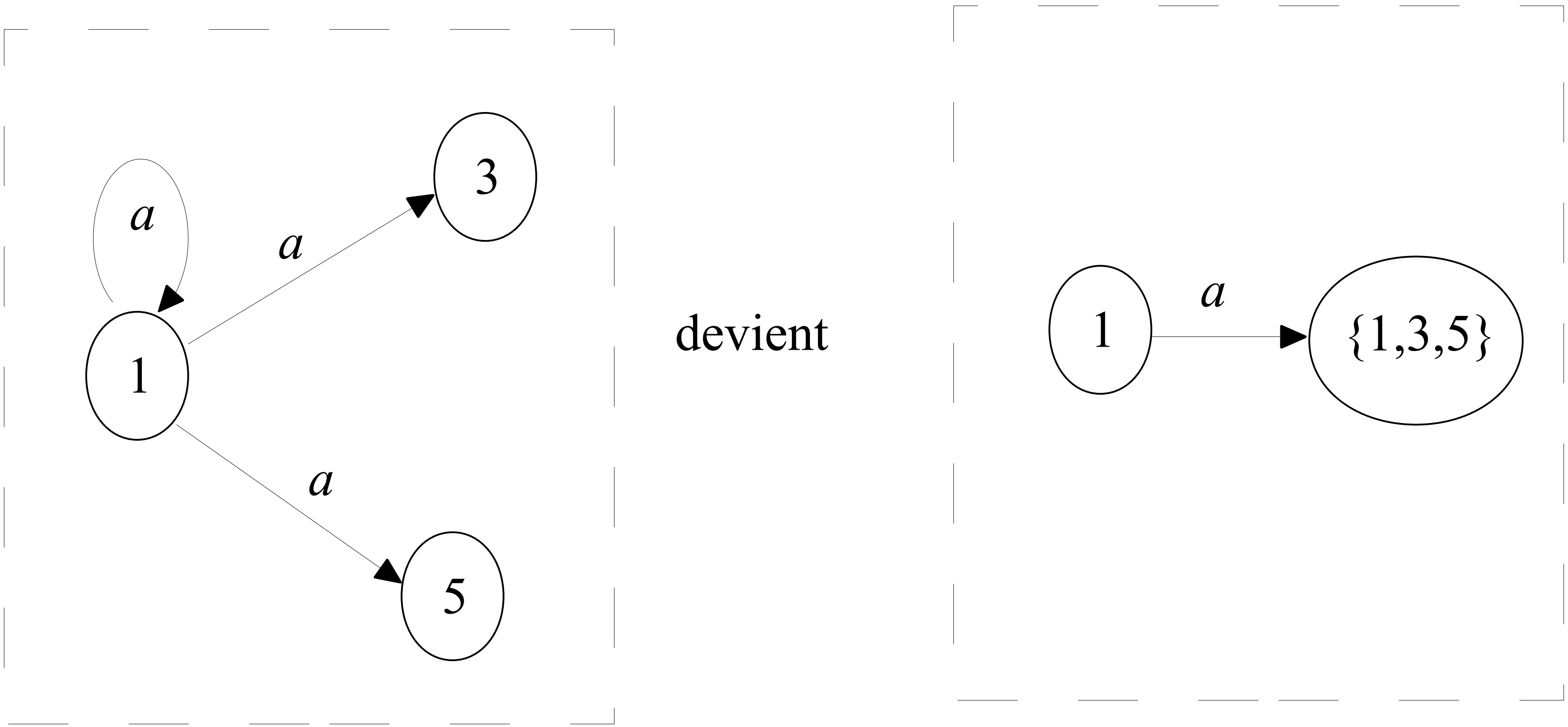
- Le fait de posséder plusieurs états initiaux est facilement résolu si on considère qu'il s'agit d'un UNIQUE ensemble d'états initiaux



