Grammaires non contextuelles

1

- AF : « machines » pour reconnaître les chaînes d'un langage
  => reconnaissance d'un langage
- ER : « notation » pour décrire les chaînes d'un langage
  => spécification d'un langage
- grammaires : règles de réécriture pour produire les chaînes d'un langage
  => génération d'un langage

2

Expressivité des grammaires

- Il existe différentes formes de grammaires qui décrivent des classes différentes de langages
- Rappel de la Classification de Chomsky :

Classes de langages	**Types de machines**	**Types de grammaires**
Réguliers	Automates finis	Type 3 : régulières
Non contextuels	Automates à pile	Type 2 : non contextuelles
Contextuels		Type 1 : contextuelles
Récursivement énumérables	Machines de Turing	Type 0 : sans restriction

3

Ce que peut exprimer une grammaire non contextuelle
et que ne peuvent pas exprimer les ER ou les AF :
la structure récursive des phrases.

Exemple :
si I1 et I2 sont des instructions
et E est une expression
alors si E alors I1 sinon I2 est une instruction

Inst → si Expr alors Instr sinon Instr

=> Les grammaires vont être parfaitement adaptées pour exprimer la syntaxe des langages de programmation

4

Présentation informelle

Les grammaires permettent d'exprimer des définitions récursives

Exemple 1 : $L = \{a^n b^n \mid n \in N\}$

Déf. récursive : <u>base</u> : $\varepsilon \in L$
<u>récur</u> : si $w \in L$ ($w = a^k b^k$)
alors $awb \in L$ ($awb = a^{k+1} b^{k+1}$)

Grammaire correspondante :

(1) ⟨mot⟩ → ε
(2) ⟨mot⟩ → a ⟨mot⟩ b

5

Exemple 2 : expressions arithmétiques (EA)

formées avec des constantes numériques, les opérateurs + et *, et des parenthèses

- Définition récursive:

<u>base</u> : une constante numérique est une EA
<u>récur</u> : si e1 et e2 sont des EA
alors e1 + e2
e1 * e2
(e1) sont des EA

- Grammaire non contextuelle correspondante :

(1) ⟨expr⟩ → const
(2) ⟨expr⟩ → ⟨expr⟩ + ⟨expr⟩
(3) ⟨expr⟩ → ⟨expr⟩ * ⟨expr⟩
(4) ⟨expr⟩ → (⟨expr⟩)

6

Définition

Une grammaire non contextuelle (GNC) est un quadruplet $G = (V_T, V_N, S, P)$ où

- V_T est un ensemble fini de symboles terminaux (ou alphabet)
- V_N est un ensemble fini de symboles non terminaux, ou catégories syntaxiques
- $S \in V_N$ est le symbole non terminal initial (start), ou axiome de la grammaire
- P est un ensemble de productions de la forme $A \to \alpha$ avec $A \in V_N$ et $\alpha \in (V_T \cup V_N)^*$

Abréviation : Un ensemble de règles ayant même tête : $A \to \alpha_1, \ldots, A \to \alpha_n$ peut s'écrire $A \to \alpha_1 | \alpha_2 | \ldots | \alpha_n$

7

Exemple : affectation (simplifiée)

P = { Inst $\to$ ident ':=' Expr
Expr $\to$ ident | nbr | Expr + Expr }
V_N = {Inst, Expr}
V_T = {ident, nbr, :=, +}
S = Inst

Inst $\Rightarrow$ ident ':=' Expr
$\Rightarrow$ ident ':=' Expr + Expr
$\Rightarrow$ ident ':=' nbr + Expr
$\Rightarrow$ ident ':=' nbr + nbr

8

Langage défini par une grammaire

- C'est l'ensemble des chaînes que l'on peut obtenir à partir du symbole initial en appliquant les règles
- Principe
  - On part du symbole initial (la chaîne en construction = S)
  - On applique une règle : on remplace la partie gauche de la règle (sa tête) par sa partie droite (son corps) dans la chaîne en construction
  - On continue à appliquer des règles jusqu'à ce que la chaîne ne contienne plus que des symboles terminaux
- Ce processus consistant à appliquer des règles pour construire des chaînes s'appelle dérivation

9

Dérivations

- Si $A \to \beta$ est une production alors on dit que $\alpha_1 A \alpha_2$ se dérive en 1 étape en $\alpha_1 \beta \alpha_2$ ce que l'on note $\alpha_1 A \alpha_2 \Rightarrow \alpha_1 \beta \alpha_2$
- On étend la relation $\Rightarrow$ pour exprimer les dérivations en zéro ou plusieurs étapes, ce que l'on note $\Rightarrow^*$

$\alpha \Rightarrow^* \alpha'$
se lit
« α se dérive en 0, 1 ou ++ étapes en α' »

10

Phrases et Langages

- Une chaîne qui peut être dérivée à partir de S est appelée:
  - une phrase si elle est constituée uniquement de symboles terminaux
  - Une protophrase si elle contient au moins un symbole non terminal

Soit $G = (V_T, V_N, S, P)$ une GNC

- Le langage engendré par G, noté L(G), est l'ensemble des chaînes de symboles terminaux ω qui peuvent être dérivées à partir de S :

$$L(G) = \{\omega \in V_T^* \ / \ S \Rightarrow^* \omega\}$$

11

Stratégies de dérivation

À chaque étape de dérivation, il faut faire 2 choix :

- Quel symbole NT remplacer ?
- Une fois choisi le symbole NT, quelle production utiliser ?

