CSE211 – Formal Languages and Automata Theory

U2L2 – Derivation and its Types

Dr. P. Saravanan
School of Computing
SASTRA Deemed University

Agenda

- Derivation using Grammar
- Types of derivation
- Examples
- Comparison of LM and RM derivation
- Definition of CFL
- Sentential Forms
- Parse Tree
- Examples for parse tree
- Yield of a parse tree

In previous class

- The above productions may be rewritten integrally as
  - $E \rightarrow \text{I} \mid E + E \mid E^*E \mid (E)$
  - $\text{I} \rightarrow a \mid b \mid \text{Ia} \mid \text{Ib} \mid \text{I0} \mid \text{I1}$
- An example of string derivations: $a*(a0+b1)$
  - $E \Rightarrow E*E$
  - $\Rightarrow I*E$
  - $\Rightarrow a*E$
  - $\Rightarrow a*(E)$
  - $\Rightarrow a*(E+E)$
  - $\Rightarrow a*(I0+E) \Rightarrow \ldots \Rightarrow a*(a0+b1)$

CFG Examples

(a) All strings in the language L : $\{a^n b^m a^{2n} \mid n, m \geq 0\}$

$S \rightarrow aSaa \mid B$

$B \rightarrow bB \mid \varepsilon$

(b) All nonempty strings that start and end with the same symbol.

$S \rightarrow aXa \mid bXb \mid a \mid b$

$X \rightarrow aX \mid bX \mid \varepsilon$

(c) All strings with more a's than b's.

$S \rightarrow Aa \mid MS \mid SMA$

$A \rightarrow Aa \mid \varepsilon$

$M \rightarrow \varepsilon \mid MM \mid bMa \mid aMb$

(d) All palindromes

$S \rightarrow aSa \mid bSb \mid a \mid b \mid \varepsilon$

Derivations Using a Grammar

- **Concept:** we use productions to generate (“infer”) strings in the language described by the grammar.
- Two ways for such *string inference:*
- *Recursive inference:*
  - bottom up (“from body to head”);
  - starting from known strings (often from terminals in productions)
- *Derivation:*
  - top down (“from head to body”) in concept;
  - as shown by the examples given before.

Derivation(head -> body)

- Show a top-down derivation of the string $w = a*(a + b00)$ using the productions of the CFG G_r described previously
- w above is not a regular expression but an arithmetic expression.

Production Rules:

$E \rightarrow I$
$E \rightarrow E + E$
$E \rightarrow E*E$
$E \rightarrow (E)$
$I \rightarrow a$
$I \rightarrow b$
$I \rightarrow Ia$
$I \rightarrow Ib$
$I \rightarrow I0$
$I \rightarrow I1$

Derivation of $a*(a + b00)$

E	$\Rightarrow E*E$	apply (3)
	$\Rightarrow I*E$	apply (1)
	$\Rightarrow a*E$	apply (5)
	$\Rightarrow a*(E)$	apply (4)
	$\Rightarrow a*(E + E)$	apply (2)
	$\Rightarrow a*(I + E)$	apply (1)
	$\Rightarrow a*(a + E)$	apply (5)
	$\Rightarrow a*(a + I)$	apply (1)
	$\Rightarrow a*(a + I0)$	apply (9)
	$\Rightarrow a*(a + I00)$	apply (9)
	$\Rightarrow a*(a + b00)$	apply (6)

Recursive Inference

- Show a bottom-up recursive inference of the string $w = a*(a + b00)$ using the productions of the CFG G_r described previously
- w above is not a regular expression but an arithmetic expression.

Derivation step	String inferred	For language of	Production used	String(s) used
(i)	a	I	5: $(I \to a)$	–
(ii)	b	I	6: $(I \to b)$	–
(iii)	$b0$	I	9: $(I \to I0)$	(ii)
(iv)	$b00$	I	9: $(I \to I0)$	(iii)
(v)	a	E	1: $(E \to I)$	(i)
(vi)	$b00$	E	1: $(E \to I)$	(iv)
(vii)	$a + b00$	E	2: $(E \to E + E)$	(v), (vi)
(viii)	$(a + b00)$	E	4: $(E \to (E))$	(vii)
(ix)	$a*(a + b00)$	E	3: $(E \to E*E)$	(v), (viii)

Fig 5.1 Inference of a string $w = a*(a+b00)$.

Notations used in derivations

- If $\alpha A \beta$ is a string of terminals and variables, and if $A \to \gamma$ is a production, then we write

$$\alpha A \beta \underset{G}{\Rightarrow} \alpha \gamma \beta$$

- to denote a *derivation*.
- For zero and more derivations, we use the following notation.

$$A \underset{G}{\overset{*}{\Rightarrow}} \gamma$$

- The label G under the double arrow may be omitted if which grammar is being used is understood.

Types of Derivations

- **Definitions:**
- *Leftmost derivation:* Replacing the leftmost variable in each derivation step (represented by the notation or, for typing convenience, also by $\Rightarrow_{lm}$)
- *Rightmost derivation:* Replacing the rightmost variable in each derivation step (represented by $\underset{rm}{\Rightarrow}$ or by $\Rightarrow_{rm}$)

Left most Derivation

- The leftmost derivation of the string $w = a*(a+b00)$ of Example 5.5 is as follows

Production Rules:

$E \to I$	(1)
$E \to E + E$	(2)
$E \to E*E$	(3)
$E \to (E)$	(4)
$I \to a$	(5)
$I \to b$	(6)
$I \to Ia$	(7)
$I \to Ib$	(8)
$I \to I0$	(9)
$I \to I1$	(10)