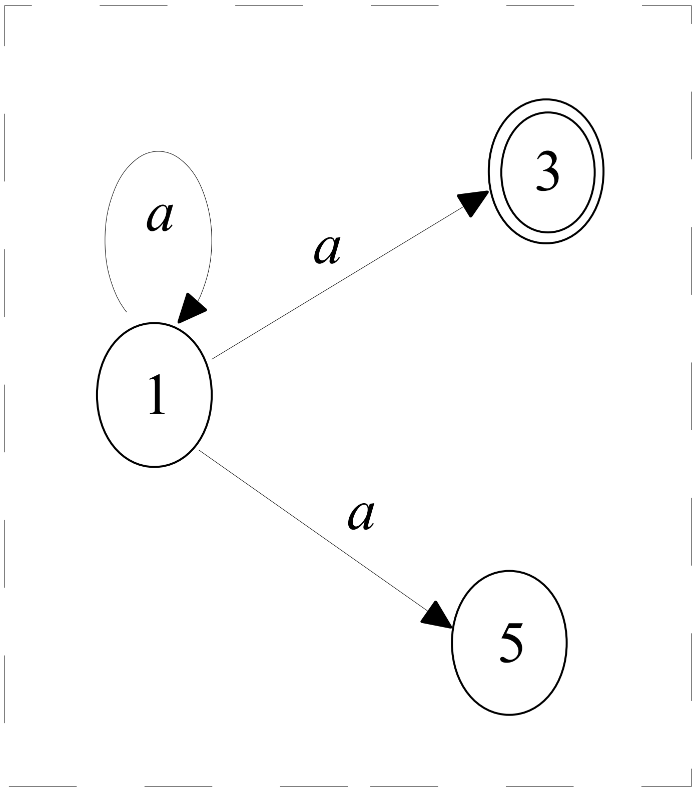
devient....

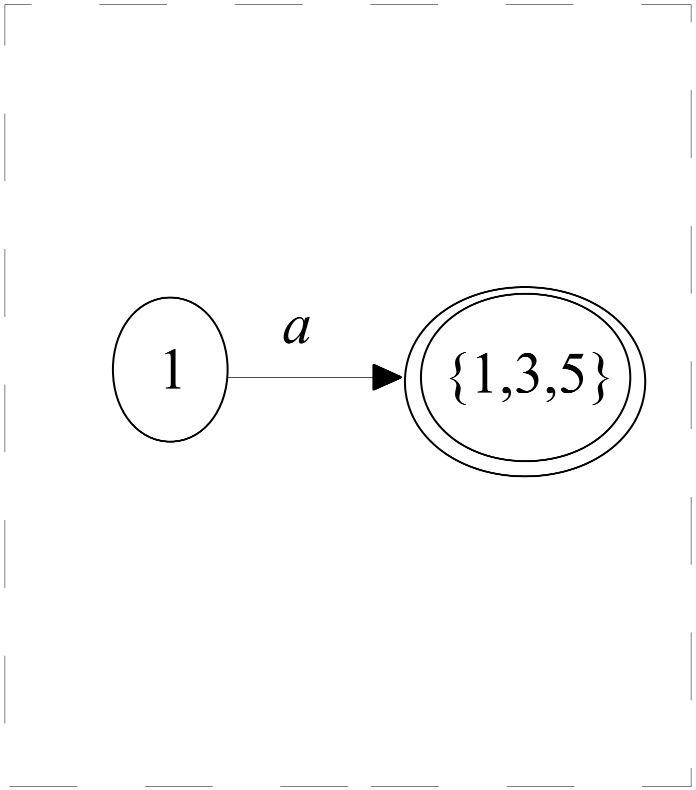


- A partir d'un état q, plusieurs transitions sont possibles par lecture d'une même lettre. Là encore il suffit de regrouper tous les états d'arrivée possibles dans un même ensemble d'états.



