Théorie des langages

Alexis Nasr

Lemme de l'étoile

Soit L un langage régulier. Il existe un entier k, appelé longueur de pompage, tel que tout mot $w \in L$ de longueur $\geq k$ peut s'écrire sous la forme $w = xyz$ avec :

1. $|xy| \leq k$,
2. $|y| > 0$,
3. $xy^iz \in L$ pour tout $i \geq 0$.

Preuve

- Soient A un AFD reconnaissant L et k le nombre d'états de A. Soit $w = w_{1,n} = w_1 \dots w_n$ un mot de L de longueur n.
- Notons

$$(q_0, w_{1,n}) \vdash (q_1, w_{2,n}) \vdash \cdots \vdash (q_{n-1}, w_{n,n}) \vdash (q_n, \varepsilon)$$

  la suite de mouvements que A effectue sur w.
- Si $n \geq k$, cette suite passe deux fois par le même état !
- Autrement dit, il existe q_i, q_j dans cette suite tels que $0 \leq i < j \leq n$ et $q_i = q_j$.
- Mais alors, pour chaque $t \geq 0$, la séquence de mouvements

$$(q_0, w_{1,n}) \vdash \cdots \vdash \{(q_{i-1,n}) \vdash \cdots \vdash (q_j, w_{j+1,n})\}^t \vdash \cdots \vdash (q_n, \varepsilon)$$

  reconnaît aussi un mot de L ($\{(q_{i-1,n}) \vdash \cdots \vdash (q_j, w_{j+1,n})\}^t$ dénote le fait que cette séquence est répétée t fois).
- Notons $x = w_{1,i}$, $y = w_{i+1,j}$ et $z = w_{j+1,n}$.
- Alors $xy^tz \in L$ pour chaque t, et $|xy| \leq k$, $|y| > 0$.

$L = a^n b^n$ n'est pas régulier

- Considérons que L est régulier, soit k la longueur de pompage.
- Soit s la chaine $a^k b^k$
- L étant régulier et $|s| > k$ alors s doit s'écrire $s = xyz$ avec $\forall i \geq 0, xy^i z \in L$
- Montrons que cela est impossible :
  1. Si y n'est composé que de a, alors $xyyz \notin L$ car $xyyz$ contient plus de a que de b
  2. Si y n'est composé que de b on aboutit aussi à une contradiction
  3. Si y contient des a et des b alors $xyyz$ n'est plus de la forme $a^n b^n$
- Donc, si on considère que L est régulier, on aboutit à une contradiction.
- L n'est donc pas régulier !

Grammaires de réécriture

Une grammaire de réécriture est un 4-uplet $\langle N, \Sigma, P, S \rangle$ où :

- N est un ensemble de symboles non terminaux, appelé l'alphabet non terminal.
- Σ est un ensemble de symboles terminaux, appelé l'alphabet terminal, tel que N et Σ soient disjoints.
- P est un sous ensemble fini de :

  $$(N \cup \Sigma)^* N (N \cup \Sigma)^* \times (N \cup \Sigma)^*$$

  un élément (α, β) de P, que l'on note $\alpha \rightarrow \beta$ est appelé une règle de production ou règle de réécriture.
  α est appelé partie gauche de la règle
  β est appelé partie droite de la règle
- S est un élément de N appelé l'axiome de la grammaire.

Notation

Pour alléger les notations, on note :

$$\alpha \rightarrow \beta_1 | \beta_2 | \ldots | \beta_n$$

les n règles :

$$\alpha \rightarrow \beta_1 \;,\; \alpha \rightarrow \beta_2 \;, \ldots ,\; \alpha \rightarrow \beta_n$$

Proto-mots d'une grammaire

Les proto-mots d'une grammaire $G = \langle N, \Sigma, P, S \rangle$ sont des mots construits sur l'alphabet $\Sigma \cup N$, on les définit récursivement de la façon suivante :

- S est une proto-mot de G
- si $\alpha\beta\gamma$ est une proto-mot de G et $\beta \to \delta \in P$ alors $\alpha\delta\gamma$ est une proto-mot de G.

Une proto-mot de G ne contenant aucun symbole non terminal est appelé un mot généré par G. Le langage généré par G, noté $L(G)$ est l'ensemble des mots générés par G.

