Automates

le cnam

- ⋆ outil théorique
- ⋆ ≃ programme, mais plus simple
- ⋆ représente un comportement dynamique et/ou automatique
- ⋆ représentation graphique
- ⋆ sorte d'esperanto de la « description » de système simple
- ⋆ une certaine catégorie de calculs bien comprise

Automates – Applications

le cnam

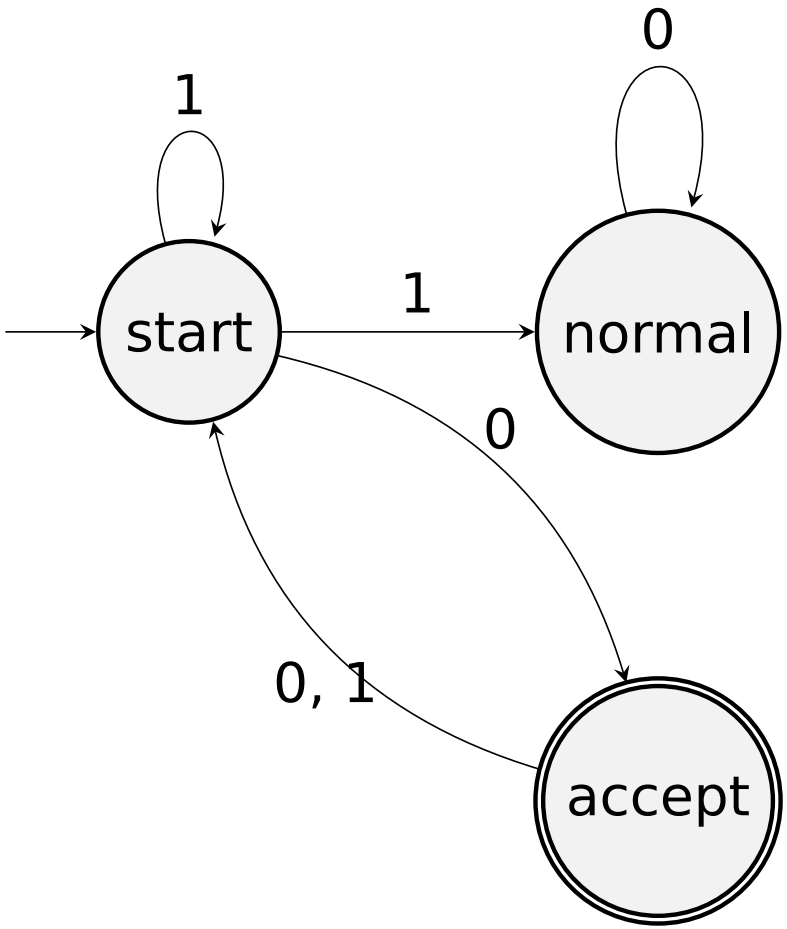
- électronique (universel)
- reconnaissance lexicale (lex, yacc, antlr)
- description/spécification de système
  (= diagramme états-transitions d'UML)
- technique de programmation systématique (problèmes simples)
- = expressions régulières

Automates – Avantages

le cnam

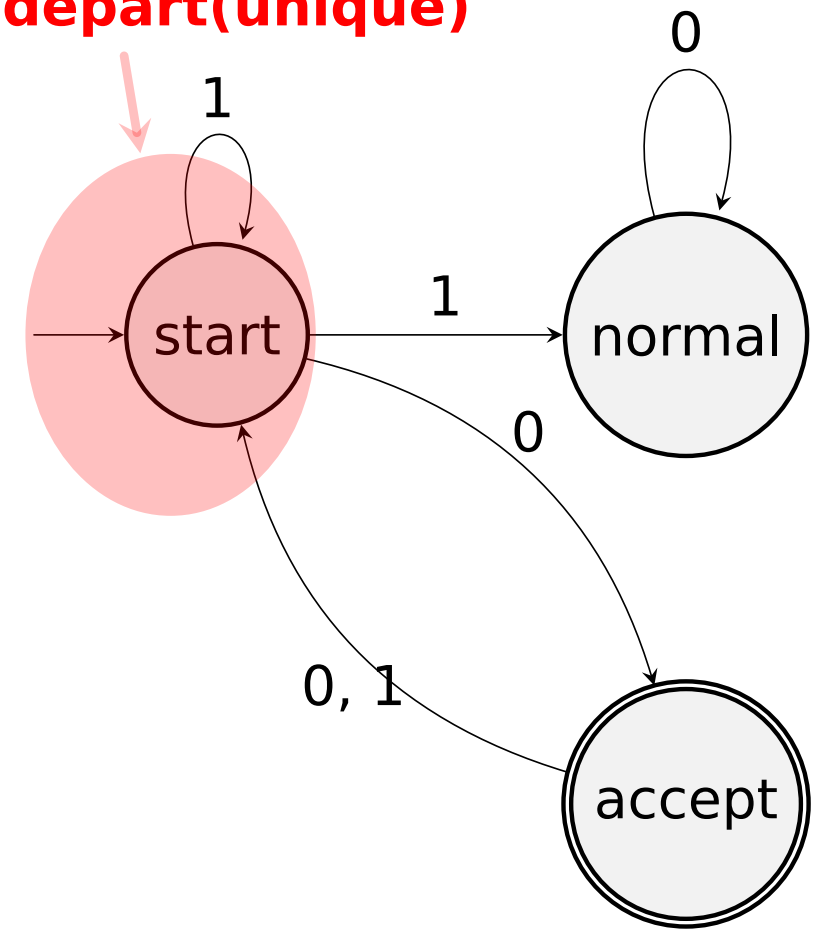
- bien compris
- nombreux algorithmes :
  - composition,
  - minimisation,
  - déterminisation,
  - etc.

