Théorie des langages
Grammaires et langages algébriques

F. Alexandre

École Supérieure d'Informatique et Applications de Lorraine

December 2, 2008

- Théorie des langages (hiérarchie de Chomsky)
- Analyse syntaxique, compilation (Module TRAD 2A)
- TAL, écriture de logiciel

Definition

On appelle grammaire algébrique un quadruplet $G = (N, T, \rightarrow, X)$ tq :

- N est un vocabulaire (fini) auxiliaire de la grammaire appelé ensemble des symboles non terminaux. Intuitivement chaque élément de N doit se comprendre comme le nom d'une catégorie syntaxique de mots sur le vocabulaire T de la grammaire.
- T est un autre vocabulaire disjoint de N et s'appelle le vocabulaire terminal. C'est sur ce vocabulaire que sont écrits les mots engendrés par la grammaire.
- $\rightarrow$ est une relation entre N et $(N \cup T)^*$ telle que pour chaque élément A de N il n'y ait qu'un nombre fini de mots α appartenant à $(N \cup T)^*$ et tels que $A \rightarrow \alpha$. Cette relation s'appelle la relation de production de la grammaire.
- X est un élément distingué de N appelé axiome de la grammaire et représentant intuitivement la catégorie syntaxique la plus large du langage que l'on veut engendrer.

Exemple 1: grammaire (très simplifiée) de la langue française

- $$\begin{aligned} N = \{ \quad & <\mathit{phrase}>, \quad <\mathit{groupe\ nominal}>, \quad <\mathit{groupe\ verbal}>, \\ & <\mathit{groupe\text{-}verbal\text{-}etre}>, \quad <\mathit{groupe\ complement}>, \\ & <\mathit{nom}>, \quad <\mathit{article}>, \quad <\mathit{verbe}>, \quad <\mathit{adjectif}>, \\ & <\mathit{preposition}>, \quad <\mathit{attribut}>, \ldots \} \end{aligned}$$
- $T = \{\mathit{le}, \ \mathit{la}, \ \mathit{chat}, \ \mathit{souris}, \ \mathit{table}, \ \mathit{est}, \ \mathit{mange}, \ \mathit{dans}, \ \mathit{sur}, \ \mathit{beau}, \ \mathit{belle}, \ldots\}$ Le vocabulaire T (l'ensemble des terminaux) de la grammaire est l'ensemble de tous les mots de la langue française.
- $X =< \mathit{phrase} >$, l'axiome est le concept syntaxique le plus général de la grammaire, ici c'est la phrase, correspondant au symbole $< \mathit{phrase} >$.

Exemple 1 (suite et fin)

Les règles de grammaire engendrant la langue française sont très nombreuses (quelques milliers à quelques dizaines de milliers). Donnons quelques exemples caractéristiques.

$$\begin{array}{lcl}
<\textit{phrase}> & \rightarrow & <\textit{groupe nominal}><\textit{groupe verbal}><\textit{groupe complement}> \mid \\
 & \rightarrow & <\textit{groupe nominal}><\textit{groupe-verbal-etre}><\textit{attribut}> \mid \ldots \\
 & & \\
<\textit{groupe nominal}> & \rightarrow & <\textit{article}><\textit{adjectif}><\textit{nom}> \mid \\
 & \rightarrow & <\textit{article}><\textit{nom}><\textit{adjectif}> \mid \\
 & \rightarrow & <\textit{article}><\textit{nom}> \mid \ldots \\
 & & \\
<\textit{article}> & \rightarrow & \textit{le} \mid \textit{la} \mid \textit{les} \mid \textit{des} \mid \ \ldots \\
<\textit{adjectif}> & \rightarrow & \textit{beau} \mid \textit{belle} \mid \textit{petit} \mid \textit{mort} \mid \textit{noir} \mid \ldots \\
<\textit{nom}> & \rightarrow & \textit{chat} \mid \textit{souris} \mid \textit{table} \mid \ \ldots \\
<\textit{preposition}> & \rightarrow & \textit{dans} \mid \textit{sur} \mid \ \ldots \\
<\textit{groupe verbal}> & \rightarrow & \textit{mange} \mid \textit{est} \mid \ \ldots \\
<\textit{groupe-verbal-etre}> & \rightarrow & \textit{est} \mid \textit{sont} \mid \ \ldots \\
<\textit{attribut}> & \rightarrow & <\textit{adjectif}> \ \mid \ \ldots \\
<\textit{groupe complement}> & \rightarrow & <\textit{groupe nominal}> \ \mid \\
 & \rightarrow & <\textit{preposition}><\textit{groupe nominal}> \ \mid \ \ldots
\end{array}$$

