Langages et Automates
Langages réguliers et Grammaires

Engel Lefaucheux

Prépas des INP

Rappels et exercices

- $L_1 \cdot L_2$ le langage obtenu en concaténant un mot de L_1 et un mot de L_2.
- $L_1^n = \underbrace{L_1 \cdot L_1 \dots L_1 \cdot L_1}_{n \text{ times}}$
- $L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \dots$

Les mots $\{\varepsilon, a, babar\}$ appartiennent-ils aux langages suivants

- $\{a, b\}^* \cdot \{r\}$
- $(\{a, b\}^* \cdot \{r\})^*$
- $((\{a, b\}^2 \setminus \{ba\})^* \cdot \{r\})^*$

Exercice retour de la commutativité

Lemma

Deux mots u et v commutent si et seulement s'ils sont puissances d'un même troisième, i.e., s'il existe un mot w et des entiers i, j tels que $u = w^i$ et $v = w^j$.

Rappel : hier on a montré le cas où $|u| = |v|$. Comment l'étendre ?

Outline

Un formalisme pour générer certains langages

Expressions régulières

- Parfois appellées expressions rationnelles
- Génère un langage "régulier"

Définition récursive sur un alphabet $\Sigma = \{a, b\}$:

- ε, a et b sont des expressions régulières pour $\{\varepsilon\}$, $\{a\}$ et $\{b\}$
- Si r_1 et r_2 sont des expressions régulières générant L_1 et L_2, alors
  - $r_1 \cdot r_2$ génère $L_1 \cdot L_2$
  - $r_1 + r_2$ génère $L_1 \cup L_2$
  - r_1^* génère L_1^*
  - (r) est une expression régulière génèrant L_1

$\longrightarrow$ les parenthèses servent à ordonner l'application des opérations

Quelques exemples

Quelles langages pour les expressions régulières suivantes :

- $(a+b)^*$
- $a+b^*$
- $a(a)^*$
- $(a^*b^*)^*$
- $(a+ab^*a)^*$

Quelles expressions rationnelles pour les langages suivants :

- les mots n'ayant que des a ou que des b
- $\{am, bm, an, cn\}$
- les mots de $\{a, i, m, o, u\}^*$ ayant *miaou* en facteur
- $\{a^n b^n \mid n \in \mathbb{N}\}$.

Outline

Une règle d'or

Si des relations existent entre les exposants apparaissant dans la description du langage, alors celui-ci n'est pas régulier.

- $\{a^n b^n \mid n \in \mathbb{N}\}$
- $\{a^n b^m c^k \mid n, m, k \in \mathbb{N} \wedge k \geq n + m\}$

ne sont pas réguliers.

Plusieurs critères formels de non-régularité

- Théorème de Myhill-Nerode (complexe)
- Lemme de l'étoile (simple, mais ne marche pas tout le temps)

Lemme de l'étoile

Theorem

Soit L un langage régulier. Il existe un entier N tel que tout mot w de L de longueur $|w| \geq N$ possède une factorisation $w = xyz$ avec $0 < |y|$ telle que

1. $0 < |xy| \leq N$ *et*
2. $xy^nz \in L$ *pour tout entier* $n \geq 0$.

Lemme de l'étoile

Theorem

Soit L un langage régulier. Il existe un entier N tel que tout mot w de L de longueur $|w| \geq N$ possède une factorisation $w = xyz$ avec $0 < |y|$ telle que

1. $0 < |xy| \leq N$ *et*
2. $xy^n z \in L$ *pour tout entier* $n \geq 0$.

Quid de $\{a^n b^n \mid n \in \mathbb{N}\}$?

Exercice

Les langages suivants sont-ils réguliers ?

- $\{a^n \mid n \text{ est un nombre premier}\}$
- $\{a^n b^m \mid n \neq m\}$
- Le langage des palindromes
- $(ab)^* \cap \{w \mid |w|_a = |w|_b\}$
- $ab(a+b)^* \cap \{w \mid |w|_a = |w|_b\}$

Exercice

Soit $L_1 = \{ab^n \mid n \in \mathbb{N}\}$ et $L_2 = \{a^n b \mid n \in \mathbb{N}\}$
Donner une expression rationelle pour $L_1 \cdot L_2, L_1 \cap L_2$ et L_1^2.

Les langages suivants sont-ils réguliers ?

