# Théorie des langages

## Automates à pile et langages algébriques

Jérôme Delobelle
jerome.delobelle@u-paris.fr

LIPADE - Université de Paris

# Automates à pile et langages algébriques

# Automates à pile et langages algébriques (hors-contextes)

**Théorème**

Un langage est algébrique si et seulement si il est reconnu par un automate à pile

# Automates à pile et langages algébriques (hors-contextes)

**Théorème**

Un langage est algébrique si et seulement si il est reconnu par un automate à pile

**Théorème**

Tout langage algébrique n'est pas reconnu par un automate à pile déterministe

# Automates à pile et langages algébriques (hors-contextes)

**Théorème**

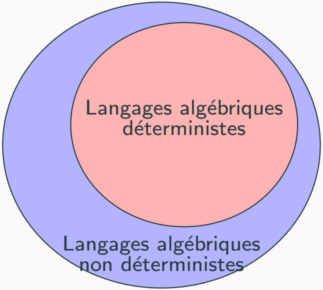
Un langage est algébrique si et seulement si il est reconnu par un automate à pile

**Théorème**

Tout langage algébrique n'est pas reconnu par un automate à pile déterministe

Il existe des langages algébriques qui sont reconnus *uniquement* par des automates à pile non déterministes

# Automates à pile et langages algébriques (hors-contextes)



# Langage algébrique déterministe

**Langage algébrique déterministe**

Un langage algébrique $L$ est déterministe s'il existe un automate à pile $M$ acceptant par état final déterministe tel que $L^F(M) = L$

# Langage algébrique déterministe

**Langage algébrique déterministe**

Un langage algébrique $L$ est déterministe s'il existe un automate à pile $M$ acceptant par état final déterministe tel que $L^F(M) = L$

- Par exemple, $\{m \in (a+b)^* | m \text{ est un palindrome}\}$ est un langage algébrique non déterministe.
  - $\rightarrow$ Intuitivement, on ne sait pas deviner où est le milieu du mot, et on ne peut donc pas écrire un automate à pile déterministe qui nous dira quand on doit arrêter d'empiler et commencer à dépiler.

# Langage algébrique déterministe

**Langage algébrique déterministe**

Un langage algébrique $L$ est déterministe s'il existe un automate à pile $M$ acceptant par état final déterministe tel que $L^F(M) = L$

- Par exemple, $\{m \in (a+b)^* | m$ est un palindrome$\}$ est un langage algébrique non déterministe.
  - $\rightarrow$ Intuitivement, on ne sait pas deviner où est le milieu du mot, et on ne peut donc pas écrire un automate à pile déterministe qui nous dira quand on doit arrêter d'empiler et commencer à dépiler.
- $\{m_1 c m_2 | m_1 m_2 \in (a+b)^*$ est un palindrome$\}$ est un langage algébrique déterministe.
  - $\rightarrow$ Intuitivement, quand on lit la lettre $c$, on sait que l'on doit commencer à dépiler.

# Automates à pile et grammaires algébriques

- Les automates à pile sont nécessaires pour reconnaître les langages algébriques
- Mais ils ne sont pas si faciles à concevoir
- Le lien entre dérivations d'une grammaire et exécution d'un automate à pile n'est pas évident à voir

? Est-il possible de dériver automatiquement un automate à pile à partir d'une grammaire algébrique ?

# Grammaire algébrique vers automate à pile

# Rappel : Forme normale de Greibach

**Forme normale de Greibach**

Une grammaire algébrique $G = \langle V, \Sigma, P, S \rangle$ est sous la forme normale de Greibach si toute production est de la forme :

$$\begin{array}{rcl} A & \rightarrow & aA_1 \ldots A_n \\ A & \rightarrow & a \end{array}$$

avec $A, A_i \in V \setminus \Sigma$, et $a \in \Sigma$

# Grammaire algébrique vers automate à pile

- On présente un algorithme de passage d'une grammaire algébrique sous forme normale de Greibach vers un automate à pile avec reconnaissance par pile vide non déterministe
- Idée :
  - Empiler l'axiome
  - A chaque symbole lu de la chaîne d'entrée, remplacer la partie gauche de la production concernée par le reste de la partie droite

# Grammaire algébrique vers automate à pile

## Grammaire algébrique vers automate à pile

Soit $L = L \setminus \epsilon$ le langage engendré par $G = \langle V, \Sigma, P, S \rangle$ sous forme normale de Greibach.

On construit l'automate à pile $M = (\Sigma, \Gamma, S, Q, q_0, F, \delta)$ tel que

- $\Gamma = V \setminus \Sigma$
- $Q = \{q_0\}$ *(automate à un seul état)*
- $F = \emptyset$ *(reconnaissance sur pile vide)*
- $S$ le symbole initial de la pile
- On construit $\delta$ itérativement de la façon suivante :
  1. $\delta \leftarrow \emptyset$
  2. Pour toute règle $A \rightarrow aA_1A_2 \dots A_n$,
     $\delta \leftarrow \delta \cup \{(q_0, a, A) \rightarrow (q_0, A_1A_2 \dots A_n)\}$
  2. Pour toute règle $A \rightarrow a$,
     $\delta \leftarrow \delta \cup \{(q_0, a, A) \rightarrow (q_0, \epsilon)\}$

# Grammaire algébrique vers automate à pile : exemple

Soit $G = \langle V, \Sigma, P, S \rangle$ avec

- $V = \{a, b, S, B\}$
- $\Sigma = \{a, b\}$
- $P = \{S \to aSB; S \to aB; B \to b\}$

