

## CSE 655, Assignment #2

Due: 21 Oct. '10

20 points.

Suppose we wish to add a **select** statement to Core as follows: a **select** statement has the structure

```
select b1 -> S1 || b2 -> S2 || ... || bn -> Sn end;
```

where  $b_1, \dots, b_n$  are boolean expressions (i.e.  $\langle cond \rangle$ 's), and  $S_1, \dots, S_n$  are  $\langle stmt seq \rangle$ 's; and the number of  $b$ 's is equal to the number of  $S$ 's. We execute the `select` as follows: if  $b_1$  evaluates to **true**, execute  $S_1$  and you are done; if  $b_1$  evaluates to **false**, and  $b_2$  to **true**, execute  $S_2$  and you are done; ...; if  $b_1, \dots, b_{(n-1)}$  evaluate to **false** and  $b_n$  to **true**, execute  $S_n$ ; if  $b_1, \dots, b_n$  all evaluate to **false**, simply go to the next statement that follows the **select** (i.e., the **select**, in this case, is a 'no-op'). Note that the "`||`" is a new terminal symbol that is used to separate each  $\langle cond \rangle, \langle stmt seq \rangle$  pair from the next pair. (Actually, it is the same symbol that we are going to use in place of "`or`" in the Core grammar.)

Now for the problems:

1. (6 points). Add the **select** statement to Core by modifying the BNF grammar appropriately. (You may use extended BNF if you wish but answering the next question might then be harder.)
2. (14 points). Using the notation from the class notes (copies of slides), write down the *Execute-select-statement* procedure and any other procedures you need to add to Core's interpreter to implement the **select** statement. Don't worry about the *Parse-select-statement* procedure.

You may assume the array representation of the parse tree when answering this question. You may also assume procedures such as *Execute-stmt-seq* or functions like *Eval-Cond* etc. already exist.

**Important Note:** The problem is *not* asking you to show how you can achieve the effect of the *select* statement by using multiple *if* statements or anything like that. You are being asked to introduce a *new* statement into the CORE language so that the CORE programmer can write, as part of his or her CORE program, statements such as:

```
select (X > 0) -> Y = 10; || (Y > 0) -> X = 10; end;
```

**Important Note 2:** The assignment is due in class on October 21. If you don't turn it in on the 21st, but turn it in by the *start* of the *next* class (Oct. 24), you will be penalised 20%. If you don't turn in the assignment by the start of class on Oct. 24, you will receive no credit for the assignment.

**Important Note 3:** The first mid-term will be on **Wednesday, October 26**. I had previously said it would be on Oct. 24 but, instead, it will be on Oct. 26. Topics for the mid-term will be everything we discuss in class before the exam. I will post a key to the problems in this assignment to the course newsgroup on October 22/23.