Exemple de phrases engendrées par la grammaire précédente :

- *le petit chat est mort*
- *le chat mange sur le table*
- *les souris sont beaux*
- *la table mange la souris*

- $$N = \{ \quad < programme >, \ < partie\ declaration >, \\ < partie\ instruction >, \ < declaration >, \ < instruction >, \\ < affectation >, \ < identificateur >, \\ < conditionnelle >, \ < iteration >, \ < procedure >, \ldots \}$$
- $T = \{A,\ B,\ \ldots,\ Z,\ a, \ldots,\ z,\ (,\ ),\ [,\ ],\ +,\ *,\ /,\ ;,\ \ldots\}$
- $X =< programme >$
- $$\begin{array}{lcl} < programme > & \rightarrow & PROGRAM < identificateur >; < partie\ declaration >; \\ & & < partie\ instruction > \\ < identificateur > & \rightarrow & \ldots \\ < partie\ declaration > & \rightarrow & \ldots \\ < partie\ instruction > & \rightarrow & < declaration > \ | \\ & \rightarrow & < declaration >; < partie\ instruction > \ | \\ & \rightarrow & \ldots \end{array}$$

Definition

Soit $G = (N,\ T,\ \rightarrow,\ X)$ une grammaire algébrique. On définit sur $(N \cup T)^*$ les relations de réécriture $\rightarrowtail$, de dérivation $\overset{*}{\rightarrowtail}$ et de dérivation stricte $\overset{+}{\rightarrowtail}$ de la manière suivante :

1. $\alpha \rightarrowtail \beta$ si et seulement si, il existe $\alpha_1 \in (N \cup T)^*$, $\alpha_2 \in (N \cup T)^*$, $A \in N$, $\gamma \in (N \cup T)^*$ tels que $\alpha = \alpha_1 A \alpha_2$ et $\beta = \alpha_1 \gamma \alpha_2$ et $A \rightarrow \gamma$ est une règle de G.
2. $\alpha \overset{*}{\rightarrowtail} \beta$ si et seulement si, il existe un entier n et une suite finie $(\alpha_i)_{0 \leq i \leq n}$ telle que $\alpha = \alpha_0$ et $\beta = \alpha_n$ et pout tout i de $[0,\ n\text{-}1]$, $\alpha_i \rightarrowtail \alpha_{i+1}$.
3. $\alpha \overset{+}{\rightarrowtail} \beta$ si et seulement si, il existe un entier n non nul et une suite finie $(\alpha_i)_{0 \leq i \leq n}$ telle que $\alpha = \alpha_0$ et $\beta = \alpha_n$ et pout tout i de $[0,\ n\text{-}1]$, $\alpha_i \rightarrowtail \alpha_{i+1}$.

La suite $(\alpha_i)_{0 \leq i \leq n}$ s'appelle une dérivation de α à β (ou une dérivation au sens large si $n > 0$).

Exemple

G définie par $N = \{X,\ A,\ B\}$, $T = \{a, b\}$, X l'axiome de la grammaire, la relation de production est définie par les règles suivantes :

$$\begin{array}{l} X \longrightarrow \varepsilon \mid aB \mid bA \\ A \longrightarrow aX \mid bAA \\ B \longrightarrow bX \mid aBB \end{array}$$

Exemples de réécritures :
$aBbA \rightarrowtail aaBBbA$
$aaBBbA \rightarrowtail aaBbXbA$
$aaBbXbA \rightarrowtail aaBb\varepsilon bA = aaBbbA$

Exemple de dérivation :

$aBbA \rightarrowtail aaBBbA \rightarrowtail aaBbXbA \rightarrowtail aaBbbA$ d'où $aBbA \overset{*}{\rightarrowtail} aaBbbA$

Definition

Soit $G = (N,\ T,\ \rightarrow,\ X)$ une grammaire algébrique.