Exemple

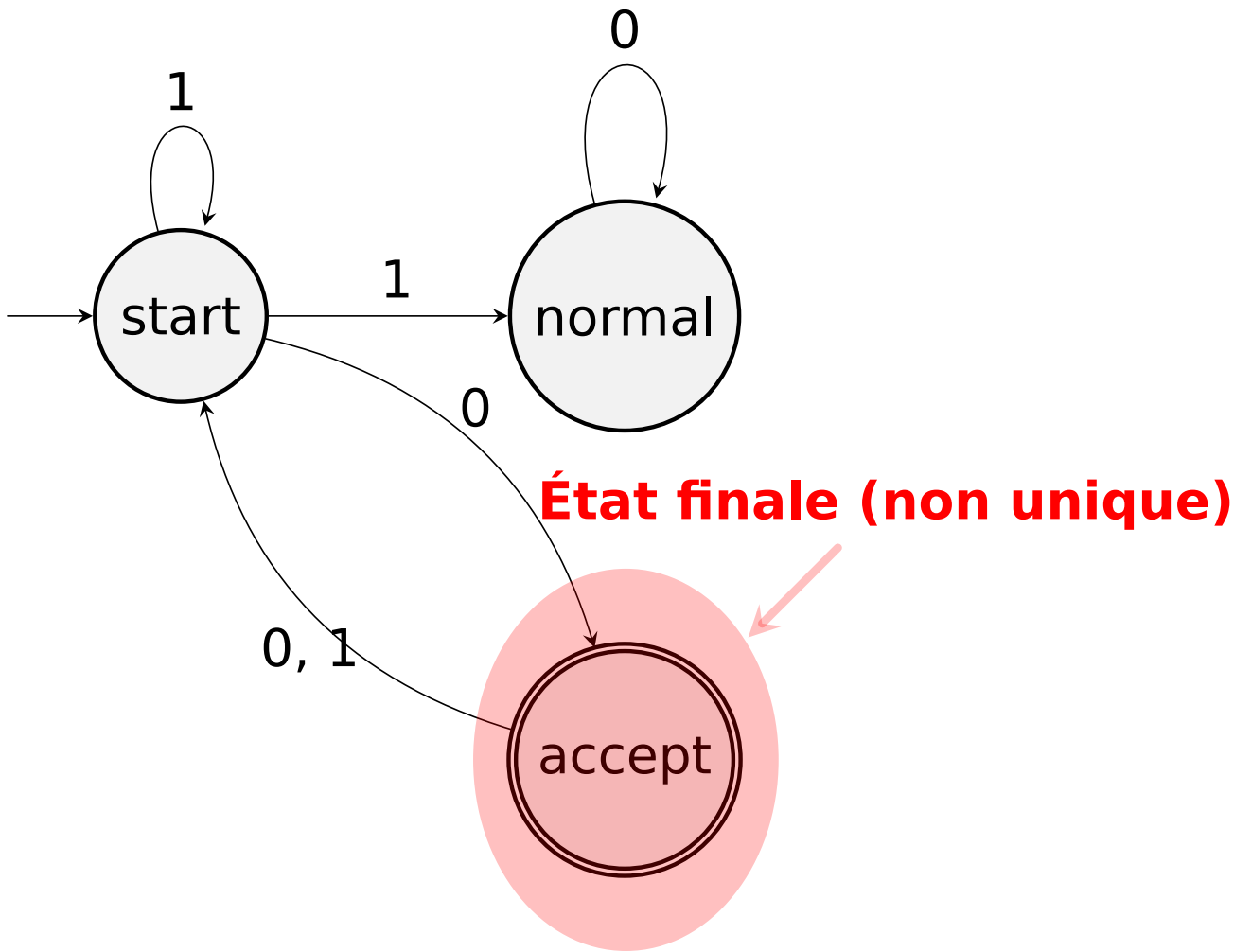


Exemple

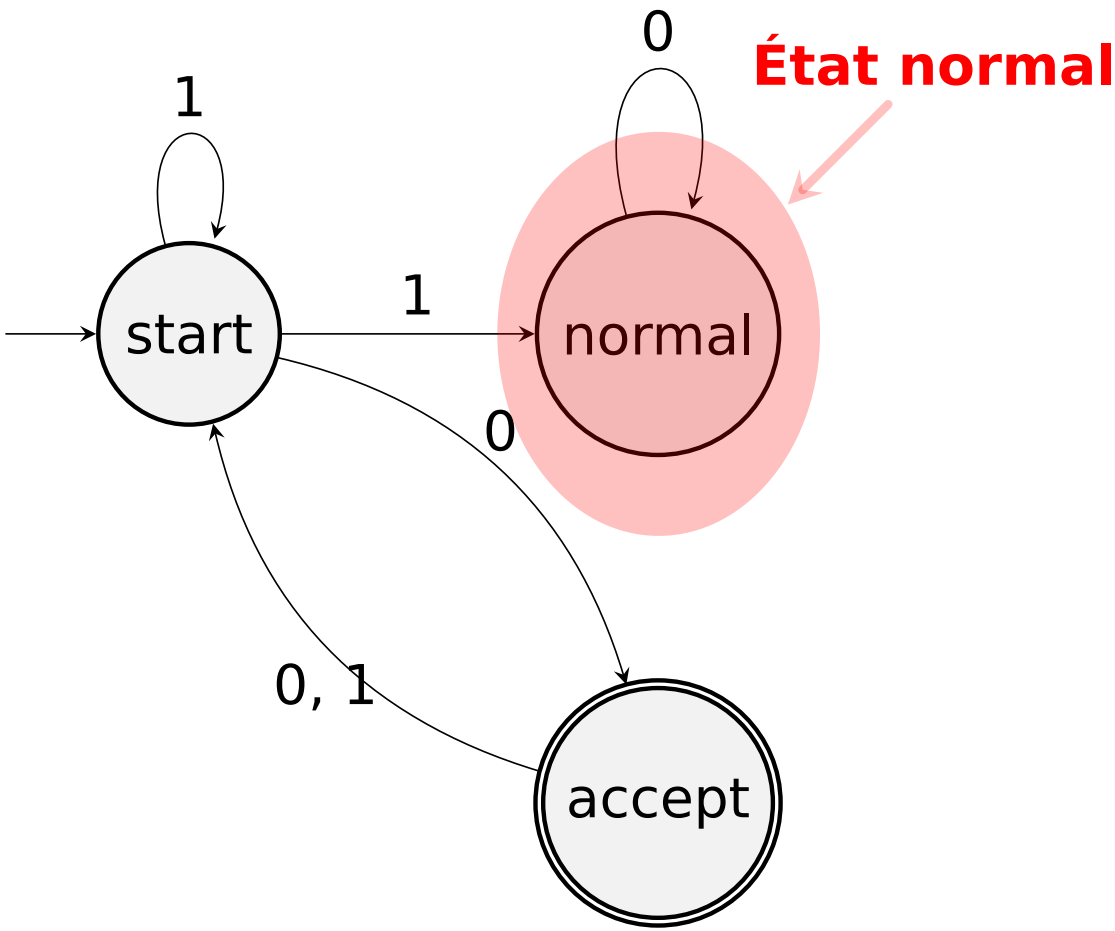
État de départ(unique)



