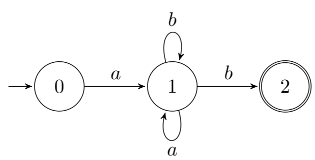
- Un **automate** est un 5-uplet $A = (\Sigma, Q, I, F, E)$ où :
  - Σ est un alphabet.
  - Q est un ensemble fini d'**états**.
  - $I \in Q$ est un ensemble d'**états initiaux**.
  - $F \subseteq Q$ est un ensemble d'**états acceptants** (ou **finaux**)
  - $E \subseteq Q \times \Sigma \times Q$ est un ensemble de **transitions**.
    On peut remplacer l'ensemble E de transitions par une **fonction de transition** $\delta : Q \times \Sigma \longrightarrow \mathcal{P}(Q)$
- - Un chemin dans A est **acceptant** s'il part d'un état initial pour aller dans un état final.
  - Un mot est **accepté** par A s'il est l'étiquette d'un chemin acceptant.
  - Le **langage** $L(A)$ **accepté** (ou **reconnu**) par A est l'ensemble des mots acceptés par A.

  Exemple : le langage $a(a+b)^*b$ est reconnaissable, car reconnu par l'automate ci-dessous.

- Pour déterminer algorithmiquement si un automate A accepte un mot $u = u_1 ... u_n$, on peut calculer de proche en proche $Q_0 = I$, $Q_1 =$ états accessibles depuis Q_0 avec la lettre u_1, $Q_2 =$ états accessibles depuis Q_0 avec la lettre u_2... et regarder si Q_n contient un état final.

- Soit $A = (\Sigma, Q, I, F, E)$ un automate.
  - A est **complet** si : $\forall q \in Q,\ \forall a \in \Sigma,\ \exists (q, a, q') \in E$
  - Un automate $(\Sigma, Q, \{q_i\}, F, E)$ est **déterministe** si :
    1. Il n'y a qu'un seul état initial q_i.
    2. $(q, a, q_1) \in E \wedge (q, a, q_2) \in E \implies q_1 = q_2$: il y a au plus une transition possible en lisant une lettre depuis un état
  - Un automate déterministe et complet possède une unique transition possible depuis un état en lisant une lettre.
    On a alors $\delta : Q \times \Sigma \longrightarrow Q$ qu'on peut étendre en $\delta^* : Q \times \Sigma^* \longrightarrow Q$ définie par :
    * $\delta^*(q, \varepsilon) = q$
    * Si $u = av$, $\delta^*(q, av) = \delta^*(\delta(q, a), v)$

    On a alors $u \in L(A) \iff \delta^*(q_i, u) \in F$.

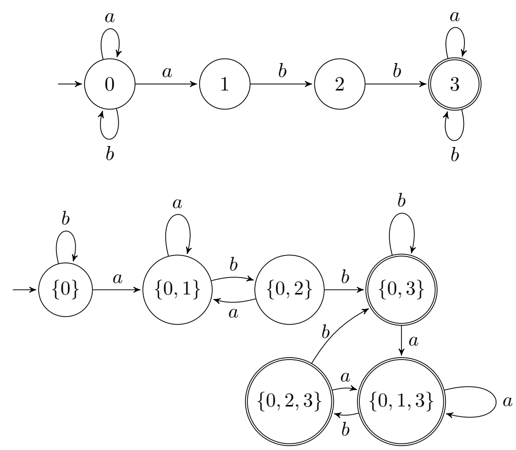
- Deux automates sont **équivalents** s'ils ont le même langage.

- Soit A un automate. Alors A est équivalent à un automate déterministe complet.

  Preuve : Utilise l'automate des parties $A' = (\Sigma, \mathcal{P}(Q), \{I\}, F', \delta')$ où $F' = \{X \subseteq Q \mid X \cap F \neq \emptyset\}$

  Remarque : Si on veut juste un automate complet (pas forcément déterministe), on peut ajouter un état poubelle vers lequel on redirige toutes les transitions manquantes. Dans l'automate des parties, cet état poubelle est $\emptyset$.

Exemple : Un automate A avec son déterminisé A'.