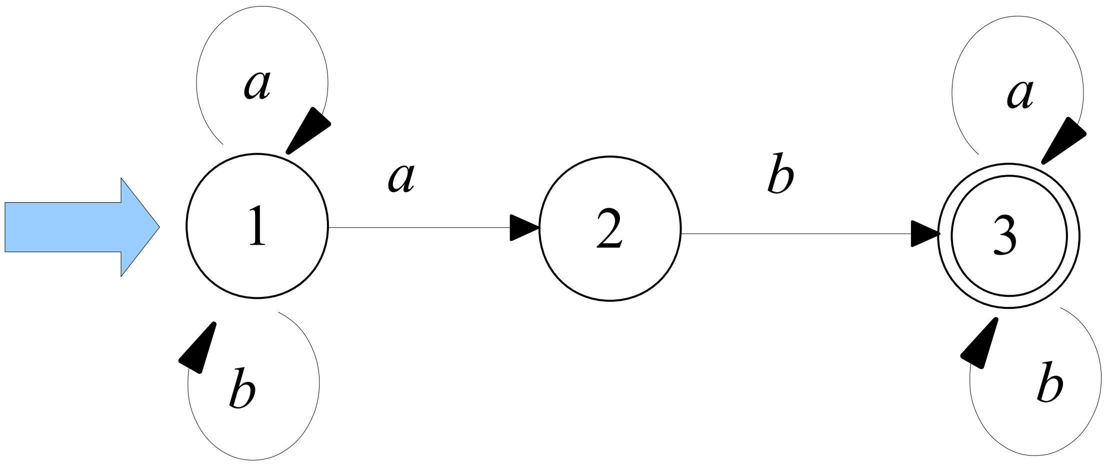
- Si une transition possible mène à un état acceptant, alors les états regroupés doivent également être acceptants :





L'idée est donc de construire à partir de l'AFN $\mathcal{A}$ donné, un automate appelé automate des parties dont les états sont constitués d'ensembles d'états de $\mathcal{A}$.
(Voir l'animation DéterminiserUnAFN)

Mettons ces idées en œuvre sur l'exemple suivant dont on a vu qu'il s'agit d'un AFN reconnaissant les mots contenant *ab*.

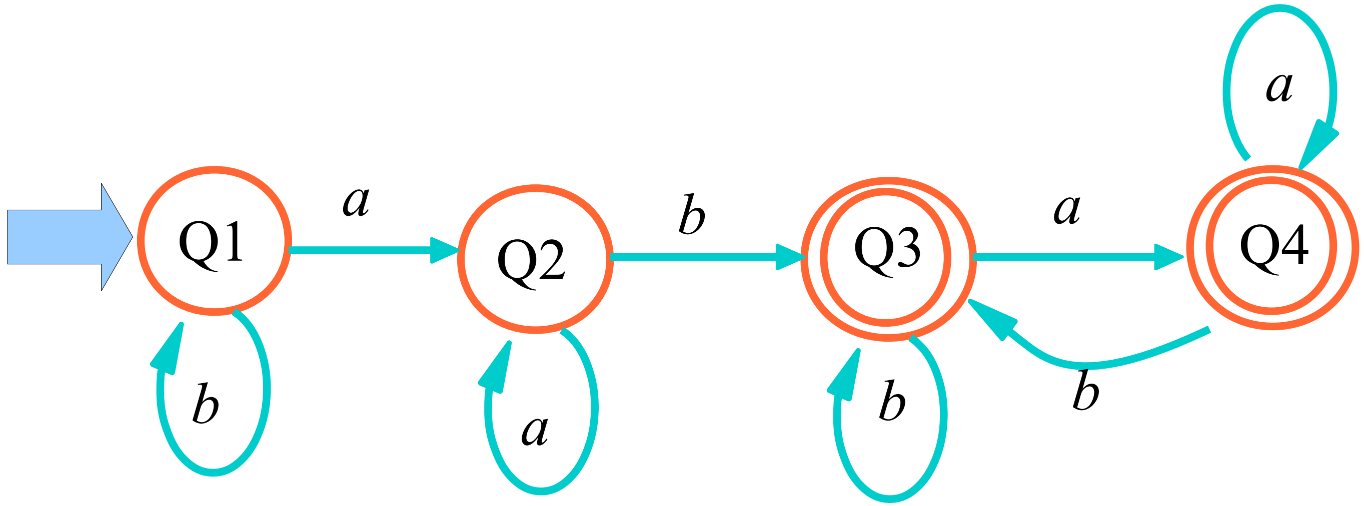


Le plus simple est de raisonner sur la table de transition et de construire progressivement les états du nouvel automate et les transitions correspondantes :

On obtient successivement :

T	*a*	*b*
Q1={1}	{1,2}	Q1
Q2={1,2}	Q2	{1,3}
Q3={1,3}	{1,2,3}	Q3
Q4={1,2,3}	Q4	Q3

Donc $\mathcal{A}$ est équivalent à l'automate déterministe suivant :



Algorithme (déterminisation d'un AFN)

On construit à partir de $\mathcal{A}$ un AFD dont les états sont constitués d'ensemble d'états de $\mathcal{A}$.

