

# CS314 Fall 2018

## Assignment 3

### 1 LL(1) Grammar and Recursive Descent Parsing

$\langle \text{program} \rangle ::= \text{def } \langle \text{funcname} \rangle \langle \text{arguments} \rangle : \backslash \mathbf{n} \langle \text{block} \rangle \mathbf{EOF}$   
 $\langle \text{funcname} \rangle ::= \mathbf{f} \mid \mathbf{g}$   
 $\langle \text{arguments} \rangle ::= ( \langle \text{variable} \rangle \langle \text{morevars} \rangle )$   
 $\langle \text{morevars} \rangle ::= , \langle \text{variable} \rangle \langle \text{morevars} \rangle \mid \epsilon$   
 $\langle \text{block} \rangle ::= \langle \text{stmtlist} \rangle$   
 $\langle \text{stmtlist} \rangle ::= \backslash \mathbf{t} \langle \text{stmt} \rangle \langle \text{morestmts} \rangle$   
 $\langle \text{morestmts} \rangle ::= \backslash \mathbf{n} \langle \text{stmtlist} \rangle \mid \epsilon$   
 $\langle \text{stmt} \rangle ::= \langle \text{assign} \rangle \mid \langle \text{ifstmt} \rangle \mid \langle \text{returnstmt} \rangle$   
 $\langle \text{assign} \rangle ::= \langle \text{variable} \rangle = \langle \text{expr} \rangle$   
 $\langle \text{condition} \rangle ::= \langle \text{variable} \rangle \leq \langle \text{expr} \rangle$   
 $\langle \text{ifstmt} \rangle ::= \mathbf{if} \langle \text{condition} \rangle : \langle \text{assign} \rangle \backslash \mathbf{n} \backslash \mathbf{t} \mathbf{else} : \langle \text{assign} \rangle$   
 $\langle \text{returnstmt} \rangle ::= \mathbf{return} \langle \text{variable} \rangle$   
 $\langle \text{expr} \rangle ::= \langle \text{term} \rangle + \langle \text{term} \rangle$   
 $\langle \text{term} \rangle ::= \langle \text{variable} \rangle \mid \langle \text{digit} \rangle$   
 $\langle \text{variable} \rangle ::= \mathbf{a} \mid \mathbf{b} \mid \mathbf{c}$   
 $\langle \text{digit} \rangle ::= \mathbf{0} \mid \mathbf{1} \mid \mathbf{2}$

$\backslash \mathbf{n}$  represents the “new line” terminal.  $\backslash \mathbf{t}$  represents the “tab” terminal.

- (a) Show that the grammar above is LL(1). Use a formal argument based on the definition of the LL(1) grammar.
- (b) Show the LL(1) parse table.

- (c) Write a recursive descent parser for the above grammar in pseudo code in the same format as that in lecture 6.