Dérivation

- L'opération qui consiste à générer une proto-mot $\alpha\delta\gamma$ à partir d'une proto-mot $\alpha\beta\gamma$ et d'une règle de production r de la forme $\beta \rightarrow \delta$ est appelée l'opération de dérivation. Elle se note à l'aide d'une double flèche :

$$\alpha\beta\gamma \Rightarrow \alpha\delta\gamma$$

- On note $\alpha \overset{k}{\Rightarrow} \beta$ pour indiquer que β se dérive de α en k étapes.
- On définit aussi les deux notations $\overset{+}{\Rightarrow}$ et $\overset{*}{\Rightarrow}$ de la façon suivante :
  - $\alpha \overset{+}{\Rightarrow} \beta \equiv \alpha \overset{k}{\Rightarrow} \beta$ avec $k > 0$
  - $\alpha \overset{*}{\Rightarrow} \beta \equiv \alpha \overset{k}{\Rightarrow} \beta$ avec $k \geq 0$

Langage généré par une grammaire

- $L(G)$ est défini de la façon suivante :

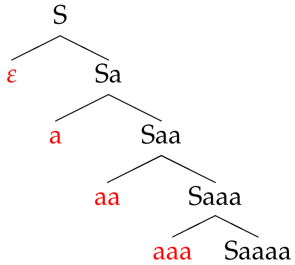
$$L(G) = \{m \in \Sigma^* | S \stackrel{+}{\Rightarrow} m\}$$

- Deux grammaires G et G' sont équivalentes si $L(G) = L(G')$.

$$L_1 = \{\varepsilon, a, aa, aaa, \ldots\}$$

$$G = \langle \{S\}, \{a\}, \{S \rightarrow Sa | \varepsilon\}, S \rangle$$

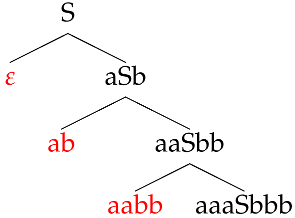
Sous-ensemble des proto-mots de G



$L_2 = \{\varepsilon, ab, aabb, aaabbb, aaaabbbb, \ldots\}$

$G = \langle \{S\}, \{a, b\}, \{S \to aSb | \varepsilon\}, S \rangle$

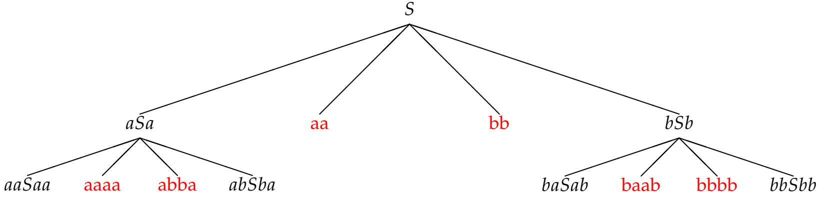
Sous-Ensemble des proto-mots de G



$$L_3 = \{aa, bb, aaaa, abba, baab, bbbb, \ldots\}$$

$$G = \langle \{S\}, \{a, b\}, \{S \rightarrow aSa|bSb|aa|bb\}, S \rangle$$

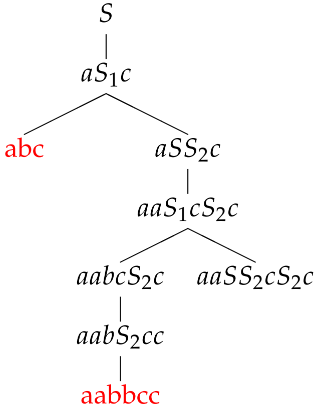
Sous-Ensemble des proto-mots de G



$L_4 = \{\varepsilon, abc, aabbcc, aaabbbccc, \ldots\}$

$G = \langle \{S, S_1, S_2\}, \{a, b, c\}, \{S \to aS_1c, S_1 \to b|SS_2, cS_2 \to S_2c, bS_2 \to bb\}, S\rangle$.

