CSC 416/565: Compilers
Fall 2015

Written Homework #2
Date Out: Tuesday, September 29
Due Date: Tuesday, October 8 (at the start of class; no late submissions accepted)

Grammars, Parses, and Derivations

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

$\boxed{5}$ 2. For the following grammar, determine which of the following strings are in the language of the grammar.

$X \rightarrow Y\texttt{a} \,|\, \texttt{b}Z$
$Y \rightarrow \texttt{d} \,|\, \texttt{e}Y\texttt{f}$
$Z \rightarrow \texttt{c}Z \,|\, \texttt{c}$

Input strings:

(a) `bc`

(b) `bccc`

(c) `bffd`

(d) `faae`

(e) `eedffa`

$\boxed{5}$ 3. Given the grammar $S \rightarrow SS \,|\, (S) \,|\, \epsilon$,

(a) Describe the language it generates.

(b) Show that it is ambiguous.

$\boxed{5}$ 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* $x = 5$.)

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

$\boxed{5}$ 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

$\boxed{10}$ 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:
$X \rightarrow \texttt{a}\,Y \mid \texttt{b}$
$Y \rightarrow \texttt{c}\,X\,\texttt{c} \mid \texttt{d}$

Input String: `acbc`

Eliminating Left Recursion

$\boxed{15}$ 7. Rewrite the following grammars to eliminate the left recursion:

(a) $X \rightarrow X\texttt{a} \mid X\texttt{b} \mid \texttt{c} \mid \texttt{d}$

(b) $X \rightarrow \texttt{a}Y \mid \texttt{b} \mid \texttt{c}X$
$Y \rightarrow Y\texttt{d} \mid \texttt{e}$

CSC416 ONLY:

(c) $X \rightarrow Y\texttt{a} \mid X\texttt{a} \mid \texttt{c}$
$Y \rightarrow Y\texttt{b} \mid X\texttt{b} \mid \texttt{d}$

CSC565 ONLY:
$X \rightarrow Y \mid Z \mid X\texttt{a} \mid \texttt{b}X$
$Y \rightarrow Z\texttt{b} \mid \texttt{a}Y \mid \texttt{d}$
$Z \rightarrow X\texttt{c} \mid Y\texttt{b} \mid \texttt{e}$

Left Factoring

$\boxed{10}$ 8. Left factor the following grammars:

(a) $X \rightarrow \texttt{b}Y\texttt{c} \mid \texttt{b}Y\texttt{e} \mid \texttt{aa}$
$Y \rightarrow \texttt{d} \mid \texttt{c}Y$

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

Top-Down Predictive Parsing

$\boxed{10}$ 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: `(id + id) * id`

LL(1) Paring Table:

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

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)?

$\boxed{15}$ (a) $X \to Y\,\texttt{a} \mid \texttt{a}$
$Y \to \texttt{c} \mid \texttt{b}\,Z$
$Z \to \texttt{c}\,Z \mid \epsilon$

$\boxed{15}$ (b) $X \to Y\,\texttt{a} \mid \texttt{a}$
$Y \to \texttt{c} \mid \texttt{b}\,Z\,Y$
$Z \to \texttt{c}\,Z \mid \epsilon$