- Le langage des palindromes sur $\Sigma = \{a, b\}$
- Le langage des palindromes sur $\Sigma = \{a\}$
- $(ab)^* \cap \{w \mid |w|_a = |w|_b\}$
- $ab(a+b)^* \cap \{w \mid |w|_a = |w|_b\}$

Theorem

Soit L un langage régulier. Il existe un entier N tel que tout mot w de L de longueur $|w| \geq N$ possède une factorisation $w = xyz$ avec $0 < |y|$ telle que

1. $0 < |xy| \leq N$ *et*
2. $xy^n z \in L$ *pour tout entier* $n \geq 0$.

Dépasser les expressions régulières

Les langages réguliers reconnaissent

- des mots issus de lexiques
- des structures simples.

Ils sont insuffisants pour

- les structures de type $\{a^n b^n \mid n \in \mathbb{N}\}$
- le langage naturel
- l'analyse de programmes

Outline

Définition formelle

Une grammaire est un quadruplet (T, N, R, S)

- T : symboles terminaux
  $\longrightarrow$ l'alphabet du langage que nous voulons créer
- N : symboles non-terminaux / temporaires
  $\longrightarrow$ symboles utilisés au cours de la dérivation
- R : ensemble des règles de la dérivation
- S : Axiome
  $\longrightarrow$ symbole de départ

$(\{a, b\}, \{S, A\}, R, S)$ où R est l'ensemble de règles suivant

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$

Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$

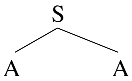
Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$



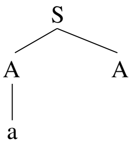
Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$



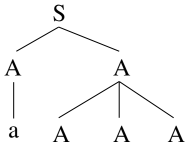
Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$

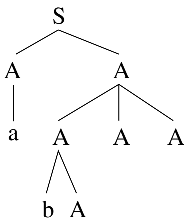


Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$
$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$



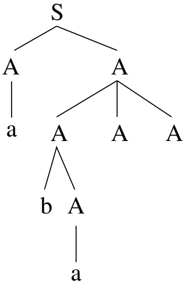
Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$



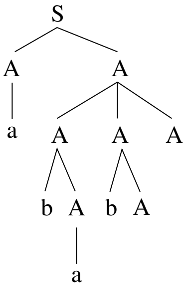
Arbre de dérivation

Représentation graphique de la dérivation

- Racine $=$ axiome
- Noeud $=$ Symbole non-terminal
- Feuille $=$ symbole terminal
- Relation parent-enfant $=$ règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$



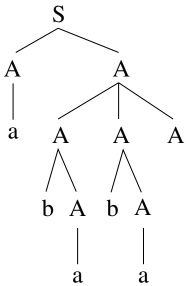
Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$



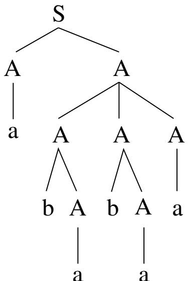
Arbre de dérivation

Représentation graphique de la dérivation

- Racine = axiome
- Noeud = Symbole non-terminal
- Feuille = symbole terminal
- Relation parent-enfant = règle

$S \rightarrow AA$

$A \rightarrow AAA \mid b\ A \mid A\ b \mid a$



Petit exemple de grammaire

On veut construire une PHRASE simple.

$$\begin{aligned}
PHRASE &\rightarrow SUJET\ \ VERBE\ \ COMPLEMENT \\
SUJET &\rightarrow ludovic \mid pierre \mid nicolas \\
VERBE &\rightarrow mange \mid porte \\
COMPLEMENT &\rightarrow ARTICLE\ \ NOM\ \ ADJECTIF \\
ARTICLE &\rightarrow un \mid le \\
NOM &\rightarrow livre \mid plat \mid chat \\
ADJECTIF &\rightarrow delicieux \mid rouge \mid doux
\end{aligned}$$

On part de PHRASE, et on remplace les termes en grâce aux règles

Petit exemple de grammaire

Construction d'une *IFSTRUC*

$$\begin{aligned}
IFSTRUC &\rightarrow if\ \ TEST\ \ then\ BLOCK\ \ else\ BLOCK \\
TEST &\rightarrow VAR\ \ \leq\ \ INT \mid TEST\ \ et\ \ TEST \\
BLOCK &\rightarrow INSTANCE \mid INSTANCE\ \ BLOCK \\
VAR &\rightarrow x \mid y \\
INT &\rightarrow 0 \mid 1 \mid INT\ \ INT \\
INSTANCE &\rightarrow incr\ \ VAR \mid decr\ \ VAR \mid IFSTRUC \mid return\ \ VAR
\end{aligned}$$

