COMP3002: End of Term Exam

Carleton University

Winter 2008

1. [2 marks] State two advantages of separating a compiler into a separate parser and code generator that interface by means of an intermediate representation.

2. [10 marks] Using the alphabet $\Sigma = \{A, \ldots, Z, a, \ldots, z, 0 \ldots, 9, \_, +\}$, write regular expressions for the following, or explain why you can't:

   (a) Identifiers like those found in Java and C that are non-empty strings over Σ that don't begin with a digit.

   (b) Decimal numbers with at least one digit before the decimal point, an optional decimal points and (if the decimal point is present) at least one digit after the decimal. Examples, 10, 4, 5.4, 0.2, 7.3.

   (c) The same as above, except that the numbers may not start with a 0 unless their value is less than 1. For example, 0.54 is ok, but 04.32 and 0332 are not.

   (d) All strings that have the same number of occurences of A and B

   (e) Pairs of numbers, separated by a `+`, that add up to 10

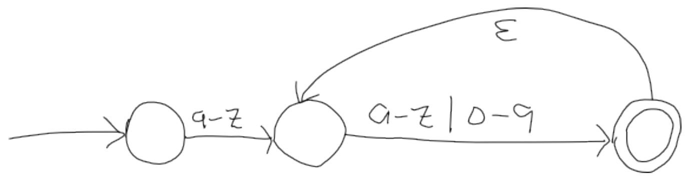
3. [6 marks] Convert the following regular expressions, over the alphabet $\Sigma = \{a, b\}$ into NFAs. Don't forget to indicate the start state and any accepting states.

   (a) `a(a|b)*`

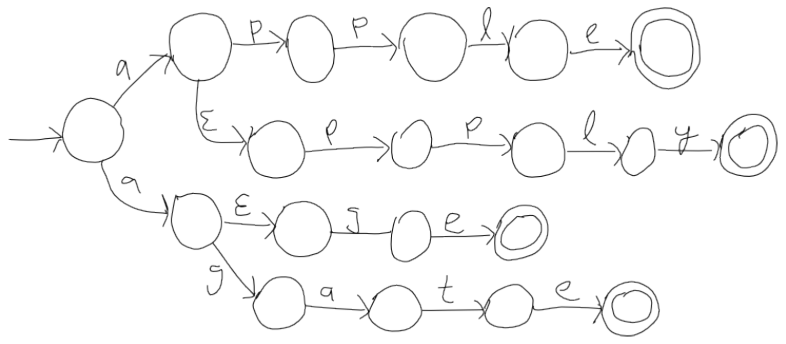
   (b) `arbitrary|arbutus`

   (c) `(a|b)*a(a|b)(a|b)`

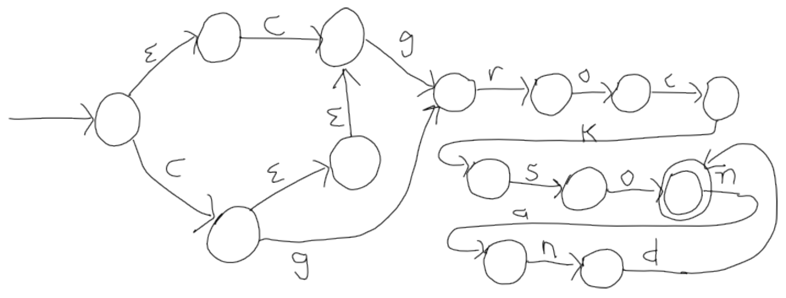
4. [6 marks] Convert the following NFAs into DFAs that have a single accepting state.

   (a) 

(b)



(c)



5. [6 marks] Consider the following context-free grammar:

```
S -> E
E -> E + E
E -> E * E
E -> ( E )
E -> id
```

   (a) Give a leftmost derivation of the string '(id + id) * id'

   (b) Give a rightmost derivation of the string '(id + id) * id'

   (c) Give at least two derivations of the string 'id + id * id' that produce different parse trees and show the resulting parse trees

6. [4 marks] Consider the following grammar

```
S  -> E
E  -> T T'
E  -> ( E )
T  -> id
T  -> id * E
T' -> + T
T' ->
```

   (a) Is this grammar LL(1)? If not, show how to make it LL(1)