Sous-Ensemble des proto-mots de G



Sens de dérivation

$G = \langle \{E, T, F\}, \{+, *, a\}, \{E \rightarrow T + E | T, T \rightarrow F * T | F, F \rightarrow a\}, E \rangle$

- Les proto-mots générées lors d'une dérivation peuvent comporter plus d'un symbole non terminal :
  $E \Rightarrow T + E \Rightarrow T + T \Rightarrow F + T \Rightarrow F + F * T \Rightarrow F + a * T \Rightarrow F + a * F \Rightarrow a + a * F \Rightarrow a + a * a$

Sens de dérivation

$G = \langle \{E, T, F\}, \{+, *, a\}, \{E \to T + E | T, T \to F * T | F, F \to a\}, E \rangle$

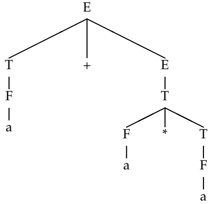
- Les proto-mots générées lors d'une dérivation peuvent comporter plus d'un symbole non terminal :
  $E \Rightarrow T + E \Rightarrow T + T \Rightarrow F + T \Rightarrow F + F * T \Rightarrow F + a * T \Rightarrow F + a * F \Rightarrow a + a * F \Rightarrow a + a * a$
- Dérivation droite : on réécrit le non terminal le plus à droite :
  $E \Rightarrow T + E \Rightarrow T + T \Rightarrow T + F * T \Rightarrow T + F * F \Rightarrow T + F * a \Rightarrow T + a * a \Rightarrow F + a * a \Rightarrow a + a * a$

Sens de dérivation

$G = \langle \{E, T, F\}, \{+, *, a\}, \{E \rightarrow T + E | T, T \rightarrow F * T | F, F \rightarrow a\}, E \rangle$

- Les proto-mots générées lors d'une dérivation peuvent comporter plus d'un symbole non terminal :
  $E \Rightarrow T + E \Rightarrow T + T \Rightarrow F + T \Rightarrow F + F * T \Rightarrow F + a * T \Rightarrow F + a * F \Rightarrow a + a * F \Rightarrow a + a * a$
- Dérivation droite : on réécrit le non terminal le plus à droite :
  $E \Rightarrow T + E \Rightarrow T + T \Rightarrow T + F * T \Rightarrow T + F * F \Rightarrow T + F * a \Rightarrow T + a * a \Rightarrow F + a * a \Rightarrow a + a * a$
- Dérivation gauche : on réécrit le non terminal le plus à gauche :
  $E \Rightarrow T + E \Rightarrow F + E \Rightarrow a + E \Rightarrow a + T \Rightarrow a + F * T \Rightarrow a + a * T \Rightarrow a + a * F \Rightarrow a + a * a$

Arbre de dérivation



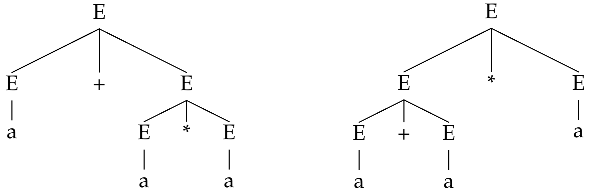
Un arbre de dérivation pour G ($G = \langle N, \Sigma, P, S \rangle$) est un arbre ordonné et étiqueté dont les étiquettes appartiennent à l'ensemble $N \cup \Sigma \cup \{\varepsilon\}$. Si un nœud de l'arbre est étiqueté par le non terminal A et ses fils sont étiquetés $X_1, X_2, ..., X_n$ alors la règle $A \rightarrow X_1, X_2, ..., X_n$ appartient à P.

Arbre de dérivation

- Un arbre de dérivation indique les règles qui ont été utilisées dans une dérivation, mais pas l'ordre dans lequel elles ont été utilisées.
- A un arbre de dérivation correspondent une seule dérivation droite et une seule dérivation gauche.

Ambiguïté

Une grammaire G est **ambiguë** s'il existe au moins un mot m dans $L(G)$ auquel correspond plus d'un arbre de dérivation.
Exemple : $E \rightarrow E + E | E * E | a$



Types de règles

Les grammaires peuvent être classées en fonction de la forme de leurs règles de production. On définit cinq types de règles de production :

- Une règle est régulière à gauche si et seulement si elle est de la forme $A \rightarrow xB$ ou $A \rightarrow x$ avec $A, B \in N$ et $x \in \Sigma$.
- Une règle est régulière à droite si et seulement si elle est de la forme $A \rightarrow Bx$ ou $A \rightarrow x$ avec $A, B \in N$ et $x \in \Sigma$.
- Une règle $A \rightarrow \alpha$ est un règle hors-contexte si et seulement si : $A \in N$ et $\alpha \in (N \cup \Sigma)^*$

