

Grammars, Parses, and Derivations

- 5 1. Write a grammar for the language of strings: $\{“()”, “(b)”, “(b:b)”, “(b:b:b)”, \text{etc.}\}$. (Do not handle the quotation symbols, but the colon(s) and parentheses symbols are indeed part of each string in the language.)

- 5 2. For the following grammar, determine which of the following strings are in the language of the grammar.

$$X \rightarrow Y\mathbf{a} \mid \mathbf{b}Z$$
$$Y \rightarrow \mathbf{d} \mid \mathbf{e}Y\mathbf{f}$$
$$Z \rightarrow \mathbf{c}Z \mid \mathbf{c}$$

Input strings:

- (a) **bc**
- (b) **bccc**
- (c) **bffd**
- (d) **faae**
- (e) **eedffa**

- 5 3. Given the grammar $S \rightarrow SS \mid (S) \mid \epsilon$,

- (a) Describe the language it generates.
- (b) Show that it is ambiguous.

- 5 4. Is the following grammar that describes conditional statements ambiguous? Why or why not? (**stm** and **exp** represent arbitrary expressions and statements that would be eventually derived if the grammar was more detailed, such as $2 < 3$ and $x = 5$.)

$$S \rightarrow \mathbf{if} (E) S$$
$$S \rightarrow \mathbf{if} (E) S \mathbf{else} S$$
$$S \rightarrow \mathbf{stm}$$
$$E \rightarrow \mathbf{exp}$$

- 5 5. Write a grammar that describes one or more **x**'s followed by an equal number of **y**'s. (Same number of **x**'s and **y**'s.)

Recursive Descent

- 10 6. Use the recursive descent parsing algorithm on the following grammar and input string. Draw the parse tree. If necessary, show any backtracking that the algorithm performed.

Grammar:

$$X \rightarrow aY \mid b$$
$$Y \rightarrow cXc \mid d$$

Input String: acbc

Eliminating Left Recursion

- 15 7. Rewrite the following grammars to eliminate the left recursion:

(a) $X \rightarrow Xa \mid Xb \mid c \mid d$

(b) $X \rightarrow aY \mid b \mid cX$
 $Y \rightarrow Yd \mid e$

CSC416 ONLY:

(c) $X \rightarrow Ya \mid Xa \mid c$
 $Y \rightarrow Yb \mid Xb \mid d$

CSC565 ONLY:

$$X \rightarrow Y \mid Z \mid Xa \mid bX$$
$$Y \rightarrow Zb \mid aY \mid d$$
$$Z \rightarrow Xc \mid Yb \mid e$$

Left Factoring

- 10 8. Left factor the following grammars:

(a) $X \rightarrow bYc \mid bYe \mid aa$
 $Y \rightarrow d \mid cY$

(b) $S \rightarrow \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S \mid \text{stm}$
 $E \rightarrow \text{exp}$

Top-Down Predictive Parsing

- 10 9. Use the predictive parsing algorithm on the following LL(1) parsing table and input string. At each step: show the stack, input string, and describe the performed action. Draw the final parse tree.

Input String: $\text{id} * (\text{id} + \text{id})$

LL(1) Parsing Table:

	id	+	*	(	)	\$
S	$S \rightarrow E\$$			$S \rightarrow E\$$		
E	$E \rightarrow TE'$			$E \rightarrow TE'$		
E'		$E' \rightarrow +TE'$			$E' \rightarrow$	$E' \rightarrow$
T	$T \rightarrow FT'$			$T \rightarrow FT'$		
T'		$T' \rightarrow$	$T' \rightarrow *FT'$		$T' \rightarrow$	$T' \rightarrow$
F	$F \rightarrow \text{id}$			$F \rightarrow (E)$		

FIRST and FOLLOW sets, LL(1) Parsing Table

10. Construct the FIRST set, FOLLOW set, and LL(1) parsing table for the following grammars. Is the grammar LL(1)?

- 15 (a) $X \rightarrow Y \mathbf{a} | \mathbf{a}$
 $Y \rightarrow \mathbf{c} | \mathbf{b} Z$
 $Z \rightarrow \mathbf{c} Z | \epsilon$

- 15 (b) $X \rightarrow Y \mathbf{a} | \mathbf{a}$
 $Y \rightarrow \mathbf{c} | \mathbf{b} Z Y$
 $Z \rightarrow \mathbf{c} Z | \epsilon$