Leftmost Derivation of $a*(a + b00)$

E	$\Rightarrow_{lm} E*E$	apply (3)
	$\Rightarrow_{lm} I*E$	apply (1)
	$\Rightarrow_{lm} a*E$	apply (5)
	$\Rightarrow_{lm} a*(E)$	apply (4)
	$\Rightarrow_{lm} a*(E + E)$	apply (2)
	$\Rightarrow_{lm} a*(I + E)$	apply (1)
	$\Rightarrow_{lm} a*(a + E)$	apply (5)
	$\Rightarrow_{lm} a*(a + I)$	apply (1)
	$\Rightarrow_{lm} a*(a + I0)$	apply (9)
	$\Rightarrow_{lm} a*(a + I00)$	apply (9)
	$\Rightarrow_{lm} a*(a + b00)$	apply (6)

Right most Derivation

- The rightmost derivation of the string $w = a*(a+b00)$ of Example 5.5 is as follows

Production Rules:

$E \to I$	(1)
$E \to E + E$	(2)
$E \to E*E$	(3)
$E \to (E)$	(4)
$I \to a$	(5)
$I \to b$	(6)
$I \to Ia$	(7)
$I \to Ib$	(8)
$I \to I0$	(9)
$I \to I1$	(10)

Rightmost Derivation of $a*(a + b00)$

E	$\Rightarrow_{rm} E*E$	apply (3)
	$\Rightarrow_{rm} E*(E)$	apply (4)
	$\Rightarrow_{rm} E*(E + E)$	apply (2)
	$\Rightarrow_{rm} E*(E + I)$	apply (1)
	$\Rightarrow_{rm} E*(E + I0)$	apply (9)
	$\Rightarrow_{rm} E*(E + I00)$	apply (9)
	$\Rightarrow_{rm} E*(E + b00)$	apply (6)
	$\Rightarrow_{rm} E*(I + b00)$	apply (1)
	$\Rightarrow_{rm} E*(a + b00)$	apply (5)
	$\Rightarrow_{rm} I*(a + b00)$	apply (1)
	$\Rightarrow_{rm} a*(a + b00)$	apply (5)

Comparison of LM &RM

Leftmost Derivation of $a*(a+b00)$

$$\begin{aligned}
E &\Rightarrow_{lm} E*E && \text{apply (3)}\\
&\Rightarrow_{lm} I*E && \text{apply (1)}\\
&\Rightarrow_{lm} a*E && \text{apply (5)}\\
&\Rightarrow_{lm} a*(E) && \text{apply (4)}\\
&\Rightarrow_{lm} a*(E+E) && \text{apply (2)}\\
&\Rightarrow_{lm} a*(I+E) && \text{apply (1)}\\
&\Rightarrow_{lm} a*(a+E) && \text{apply (5)}\\
&\Rightarrow_{lm} a*(a+I) && \text{apply (1)}\\
&\Rightarrow_{lm} a*(a+I0) && \text{apply (9)}\\
&\Rightarrow_{lm} a*(a+I00) && \text{apply (9)}\\
&\Rightarrow_{lm} a*(a+b00) && \text{apply (6)}
\end{aligned}$$

Rightmost Derivation of $a*(a+b00)$

$$\begin{aligned}
E &\Rightarrow_{rm} E*E && \text{apply (3)}\\
&\Rightarrow_{rm} E*(E) && \text{apply (4)}\\
&\Rightarrow_{rm} E*(E+E) && \text{apply (2)}\\
&\Rightarrow_{rm} E*(E+I) && \text{apply (1)}\\
&\Rightarrow_{rm} E*(E+I0) && \text{apply (9)}\\
&\Rightarrow_{rm} E*(E+I00) && \text{apply (9)}\\
&\Rightarrow_{rm} E*(E+b00) && \text{apply (6)}\\
&\Rightarrow_{rm} E*(I+b00) && \text{apply (1)}\\
&\Rightarrow_{rm} E*(a+b00) && \text{apply (5)}\\
&\Rightarrow_{rm} I*(a+b00) && \text{apply (1)}\\
&\Rightarrow_{rm} a*(a+b00) && \text{apply (5)}
\end{aligned}$$

The Language of a Grammar

- **Definition :**
- The language $L(G)$ of a CFG $G = (V, T, P, S)$ is

$$L(G) = \{w \mid w \in T^*, S \overset{*}{\underset{G}{\Rightarrow}} w\}$$

- The language of a CFG is called a context-free language (CFL)
-
- Theorem 5.7 in the text book shows a typical way to prove that a given grammar really generates the desired language

Sentential forms

- Derivations from the start symbol are called *sentential forms*.
  - Given a CFG $G = (V, T, P, S)$, if $S \overset{*}{\Rightarrow} \alpha$ with $\alpha \in (V \cup T)^*$, then α is a sentential form.
  - If $S \underset{lm}{\overset{*}{\Rightarrow}} \alpha$ where $\alpha \in (V \cup T)^*$, then α is a left-sentential form.
  - If $S \underset{rm}{\overset{*}{\Rightarrow}} \alpha$ where $\alpha \in (V \cup T)^*$, then α is a right-sentential form.

Summary

- Derivation using Grammar
- Types of derivation
- Examples
- Comparison of LM and RM derivation
- Definition of CFL
- Sentential Forms

References

- John E. Hopcroft, Rajeev Motwani and Jeffrey D. Ullman, *Introduction to Automata Theory*, Languages, and Computation, Pearson, 3rd Edition, 2011.
- Peter Linz, An Introduction to Formal Languages and Automata, Jones and Bartle Learning International, United Kingdom, 6th Edition, 2016.

Next Class:

Parse Tree and Its Type

THANK YOU.