Types de règles

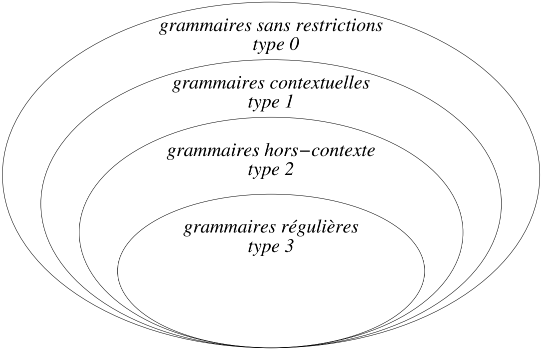
- Une règle $\alpha \rightarrow \beta$ est une règle contextuelle si et seulement si : $\alpha = gAd$ et $\beta = gBd$ avec $g, d, B \in (N \cup \Sigma)^*$ et $A \in N$.
  Le nom "contextuelle" provient du fait que A se réecrit B uniquement dans le contexte g_d.
- Une règle $\alpha \rightarrow \beta$ est une règle sans restriction si et seulement si : $|\alpha| \geq 1$

Type d'une grammaire

Une grammaire est :

- régulière ou de type 3 si elle est régulière à droite ou régulière à gauche. Une grammaire est régulière à gauche si toutes ses règles sont régulières à gauche et une grammaire est régulière à droite si toutes ses règles sont régulières à droite.
- hors contexte ou de type 2 si toutes ses règles de production sont hors contexte.
- dépendante du contexte ou de type 1 si toutes ses règles de production sont dépendantes du contexte.
- sans restrictions ou de type 0 si toutes ses règles de production sont sans restrictions.

Hiérarchie de Chomsky



Type d'un langage

Un langage pouvant être généré par une grammaire de type x et pas par une grammaire d'un type supérieur dans la hiérarchie, est appelé un langage de type x.

Type	Nom
3	régulier
2	hors contexte
1	dépendant du contexte
0	récursivement énumérable

Exemples de langages réguliers

$$L = \{m \in \{a,b\}^*\}$$

$$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0\}$$

$$L = \{m \in \{a,b\}^* | m = xaaa \text{ avec } x \in \{a,b\}^*\}$$

$$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0 \text{ et} |m|_b \mod 2 = 0\}$$

Exemples de langages réguliers

$L = \{m \in \{a,b\}^*\}$

$G = \langle \{S\}, \{a,b\}, \{S \rightarrow aS|bS|\varepsilon\}, S \rangle$

$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0\}$

$L = \{m \in \{a,b\}^* | m = xaaa \text{ avec } x \in \{a,b\}^*\}$

$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0 \text{ et} |m|_b \mod 2 = 0\}$

Exemples de langages réguliers

$L = \{m \in \{a,b\}^*\}$

$G = \langle \{S\}, \{a,b\}, \{S \to aS|bS|\varepsilon\}, S\rangle$

$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0\}$

$G = \langle \{S,T\}, \{a,b\}, \{S \to aT|bS|\varepsilon, T \to aS|bT\}, S\rangle$

$L = \{m \in \{a,b\}^* | m = xaaa \text{ avec } x \in \{a,b\}^*\}$

$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0 \text{ et} |m|_b \mod 2 = 0\}$

Exemples de langages réguliers

$L = \{m \in \{a,b\}^*\}$

$G = \langle \{S\}, \{a,b\}, \{S \to aS|bS|\varepsilon\}, S \rangle$

$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0\}$

$G = \langle \{S,T\}, \{a,b\}, \{S \to aT|bS|\varepsilon, T \to aS|bT\}, S \rangle$

$L = \{m \in \{a,b\}^* | m = xaaa \text{ avec } x \in \{a,b\}^*\}$

$G = \langle \{S,T,U\}, \{a,b\}, \{S \to aS|bS|aT, T \to aU, U \to a\}, S \rangle$

$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0 \text{ et} |m|_b \mod 2 = 0\}$

Exemples de langages réguliers

$L = \{m \in \{a,b\}^*\}$

$G = \langle \{S\}, \{a,b\}, \{S \rightarrow aS|bS|\varepsilon\}, S\rangle$

$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0\}$

$G = \langle \{S,T\}, \{a,b\}, \{S \rightarrow aT|bS|\varepsilon, T \rightarrow aS|bT\}, S\rangle$

