Automate de Glushkov : rationnel $\implies$ reconnaissable

- Une expression rationnelle est **linéaire** si chaque lettre y apparaît au plus une fois : $a(d+c)^*b$ est linéaire mais pas $ac(a+b)$.

- Soit L un langage. On définit :
  - $P(L) = \{a \in \Sigma \mid a\Sigma^* \cap L \neq \emptyset\}$ (premières lettres des mots de L)
  - $S(L) = \{a \in \Sigma \mid \Sigma^* a \cap L \neq \emptyset\}$ (dernières lettres des mots de L)
  - $F(L) = \{u \in \Sigma^2 \mid \Sigma^* u \Sigma^* \cap L \neq \emptyset\}$ (facteurs de longueur 2 des mots de L)
  - L est **local** si, pour tout mot $u = u_1 u_2 ... u_n \neq \varepsilon$:

    $$u \in L \iff u_1 \in P(L) \land u_n \in S(L) \land \forall k, u_k u_{k+1} \in F(L)$$

    Il suffit donc de regarder la première lettre lettre, la dernière lettre et les facteurs de taille 2 pour savoir si un mot appartient à un langage local.

    Remarques :
    - $\implies$ est toujours vrai donc il suffit de prouver $\impliedby$.
    - Définition équivalente :

      $$L \text{ local} \iff L \setminus \{\varepsilon\} = (P(L) \cap S(L)) \setminus N(L)$$

      où $N(L) = \Sigma^2 \setminus F(L)$.

    Exemples :
    - Si $L_2 = (ab)^*$ alors $P(L_2) = \{a\}$, $S(L_2) = \{b\}$ et $F(L_2) = \{ab, ba\}$. De plus si $u = u_1 u_2 ... u_n \neq \varepsilon$ avec $u_1 \in P(L), u_n \in S(L)$, et $\forall k, u_k u_{k+1} \in F(L)$ alors $u_1 = a$, $u_n = b$ et on montre (par récurrence) que $u = abab...ab \in {}_2$. Donc L_2 est local.
    - Si $L_3 = a^* + (ab)^*$ alors $P(L_3) = \{a\}$, $S(L_3) = \{a, b\}$, $F(L_3) = \{aa, ab, ba\}$. Soit $u = aab$. La première lettre de u est dans $P(L_3)$, la dernière dans $S(L_3)$ et les facteurs de u sont aa et ba qui appartiennent à $F(L_3)$. Mais $u \notin L_3$, ce qui montre que L_3 n'est pas local.

- Un automate déterministe (Σ, Q, q_0, F, E) est **local** si toutes les transitions étiquetées par une même lettre aboutissent au même état : $(q_1, a, q_2) \in E \land (q_3, a, q_4) \in E \implies q_2 = q_4$

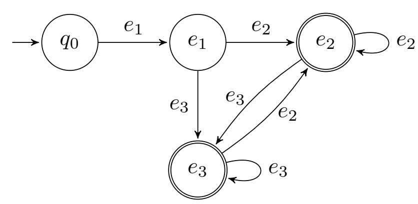
- Un langage local L est reconnu par un automate local.

  Preuve : L est reconnu par (Σ, Q, q_0, F, E) où :
  - $Q = \Sigma \cup \{q_0\}$: un état correspond à la dernière lettre lue
  - $F = S(L)$ si $\varepsilon \notin L$, sinon $F = S(L) \cup \{q_0\}$.
  - $E = \{(q_0, a, a) \mid a \in P(L)\} \cup \{(a, b, b) \mid ab \in F(L)\}$

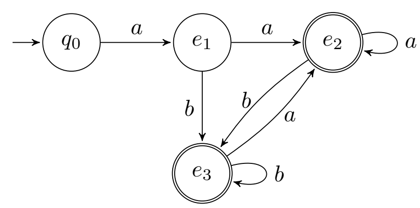
- L'**algorithme de Berry-Sethi** permet de construire un automate à partir d'une expression rationnelle e.

Exemple avec $e = a(a+b)^*$:

1. On linéarise e en e', en remplaçant chaque occurrence de lettre dans e par une nouvelle lettre : $e' = e_1(e_2 + e_3)^*$
2. On peut montrer que $L(e')$ est un langage local.
3. Un langage local est reconnu par l'automate local $A = (\Sigma, Q, q_0, F, E)$



4. On fait le remplacement inverse de 1. sur les transitions de A pour obtenir un automate reconnaissant $L(e)$:



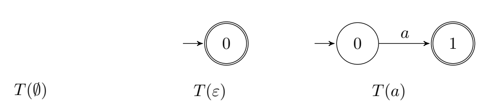
Automate de Thompson : rationnel $\implies$ reconnaissable

- Une ε-transition est une transition étiquetée par ε.