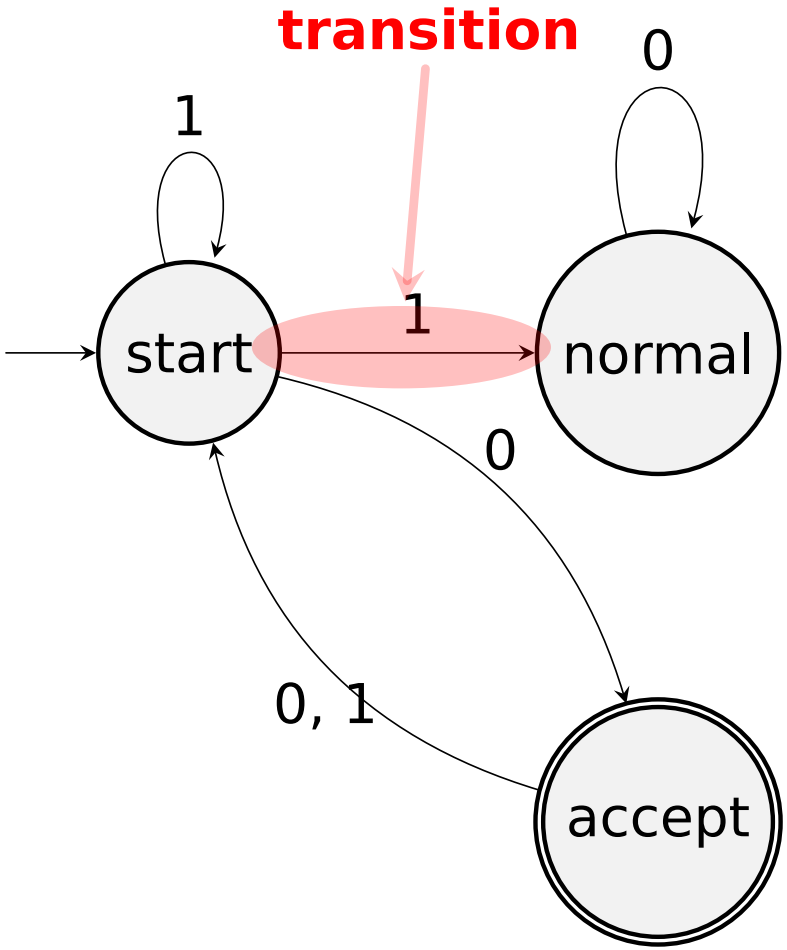
Exemple



Exemple

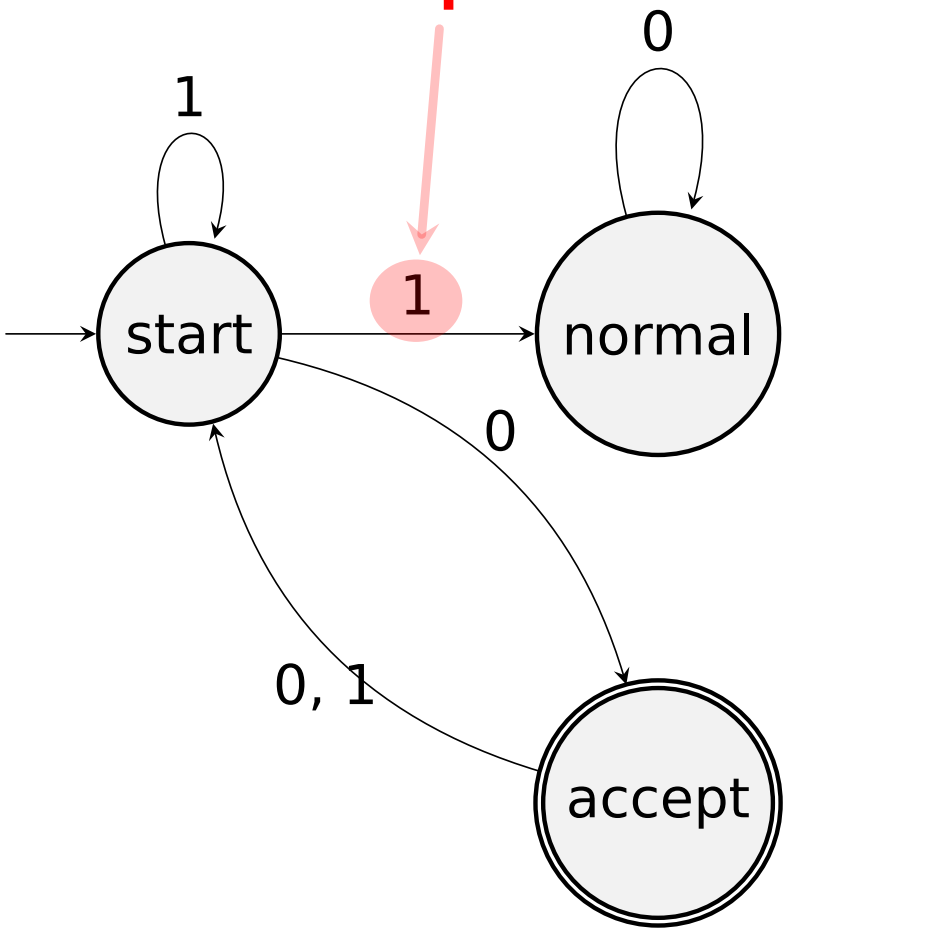


Exemple



Exemple

caractère « lu » par la transition



- entrée : un automate A et un « mot » m à « reconnaître »
- on démarre dans l'état initial
- on lit (consomme) la première lettre de m
- on suit une transition étiquetée par cette lettre
  - pas nécessairement unique
  - on arrive dans un nouvel état
- Si il reste des lettres dans m
  - Alors retour à l'étape 1
  - Sinon sommes nous sur un état final ?
    - Oui. OK mot reconnu
    - NON. Revenir en arrière et essayer un autre choix de transition

Il suffit qu'il y ait un « chemin » de l'état initial vers un état final.

Site de démonstration des automates

le cnam

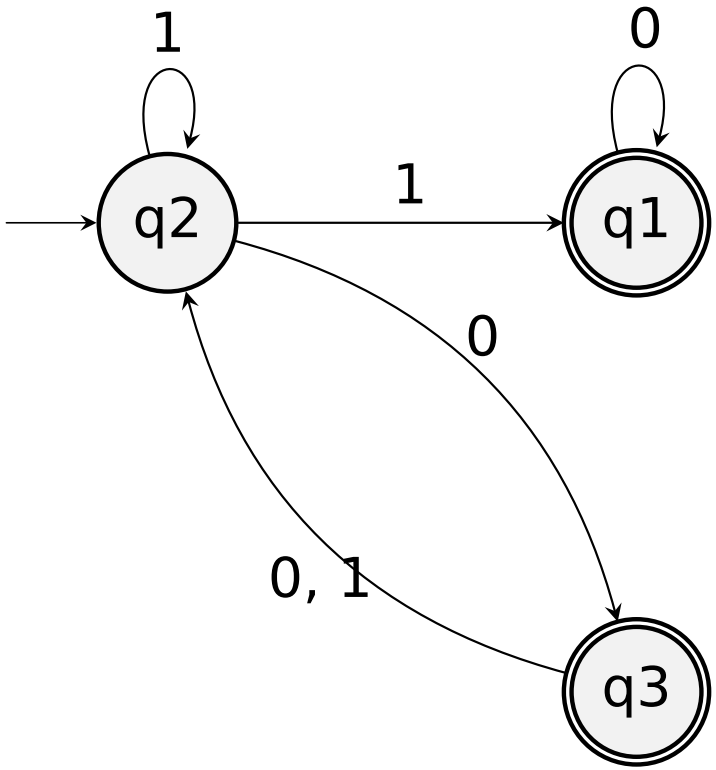
http://ivanzuzak.info/noam/webapps/fsm_simulator/

- `Input automaton` ⇒ `Generate random NFA` ⇒ `Create automaton`
- taper une chaine dans la zone de droite
- exécuter pas à pas la reconnaissance.
- états allumés = tous les états dans lesquels ont peut se trouver
- tous les état éteints = pas d'état atteignable
  transition impossible, mot non reconnu
- Quand le mot est consommé : Y a-t-il un état final allumé ?
  Oui : OK. Non : mot non reconnu.