$L = \{m \in \{a,b\}^* | m = xaaa \text{ avec } x \in \{a,b\}^*\}$

$G = \langle \{S,T,U\}, \{a,b\}, \{S \rightarrow aS|bS|aT, T \rightarrow aU, U \rightarrow$
$a\}, S\rangle$
$L = \{m \in \{a,b\}^* \mid |m|_a \mod 2 = 0 \text{ et} |m|_b \mod 2 = 0\}$

$G = \langle \{S,T,U,V\}, \{a,b\},$
$\{S \rightarrow aT|bU, T \rightarrow aS|bV, V \rightarrow aU|bT, U \rightarrow aV|bS|\varepsilon\}, S\rangle$

Exemples de langages hors-contexte

$L = \{a^n b^n \mid n \geq 0\}$

$L = \{mm^{-1} \mid m \in \{a, b\}^*\}$ (langage miroir)

Exemples de langages hors-contexte

$L = \{a^n b^n \mid n \geq 0\}$

$G = \langle \{S\}, \{a, b\}, \{S \rightarrow aSb | \varepsilon\}, S \rangle$

$L = \{mm^{-1} \mid m \in \{a, b\}^*\}$ (langage miroir)

Exemples de langages hors-contexte

$L = \{a^n b^n \mid n \geq 0\}$

$G = \langle \{S\}, \{a, b\}, \{S \rightarrow aSb|\varepsilon\}, S \rangle$

$L = \{mm^{-1} \mid m \in \{a, b\}^*\}$ (langage miroir)

$G = \langle \{S\}, \{a, b\}, \{S \rightarrow aSa|bSb|aa|bb\}, S \rangle$

Exemples de langages contextuels

$$L = \{a^n b^n c^n \mid n \geq 0\}$$

Exemples de langages contextuels

$L = \{a^n b^n c^n \mid n \geq 0\}$

$G = \langle \{S, B, W, X\}, \{a, b, c\}, \{\, S \rightarrow abc,$
$S \rightarrow aSBc,$
$cB \rightarrow WB,$
$WB \rightarrow WX,$
$WX \rightarrow BX,$
$BX \rightarrow BC,$
$bB \rightarrow bb\},$
$S \rangle$

Dérivation de $a^3b^3c^3$

$$
\begin{aligned}
S &\Rightarrow_2 aSBc \\
&\Rightarrow_2 aaSBcBc \\
&\Rightarrow_1 aaabcBcBc \\
&\Rightarrow_3 aaabWBcBc \\
&\Rightarrow_4 aaabWXcBc \\
&\Rightarrow_5 aaabBXcBc \\
&\Rightarrow_6 aaabBccBc \\
&\Rightarrow_3 aaabBcWBc \\
&\Rightarrow_4 aaabBcWXc \\
&\Rightarrow_5 aaabBcBXc \\
&\Rightarrow_6 aaabBcBcc \\
&\Rightarrow_3 aaabBWBcc \\
&\Rightarrow_4 aaabBWXcc \\
&\Rightarrow_5 aaabBBXcc \\
&\Rightarrow_6 aaabBBccc \\
&\Rightarrow_7 aaabbBccc \\
&\Rightarrow_7 aaabbbccc
\end{aligned}
$$

Exemples de langages récursivement énumérables

- $L = \{m\sharp m \mid \text{avec} m \in \{a,b\}^*\}$
- $L = \{a^{2^n} \mid \text{avec } n \geq 0\}$
- $L = \{\sharp x_1 \sharp x_2 \sharp \ldots \sharp x_l \mid x_i \in \{0,1\}^* \text{ et } x_i \neq x_j \text{ pour tout } i \neq j\}$

Grammaire v/s Reconnaisseur

- Une grammaire d'un langage L permet de générer tous les mots appartenant à L.
- Un reconnaisseur pour un langage L est un programme qui prend en entrée un mot m et répond oui si m appartient à L et non sinon.
- Pour chaque classe de grammaire, il existe une classe de reconnaisseurs qui définit la même classe de langages.

Type de grammaire	Type de reconnaisseur
régulière	Automate fini
hors contexte	Automate à pile
dépendantes du contexte	*Linear Bounded Automaton*
sans restriction	Machine de Turing