Langages et Automates
Langages et Grammaires

Engel Lefaucheux

Prépas des INP

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

Objectif de ce cours

- Découvrir la notion de grammaire
  - Définitions
  - Dérivations
  - Classes de grammaire
- Simplification d'une grammaire
  - Forme normale

Plan

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

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

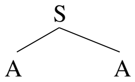
Arbre de dérivation

Représentation graphique de la dérivation

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

$S \to AA$

$A \to AAA \mid b\ A \mid A\ b \mid a$



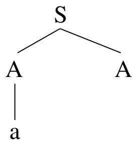
Arbre de dérivation

Représentation graphique de la dérivation

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

$S \rightarrow AA$

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



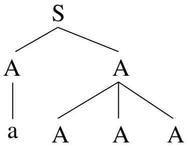
Arbre de dérivation

Représentation graphique de la dérivation

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

$S \rightarrow AA$

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



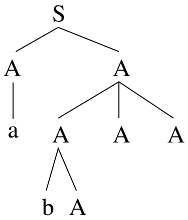
Arbre de dérivation

Représentation graphique de la dérivation

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

$S \rightarrow AA$

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



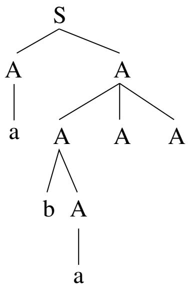
Arbre de dérivation

Représentation graphique de la dérivation

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

$S \rightarrow AA$

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



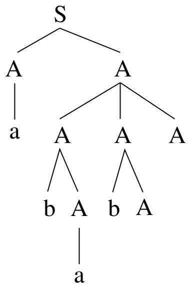
Arbre de dérivation

Représentation graphique de la dérivation

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

$S \rightarrow AA$

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



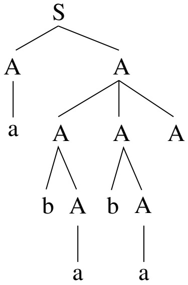
Arbre de dérivation

Représentation graphique de la dérivation

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

$S \rightarrow AA$

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



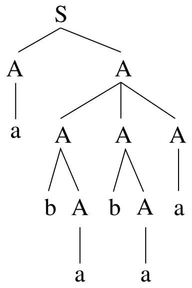
Arbre de dérivation

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

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^*

Outline

Simplifier une grammaire

$\Rightarrow$ Supprimer les éléments **inutiles** de la grammaire

- Symboles **improductifs**
  - A est improductif s'il n'y a pas de $m \in T^*$ tel que $A \xrightarrow{*} m$
- Symboles **inaccessibles**
  - A est inaccessible s'il n'y a pas de α et β tels que $S \xrightarrow{*} \alpha A \beta$
- ϵ-productions
  - Une ϵ-production est une dérivation telle que $A \xrightarrow{*} \epsilon$
- Production simple
  - $A \rightarrow B$ est une production simple si $A \in N$ et $B \in N$

$\Rightarrow$ Pour toute grammaire, il existe une grammaire équivalente sans symboles improductifs ni inaccessibles, sans ϵ-productions ni productions simples

Élimination des symboles improductifs

- Calcul des symboles productifs (grammaire de type 2 ou 3)
  - Soit $P_1 = \{A \in N \mid \exists w \in T^*, A \to w \in R\}$
  - Tant que $P_i \neq P_{i+1}$
    $P_{i+1} = P_i \cup \{A \in N \mid \exists w \in (T \cup P_i)^*, A \to w \in R\}$

Que contient P_i ?

Élimination des symboles improductifs

▷ Calcul des symboles productifs (grammaire de type 2 ou 3)

- Soit $P_1 = \{A \in N \mid \exists w \in T^*, A \to w \in R\}$
- Tant que $P_i \neq P_{i+1}$
  $P_{i+1} = P_i \cup \{A \in N \mid \exists w \in (T \cup P_i)^*, A \to w \in R\}$

Que contient P_i ?

