CS314 Fall 2018

Assignment 3

1 LL(1) Grammar and Recursive Descent Parsing

<program> ::= **def** <funcname> <arguments> **:** **\n** <block> **EOF**

<funcname> ::= **f** | **g**

<arguments> ::= **(** <variable> <morevars> **)**

<morevars> ::= **,** <variable> <morevars> | ϵ

<block> ::= <stmtlist>

<stmtlist> ::= **\t** <stmt> <morestmts>

<morestmts> ::= **\n** <stmtlist> | ϵ

<stmt> ::= <assign> | <ifstmt> | <returnstmt>

<assign> ::= <variable> **=** <expr>

<condition> ::= <variable> **<=** <expr>

<ifstmt> ::= **if** <condition> **:** <assign> **\n** **\t** **else** **:** <assign>

<returnstmt> ::= **return** <variable>

<expr> ::= <term> + <term>

<term> ::= <variable> | <digit>

<variable> :: = a | b | c

<digit> :: = 0 | 1 | 2

\n represents the "new line" terminal. **\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.