## Grammaire algébrique vers automate à pile : exemple

Soit $G = \langle V, \Sigma, P, S \rangle$ avec

- $V = \{a, b, S, B\}$
- $\Sigma = \{a, b\}$
- $P = \{S \rightarrow aSB; S \rightarrow aB; B \rightarrow b\}$

On construit l'automate $M = (\Sigma, \Gamma, S, Q, q_0, F, \delta)$ tel que :

# Grammaire algébrique vers automate à pile : exemple

Soit $G = \langle V, \Sigma, P, S \rangle$ avec

- $V = \{a, b, S, B\}$
- $\Sigma = \{a, b\}$
- $P = \{S \rightarrow aSB; S \rightarrow aB; B \rightarrow b\}$

On construit l'automate $M = (\Sigma, \Gamma, S, Q, q_0, F, \delta)$ tel que :

- $\Gamma = \{S, B\}$
- $Q = \{q_0\}$, $F = \emptyset$
- $\delta$ contient
  - $(q_0, a, S) \rightarrow (q_0, SB)$
  - $(q_0, a, S) \rightarrow (q_0, B)$
  - $(q_0, b, B) \rightarrow (q_0, \epsilon)$

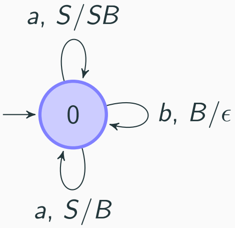
# Grammaire algébrique vers automate à pile : exemple

Soit $G = \langle V, \Sigma, P, S \rangle$ avec

- $V = \{a, b, S, B\}$
- $\Sigma = \{a, b\}$
- $P = \{S \rightarrow aSB; S \rightarrow aB; B \rightarrow b\}$

On construit l'automate $M = (\Sigma, \Gamma, S, Q, q_0, F, \delta)$ tel que :

- $\Gamma = \{S, B\}$
- $Q = \{q_0\}$, $F = \emptyset$
- $\delta$ contient
  - $(q_0, a, S) \rightarrow (q_0, SB)$
  - $(q_0, a, S) \rightarrow (q_0, B)$
  - $(q_0, b, B) \rightarrow (q_0, \epsilon)$



# Grammaire algébrique vers automate à pile : exemple

Soit $G = \langle V, \Sigma, P, S \rangle$ avec $V = \{a, +, *, (, ), S, A, B, C\}$,
$\Sigma = \{a, b, +, *, (, )\}$

$$\begin{array}{lllllll} S & \to & a & \quad & S & \to & (SC \\ S & \to & aAS & & A & \to & + \\ S & \to & (SCAS & & B & \to & * \\ S & \to & aBS & & C & \to & ) \\ S & \to & (SCBS & & & & \end{array}$$

# Grammaire algébrique vers automate à pile : exemple

Soit $G = \langle V, \Sigma, P, S \rangle$ avec $V = \{a, +, *, (, ), S, A, B, C\}$,
$\Sigma = \{a, b, +, *, (, )\}$

$$
\begin{array}{lcllcl}
S & \rightarrow & a & S & \rightarrow & (SC \\
S & \rightarrow & aAS & A & \rightarrow & + \\
S & \rightarrow & (SCAS & B & \rightarrow & * \\
S & \rightarrow & aBS & C & \rightarrow & ) \\
S & \rightarrow & (SCBS & & &
\end{array}
$$

On construit l'automate $M = (\Sigma, \Gamma, S, Q, q_0, F, \delta)$ tel que
$\Gamma = \{S, A, B, C\}$, $Q = \{q_0\}$, $F = \emptyset$ et $\delta$ contient

$$
\begin{array}{lcllcl}
(q_0, a, S) & \rightarrow & (q_0, \epsilon) & (q_0, (, S) & \rightarrow & (q_0, SC) \\
(q_0, a, S) & \rightarrow & (q_0, AS) & (q_0, +, A) & \rightarrow & (q_0, \epsilon) \\
(q_0, (, S) & \rightarrow & (q_0, SCAS) & (q_0, *, B) & \rightarrow & (q_0, \epsilon) \\
(q_0, a, S) & \rightarrow & (q_0, BS) & (q_0, ), C) & \rightarrow & (q_0, \epsilon) \\
(q_0, (, S) & \rightarrow & (q_0, SCBS) & & &
\end{array}
$$

# Automate à pile vers grammaire algébrique

# Automate à pile vers grammaire algébrique

- On présente un algorithme de passage d'un automate à pile à une grammaire algébrique
- A tout couple $(p, q)$ d'états de l'automate, et à tout $X$ de la pile, on associe un non terminal de la forme $\langle p, X, q\rangle$
- On associe toutes les lectures possibles dans l'automate pour obtenir la grammaire *sans savoir a priori lesquelles vont vider la pile*
- A la fin, on nettoie la grammaire obtenue

# Automate à pile vers grammaire algébrique : algorithme

**Algorithme : automate à pile par pile vide vers grammaire**

Soit $M = (\Sigma, \Gamma, Z_0, Q, q_0, \emptyset, \delta)$. On veut construire $G = \langle V, \Sigma, P, S\rangle$ telle que $\mathcal{L}(G) = \mathcal{L}(M)$.