- On part de Q1=I l'ensemble des états initiaux de $\mathcal{A}$.

- tant qu'il reste un état Q dont on n'a pas étudier les transitions :
    - Pour chaque lettre l de l'alphabet :
    - On définit Q'=T(Q,l) regroupant tous les états accessibles à partir des états contenus dans Q par lecture de la lettre l.

- Un état Q est acceptant s'il contient un état acceptant de $\mathcal{A}$.

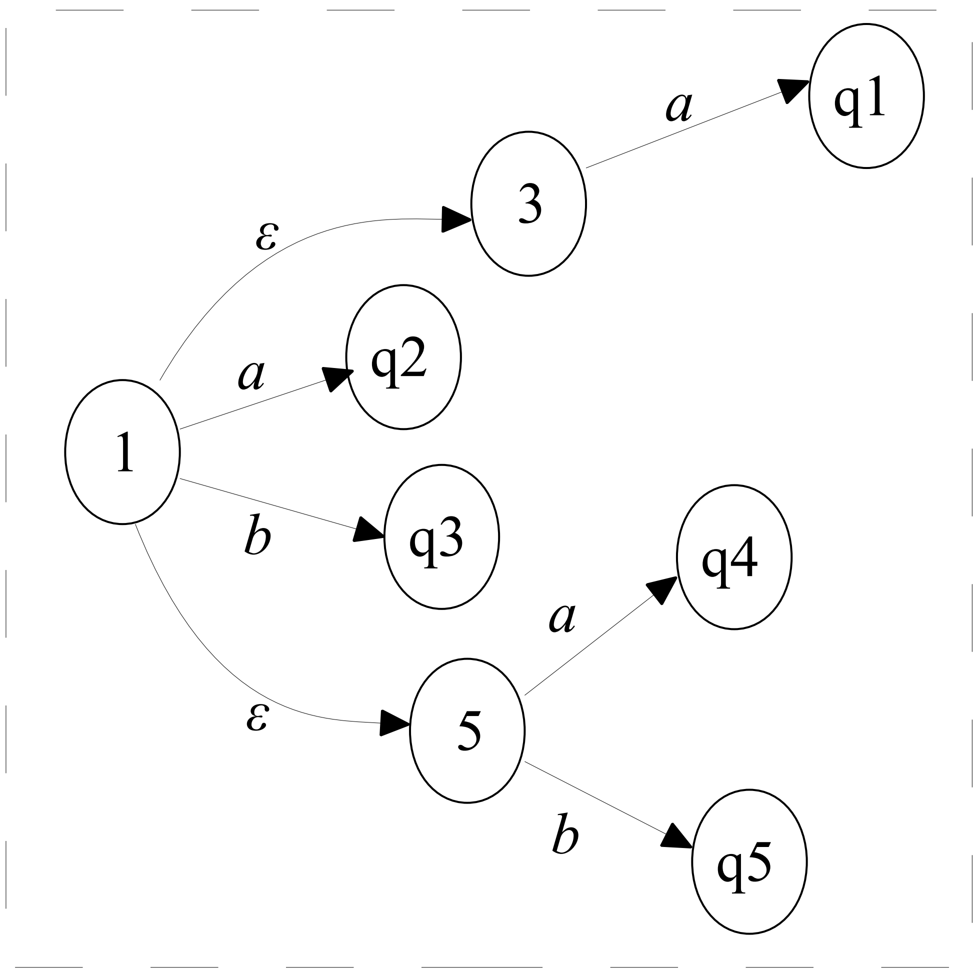
Déterminisation d'un AFNε

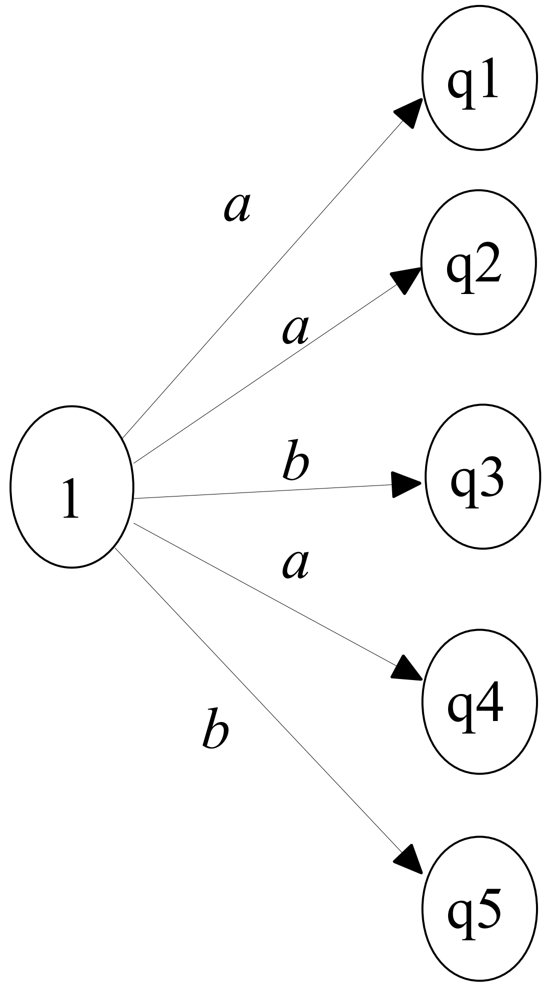
Théorème

Pour tout automate non déterministe à $\mathcal{A} = (Q, \Sigma ,T,I,A)$, il existe un automate fini non déterministe $\mathcal{A}'$ équivalent à $\mathcal{A}$. (donc aussi un automate fini déterministe équivalent).

La encore, on ne change donc pas les langages automatiquement acceptables en autorisant des transitions spontanées

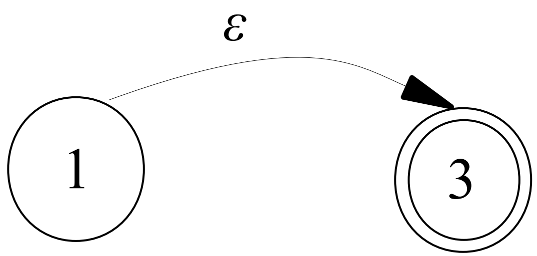
- L'idée est cette fois-ci, non plus de regrouper les états, mais d'étendre les transitions à tous les états que l'on peut atteindre par ε–transition





- D'autre part :

  Si un état peut passer dans un état acceptant par ε–transition, il doit être lui-même considéré comme acceptant.



On peut alors définir proprement ...

Définition

Soit $\mathcal{A}$ un Automate non déterministe à ε–transitions.