Définition

$A = ($, , , , $)$

$A = ($, , , , $)$

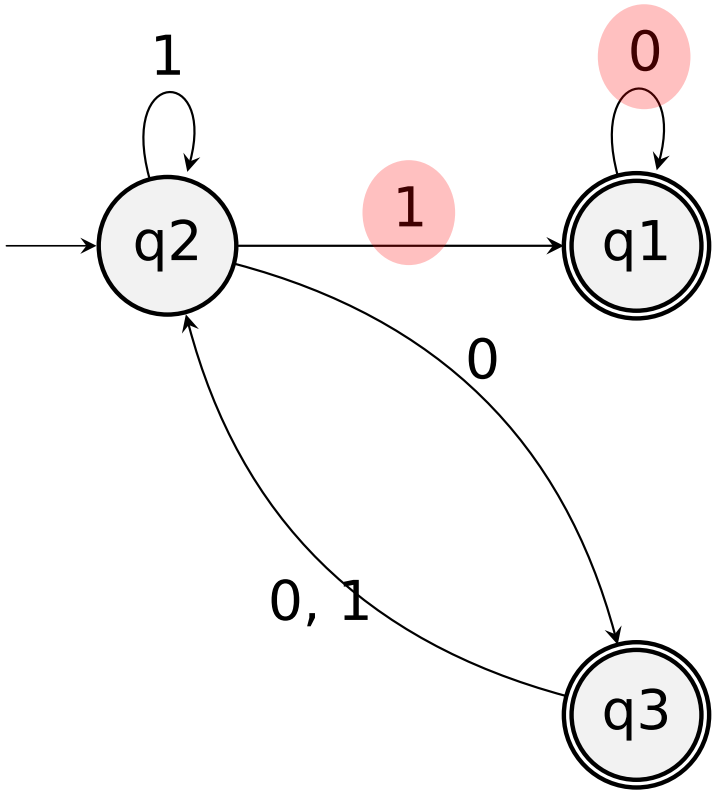


Définition

$A = (\quad \Sigma \quad , \quad , \quad , \quad , \quad)$

$A = (\quad \{0, 1\} \quad , \quad , \quad , \quad , \quad)$

alphabet
(ensemble)

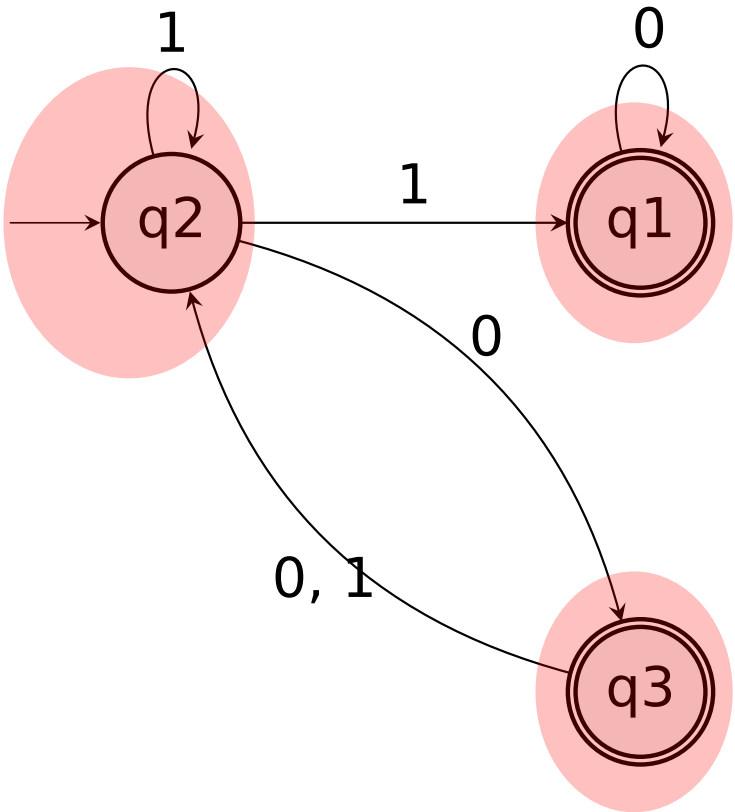


Définition

$$A = (\quad \Sigma \quad , \quad Q \quad , \quad , \quad , \quad)$$

$$A = (\quad \{0,1\} \quad , \{q2, q3, q1\} , \quad , \quad , \quad)$$

alphabet (ensemble) — états ensemble

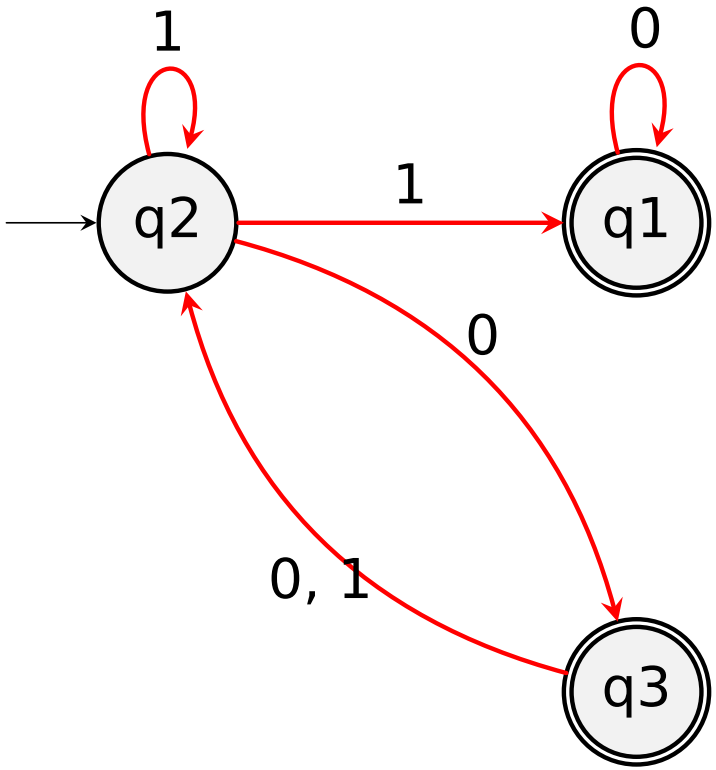


Définition

$A = ($ Σ , Q , δ , , $)$

$A = ($ $\{0, 1\}$, $\{q2, q3, q1\}$, $\ldots$, , $)$

alphabet (ensemble) — états ensemble — transitions fonction



$\delta = \{ \; (q1, 0, q1);$
$(q2, 1, q1);$
$(q2, 1, q2);$
$(q2, 0, q3);$
$(q3, 0, 1, q2) \; \}$