- Soit L un langage reconnaissable. Alors $\overline{L}$ $(= \Sigma^* \backslash L)$ est reconnaissable.

  Preuve : Soit $A = (\Sigma, Q, q_i, F, \delta)$ un automate déterministe complet reconnaissant L. Alors $A' = (\Sigma, Q, q_i, Q \backslash F, \delta)$ (on inverse états finaux et non-finaux) reconnaît $\overline{L}$.

- Soient L_1 et L_2 des langages reconnaissables. Alors :
  - $L_1 \cap L_2$ est reconnaissable.
  - $L_1 \cup L_2$ est reconnaissable.
  - $L_1 \backslash L_2$ est reconnaissable.

  Preuve : Soient $A_1 = (Q_1, q_1, F_1, \delta_1)$ et $A_2 = (Q_2, q_2, F_2, \delta_2)$ des automates finis déterministes complets reconnaissants L_1 et L_2. Soit $A = (Q_1 \times Q_2, (q_1, q_2), F, \delta)$ (**automate produit**) où :
  - $F = F_1 \times F_2$: A reconnait $L_1 \cap L_2$.
  - $F = \{(q_1, q_2) \mid q_1 \in F_1 \text{ ou } q_2 \in F_2\}$: A reconnait $L_1 \cup L_2$.
  - $F = \{(q_1, q_2) \mid q_1 \in F_1 \text{ et } q_2 \notin F_2\}$: A reconnait $L_1 \setminus L_2$.

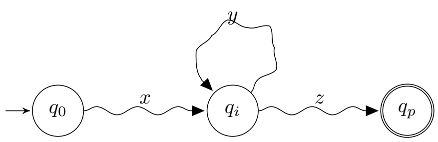
Remarque : Comme l'ensemble des langages reconnaissables est égal à l'ensemble des langages rationnels, l'ensemble des langages rationnels est aussi stable par complémentaire, intersection et différence.
Il n'y a pas de stabilité par inclusion (L rationnel et $L' \subseteq L$ n'implique pas forcément L' rationnel).

- (**Lemme de l'étoile**) Soit L un langage reconnaissable par un automate à n états.
  Si $u \in L$ et $|u| \geq n$ alors il existe des mots x, y, z tels que :
  - $u = xyz$
  - $|xy| \leq n$
  - $y \neq \varepsilon$
  - $xy^*z \subseteq L$ (c'est-à-dire : $\forall k \in \mathbb{N},\ xy^kz \in L$)

  Preuve : Soit $A = (\Sigma, Q, I, F, \delta)$ un automate reconnaissant L et $n = |Q|$.
  Soit $u \in L$ tel que $|u| \geq n$.
  u est donc l'étiquette d'un chemin acceptant C :

  $$q_0 \in I \xrightarrow{u_0} q_1 \xrightarrow{u_1} ... \xrightarrow{u_{p-1}} q_p \in F$$

  C a $p + 1 > n$ sommets donc passe deux fois par un même état $q_i = q_j$ avec $i < n$. La partie de C entre q_i et q_j forme donc un cycle.

  Soit $x = u_0u_1...u_{i-1}$, $y = u_i...u_j$ et $z = u_{j+1}...u_{p-1}$.
  xy^kz est l'étiquette du chemin acceptant obtenu à partir de C en passant k fois dans le cycle.

Application : $L_1 = \{a^nb^n \mid n \in \mathbb{N}\}$ n'est pas reconnaissable.
Preuve : Supposons L_1 reconnaissable par un automate à n états. Soit $u = a^nb^n$. Clairement, $u \in L_1$ et $|u| \geq n$.
D'après le lemme de l'étoile : il existe x, y, z tels que $u = xyz$, $|xy| \leq n$, $y \neq \varepsilon$ et $xy^*z \subseteq L$.
Comme $|xy| \leq n$, x et y ne contiennent que des a : $x = a^i$ et $y = a^j$. Comme $y \neq \varepsilon$, $j > 0$.
En prenant $k = 0$: $xy^0z = xz = a^{n-j}b^n$.
Or $j > 0$ donc $a^{n-j}b^n \notin L_1$: absurde.