1. $V \setminus \Sigma = \{\langle q, X, p\rangle | p$ et $q \in Q$, $X \in \Gamma\} \cup \{S\}$

# Automate à pile vers grammaire algébrique : algorithme

**Algorithme : automate à pile par pile vide vers grammaire**

Soit $M = (\Sigma, \Gamma, Z_0, Q, q_0, \emptyset, \delta)$. On veut construire $G = \langle V, \Sigma, P, S \rangle$ telle que $\mathcal{L}(G) = \mathcal{L}(M)$.

1. $V \setminus \Sigma = \{\langle q, X, p \rangle | p \text{ et } q \in Q,\ X \in \Gamma\} \cup \{S\}$
2. $P \leftarrow \emptyset$

# Automate à pile vers grammaire algébrique : algorithme

**Algorithme : automate à pile par pile vide vers grammaire**

Soit $M = (\Sigma, \Gamma, Z_0, Q, q_0, \emptyset, \delta)$. On veut construire $G = \langle V, \Sigma, P, S \rangle$ telle que $\mathcal{L}(G) = \mathcal{L}(M)$.

1. $V \setminus \Sigma = \{\langle q, X, p \rangle | p \text{ et } q \in Q,\ X \in \Gamma\} \cup \{S\}$
2. $P \leftarrow \emptyset$
3. Pour tout état $q \in Q$, $P \leftarrow P \cup \{S \rightarrow \langle q_0, Z_0, q \rangle\}$

# Automate à pile vers grammaire algébrique : algorithme

**Algorithme : automate à pile par pile vide vers grammaire**

Soit $M = (\Sigma, \Gamma, Z_0, Q, q_0, \emptyset, \delta)$. On veut construire $G = \langle V, \Sigma, P, S \rangle$ telle que $\mathcal{L}(G) = \mathcal{L}(M)$.

1. $V \setminus \Sigma = \{\langle q, X, p \rangle | p$ et $q \in Q$, $X \in \Gamma\} \cup \{S\}$
2. $P \leftarrow \emptyset$
3. Pour tout état $q \in Q$, $P \leftarrow P \cup \{S \rightarrow \langle q_0, Z_0, q \rangle\}$
4. Pour toute transition $(q, a, X) \rightarrow (p, \epsilon)$ de $\delta$ faire
   $P \leftarrow P \cup \{\langle q, X, p \rangle \rightarrow a\}$

# Automate à pile vers grammaire algébrique : algorithme

**Algorithme : automate à pile par pile vide vers grammaire**

Soit $M = (\Sigma, \Gamma, Z_0, Q, q_0, \emptyset, \delta)$. On veut construire $G = \langle V, \Sigma, P, S \rangle$ telle que $\mathcal{L}(G) = \mathcal{L}(M)$.

1. $V \setminus \Sigma = \{\langle q, X, p\rangle | p$ et $q \in Q$, $X \in \Gamma\} \cup \{S\}$
2. $P \leftarrow \emptyset$
3. Pour tout état $q \in Q$, $P \leftarrow P \cup \{S \rightarrow \langle q_0, Z_0, q\rangle\}$
4. Pour toute transition $(q, a, X) \rightarrow (p, \epsilon)$ de $\delta$ faire $P \leftarrow P \cup \{\langle q, X, p\rangle \rightarrow a\}$
5. Pour toute transition $(q, a, X) \rightarrow (p, B_m \ldots B_1)$ de $\delta$ faire

# Automate à pile vers grammaire algébrique : algorithme

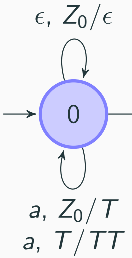
**Algorithme : automate à pile par pile vide vers grammaire**

Soit $M = (\Sigma, \Gamma, Z_0, Q, q_0, \emptyset, \delta)$. On veut construire $G = \langle V, \Sigma, P, S\rangle$ telle que $\mathcal{L}(G) = \mathcal{L}(M)$.

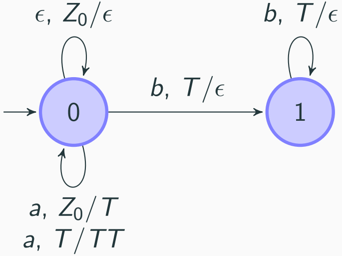
1. $V \setminus \Sigma = \{\langle q, X, p\rangle | p$ et $q \in Q$, $X \in \Gamma\} \cup \{S\}$
2. $P \leftarrow \emptyset$
3. Pour tout état $q \in Q$, $P \leftarrow P \cup \{S \to \langle q_0, Z_0, q\rangle\}$
4. Pour toute transition $(q, a, X) \to (p, \epsilon)$ de $\delta$ faire
   $P \leftarrow P \cup \{\langle q, X, p\rangle \to a\}$
5. Pour toute transition $(q, a, X) \to (p, B_m \dots B_1)$ de $\delta$ faire
   - Pour tout $m$-uplet d'états $q_1, \dots, q_m$ de $Q$ faire

     $P \leftarrow P \cup \{\langle q, X, q_m\rangle \to a\langle p, B_m, q_1\rangle\langle q_1, B_{m-1}, q_2\rangle \dots \langle q_{m-1}, B_1, q_m\rangle\}$

# Automate à pile vers grammaire algébrique : exemple



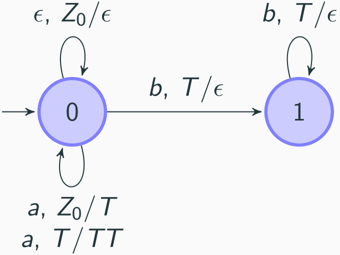
# Automate à pile vers grammaire algébrique : exemple




