COSC 561

Midterm 2 Study Guide

1. Use Figure 4.37 and the expression grammar 4.1 to parse the string **(id * id) + id** using the LR parsing algorithm.

2. Given the following grammar construct the LR(0) sets of items.

   1. S → XYa
   2. X → Yb
   3. X → b
   4. Y → c
   5. Y → ϵ

3. Use the grammar and the sets of items in the last problem to construct the SLR parsing table.

4. Given the following grammar construct the LR(1) sets of items.

   1. S → XYa
   2. X → Yb
   3. X → b
   4. Y → c
   5. Y → ϵ

5. Use the grammar and the set of items in the last problem to construct the canonical parsing table.

6. Draw the DAG representation for the following expression (as shown in Figure 6.3). Provide the quadruple, triple and indirect triple representations of the expression.

$$(a * c) + b * (a * c) + b - (a * c)$$

7. Give the three-address code that could be emitted to translate the following assignment statement. However, you may **not** use array index operations. Assume that `a` is an integer array with 5 rows and 10 columns. Assume that `v` is an integer variable and `i` and `j` are integer parameters passed to the function. Provide solutions for when 1) the arrays are stored in *row major* order, and 2) the arrays are stored in *column major* order.

```
v = a[i][j];
```

8. Use Figures 6.36 and 6.37 to translate the following statement in two passes (no backpatching).

$$if(x > 10 \quad \&\& \;\; y! = 0 \quad || \quad x == y) \quad x = y;$$

9. Use Figures 6.43 and 6.46 to translate the expression in a single pass (use backpatching). Start outputting the code from address 20.

$$if(x > 10 \quad \&\& \quad y! = 0 \quad || \quad x == y) \quad x = y;$$

10. Answer the following:

    (a) Why is bottom-up parsing (with one symbol of lookahead) more powerful than top-down parsing (with one lookahead symbol)?

    (b) Explain the different error recovery strategies.

    (c) What is the general strategy of panic-mode recovery? What are its main advantages?

    (d) Give the advantages and disadvantages of using a three-address form of intermediate representation over a zero-address representation.

    (e) Describe the three strategies to translate large *switch* statements. What are the characteristics of each approach.

    (f) What is static checking? Why is static checking preferable to dynamic checking?

    (g) Describe the rules for type checking.

    (h) What is coercion, overloading, and polymorphism? Give an example of each in the C language.

    (i) Understand the purpose of the *backpatch* routine in the csem assignment.

    (j) Be able to apply the backpatch routine in a semantic routine for statements such as *if*, *if-else*, and *while*.