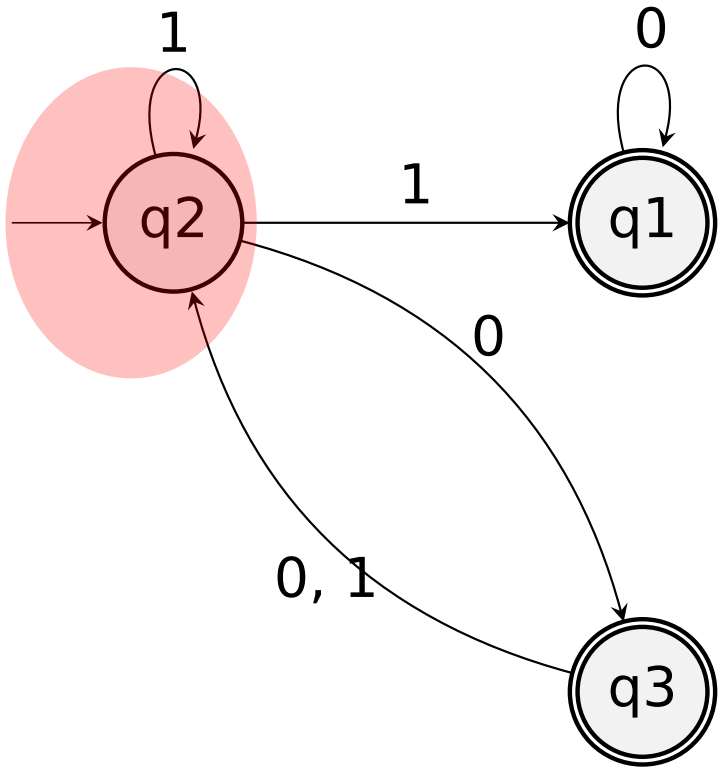
Définition

le cnam

$$A = (\ \Sigma\ ,\ Q\ ,\ \delta\ ,\ I\ ,\ )$$

$$A = (\ \{0,1\}\ ,\ \{q2, q3, q1\}\ ,\ \ldots\ ,\ q2\ ,\ )$$

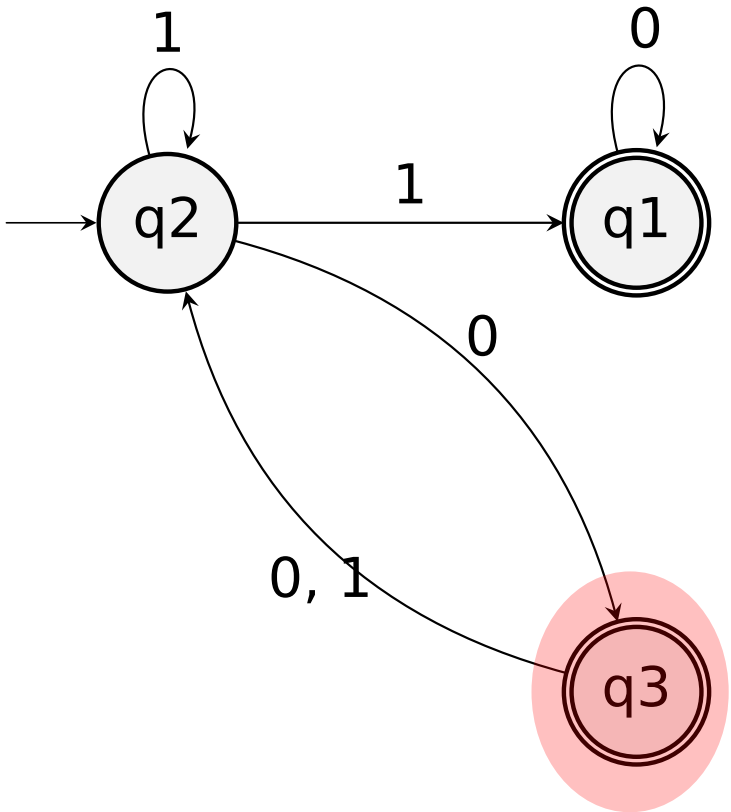
alphabet	états	transitions	ét. initial
(ensemble)	ensemble	fonction	singleton



Définition

$$A = (\ \Sigma\ ,\ Q\ ,\ \delta\ ,\ I\ ,\ F)$$
$$A = (\ \{0,1\}\ ,\ \{q2, q3, q1\}\ ,\ \ldots\ ,\ q2\ ,\ \{q1, q3\})$$

alphabet	états	transitions	ét. initial	ét. finaux
(ensemble)	ensemble	fonction	singleton	ensemble



Définition

Définition (Mot)

Un mot m sur un alphabet $\mathcal{A}$ est une séquence finie (éventuellement vide) d'éléments de $\mathcal{A}$.

Exemple : alphabet $\mathcal{A} = \{a, b, c\}$, trois mots : $aabbc$, ϵ, aaa

Définition (Langage)

Un langage $\mathcal{L}$ sur un alphabet $\mathcal{A}$ est un ensemble de mots sur $\mathcal{A}$.

Exemple : alphabet $\mathcal{A} = \{a, b, c\}$, deux langages : $\mathcal{L}_1 = \{ab, ba, abc\}$, $\mathcal{L}_2 = \{\epsilon, a, aa, aaa \ldots\}$

Notation

Langage accepté par un automate A : $L(A)$ ou $\mathcal{L}(A)$.

Définition

mot vide

Définition (Mot)

Un mot m sur un alphabet $\mathcal{A}$ est une séquence finie (éventuellement vide) d'éléments de $\mathcal{A}$.

Exemple : alphabet $\mathcal{A} = \{a, b, c\}$, trois mots : $aabbc$, ϵ, aaa

Définition (Langage)

Un langage $\mathcal{L}$ sur un alphabet $\mathcal{A}$ est un ensemble de mots sur $\mathcal{A}$.

Exemple : alphabet $\mathcal{A} = \{a, b, c\}$, deux langages : $\mathcal{L}_1 = \{ab, ba, abc\}$, $\mathcal{L}_2 = \{\epsilon, a, aa, aaa \ldots\}$

Notation

Langage accepté par un automate A : $L(A)$ ou $\mathcal{L}(A)$.