1. $V \setminus \Sigma = \{\langle q, X, p\rangle | p \text{ et } q \in Q,\ X \in \Gamma\} \cup \{S\}$

   $V \setminus \Sigma = \{S, \langle q_0, Z_0, q_0\rangle, \langle q_0, Z_0, q_1\rangle, \langle q_1, Z_0, q_0\rangle, \langle q_1, Z_0, q_1\rangle,$
   $\langle q_0, T, q_0\rangle, \langle q_0, T, q_1\rangle, \langle q_1, T, q_0\rangle, \langle q_1, T, q_1\rangle\}$

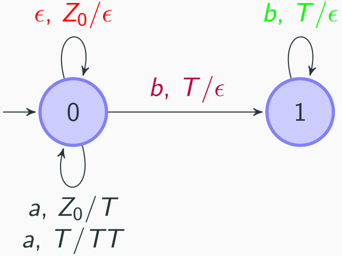
# Automate à pile vers grammaire algébrique : exemple




2. $P \leftarrow \emptyset$
3. Pour tout état $q \in Q$, $P \leftarrow P \cup \{S \to \langle q_0, Z_0, q\rangle\}$
   $P$ reçoit

$$\begin{aligned} S &\to \langle q_0, Z_0, q_0\rangle \\ S &\to \langle q_0, Z_0, q_1\rangle \end{aligned}$$

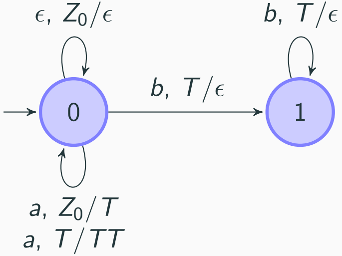
# Automate à pile vers grammaire algébrique : exemple




4. Pour toute transition $(q, a, X) \to (p, \epsilon)$ de $\delta$ faire
   $P \leftarrow P \cup \{\langle q, X, p\rangle \to a\}$
   $P$ reçoit

$$\begin{aligned}\langle q_0, Z_0, q_0\rangle &\to \epsilon\\ \langle q_0, T, q_1\rangle &\to b\\ \langle q_1, T, q_1\rangle &\to b\end{aligned}$$

# Automate à pile vers grammaire algébrique : exemple

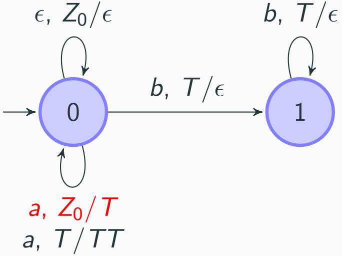



5. Pour toute transition $(q, a, X) \to (p, B_m \dots B_1)$ de $\delta$ faire

    Pour tout $m$-uplet d'états $q_1, \dots, q_m$ de $Q$ faire

    $P \leftarrow P \cup \{\langle q, X, q_m\rangle \to a\langle p, B_1, q_1\rangle\langle q_1, B_2, q_2\rangle \dots \langle q_{m-1}, B_m, q_m\rangle\}$

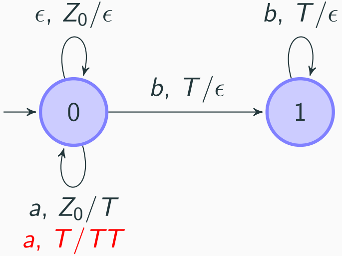
# Automate à pile vers grammaire algébrique : exemple




5. Pour toute transition $(q, a, X) \to (p, B_m \dots B_1)$ de $\delta$ faire
   Pour tout $m$-uplet d'états $q_1, \dots, q_m$ de $Q$ faire
   $P \leftarrow P \cup \{\langle q, X, q_m\rangle \to a\langle p, B_1, q_1\rangle\langle q_1, B_2, q_2\rangle \dots \langle q_{m-1}, B_m, q_m\rangle\}$

$$\begin{aligned}
\langle q_0, Z_0, q_0\rangle &\to a\langle q_0, T, q_0\rangle \\
\langle q_0, Z_0, q_1\rangle &\to a\langle q_0, T, q_1\rangle
\end{aligned}$$

# Automate à pile vers grammaire algébrique : exemple




5. Pour toute transition $(q, a, X) \to (p, B_m \dots B_1)$ de $\delta$ faire

   Pour tout $m$-uplet d'états $q_1, \dots, q_m$ de $Q$ faire

   $P \leftarrow P \cup \{\langle q, X, q_m\rangle \to a\langle p, B_1, q_1\rangle\langle q_1, B_2, q_2\rangle \dots \langle q_{m-1}, B_m, q_m\rangle\}$

$$\begin{aligned}
\langle q_0, T, q_0\rangle &\to a\langle q_0, T, q_0\rangle\langle q_0, T, q_0\rangle \\
\langle q_0, T, q_0\rangle &\to a\langle q_0, T, q_1\rangle\langle q_1, T, q_0\rangle \\
\langle q_0, T, q_1\rangle &\to a\langle q_0, T, q_0\rangle\langle q_0, T, q_1\rangle \\
\langle q_0, T, q_1\rangle &\to a\langle q_0, T, q_1\rangle\langle q_1, T, q_1\rangle
\end{aligned}$$

# Automate à pile vers grammaire algébrique : exemple

On récapitule :