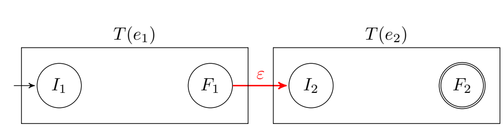
- Un automate avec ε-transitions est équivalent à un automate sans ε-transitions.

  Preuve : Si $A = (\Sigma, Q, I, F, \delta)$ est un automate avec ε-transitions, on définit $A' = (\Sigma, Q, I', F, \delta')$ où :
  - I' est l'ensemble des états atteignables depuis un état de I en utilisant uniquement des ε-transitions.
  - $\delta'(q, a)$ est l'ensemble des états q' tel qu'il existe un chemin de q à q' dans A étiqueté par un a et un nombre quelconque de ε (ce qui peut être trouvé par un parcours de graphe).

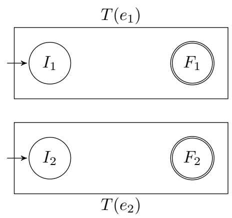
- L'**automate de Thompson** est construit récursivement à partir d'une expression rationnelle e :
  - Cas de base :



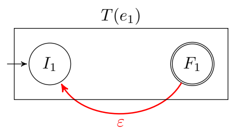
  - $T(e_1 e_2)$: ajout d'une ε-transition depuis chaque état final de $T(e_1)$ vers chaque état initial de $T(e_2)$.



  - $T(e_1 | e_2)$: union des états initiaux et des états finaux.



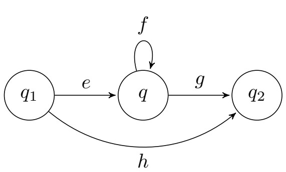
– $T(e_1^*)$: ajout d'une ε-transition depuis chaque état final vers chaque état initial.



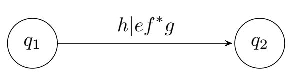
Élimination des états : reconnaissable $\implies$ rationnel

- Tout automate est équivalent à un automate avec un unique état initial sans transition entrante et un unique état final sans transition sortante.
  Preuve : On ajoute un état initial q_i et un état final q_f et des transitions ε depuis q_i vers les états initiaux et depuis les états finaux vers q_f.

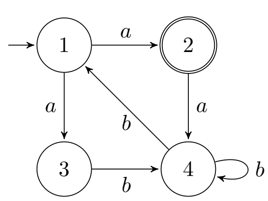
- **Méthode d'élimination des états** : On considère un automate A comme dans le point précédent. Tant que A possède au moins 3 états, on choisit un état $q \notin \{q_i, q_f\}$ et on supprime q en modifiant les transitions :



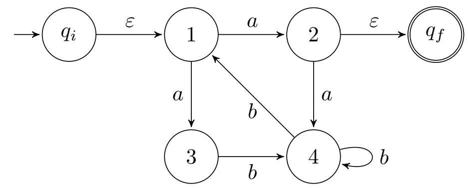
en



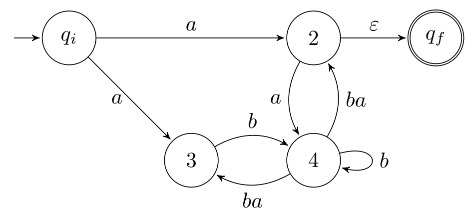
Exemple :



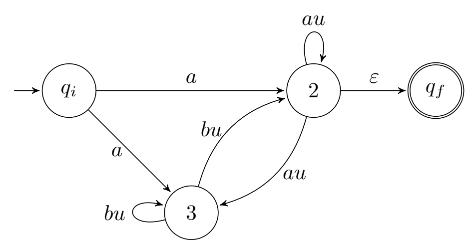
1. On commence par se ramener à un automate avec un état initial sans transition entrante et un état final sans transition sortante :



2. Suppression de l'état 1 :

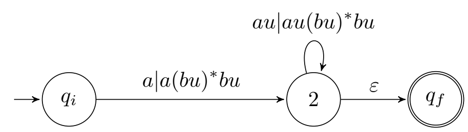


3. Suppression de l'état 4 :

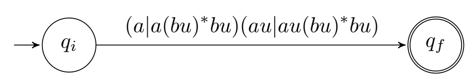


Avec $u = b^*ba$.

4. Suppression de l'état 3 :



5. Suppression de l'état 2 :



On obtient l'expression rationnelle $a|a(bu)^*bu(au|au(bu)^*bu)$, où $u = b^*ba$.

Rationnel $\iff$ reconnaissable

- **Théorème de Kleene** : un langage est rationnel si et seulement si il est reconnaissable par un automate.
- Les théorèmes sur les automates s'appliquent aussi aux langages rationnels, et inversement. Notamment, les langages rationnels sont stables par union, concaténation, étoile, intersection, complémentaire, différence.