CS314 Spring 2023

Homework 3

Due Tuesday, February 21, 11:59pm

No deadine extension possible due to Midterm on Friday, Feb. 24

submission: pdf file through canvas

1 Problem — LL(1) Recursive Descent Parsing

<program> ::= **prog** <block> **.**

<block> ::= **begin** <stmtlist> **end**

<stmtlist> ::= <stmt> <morestmts>

<morestmts> ::= **;** <stmtlist> |ϵ

<stmt> ::= <assign> |<ifstmt> |
<repeatstmt> |<block>

<assign> ::= <var> = <expr>

<ifstmt> ::= **if** <testexpr> **then** <stmt> **else** <stmt>

<repeatstmt> ::= **repeat** <stmt> **until** <testexpr>

<testexpr> ::= <var> <= <expr>

<expr> ::= + <expr> <expr> |
− <expr> <expr> |
∗ <expr> <expr> |
<var> |
<digit>

<var> :: = a | b | c

<digit> :: = 0 | 1 | 2

1. Show that the grammar above is LL(1). Use a formal argument based on the definition of the LL(1) property.
2. Show the LL(1) parse table.
3. Write a recursive descent parser for the above grammar in an imperative C-like pseudo code as used in class (see lecture 9).

4. Extend your recursive descent parser such that it prints the total number of binary operators (+, −, ∗, <=) in the program. For the program listed below, your parser should print '7 binary operators'.

```
program
begin
  if b <= 0 then
    begin
      a = * a + b c
    end;
    c = + a b
  else
    repeat
    begin
      a = + a b;
      c = - a 1
    end
    until a <= 1
end.
```