Définition

Définition (Langage régulier)

Un langage $\mathcal{L}$ est régulier ssi **il existe** un automate (fini) A tel $L(A) = \mathcal{L}$.

Définition (Automate déterministe)

Un automate A est déterministe si il n'existe pas d'état ayant deux transitions sortantes avec le même caractère.

$$\nexists (q, \alpha, q'), (q, \alpha, q'') \in \delta \wedge q' \neq q''.$$

Définition

Définition (Automates équivalents)

Automates A et B équivalents ssi $L(A) = L(B)$ (même langage reconnu)

Théorème (Déterminisation)

Pour tout automate fini A il existe une automate ***déterministe*** *équivalent A'.*

Les langages réguliers sont réguliers

le cnam

- $\emptyset$ est régulier
- $\{\epsilon\}$ est régulier
- $\{a\}$ est régulier ($a \in \Sigma$)
- Étant donné L_1 et L_2 deux langages réguliers (A_1 et A_2) :
  - $L_1 \cup L_2$ est un langage régulier
  - $L_1 \cap L_2$ est un langage régulier
  - $L_1 \setminus L_2$ est un langage régulier
  - $L_1 L_2$ est un langage régulier
  - L_1^* est un langage régulier
  - $\overline{L_1}$ est un langage régulier

(Rappel $A = (\Sigma, Q, \delta, I, F)$)

Les langages réguliers sont réguliers

le cnam

- $\emptyset$ est régulier
- $\{\epsilon\}$ est régulier
- $\{a\}$ est régulier ($a \in \Sigma$)
- Étant donné L_1 et L_2 deux langages réguliers (A_1 et A_2) :
  - $L_1 \cup L_2$ est un langage régulier
  - $L_1 \cap L_2$ est un langage régulier
  - $L_1 \setminus L_2$ est un langage régulier
  - $L_1 L_2$ est un langage régulier
  - L_1^* est un langage régulier
  - $\overline{L_1}$ est un langage régulier

Union ensembliste

(Rappel $A = (\Sigma, Q, \delta, I, F)$)

Les langages réguliers sont réguliers

le cnam

- $\emptyset$ est régulier
- $\{\epsilon\}$ est régulier
- $\{a\}$ est régulier ($a \in \Sigma$)
- Étant donné L_1 et L_2 deux langages réguliers (A_1 et A_2) :
  - $L_1 \cup L_2$ est un langage régulier
  - $L_1 \cap L_2$ est un langage régulier
  - $L_1 \setminus L_2$ est un langage régulier
  - $L_1 L_2$ est un langage régulier
  - L_1^* est un langage régulier
  - $\overline{L_1}$ est un langage régulier

Intersection ensembliste

(Rappel $A = (\Sigma, Q, \delta, I, F)$)

Les langages réguliers sont réguliers

le cnam

- $\emptyset$ est régulier
- $\{\epsilon\}$ est régulier
- $\{a\}$ est régulier ($a \in \Sigma$)
- Étant donné L_1 et L_2 deux langages réguliers (A_1 et A_2) :
  - $L_1 \cup L_2$ est un langage régulier
  - $L_1 \cap L_2$ est un langage régulier
  - $L_1 \setminus L_2$ est un langage régulier
  - $L_1 L_2$ est un langage régulier
  - L_1^* est un langage régulier
  - $\overline{L_1}$ est un langage régulier

Différence ensembliste

(Rappel $A = (\Sigma, Q, \delta, I, F)$)

Les langages réguliers sont réguliers

le cnam

- $\emptyset$ est régulier
- $\{\epsilon\}$ est régulier
- $\{a\}$ est régulier ($a \in \Sigma$)
- Étant donné L_1 et L_2 deux langages réguliers (A_1 et A_2) :
  - $L_1 \cup L_2$ est un langage régulier
  - $L_1 \cap L_2$ est un langage régulier
  - $L_1 \setminus L_2$ est un langage régulier
  - $L_1 L_2$ est un langage régulier
  - L_1^* est un langage régulier
  - $\overline{L_1}$ est un langage régulier

concaténation de langage

(Rappel $A = (\Sigma, Q, \delta, I, F)$)

Les langages réguliers sont réguliers

le cnam

- $\emptyset$ est régulier
- $\{\epsilon\}$ est régulier
- $\{a\}$ est régulier ($a \in \Sigma$)
- Étant donné L_1 et L_2 deux langages réguliers (A_1 et A_2) :
  - $L_1 \cup L_2$ est un langage régulier
  - $L_1 \cap L_2$ est un langage régulier
  - $L_1 \setminus L_2$ est un langage régulier
  - $L_1 L_2$ est un langage régulier
  - L_1^* est un langage régulier
  - $\overline{L_1}$ est un langage régulier

Clôture sous concaténation

(Rappel $A = (\Sigma, Q, \delta, I, F)$)

Les langages réguliers sont réguliers

le cnam

- $\emptyset$ est régulier
- $\{\epsilon\}$ est régulier
- $\{a\}$ est régulier ($a \in \Sigma$)
- Étant donné L_1 et L_2 deux langages réguliers (A_1 et A_2) :
  - $L_1 \cup L_2$ est un langage régulier
  - $L_1 \cap L_2$ est un langage régulier
  - $L_1 \setminus L_2$ est un langage régulier
  - $L_1 L_2$ est un langage régulier
  - L_1^* est un langage régulier
  - $\overline{L_1}$ est un langage régulier

Complémentaire d'un langage

(Rappel $A = (\Sigma, Q, \delta, I, F)$)

Concaténation

Notation

On note $m_1 m_2$ le mot composé des lettre du mot m_1 suivi des lettres du mot m_2.

But :

Étant donnés deux langages réguliers L_1 et L_2, on construit le langage suivant :

$$\{m = m_1 m_2 \text{ t.q. } m_1 \in L_1 \wedge m_2 \in L_2\}.$$

Solution

« Brancher » les états sortant d'automate de L_1 sur les états entrant de L_2.

Clôture sous concaténation

le cnam

But

Étant donnés un langage régulier L, on construit le langage suivant :

$$\{\exists p \in \mathbb{N}, m = m_1 m_2 \cdots_p \text{t.q.} m_i \in L\}.$$

Exemple : $L = \{aa, bb\}$.
$L^* = \{\epsilon, aa, bb, aaaa, aabb, bbbb, aaaaaa, \ldots\}$.