- $V \setminus \Sigma = \{S, \langle q_0, Z_0, q_0\rangle, \langle q_0, Z_0, q_1\rangle, \langle q_1, Z_0, q_0\rangle, \langle q_1, Z_0, q_1\rangle, \langle q_0, T, q_0\rangle, \langle q_0, T, q_1\rangle, \langle q_1, T, q_0\rangle, \langle q_1, T, q_1\rangle\}$

$$
\begin{array}{lcl}
S & \to & \langle q_0, Z_0, q_0\rangle \\
S & \to & \langle q_0, Z_0, q_1\rangle \\
\langle q_0, Z_0, q_0\rangle & \to & \epsilon \\
\langle q_0, T, q_1\rangle & \to & b \\
\langle q_1, T, q_1\rangle & \to & b \\
\langle q_0, Z_0, q_0\rangle & \to & a\langle q_0, T, q_0\rangle \\
\langle q_0, Z_0, q_1\rangle & \to & a\langle q_0, T, q_1\rangle \\
\langle q_0, T, q_0\rangle & \to & a\langle q_0, T, q_0\rangle\langle q_0, T, q_0\rangle \\
\langle q_0, T, q_0\rangle & \to & a\langle q_0, T, q_1\rangle\langle q_1, T, q_0\rangle \\
\langle q_0, T, q_1\rangle & \to & a\langle q_0, T, q_0\rangle\langle q_0, T, q_1\rangle \\
\langle q_0, T, q_1\rangle & \to & a\langle q_0, T, q_1\rangle\langle q_1, T, q_1\rangle
\end{array}
$$

# Automate à pile vers grammaire algébrique : exemple

On renomme :

- $V \setminus \Sigma = \{S, \langle q_0, Z_0, q_0\rangle, \langle q_0, Z_0, q_1\rangle, \langle q_1, Z_0, q_0\rangle, \langle q_1, Z_0, q_1\rangle, \langle q_0, T, q_0\rangle, \langle q_0, T, q_1\rangle, \langle q_1, T, q_0\rangle, \langle q_1, T, q_1\rangle\}$

$$
\begin{array}{lcl}
S & \rightarrow & \langle q_0, Z_0, q_0\rangle \\
S & \rightarrow & \langle q_0, Z_0, q_1\rangle \\
\langle q_0, Z_0, q_0\rangle & \rightarrow & \epsilon \\
\langle q_0, T, q_1\rangle & \rightarrow & b \\
\langle q_1, T, q_1\rangle & \rightarrow & b \\
\langle q_0, Z_0, q_0\rangle & \rightarrow & a\langle q_0, T, q_0\rangle \\
\langle q_0, Z_0, q_1\rangle & \rightarrow & a\langle q_0, T, q_1\rangle \\
\langle q_0, T, q_0\rangle & \rightarrow & a\langle q_0, T, q_0\rangle\langle q_0, T, q_0\rangle \\
\langle q_0, T, q_0\rangle & \rightarrow & a\langle q_0, T, q_1\rangle\langle q_1, T, q_0\rangle \\
\langle q_0, T, q_1\rangle & \rightarrow & a\langle q_0, T, q_0\rangle\langle q_0, T, q_1\rangle \\
\langle q_0, T, q_1\rangle & \rightarrow & a\langle q_0, T, q_1\rangle\langle q_1, T, q_1\rangle
\end{array}
$$

# Automate à pile vers grammaire algébrique : exemple

On renomme :

- $V \setminus \Sigma = \{S, A, \langle q_0, Z_0, q_1\rangle, \langle q_1, Z_0, q_0\rangle, \langle q_1, Z_0, q_1\rangle, \langle q_0, T, q_0\rangle, \langle q_0, T, q_1\rangle, \langle q_1, T, q_0\rangle, \langle q_1, T, q_1\rangle\}$

$$
\begin{array}{lcl}
S & \rightarrow & A \\
S & \rightarrow & \langle q_0, Z_0, q_1\rangle \\
A & \rightarrow & \epsilon \\
\langle q_0, T, q_1\rangle & \rightarrow & b \\
\langle q_1, T, q_1\rangle & \rightarrow & b \\
A & \rightarrow & a\langle q_0, T, q_0\rangle \\
\langle q_0, Z_0, q_1\rangle & \rightarrow & a\langle q_0, T, q_1\rangle \\
\langle q_0, T, q_0\rangle & \rightarrow & a\langle q_0, T, q_0\rangle\langle q_0, T, q_0\rangle \\
\langle q_0, T, q_0\rangle & \rightarrow & a\langle q_0, T, q_1\rangle\langle q_1, T, q_0\rangle \\
\langle q_0, T, q_1\rangle & \rightarrow & a\langle q_0, T, q_0\rangle\langle q_0, T, q_1\rangle \\
\langle q_0, T, q_1\rangle & \rightarrow & a\langle q_0, T, q_1\rangle\langle q_1, T, q_1\rangle
\end{array}
$$

# Automate à pile vers grammaire algébrique : exemple

On renomme :

- $V \setminus \Sigma = \{S, A, B, C, D, E, F, G, H\}$

$$\begin{array}{lllllll} S & \to & A & \quad & B & \to & aF \\ S & \to & B & & E & \to & aEE \\ A & \to & \epsilon & & E & \to & aFG \\ F & \to & b & & F & \to & aEF \\ H & \to & b & & F & \to & aFH \\ A & \to & aE & & & & \end{array}$$

