

CSE 105—Theory of Computability

Lecture 6—April 20, 2006
Context-Free Grammars

Instructor: Neil Rhodes

Hierarchy

Example

Rules (productions)

- $S \rightarrow 0S1$
- $S \rightarrow B$
- $B \rightarrow \#$

Variables (Non-terminals)

- S
- B

Start variable

- S

Terminals

- 0
- 1
- $\#$

Derivation (generates a string)

- $S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000B111 \Rightarrow 000\#111$

Language of the grammar

- All possible strings that can be generated from the grammar

Types of Grammars

Context-sensitive grammar

- Left-hand-side can contain an entire pattern
- The language of the grammar is *context-sensitive language*
 - Equivalent to a Turing Machine with a linear amount of space

Context-free grammar

- Single variable on the left-hand-side of a rule
- The language of the grammar is *context-free language*
 - Equivalent to an FSA with an extra stack

Linear Grammar

- Right-hand-side can have at most one variable
 - Left-linear grammar: variable must occur at beginning of the right-hand-side
 - Right-linear grammar: variable must occur at end of the right-hand-side
 - The languages of these two grammars are regular

Example Right-Linear Grammar

Odd number of 0's or 1's but not both

- $S \rightarrow 0T$
- $T \rightarrow 0S$
- $T \rightarrow \epsilon$
- $T \rightarrow 1V$
- $S \rightarrow 1U$
- $U \rightarrow 1S$
- $U \rightarrow \epsilon$
- $U \rightarrow 0V$
- $V \rightarrow 0U$
- $V \rightarrow 1T$

Derivation as Parse Tree

Example:

$S \rightarrow 0S1$

$S \rightarrow B$

$B \rightarrow \#$

Example of Nested Parentheses

Rules

- $S \rightarrow (S) \mid SS \mid \epsilon$

Equal 0's and 1's

Closure properties

Union

Concatenation

Star

Ambiguity

If more than one parse tree exists for a given string generated by a grammar, then the string is *derived ambiguously* and the grammar is *ambiguous*.

Example:

- $S = S \times S \mid S + S \mid (S) \mid a$

Alternate definition: if more than one *leftmost derivation* for the string generated by a grammar

- Leftmost derivation: replace left-most variable at every step

Alternate non-ambiguous grammar

Do product before sum:

- $\langle \text{expr} \rangle \rightarrow \langle \text{expr} \rangle + \langle \text{term} \rangle \mid \langle \text{term} \rangle$
- $\langle \text{term} \rangle \rightarrow \langle \text{term} \rangle \times \langle \text{factor} \rangle \mid \langle \text{factor} \rangle$
- $\langle \text{factor} \rangle \rightarrow (\langle \text{expr} \rangle) \mid a$

Inherently Ambiguous

If all grammars for a language are ambiguous, the language is *inherently ambiguous*

- Example: $\{a^i b^j c^k \mid i=j \text{ or } j=k\}$

Constructing grammars

x^*

- $A \rightarrow Ax \mid \epsilon$

x^ny^n

- $A \rightarrow xAy \mid \epsilon$

$x^{2n}y^n$

- $A \rightarrow xxaY \mid \epsilon$

$x^ny^m: n > m$

- $S \rightarrow AB$
- $A \rightarrow Ax \mid \epsilon$
- $B \rightarrow xBy \mid \epsilon$

Union of two CFLs

- $S \rightarrow S_1 \mid S_2$

Concatenation of two CFLs

- $S \rightarrow S_1S_2$