- On dit qu'un mot α de T^* est engendré par la grammaire G à partir de A de N si et seulement si $A \overset{*}{\rightarrowtail} \alpha$.
- On dit qu'un mot α de T^* est engendré par la grammaire G si et seulement si $X \overset{*}{\rightarrowtail} \alpha$ (α est engendré par G à partir de l'axiome de la grammaire).
- Le langage engendré par la grammaire G à partir de A ($A \in N$) est l'ensemble des mots engendrés par G à partir de A, on le note $L(G,\ A)$
- Le langage engendré par la grammaire G est le langage engendré par la grammaire à partir de l'axiome de la grammaire, on le note $L(G)$. On a

  $L(G) = L(G,\ X) = \{\alpha\ ;\ \alpha \in T^* \text{ et } X \overset{*}{\rightarrowtail} \alpha\}$.
- Un langage engendré par une grammaire algébrique s'appelle un langage algébrique (ou hors-contexte).

- Soit la grammaire suivante $G = (\{X\},\ \{a, b\},\ \ \rightarrow,\ X)$, où les règles de production sont définies par $X \rightarrow aXb \mid \varepsilon$
  On a $L(G) = \{a^n b^n\ ; n \geq 0\}$.
- Soit la grammaire suivante $G = (\{X\},\ \{a, b\},\ \ \rightarrow,\ X)$ où les règles de production sont $X \rightarrow aXa \mid bXb \mid \varepsilon$.
  Le langage engendré par G est $L(G) = \{w\widetilde{w}\ ;\ w \in \{a, b\}^*\}$ ($\widetilde{w}$ est le mot obtenu en renversant le mot w).

Soit la grammaire algébrique $G = (\{X\},\ \{a,b\},\ \ \rightarrow,\ X)$, où $\rightarrow$ est défini par

$$X \rightarrow XX\ |1|\ aXa\ |2|\ bXb\ |3|\ \varepsilon|4|$$

Soit le mot *aabaab*, on veut montrer que $aabaab \in L(G)$

$X \overset{1}{\rightarrowtail} XX \overset{2}{\rightarrowtail} aXaX \overset{4}{\rightarrowtail} aaX \overset{3}{\rightarrowtail} aabXb \overset{2}{\rightarrowtail} aabaXab \overset{4}{\rightarrowtail} aabaab$

$X \overset{1}{\rightarrowtail} XX \overset{3}{\rightarrowtail} XbXb \overset{2}{\rightarrowtail} XbaXab \overset{4}{\rightarrowtail} Xbaab \overset{2}{\rightarrowtail} aXabaab \overset{4}{\rightarrowtail} aabaab$

il y 8 autres dérivations possibles pour générer le mot *aabaab* à partir de l'axiome X.

Remarque : le fait que l'ordre dans lequel les règles de productions sont appliquées n'est pas important est une caractéristique des grammaires algébriques, aussi appelées grammaires hors-contexte (car un non-terminal peut être remplacé indépendamment de la chaîne de caractères qui l'entoure).

Definition

Soit $G = (N,\ T,\ \rightarrow,\ X)$, un arbre syntaxique pour la grammaire G est un arbre étiqueté par les éléments de $N \cup T \cup \{\varepsilon\}$ qui satisfait les conditions suivantes :

- la racine de l'arbre est étiquetée par X, l'axiome de G
- chaque nœud interne est étiqueté par un élément de N. Chaque feuille est étiquetée par un élément de T ou par ε.
- pour tout nœud interne, si son étiquette est un non-terminal A et si ses descendants immédiats sont les nœuds $n_1, \ldots, n_k$ ayant respectivement pour étiquettes $X_1, \ldots, X_k$ alors $A \rightarrow X_1 \ldots X_k$ doit être une règle de production de G.
- si un nœud est étiqueté par ε, alors ce nœud est le seul descendant immédiat de son prédécesseur (cette dernière règle évite l'introduction d'instances inutiles de ε dans l'arbre syntaxique).

Definition

Le mot généré par un arbre syntaxique est celui obtenu par la concaténation des feuilles de l'arbre prises de gauche à droite.