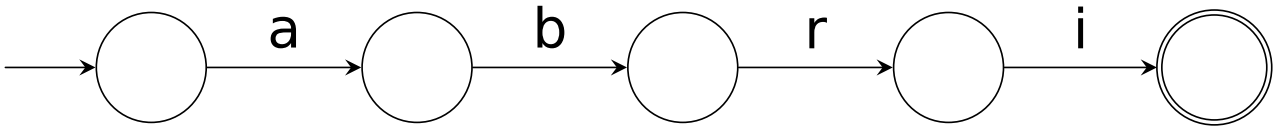
Solution

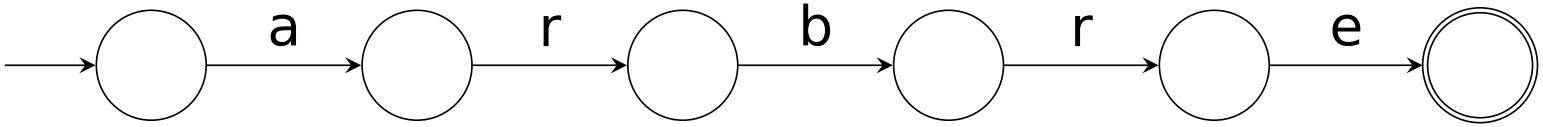
« Brancher » les états sortant à l'état initial.

Exemples d'opérations

le cnam

Un automate par mot du dictionnaire



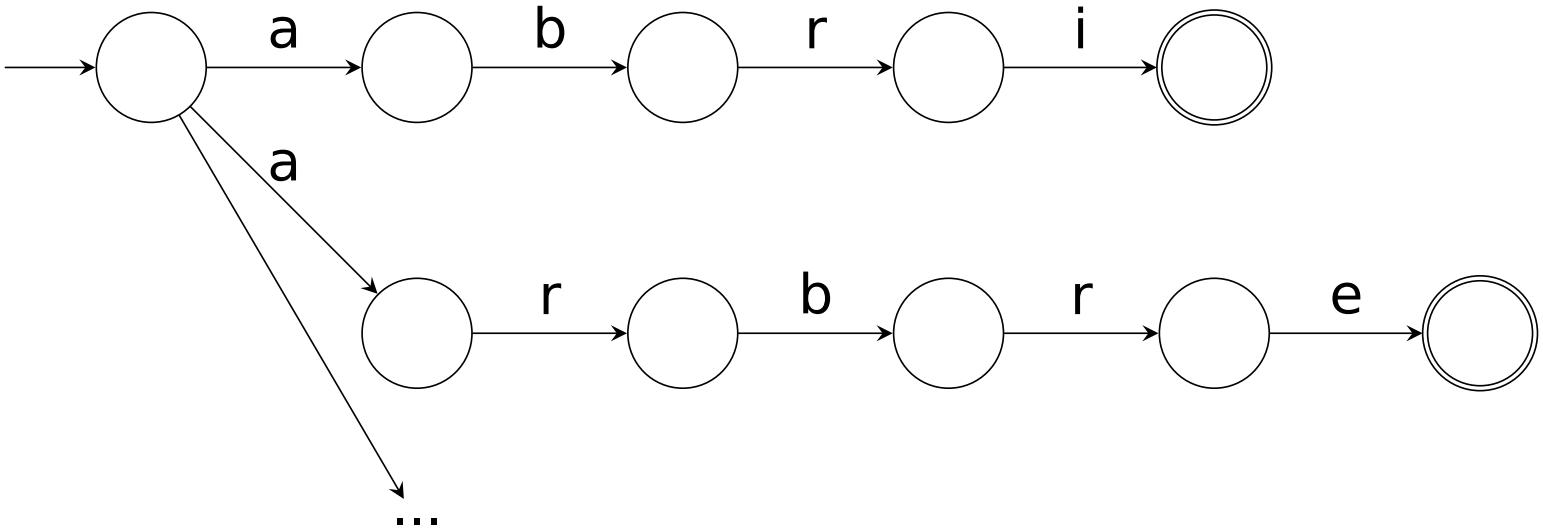


...

Exemples d'opérations

le cnam

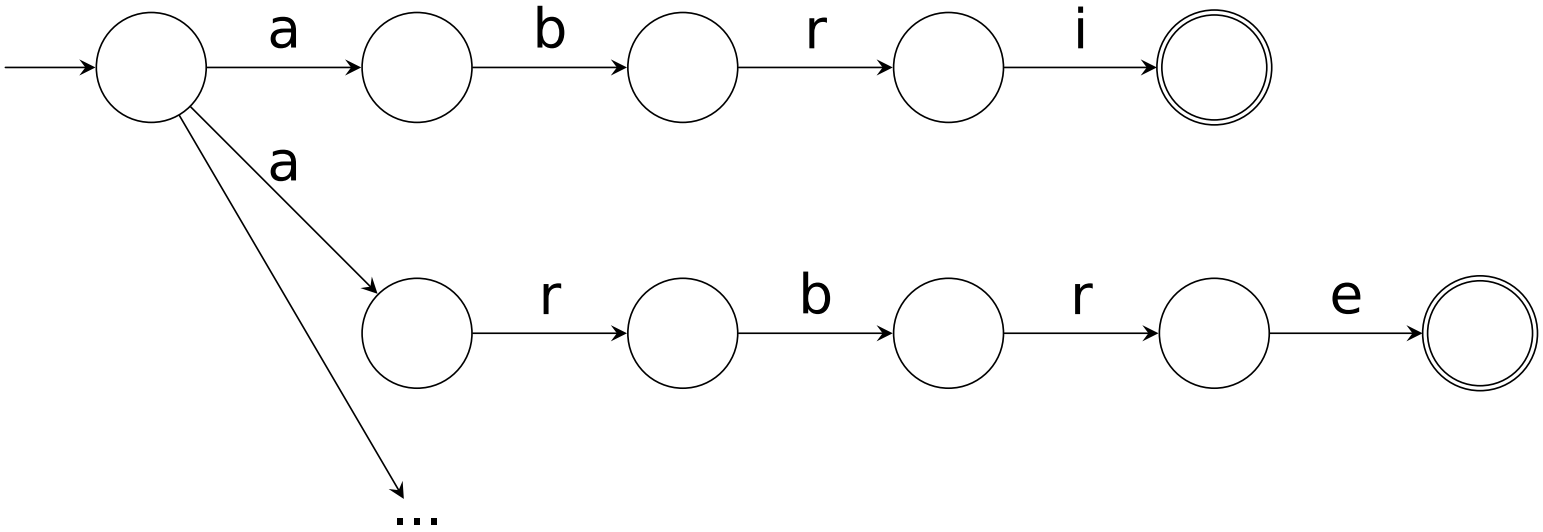
Un seul automate : Union

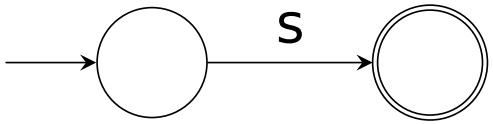


Exemples d'opérations

le cnam

Ajout du pluriel :

