EECS665

Compiler Construction

Drew Davidson
Ruturaj Vaidya

Lecture: LEEP2 G415
MWF 3:00-3:50

Lab: Eaton 1005B

ANOUNCEMENTS | LAB | SCHEDULE | MATERIALS | ASSIGNMENTS

Homework 5

Due on October 10th @ 3:00 PM (in class, to Drew, or at Engineering front office)

Not accepted late

ALL homework must be done individually

For this homework you will define a syntax-directed translation for the CFG given below, which defines a very simple programming language.

```
program → MAIN LPAREN RPAREN LCURLY list RCU

list → list oneItem
     | epsilon

oneItem → decl
        | stmt
```

```
decl → BOOL ID SEMICOLON
     | INT ID SEMICOLON

stmt → ID ASSIGN exp SEMICOLON
     | IF LPAREN exp RPAREN stmt
     | LCURLY list RCURLY

exp → exp PLUS exp
    | exp LESS exp
    | exp EQUALS exp
    | ID
    | BOOLLITERAL
    | INTLITERAL
```

Question 1

Write a syntax-directed translation for the CFG given above, so that the translation of an input

program is the *set* of names of variables used somewhere in the program. Note: here the term *use* is in contrast to *declaration*; any appearance of a variable that is not a variable declaration counts as a *use* of that variable. For the example code in Question 2, the translation should be { x, a, b }.

Your translation rules should use the following notation:

- { } is an empty set
- { ID.value } is a set containing the variable whose name is the value associated with this ID token
- S1 ∩ S2 is the intersection of sets S1 and S2
- S1 ∪ S2 is the union of sets S1 and S2
- S1 - S2 is the set of all items that are in S1 but not in S2

Note that you should not try to use something like "{ a, b }" to mean a set with two elements; instead, use set union to combine two sets that each contain one element.

Use the notation that was used in class and in the on-line readings; i.e., use nonterminal.trans to mean the translation of a nonterminal, and terminal.value to mean the value of a terminal. Assume that ID.value is a String (the name of the identifier). Use subscripts for translation rules that include the same nonterminal or the same terminal more than once.

Question 2

Draw a parse tree for the program given below and annotate each nonterminal in the tree with its translation.

```
main ( ) {
                                    int x;
                                    bool y;
                                    x = a;
                                    int z;
                                    if (x == a) {
                                        int x;
                                        b = x < 18;
                                    }
                                }
```