Example

Soit la grammaire algébrique $G = (\{X\},\ \{a, b\},\ \rightarrow,\ X)$, où $\rightarrow$ est défini par

$$X \rightarrow XX \mid aXa \mid bXb \mid \varepsilon$$

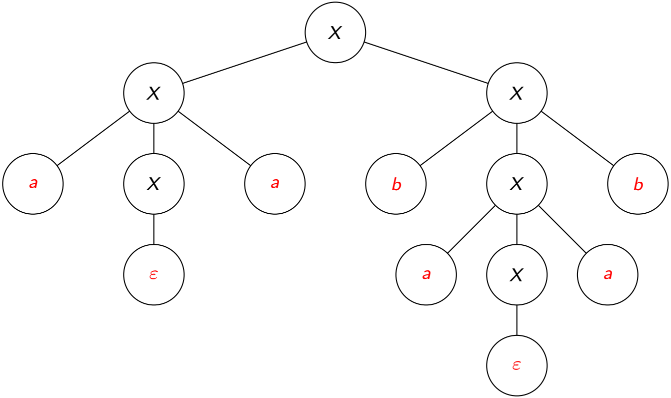
Dessiner l'arbre syntaxique générant le mot *aabaab*.

Proposition

Soit G une grammaire algébrique, un mot w est généré par G si et seulement si il existe un arbre syntaxique de la grammaire G qui génère w.

Exemple d'arbre syntaxique

Un arbre syntaxique du mot *aabaab*



Definition

Une grammaire G est ambiguë si et seulement si il existe un mot $w \in L(G)$ engendré par deux arbres syntaxiques différents.

Definition

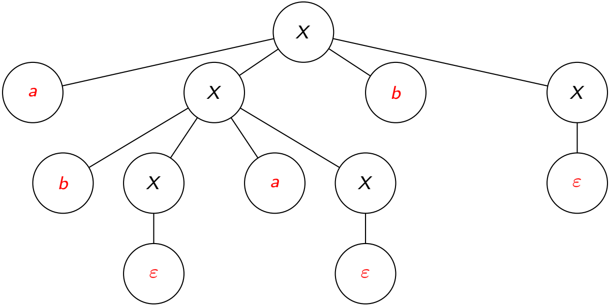
Soit L un langage algébrique. On dit que L possède une ambiguïté inhérente si et seulement si toute grammaire algébrique engendrant L est ambiguë.

Example

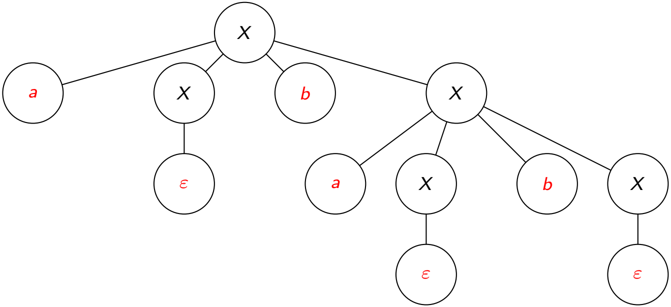
Soit $G = (\{X\}, \ \{a, \ b\}, \ \ \rightarrow, \ X)$, la grammaire définie par les règles de production suivantes : $X \rightarrow aXbX \ |bXaX \ |\varepsilon$, est ambiguë.
Le langage $L(G)$ ne possède pas d'ambiguïté inhérente (on peut montrer qu'il existe au moins une grammaire non ambiguë qui l'engendre).

Exemple : ambiguïté

Soit $G = (\{X\},\ \{a,\ b\},\ \ \rightarrow,\ X)$, la grammaire définie par les règles $X \rightarrow aXbX\ |bXaX\ |\varepsilon$ est ambiguë. Arbre syntaxique du mot *abab*



Deuxième arbre syntaxique de *abab*



Le mot *abab* a deux arbres syntaxiques, il est ambigü, la grammaire est donc ambiguë.

Langages de programmation classiques (PASCAL, LISP, FORTRAN, C,...) sont en première approximation des expressions arithmétiques compliquées ...
On s'intéresse ici à la **syntaxe** des expressions arithmétiques. Pour construire les expressions arithmétiques on a besoin de noms de fonctions. Les symboles de fonctions sont classés selon leur **arité** (nombre d'arguments de la fonction):

- $\{a, \ b, \ c\}$ les symboles de constantes (d'arité 0).
- $\{s, \ f, \ g\}$ les symboles d'arité 1.
- $\{+, \ *, \ /, -\}$ les opérateurs binaires.
- $\{\psi, \ \varphi\}$ les opérateurs d'arité 3.

On dispose d'un ensemble de symboles de variables $X = \{x, \ y, \ z, \ x', \ y', \ z'\}$.

L'ensemble des symboles terminaux est constitué de l'ensemble des symboles de fonctions et des symboles de variables (plus éventuellement des symboles de ponctuation et des parenthèses selon les cas).