Pour chaque état q de $\mathcal{A}$ on définit la clôture de q par ε–transitions comme l'ensemble des états accessibles à partir de q sans lire de lettre.

Elle contient donc q lui-même et les états accessibles en n'utilisant que des ε–transitions. On la note C(q).

Dans notre exemple précédent C(1)={1,3,5}

Définition

On définit alors T*(q,l), les transitions étendues à partir de q par lecture de la lettre l comme l'ensemble des états accessibles à partir d'un état appartenant à C(q) par lecture de la lettre l.

Dans notre exemple précédent T*(1,a)={q1, q2, q4}

Un état peut être considéré comme acceptant si sa clôture contient un état acceptant.

Algorithme (déterminisation d'un AFN à ε-transitions.)

On construit à partir de $\mathcal{A}$ un AFN $\mathcal{A}'$ en modifiant les transitions et les états acceptants :

- On commence par calculer toutes les clôtures de tous les états.
- On ajoute (récursivement) à l'ensemble des états acceptants, tous les états dont la clôture contient au moins un état acceptant.
- On remplace les fonctions de transition par les fonctions de transition étendues.

On obtient un AFN que l'on sait assez bien déterminiser depuis tout à l'heure...

Exercice : Donner un automate déterministe équivalent à l'automate suivant (dont on sait qu'il reconnaît le langage a*b*)