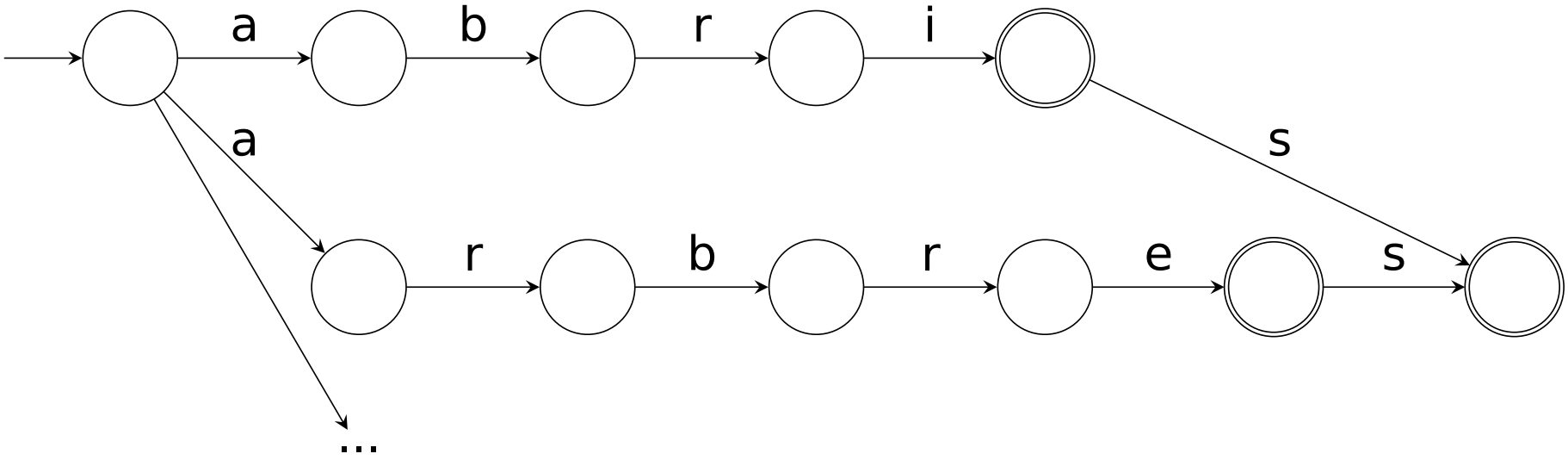


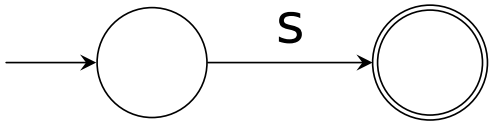


Exemples d'opérations

le cnam

Ajout du pluriel : concaténation





Expression régulière

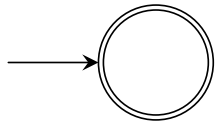
le cnam

- Une autre façon de définir un langage
- Basé sur les opérations ensemblistes :
- $\emptyset$ représente l'ensemble vide
- ϵ représente$\{\epsilon\}$
- a (pour $a \in \Sigma$) représente $\{a\}$
- si w_1 et w_2 sont deux expressions régulières dénotant L_1 et L_2 alors :
  - $w_1 w_2$ est une expression régulière et représente l'ensemble $L_1 L_2$
  - $w_1 | w_2$ est une expression régulière et représente l'ensemble $L_1 \cup L_2$
  - w_1^* est une expression régulière et représente l'ensemble L_1^*

Reconstruire l'automate

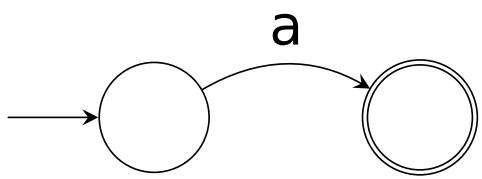
le cnam

ϵ



Reconstruire l’automate

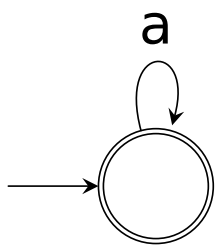
a



Reconstruire l'automate

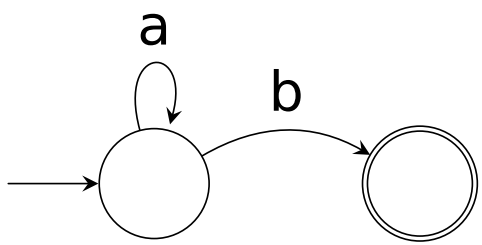
le cnam

$a*$



Reconstruire l'automate

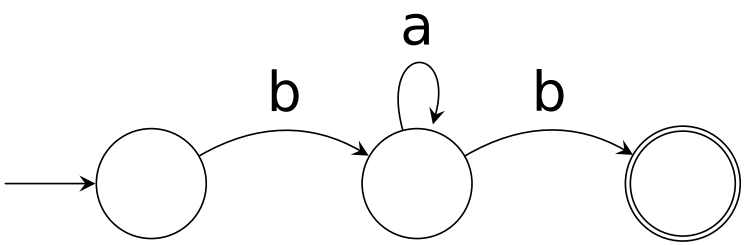
$a^* b$



Reconstruire l'automate

le cnam

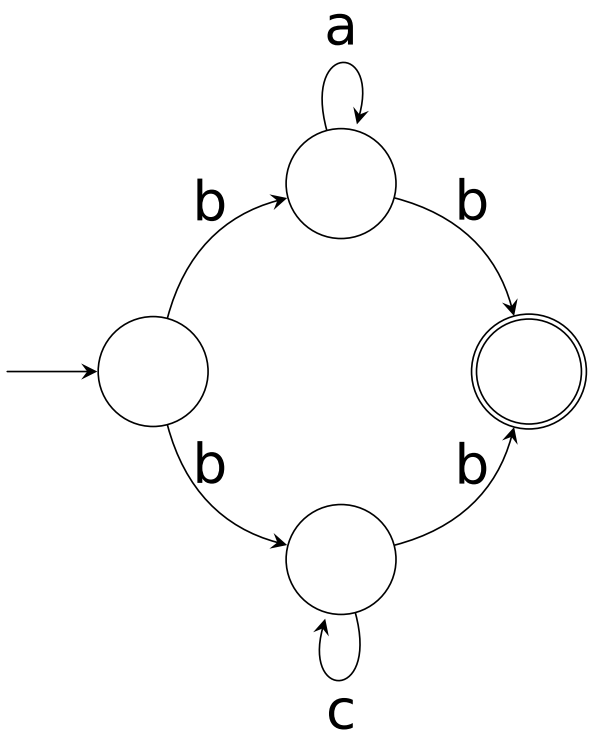
$$ba^*b$$



Reconstruire l'automate

le cnam

$$b(a^*|c^*)b$$



Reconstruire l'automate

le cnam

$$b(a^*|c)b$$