- $G = (\{X,\ A,\ B,\ C,\ D,\ V\},\ T,\ \ \rightarrow,\ X)$ telle que
  - $T = \{x,\ y,\ z,\ x',\ y',\ z'\} \cup \{a,\ b,\ c\} \cup \{s,\ f,\ g\} \cup \{+,\ *,\ /, -\} \cup \{\psi,\ \varphi\} \cup \{(,\ ),\ ,\}$
  - $\rightarrow$ est définie par :

$$
\begin{array}{l}
X \rightarrow V \ |A \ |B(X) \ |C(X,\ X) \ |D(X,\ X,\ X) \\
V \rightarrow x \ |y \ |z \ |x' \ |y' \ |z' \\
A \rightarrow a \ |b \ |c \\
B \rightarrow s \ |f \ |g \\
C \rightarrow + \ | * \ |/ \ |- \\
D \rightarrow \psi \ |\varphi
\end{array}
$$

- $*(\phi(s(x),\ f(+(a,y)),b),/(\psi(a,z,+(y,c)),*(x,y)))$ est un mot engendré par G

Remarque : les parenthèses et les virgules séparent les opérandes des fonctions, elles permettent une meilleure lisibilité des formules arithmétiques.

- $G = (\{X,\ A,\ B,\ C,\ D,\ V\},\ T,\ \rightarrow,\ X)$ telle que
  - $T = \{x,\ y,\ z,\ x',\ y',\ z'\} \cup \{a,\ b,\ c\} \cup \{s,\ f,\ g\} \cup \{+,\ *,\ /, -\} \cup \{\psi,\ \varphi\}$
  - $\rightarrow$ est définie par :

$$\begin{array}{l} X \rightarrow V\ |A\ |BX\ |CXX\ |DXXX \\ V \rightarrow x\ |y\ |z\ |x'\ |y'\ |z' \\ A \rightarrow a\ |b\ |c \\ B \rightarrow s\ |f\ |g \\ C \rightarrow +\ |*\ |/\ |- \\ D \rightarrow \psi\ |\varphi \end{array}$$

- $*\phi sxf + ayb/\psi az + yc * xy$ est un mot engendré par G

Remarque : notation minimale qui consiste à écrire les opérateurs avant les opérandes. Cette notation est bien adaptée si l'on veut démontrer des résultats théoriques sur les expressions arithmétiques.

- $G = (\{X,\ A,\ B,\ C,\ D,\ V\},\ T,\ \longrightarrow,\ X)$ telle que
  - $T = \{x,\ y,\ z,\ x',\ y',\ z'\} \cup \{a,\ b,\ c\} \cup \{s,\ f,\ g\} \cup \{+,\ *,\ /, -\} \cup \{\psi,\ \varphi\}$
  - $\longrightarrow$ est définie par :

$$\begin{array}{l} X \longrightarrow V \mid A \mid XB \mid XXC \mid XXXD \\ V \longrightarrow x \mid y \mid z \mid x' \mid y' \mid z' \\ A \longrightarrow a \mid b \mid c \\ B \longrightarrow s \mid f \mid g \\ C \longrightarrow + \mid * \mid / \mid - \\ D \longrightarrow \psi \mid \varphi \end{array}$$

- $xsay + fb\phi azyc + \psi xy * /*$ est un mot engendré par G

Remarque : les opérandes sont écrits avant les opérateurs. Cette notation est pratique si l'on veut évaluer les expressions, elle a été utilisée par certaine marque de calculatrice.

C'est la notation usuelle qui consiste à noter les opérateurs binaires sous forme infixée, elle nécessite d'utiliser des parenthèses et des virgules. Les opérateurs d'arité 1 ou supérieures à 2 sont sous forme préfixée.
L'écriture de ces grammaires se fera en exercice.

Exemple de mot : $\phi(s(x),\ f(a+y),\ b) * (\psi(a,\ z,\ y+c)/(x*y))$

On considère deux vocabulaires distincts V et $\overline{V}$ tels que V et $\overline{V}$ soit en bijection (V représentent l'ensemble des parenthèses ouvrantes et $\overline{V}$ l'ensemble des parenthèses fermantes).