Exercices

Quel langage pour les grammaires suivantes commençant par S

1. $S \rightarrow \varepsilon \mid T$
   $T \rightarrow a\ b \mid a\ T\ b$
2. $S \rightarrow \varepsilon \mid A$
   $A \rightarrow a\ B\ C \mid a\ A\ B\ C$
   $C\ B \rightarrow B\ C$
   $A\ B \rightarrow A\ b$
   $a\ B \rightarrow A\ b$
   $B\ B \rightarrow B\ b$
   $C \rightarrow c$

Construisez une grammaire pour le langage des palindromes sur $\Sigma = \{a, b\}$

Outline

Hiérarchie de Chomsky

Type 0 Pas de restrictions sur les règles

Type 1 règles de la forme
$u\ A\ v \ \rightarrow\ u\ w\ v$ avec $A \in N$ et $u, v, w \in (N \cup T)^*$

Type 2 règles de la forme
$A \ \rightarrow\ w$ avec $A \in N$ et $w \in (N \cup T)^*$

Type 3 Toutes les règles sont soit de la forme
$A \ \rightarrow\ a\ B$ ou $A \ \rightarrow\ a$ (grammaire à droite)
soit de la forme
$A \ \rightarrow\ B\ a$ ou $A \ \rightarrow\ a$ (grammaire à gauche)

Hiérarchie de Chomsky

Type 0 Pas de restrictions sur les règles

Type 1 règles de la forme
$u\ A\ v \ \rightarrow\ u\ w\ v$ avec $A \in N$ et $u, v, w \in (N \cup T)^*$

Type 2 règles de la forme
$A \ \rightarrow\ w$ avec $A \in N$ et $w \in (N \cup T)^*$

Type 3 Toutes les règles sont soit de la forme
$A \ \rightarrow\ a\ B$ ou $A \ \rightarrow\ a$ (grammaire à droite)
soit de la forme
$A \ \rightarrow\ B\ a$ ou $A \ \rightarrow\ a$ (grammaire à gauche)

Grammaire de type 0 $\Longleftrightarrow$ Machine de Turing

Hiérarchie de Chomsky

Type 0 Pas de restrictions sur les règles

Type 1 règles de la forme
$u\ A\ v \ \rightarrow \ u\ w\ v$ avec $A \in N$ et $u, v, w \in (N \cup T)^*$

Type 2 règles de la forme
$A \ \rightarrow \ w$ avec $A \in N$ et $w \in (N \cup T)^*$

Type 3 Toutes les règles sont soit de la forme
$A \ \rightarrow \ a\ B$ ou $A \ \rightarrow \ a$ (grammaire à droite)
soit de la forme
$A \ \rightarrow \ B\ a$ ou $A \ \rightarrow \ a$ (grammaire à gauche)

Grammaire de type 0 $\Longleftrightarrow$ Machine de Turing
Grammaire de type ? $\Longleftrightarrow$ Expression régulière

Hiérarchie de Chomsky

Type 0 Pas de restrictions sur les règles

Type 1 règles de la forme
$u\ A\ v \ \rightarrow\ u\ w\ v$ avec $A \in N$ et $u, v, w \in (N \cup T)^*$

Type 2 règles de la forme
$A \ \rightarrow\ w$ avec $A \in N$ et $w \in (N \cup T)^*$

Type 3 Toutes les règles sont soit de la forme
$A \ \rightarrow\ a\ B$ ou $A \ \rightarrow\ a$ (grammaire à droite)
soit de la forme
$A \ \rightarrow\ B\ a$ ou $A \ \rightarrow\ a$ (grammaire à gauche)

Grammaire de type 0 $\Longleftrightarrow$ Machine de Turing
Grammaire de type 3 $\Longleftrightarrow$ Expression régulière

Exercices

Représenter les langages suivant avec une grammaire de type 3

- $baab^*$
- $b(aab)^*$

De quel type est la grammaire
$S \rightarrow aU \mid c$
$U \rightarrow Sb \mid d$
Quel est son langage ?

L_1 et L_2 langages de grammaire G_1 et G_2
Informellement, comment construire une grammaire pour
$L_1 \cup L_2$, $L_1 \cdot L_2$ et L_1^*

Exercices

Représenter les langages suivant avec une grammaire de type 3

- $baab^*$
- $b(aab)^*$

De quel type est la grammaire
$S \rightarrow aU \mid c$
$U \rightarrow Sb \mid d$

Quel est son langage ? $\{a^n cb^n, a^{n+1} db^n \mid n \in \mathbb{N}\}$

L_1 et L_2 langages de grammaire G_1 et G_2
Informellement, comment construire une grammaire pour $L_1 \cup L_2$, $L_1 \cdot L_2$ et L_1^*