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V \setminus \Sigma = \{S, A, B, C, D, E, F, G, H\}$

$$
\begin{array}{lllllll}
S & \rightarrow & A & \quad & B & \rightarrow & aF \\
S & \rightarrow & B & & E & \rightarrow & aEE \\
A & \rightarrow & \epsilon & & E & \rightarrow & aFG \\
F & \rightarrow & b & & F & \rightarrow & aEF \\
H & \rightarrow & b & & F & \rightarrow & aFH \\
A & \rightarrow & aE & & & &
\end{array}
$$

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V \setminus \Sigma = \{S, A, B, C, D, E, F, G, H\}$

$$
\begin{array}{lllllll}
S & \rightarrow & A & \quad & B & \rightarrow & aF \\
S & \rightarrow & B & & E & \rightarrow & aEE \\
A & \rightarrow & \epsilon & & E & \rightarrow & aFG \\
F & \rightarrow & b & & F & \rightarrow & aEF \\
H & \rightarrow & b & & F & \rightarrow & aFH \\
A & \rightarrow & aE & & & &
\end{array}
$$

- Symboles productifs : $\{A, F, H, S, B\}$

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V \setminus \Sigma = \{S, A, B, C, D, E, F, G, H\}$

$$
\begin{array}{lllllll}
S & \rightarrow & A & \quad & B & \rightarrow & aF \\
S & \rightarrow & B & & E & \rightarrow & aEE \\
A & \rightarrow & \epsilon & & E & \rightarrow & aFG \\
F & \rightarrow & b & & F & \rightarrow & aEF \\
H & \rightarrow & b & & F & \rightarrow & aFH \\
A & \rightarrow & aE & & & &
\end{array}
$$

- Symboles productifs : $\{A, F, H, S, B\}$

  $\rightarrow$ On supprime $C, D, E, G$

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V = \{S, A, B, F, H\}$

$$\begin{array}{lclclcl} S & \rightarrow & A & \quad & B & \rightarrow & aF \\ S & \rightarrow & B & & & & \\ A & \rightarrow & \epsilon & & & & \\ F & \rightarrow & b & & & & \\ H & \rightarrow & b & & F & \rightarrow & aFH \end{array}$$

- Symboles productifs : $\{A, F, H, S, B\}$

  $\rightarrow$ On supprime $C, D, E, G$

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V \setminus \Sigma = \{S, A, B, F, H\}$

$$\begin{array}{lllllll} S & \rightarrow & A & \quad & B & \rightarrow & aF \\ S & \rightarrow & B & & F & \rightarrow & b \\ A & \rightarrow & \epsilon & & F & \rightarrow & aFH \\ H & \rightarrow & b & & & & \end{array}$$

- Symboles productifs : $\{A, F, H, S, B\}$
  $\rightarrow$ On supprime $C, D, E, G$
- Symboles accessibles : $\{S, A, B, F, H\}$

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V \setminus \Sigma = \{S, A, B, F, H\}$

$$\begin{array}{lllllll}
S & \rightarrow & A & \quad & B & \rightarrow & aF \\
S & \rightarrow & B & & F & \rightarrow & b \\
A & \rightarrow & \epsilon & & F & \rightarrow & aFH \\
H & \rightarrow & b & & & &
\end{array}$$

- Symboles productifs : $\{A, F, H, S, B\}$
  - $\rightarrow$ On supprime $C, D, E, G$
- Symboles accessibles : $\{S, A, B, F, H\}$
- Suppression des règles unitaires

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V \setminus \Sigma = \{S, A, B, F, H\}$

$$\begin{array}{lllllll} S & \rightarrow & \epsilon & \quad & B & \rightarrow & aF \\ S & \rightarrow & aF & & F & \rightarrow & b \\ A & \rightarrow & \epsilon & & F & \rightarrow & aFH \\ H & \rightarrow & b & & & & \end{array}$$

- Symboles productifs : $\{A, F, H, S, B\}$
  $\rightarrow$ On supprime $C, D, E, G$
- Symboles accessibles : $\{S, A, B, F, H\}$
- Suppression des règles unitaires

# Automate à pile vers grammaire algébrique : exemple

On réduit :

- $V \setminus \Sigma = \{S, A, B, F, H\}$

$$
\begin{array}{lllllll}
S & \rightarrow & \epsilon & \quad & B & \rightarrow & aF \\
S & \rightarrow & aF & & F & \rightarrow & b \\
A & \rightarrow & \epsilon & & F & \rightarrow & aFH \\
H & \rightarrow & b & & & &
\end{array}
$$

- Symboles productifs : $\{A, F, H, S, B\}$
  $\rightarrow$ On supprime $C, D, E, G$
- Symboles accessibles : $\{S, A, B, F, H\}$
- Suppression des règles unitaires
- **On conserve les $\epsilon$-règles**

# Automate à pile vers grammaire algébrique : exemple

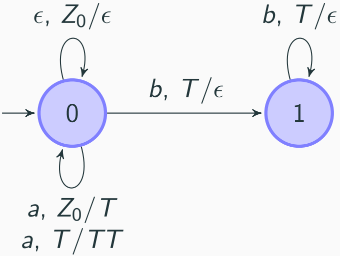
On réduit :

- $V \setminus \Sigma = \{S, A, B, F, H\}$

$$\begin{array}{lllllll} S & \rightarrow & \epsilon & & & & \\ S & \rightarrow & aF & & F & \rightarrow & b \\ & & & & F & \rightarrow & aFH \\ H & \rightarrow & b & & & & \end{array}$$

