CS 241 Week 12 Tutorial

Extending WLP4

Spring 2015

Summary

- Extending WLP4

Extending WLP4

We want to add a Boolean type to WLP4. The new type should allow us to do the following:

- Declare Boolean variables, which must be initialized to "true" or "false" (case-sensitive).
- Assign values to Boolean variables.
- Write Boolean expressions using the logical AND operator (&&), logical OR operator (||) and logical NOT operator (!) that take Boolean values as operands. The order of operations is NOT followed by AND followed by OR.
- Write test expressions using the comparison operators that evaluate to Boolean values.
- Use Boolean variables or expressions as the test condition for if statements and while loops.

We will not allow:

- Declaring pointers to Booleans.
- Allocating arrays of Booleans.
- Using a Boolean as one of the parameters of wain.
- Printing a Boolean with println.
- Returning a Boolean from any procedure.
- Arithmetic expressions where one operand is an integer or pointer and the other is a Boolean. (e.g. $true + 5$)
- Boolean expressions where one operand is a Boolean and the other is an integer or pointer. (e.g. $(x > y)$ && 7)
- Comparisons between Booleans, other than == and !=.
- Using "0" and "1" rather than "true" and "false" as Boolean values.

Here is a program that demonstrates some of the things we should be able to do with our new Boolean type.

```
int wain(int x, int y) {
    bool a = true;
    bool b = false;
    a = x > y;
    if (a) {
        println(x);
    } else {
        println(y);
    }
    b = true;
    while ((b || a) && (x < y)) {
        if (true) {
            b = !((a || b) && a);
        } else {}
    }
    return y;
}
```

Think about what steps we would need to take to implement this new type.

- What changes do we need to make to the lexical syntax of WLP4?
  (That is, the list of valid tokens recognized by the scanner.)
- What changes do we need to make to the context-free syntax of WLP4?
  (That is, the context-free grammar that the parser is based on.)
- What changes do we need to make to the context-sensitive syntax of WLP4?
  (That is, the set of type rules that the semantic analyzer uses.)
- How would we generate code for the declaration of a Boolean?
- How would we generate code for the logical AND (&&) operator?
- How would we generate code for the logical NOT (!) operator?
- How would we generate code for the logical OR (||) operator?
- Suppose we decided that we do want to be able to print Boolean values with println, and we want them to be printed as the strings “true” or “false” rather than “1” or “0”. How would we accomplish this?