Une dérivation gauche (resp. droite) est une dérivation où, à chaque étape de dérivation, c'est le symbole non terminal le plus à gauche (resp. droite) qui est remplacé

Si α se dérive en β par une dérivation gauche (resp. droite), on le note $\alpha \Rightarrow_g^* \beta$ (resp. $\alpha \Rightarrow_d^* \beta$)

Important : La stratégie de dérivation ne change pas les phrases que l'on peut produire

12

Arbre d'analyse ou Arbre de dérivation

- C'est une représentation graphique d'une dérivation où on ne précise pas l'ordre de remplacement des symboles non terminaux

- Soit $G = (V_T, V_N, S, P)$ une GNC,
  un arbre d'analyse pour G est tel que :
  - La racine est le symbole initial S
  - Chaque feuille est soit ε, soit un symbole terminal
  - Chaque nœud interne est un symbole non terminal X et ses fils sont les symboles de la partie droite d'une règle $X \rightarrow X_1X_2 \ldots X_n$

  Les feuilles de l'arbre, lues de gauche à droite, forment une phrase ω de L(G)

- L'ordre de remplacement des symboles NT n'étant pas précisé, un arbre d'analyse peut représenter plusieurs dérivations possibles d'une même chaîne

- Par contre, il représente une unique dérivation gauche (ou droite)

=> Un arbre d'analyse est équivalent à une dérivation gauche :

Un arbre $\rightarrow$ une dérivation gauche unique
Une dérivation gauche $\rightarrow$ un arbre unique

Mais il peut arriver qu'une chaîne soit obtenue par ++ dérivations gauches (avoir ++ arbres d'analyse), dans ce cas, la chaîne est dite ambiguë

Les formulations suivantes sont équivalentes :

- ω est engendré par G : $\omega \in L(G)$

- Il existe une dérivation pour ω : $S \Rightarrow^* \omega$

- Il existe une dérivation gauche pour ω : $S \Rightarrow_g^* \omega$

- Il existe un arbre d'analyse pour ω

Langages non contextuels

- Les langages générés par les GNC sont appelés les langages non contextuels

exercice

Pour chacun des langages suivants, donner une grammaire et dire à quelle classe appartient le langage (régulier, non contextuel, ou au delà) :

1. a^*b^*
2. $\{\omega \in \{a,b\}^* \ / \ |\omega|_a \bmod 2 = 0\}$
3. $\{\omega\omega^R \ / \ \omega \in \{a,b\}^*\}$
4. $\{a^nb^pc^pd^n \ / \ n,p \geq 1\}$
5. $\{\omega\omega \ / \ \omega \in \{a,b\}^*\}$

Construction d'une GNC à partir d'un AF

Soit $M = (\Sigma, Q, q_0, F, \delta)$ un AFD (ou un AFN)
On construit $G = (V_T, V_N, S, P)$ une GNC
telle que L(G)=L(M) :

- $V_T = \Sigma$
- $V_N = Q$
- $S = q_0$
- P est constitué des productions :

  $q_i \rightarrow a\, q_j$ dès que $\delta(q_i,a) = q_j$
  et $q_i \rightarrow \varepsilon$ si $q_i \in F$

Langages réguliers et langages non contextuels

- Pour tout AF, on peut construire une GNC qui génère le même langage

  => Les GNC sont au moins aussi puissantes que les AF

- Mais sont-elles plus puissantes que les AF ?

  => Oui, car il existe des langages non contextuels (pour lesquels il existe une GNC) qui ne sont pas réguliers (pour lesquels il n'existe pas d'ER ou d'AF)

  Exemple : $L = \{a^n b^n \mid n \in N\}$

  il existe une GNC pour L, mais il n'existe pas d'ER ni d'AF

=> Les langages réguliers sont un sous-ensemble strict des langages non contextuels

19

Les différentes formes de grammaires

- Les grammaires régulières (type 3) ont des règles de la forme $A \rightarrow aB$ ou $A \rightarrow a$ ou $A \rightarrow \varepsilon$
- Les grammaires non contextuelles (type 2) ont des règles de la forme $A \rightarrow \alpha$
- Les grammaires contextuelles (type 1) ont des règles de la forme $\alpha \rightarrow \beta$ avec $\alpha \neq \varepsilon$ et $|\alpha| \leq |\beta|$
- Les grammaires contextuelles sans restriction (type 0) ont des règles de la forme $\alpha \rightarrow \beta$ avec $\alpha \neq \varepsilon$

20