$\Longrightarrow$ Les symboles de $N \setminus P$ sont improductifs

$\Longrightarrow$ Enlever ces symboles et les règles associées

Élimination des symboles inaccessibles

▷ Calcul des symboles accessibles (grammaire de type 2 ou 3)

- Soit $C_1 = \{S\}$
- Tant que $C_i \neq C_{i+1}$
  $C_{i+1} = C_i \cup \{A \in N \mid \exists u, v \in (N \cup T)^*, X \in C_i, X \rightarrow uAw \in R\}$

Que contient C_i ?

Élimination des symboles inaccessibles

- Calcul des symboles accessibles (grammaire de type 2 ou 3)
  - Soit $C_1 = \{S\}$
  - Tant que $C_i \neq C_{i+1}$
    $C_{i+1} = C_i \cup \{A \in N \mid \exists u, v \in (N \cup T)^*, X \in C_i, X \rightarrow uAw \in R\}$

Que contient C_i ?

$\Longrightarrow$ Les symboles de $N \setminus C$ sont inaccessibles.

$\Longrightarrow$ Enlever ces symboles et les règles associées

Élimination des ε-production

▷ Calcul des symboles annulables (grammaire de type 2 ou 3)

- Soit $U_1 = \{A \in N \mid A \to \varepsilon \in R\}$
- Tant que $U_i \neq U_{i+1}$
  $U_{i+1} = U_i \cup \{A \in N \mid \exists u \in (U_i)^*, A \to \alpha \in R\}$

$\Longrightarrow$ Les symboles de U sont annulables.

$\Longrightarrow$ Modification de la grammaire

- Pour $A \in U$, remplacer les règles $X \to uAv$
  par $X \to uAv \mid uv$
- Supprimer les règles $A \to \varepsilon$ (sauf pour S)

Équivalences et productions simples

- Productions simples, dérivations et classes d'équivalences
  - Production simple : toute règle $A \to B$ avec $B \in N$
  - Soit la relation $\geqslant$ telle que $A \geqslant B$ si $A \xrightarrow{*} B$
  - Soit la relation $\approx$ telle que $A \approx B$ si $A \geqslant B$ et $B \geqslant A$
  - **Classes d'équivalences**
    - Si $A \approx B$, tout ce qui est dérivé de A peut l'être de B
    - Relation réflexive, symétrique et transitive
    - L'ensemble des classes est une **partition** de N
- Modification de la grammaire
  - On conserve les productions non-simples
  - Pour chaque classe d'équivalence
    - $\Rightarrow$ Choisir un symbole qui remplace tous les autres
  - $\Rightarrow$ Pour chaque dérivation $A \xrightarrow{*} B$
    - Pour chaque $B \to \beta$, ajouter $A \to \beta$

Exemple de simplification de grammaire

- Grammaire
  1. $S \rightarrow T|U$
  2. $U \rightarrow aYb|V$
  3. $V \rightarrow W$
  4. $X \rightarrow W|a$
  5. $Y \rightarrow Z$
  6. $Z \rightarrow c|\epsilon$
- Étapes
  - Symboles productifs :
  - Symboles accessibles :
  - ϵ-productions :
  - Productions simples :

Exemple de simplification de grammaire

- Grammaire
  1. $S \to T|U$
  2. $U \to aYb|V$
  3. $V \to W$
  4. $X \to W|a$
  5. $Y \to Z$
  6. $Z \to c|\epsilon$
- Étapes
  - Symboles productifs : $\{X, Z, Y, U, S\} \Rightarrow$ retirer T, V et W
  - Symboles accessibles :
  - ϵ-productions :
  - Productions simples :

Exemple de simplification de grammaire

- Grammaire
  1. $S \rightarrow U$
  2. $U \rightarrow aYb$
  3.
  4. $X \rightarrow a$
  5. $Y \rightarrow Z$
  6. $Z \rightarrow c|\epsilon$
- Étapes
  - Symboles productifs : $\{X, Z, Y, U, S\} \Rightarrow$ retirer T, V et W
  - Symboles accessibles : $\{S, U, Y, Z\} \Rightarrow$ retirer X
  - ϵ-productions :
  - Productions simples :