Definition

On appelle mot bien parenthésé sur V un mot α sur $(V \cup \overline{V})^*$ tel que $\pi(\alpha) = \varepsilon$ où l'application de $(V \cup \overline{V})^*$ dans $V^* \cup \{0\}$ est définie par :

1. $\pi(\varepsilon) = \varepsilon$
2. $a \in V \Rightarrow \pi(\alpha a) = \mathit{si}\ \pi(\alpha) = 0\ \mathit{alors}\ 0\ \mathit{sinon}\ \pi(\alpha)a\ \mathit{fsi}$
3. $$\overline{a} \in \overline{V} \Rightarrow \pi(\alpha\overline{a}) = \begin{array}{l} \mathit{si}\ \pi(\alpha) = 0 \\ \mathit{alors}\ 0 \\ \mathit{sinon}\ \mathit{si}\ \pi(\alpha) = \gamma a\ \mathit{alors}\ \gamma\ \mathit{sinon}\ 0\ \mathit{fsi} \\ \mathit{fsi} \end{array}$$

Exemple

- Soit la grammaire suivante $G = (\{X\},\ \{a,b\},\ \ \rightarrow,\ X)$, où les règles de production sont définies par $X \rightarrow aXb \mid \varepsilon$

  On a $L(G) = \{a^n b^n\ ; n \geq 0\}$.

- Soit la grammaire suivante $G = (\{X\},\ \{a,b\},\ \ \rightarrow,\ X)$ où les règles de production sont

  $X \rightarrow aXa \mid bXb \mid \varepsilon$. Montrons que ce langage engendré par G est

  $L(G) = \{w\widetilde{w}\ ;\ w \in \{a,b\}^*\}$ ($\widetilde{w}$ est le mot obtenu en renversant le mot w).

  Preuve :

  Soit un mot $u \in \{a,b\}^*$ de $L(G)$, on va montrer que u s'écrit $w\widetilde{w}$. $u \in L(G)$ donc $X \overset{*}{\rightarrowtail} u$, on effectue un raisonnement par récurrence sur la longueur n de la dérivation.

  - si $n = 1$ alors $X \rightarrowtail \varepsilon$ d'où $u = \varepsilon$ et $u = \varepsilon\widetilde{\varepsilon}$
  - si $n > 1$ alors on a l'une des deux dérivations suivantes :
    $X \rightarrowtail aXa \rightarrowtail \ldots \rightarrowtail u$ ou
    $X \rightarrowtail bXb \rightarrowtail \ldots \rightarrowtail u$
    La première dérivation implique qu'il existe un mot v de $\{a,\ b\}^*$ tel que $u = ava$, on a donc
    $X \rightarrowtail aXa \rightarrowtail \ldots \rightarrowtail ava$, et la dérivation $X \rightarrowtail \ldots \rightarrowtail v$ a comme longueur $(n-1)$, par hypothèse de récurrence il existe donc un mot s sur $\{a,b\}$ tel que $v = s\widetilde{s}$ et par conséquent
    $u = as\widetilde{s}a$, comme $\widetilde{as} = \widetilde{s}a$, en posant $w = as$, on a $u = w\widetilde{w}$.

    Même raisonnement avec la deuxième dérivation ...

Réciproque : soit un mot u de la forme $w\widetilde{w}$, montrons qu'il est engendré par G, on raisonne par récurrence sur la longueur du mot u.

- si $|u| = 0$ alors $u = \varepsilon$, ε est engendré par G car $X \rightarrowtail \varepsilon$
- si $|u| > 0$ alors $u = av\widetilde{v}a$ ou $u = bv\widetilde{v}b$, on suppose que $u = av\widetilde{v}a$ (l'autre cas se traite de façon identique),
  comme $|v\widetilde{v}| < |u|$ par hypothèse de récurrence $v\widetilde{v}$ est engendré par G,
  donc il existe une dérivation de la forme $X \rightarrowtail \ldots \rightarrowtail v\widetilde{v}$

  et on a $X \rightarrowtail aXa \rightarrowtail \ldots \rightarrowtail av\widetilde{v}a = u$, d'où $u \in L(G)$

Definition

Soit $G = (N,\ T,\ \rightarrow,\ X)$ une grammaire algébrique, on dit que G est une grammaire linéaire à droite si toutes ses règles sont de la forme :

$$A \rightarrow \alpha B \quad ou \quad A \rightarrow \alpha$$

où $A,\ B \in N$ et $\alpha \in T^*$.

Remarque

Les membres droits des règles de grammaires linéaires à droite contiennent au plus un non terminal qui est situé à droite.