- Symboles productifs : $\{A, F, H, S, B\}$
  $\rightarrow$ On supprime $C, D, E, G$
- Symboles accessibles : $\{S, A, B, F, H\}$
- Suppression des règles unitaires
- **On conserve les $\epsilon$-règles**
- Symboles accessibles : $\{S, F, H\}$ : on supprime $A$ et $B$

# Automate à pile vers grammaire algébrique : exemple




On obtient :

$$\begin{array}{lcl} S & \rightarrow & \epsilon \\ S & \rightarrow & aF \\ F & \rightarrow & b \\ F & \rightarrow & aFH \\ H & \rightarrow & b \end{array}$$

# Caractérisation des langages algébriques

# Rappel sur les langages algébriques

La classe des langages algébriques est close pour :

- L'union
- La concaténation
- L'opération étoile

Mais, elle n'est pas close pour :

- L'intersection
- Le complémentaire

# Caractérisation des langages algébriques

Un langage algébrique peut être caractérisé par :

1. Un automate à pile
    - Déterministe, si le langage est déterministe
    - Non déterministe autrement
2. Une grammaire hors-contexte

# Caractérisation des langages algébriques

Un langage algébrique peut être caractérisé par :

1. Un automate à pile
   - Déterministe, si le langage est déterministe
   - Non déterministe autrement
2. Une grammaire hors-contexte

$\Rightarrow$ Pour démontrer qu'un langage est algébrique, il suffit donc de le décrire à l'aide de l'une de ces caractérisations

# Caractérisation des langages algébriques

Un langage algébrique peut être caractérisé par :

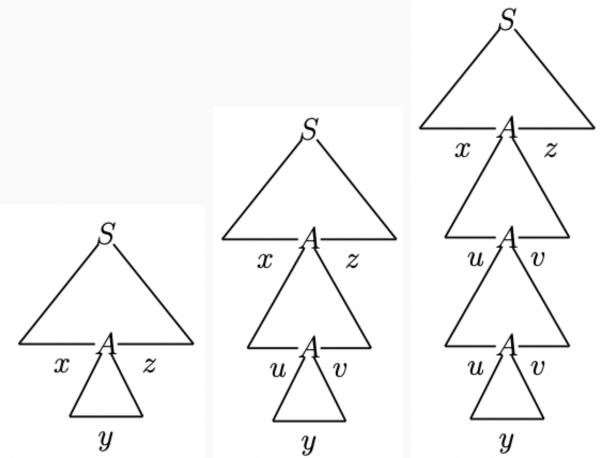
1. Un automate à pile
   - Déterministe, si le langage est déterministe
   - Non déterministe autrement
2. Une grammaire hors-contexte

$\Rightarrow$ Pour démontrer qu'un langage est algébrique, il suffit donc de le décrire à l'aide de l'une de ces caractérisations

$\Rightarrow$ Mais comment montrer qu'un langage n'est pas algébrique ?

# Idée

$$S \rightarrow xAz$$
$$A \rightarrow uAv|y$$




0 itération 1 itération 2 itérations . . .

**Lemme d'itération pour les langages algébriques (dit aussi théorème de pompage ou théorème du gonflement)**

Soit $L$ un langage algébrique sur l'alphabet $\Sigma$.

Alors, il existe $p \geq 0$ tel que $\forall w \in L$ avec $|w| \geq p$, il existe $x, u, y, z, t \in \Sigma^*$, tels que $w = xuyzt$, et

1. $|uz| > 0$
2. $|uyz| \leq p$
3. $\forall n \geq 0,\ xu^nyz^nt \in L$

# Lemme d'itération pour les langages algébriques

**Lemme d'itération pour les langages algébriques (dit aussi théorème de pompage ou théorème du gonflement)**

Soit $L$ un langage algébrique sur l'alphabet $\Sigma$.

Alors, il existe $p \geq 0$ tel que $\forall w \in L$ avec $|w| \geq p$, il existe $x, u, y, z, t \in \Sigma^*$, tels que $w = xuyzt$, et

1. $|uz| > 0$
2. $|uyz| \leq p$
3. $\forall n \geq 0,\ xu^n yz^n t \in L$

- Comme pour les langages rationnels, le théorème de pompage sert à montrer la **non-algébricité** d'un langage.
- On suppose qu'un langage $L$ est algébrique, et on cherche une contradiction

# Comment ça marche ?

- On suppose que $L$ est algébrique
- On considère $w \in L$ de longueur $\geq p$
- Quelle que soit la décomposition de $w$ en $xuyzt$ qui vérifie :
  1. $|uz| > 0$
  2. $|uyz| \leq p$
- Si on trouve un $n \geq 0$ tel que $xu^nyz^nt \notin L$
- Alors $L$ n'est pas algébrique

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n \mid n \geq 0\}$. Supposons $L$ algébrique.

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n |\ n \geq 0\}$. Supposons $L$ algébrique.

Il existe donc $p \geq 0$ tel que $w \in L$ et $|w| \geq p$, et il est possible de décomposer $w = xuyzt$, avec $|uz| > 0$, et $|uyz| \leq p$.
Alors $\forall n \geq 0$, $xu^n yz^n t \in L$.

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n \mid n \geq 0\}$. Supposons $L$ algébrique.

Il existe donc $p \geq 0$ tel que $w \in L$ et $|w| \geq p$, et il est possible de décomposer $w = xuyzt$, avec $|uz| > 0$, et $|uyz| \leq p$.
Alors $\forall n \geq 0$, $xu^n yz^n t \in L$.