Exemple de simplification de grammaire

- Grammaire
  1. $S \rightarrow U$
  2. $U \rightarrow aYb$
  3.
  4.
  5. $Y \rightarrow Z$
  6. $Z \rightarrow c|\epsilon$
- Étapes
  - Symboles productifs : $\{X, Z, Y, U, S\} \Rightarrow$ retirer T, V et W
  - Symboles accessibles : $\{S, U, Y, Z\} \Rightarrow$ retirer X
  - ϵ-productions :
  - Productions simples :

Exemple de simplification de grammaire

- Grammaire
  1. $S \rightarrow U$
  2. $U \rightarrow aYb$
  3.
  4.
  5. $Y \rightarrow Z$
  6. $Z \rightarrow c|\epsilon$
- Étapes
  - Symboles productifs : $\{X, Z, Y, U, S\} \Rightarrow$ retirer T, V et W
  - Symboles accessibles : $\{S, U, Y, Z\} \Rightarrow$ retirer X
  - ϵ-productions : $\{Z, Y\} \Rightarrow$ modifier 6, 2
  - Productions simples :

Exemple de simplification de grammaire

- Grammaire
  1. $S \rightarrow U$
  2. $U \rightarrow aYb|ab$
  3.
  4.
  5. $Y \rightarrow Z$
  6. $Z \rightarrow c$
- Étapes
  - Symboles productifs : $\{X, Z, Y, U, S\}$ $\Rightarrow$ retirer T, V et W
  - Symboles accessibles : $\{S, U, Y, Z\}$ $\Rightarrow$ retirer X
  - ϵ-productions : $\{Z, Y\}$ $\Rightarrow$ modifier 6, 2
  - Productions simples :

Exemple de simplification de grammaire

- Grammaire
  1. $S \rightarrow U$
  2. $U \rightarrow aYb|ab$
  3.
  4.
  5. $Y \rightarrow Z$
  6. $Z \rightarrow c$
- Étapes
  - Symboles productifs : $\{X, Z, Y, U, S\} \Rightarrow$ retirer T, V et W
  - Symboles accessibles : $\{S, U, Y, Z\} \Rightarrow$ retirer X
  - ϵ-productions : $\{Z, Y\} \Rightarrow$ modifier 6, 2
  - Productions simples : $S \rightarrow U$ et $Y \rightarrow Z \Rightarrow$ modifier 1, 2, 5, 6

Exemple de simplification de grammaire

- Grammaire
  1. $S \rightarrow aYb|ab$
  2.
  3.
  4.
  5. $Y \rightarrow c$
  6.
- Étapes
  - Symboles productifs : $\{X, Z, Y, U, S\} \Rightarrow$ retirer T, V et W
  - Symboles accessibles : $\{S, U, Y, Z\} \Rightarrow$ retirer X
  - ϵ-productions : $\{Z, Y\} \Rightarrow$ modifier 6, 2
  - Productions simples : $S \rightarrow U$ et $Y \rightarrow Z \Rightarrow$ modifier 1, 2, 5, 6

Exercice

- Réduire les grammaires suivantes
  - G_1
    - $S \rightarrow bSc|bTc|a|\epsilon$
    - $T \rightarrow U$
    - $U \rightarrow bUc|T$
    - $V \rightarrow U|bc$
  - G_2
    - $S \rightarrow UXT$
    - $T \rightarrow b$
    - $U \rightarrow aV|aXTXb$
    - $V \rightarrow cV|aWT$
    - $W \rightarrow V$
    - $X \rightarrow ab|\epsilon$
    - $Y \rightarrow cZ$
    - $Z \rightarrow aa$

Formes normales

- **Forme normale de Chomsky** : toutes les règles sont de la forme :
  $A \to BC$ avec $A, B, C \in N$
  ou
  $A \to a$ avec $a \in T$

- **Forme normale de Greibach** : toutes les règles sont de la forme :
  $A \to aw$ avec $w \in N^*$

Pour tout langage hors-contexte (notamment les langages réguliers) il existe une grammaire en forme normale de Chomsky et une grammaire en forme normale de Greibach qui le génèrent