Théorème

Un langage est régulier si et seulement si il est généré par une grammaire linéaire à droite.

Preuve

Soit L un langage régulier et soit $\mathcal{A} = (A, Q, q_0, \delta, T)$ un automate reconnaissant L. La grammaire $G = (N, T, \rightarrow, X)$ telle que

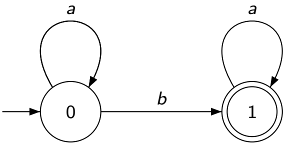
- T l'ensemble des terminaux de la grammaire est l'ensemble des lettres de l'alphabet A ($T = A$)
- L'ensemble N des non-terminaux de la grammaire est égal à l'ensemble Q des états de l'automate ($N = Q$)
- l'axiome X de la grammaire est l'état initial de l'automate q_0, ($X = q_0$)
- on associe la règle $A \rightarrow aB$ à la transition $\delta(A, a) = B$
  on associe la règle $A \rightarrow \varepsilon$ à chaque état final A de l'automate

Corollaire

Tout langage régulier est un langage algébrique.

Exemple

Soit $\mathcal{A} = (\{a, b\},\ \{0,\ 1\}, 0,\ \delta,\ \{1\})$ tel que $\delta(0, a) = 0$, $\delta(0, b) = 1$, $\delta(1, a) = 1$



La grammaire correspondant à cet automate est $G = (\{X, A\},\ \{a, b\},\ \ \rightarrow,\ X)$ est
X correspond à l'état 0 et A à l'état 1.
Les règles sont :

$$\begin{array}{llll}
X & \rightarrow & aX & (\delta(0, a) = 0) \\
X & \rightarrow & bA & (\delta(0, b) = 1) \\
A & \rightarrow & aA & (\delta(1, a) = 1) \\
A & \rightarrow & \varepsilon & (A \textit{ est un etat terminal})
\end{array}$$

Théorèmes de “gonflement” (pumping lemma) (lemme de l’étoile)

Lemme de l’étoile

Soient L un langage régulier infini et un automate fini comportant k états reconnaissant L. Soit α un mot de L tel que $|\alpha| \geq k$, alors $\exists x, u, y$ avec $u \neq \varepsilon$ et $|xu| \leq k$ tels que $\alpha = xuy$ et $(\forall n \in \mathbb{N})$ $(xu^n y \in L)$.

Justification

Soit un automate à k états et soit α un mot tel que $|\alpha| \geq k$, reconnu par l’automate donné

$$q_0 \xrightarrow{x} q_m \xrightarrow{u} q_m \xrightarrow{y} q_f$$

$\alpha = xuy$, comme on passe deux fois par l’état q_m, on peut y passer un nombre n quelconque de fois, donc le mot $xu^n y$ est reconnu par l’automate pour tout $n \in \mathbb{N}$.

Soit le langage $L = \{a^i b^i,\ i \in \mathbb{N}\}$, montrons que L n'est pas un langage régulier en appliquant le lemme de l'étoile.
Soit $\alpha = a^m b^m \in L$, supposons que $\alpha = xuy$, pour certains x, u et y, il faudrait alors montrer que $xu^n y \in L$ pour tout n, voyons ce que peut être u.

- si u ne contient que des a, ce n'est pas possible car $xu^n y$ contiendrait plus de a que de b pour n suffisamment grand
- si u ne contient que des b, c'est impossible pour la même raison
- si u contient des a et des b, en considérant u^n il y aura des b avant des a, donc le mot $xu^n y$ n'appartient pas à L

Conclusion : L n'est pas régulier.

On a vu que L est algébrique car L est reconnu par la grammaire algébrique

$(\{X\},\ \{a, b\},\ \ \rightarrow,\ X)$ dont les règles sont : $X \rightarrow \varepsilon$ et $X \rightarrow aXb$.

Proposition

Si L et L' sont des langages algébriques alors $L \cup L'$, LL', L^* sont des langages algébriques algébriques.

Remarques

- Pour montrer la proposition il suffit de construire des grammaires algébriques engendrant les langages $L \cup L'$, LL' et L^* à partir de grammaires engendrant L et L'.
- L'intersection de deux langages algébriques n'est en général pas un langage algébrique. De même le complémentaire d'un langage algébrique n'est en général pas un langage algébrique.
- On peut montrer que l'intersection d'un langage régulier et d'un langage algébrique est un langage algébrique.