Soit $w = a^p b^p c^p = xuyzt$. On a bien $|w| = 3p \geq p$.

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n \mid n \geq 0\}$. Supposons $L$ algébrique.

Il existe donc $p \geq 0$ tel que $w \in L$ et $|w| \geq p$, et il est possible de décomposer $w = xuyzt$, avec $|uz| > 0$, et $|uyz| \leq p$.
Alors $\forall n \geq 0$, $xu^n yz^n t \in L$.

Soit $w = a^p b^p c^p = xuyzt$. On a bien $|w| = 3p \geq p$.

---

Clairement, chacun des mots $u$ et $z$ ne contient qu'une seule des lettres $a, b$ ou $c$ sinon le mot ne fait plus partie du langage lorsque on va itérer $u$ et/ou $z$. Par exemple, si

$w = \underbrace{a^r}_{x}\underbrace{a^s b^i}_{u}\underbrace{b^j}_{y}\underbrace{b^k}_{v}\underbrace{c^p}_{t}$, avec $r + s = j + i + k = p$ et $s + i + k > 0$

On a donc $\forall n \geq 0$, $xu^n yv^n t \in L$.
Prenons $n = 2$. On a : $a^r a^s b^i a^s b^i b^j b^k b^k c^p \notin L$.
Contradiction.

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n |\ n \geq 0\}$. Supposons $L$ algébrique.

Il existe donc $p \geq 0$ tel que $w \in L$ et $|w| \geq p$, et il est possible de décomposer $w = xuyzt$, avec $|uz| > 0$, et $|uyz| \leq p$.
Alors $\forall n \geq 0$, $xu^n yz^n t \in L$.

Soit $w = a^p b^p c^p = xuyzt$. On a bien $|w| = 3p \geq p$.

---

On a donc $u, v \in \{a^*, b^*, c^*\}$.
Les configurations où $v$ contient des symboles censés être avant sont impossibles car on veut avoir des $a$ puis des $b$ puis des $c$. Par exemple, $w = \underbrace{a^r}_{x}\underbrace{c^i}_{u}\underbrace{b^p}_{y}\underbrace{a^s}_{v}\underbrace{c^j}_{t}$ avec $r + s = i + j = p$ et $i + s > 0$.

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n | \ n \geq 0\}$. Supposons $L$ algébrique.

Il existe donc $p \geq 0$ tel que $w \in L$ et $|w| \geq p$, et il est possible de décomposer $w = xuyzt$, avec $|uz| > 0$, et $|uyz| \leq p$.
Alors $\forall n \geq 0$, $xu^n yz^n t \in L$.

Soit $w = a^p b^p c^p = xuyzt$. On a bien $|w| = 3p \geq p$.

---

On a donc $u, v \in \{a^*, b^*, c^*\}$.
Prenons les configurations où $u$ et $v$ concernent le même symbole. Par exemple,
$w = \underbrace{a^i}_{x} \underbrace{a^j}_{u} \underbrace{a^k}_{y} \underbrace{a^l}_{v} \underbrace{a^m b^p c^p}_{t}$ avec $i + j + k + l + m = p$ et $j + l > 0$.
On a donc $\forall n \geq 0$, $xu^n yv^n t \in L$.
Prenons $n = 2$. On a : $a^i a^j a^j a^k a^l a^l b^p c^p \notin L$.
Contradiction.
($\Rightarrow$ Idem pour $b$ et $c$)

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n |\ n \geq 0\}$. Supposons $L$ algébrique.

Il existe donc $p \geq 0$ tel que $w \in L$ et $|w| \geq p$, et il est possible de décomposer $w = xuyzt$, avec $|uz| > 0$, et $|uyz| \leq p$.
Alors $\forall n \geq 0$, $xu^n yz^n t \in L$.

Soit $w = a^p b^p c^p = xuyzt$. On a bien $|w| = 3p \geq p$.

---

On a donc $u, v \in \{a^*, b^*, c^*\}$.
Enfin prenons les configurations où $u$ et $v$ concernent deux symboles différents mais qui se "suivent". Par exemple avec $u = a^*$ et $v = b^*$,
$w = \underbrace{a^i}_{x} \underbrace{a^j}_{u} \underbrace{a^k b^l}_{y} \underbrace{b^m}_{v} \underbrace{b^o c^p}_{t}$ avec $i + j + k = l + m + o = p$ et $j + m > 0$.
On a donc $\forall n \geq 0$, $xu^n yv^n t \in L$.
Prenons $n = 2$. On a : $a^i a^j a^j a^k b^l b^m b^m b^o c^p \notin L$ (pas assez de $c$).
Contradiction.
($\Rightarrow$ Idem pour $u = a^* + v = c^*$ et $u = b^* + v = c^*$)

# Montrer qu'un langage n'est pas algébrique : exemple

Soit $\Sigma = \{a, b, c\}$, $L = \{a^n b^n c^n |\ n \geq 0\}$. Supposons $L$ algébrique.

Il existe donc $p \geq 0$ tel que $w \in L$ et $|w| \geq p$, et il est possible de décomposer $w = xuyzt$, avec $|uz| > 0$, et $|uyz| \leq p$.
Alors $\forall n \geq 0$, $xu^n yz^n t \in L$.

Soit $w = a^p b^p c^p = xuyzt$. On a bien $|w| = 3p \geq p$.

---

Le théorème de gonflement n'est pas vérifié. Ce langage n'est donc pas un langage algébrique.