   (b) Give an LL(1) parse table for the resulting grammar

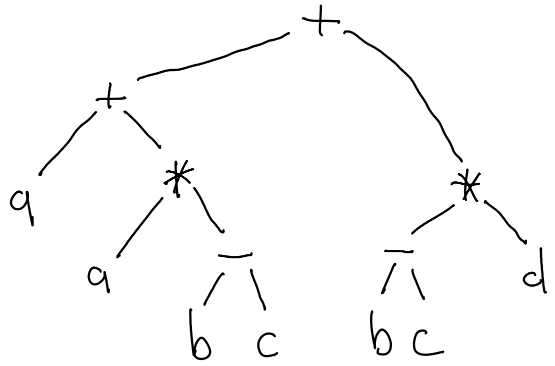
7. [6 marks] Consider the alphabet $\Sigma = \{\text{num}, \text{op}\}$. A polish-notation expression is either a number (num), or two expressions followed by an operator (op).

   (a) Write a context-free grammar for the language of polish-notation expressions.

   (b) Try to write an equivalent grammar with no left recursion

   (c) Try to make your grammar into an LL(1) grammar

8. [4 marks] Consider the following parse tree:



   (a) Generate simple 3 address code that evaluates this parse tree

   (b) Convert this parse tree into a parse DAG using the method described in class

9. [2 marks] Consider the following code:

```
 1.  int odd_factorial(int q) {
 2.    int factorial(int n) {
 3.      if (n == 1) return q;
 4.      int t = factorial(n-1);
 5.      return n * t;
 6.    }
 7.    if (q % 2 == 0)
 8.      return q;
 9.    return factorial(t);
10. }
```

   (a) Consider calling the function `odd_factorial(5)`. Show the state of the stack, including parent frame pointers, during the *execution* of the return statement in line 3.

10. [4 marks] Consider the following JVM code:

```
 1.   getstatic java/lang/System/out Ljava/io/PrintStream;
 2.   iload 0
 3.   ifeq false_label
 4.   ldc "true"
 5.   goto print_it
 6. false_label:
 7.   ldc "false"
 8. print_it:
 9.   invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V
10.   return
11.   goto false_label
```

(a) Split this code into its basic blocks (just use line numbers)

(b) Construct the flow graph for the code

11. [6 marks] Consider the following basic block of 3-address instructions:

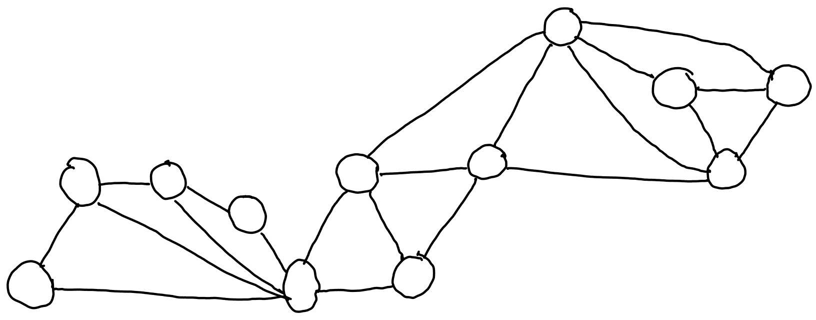
```
1. a := b + c
2. x := a + b
3. b := a - d
4. c := b + c
5. d := a - d
6. y := a - d
```

(a) Write the next-use information for each line of the basic block

(b) Construct the DAG representation of this basic block

(c) Assuming c is the only live variable on exit from this block, show which nodes of the DAG correspond to dead code

12. [3 marks] Consider the following register allocation graph.

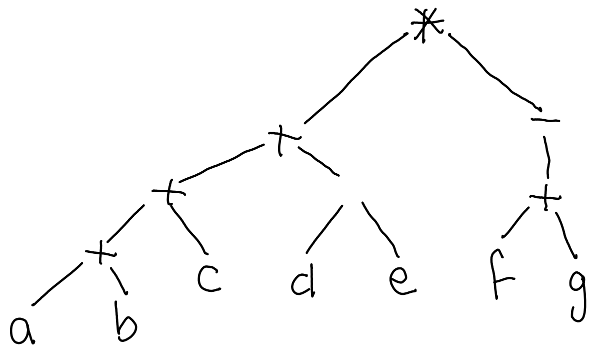


Illustrate the operation of the recursive coloring algorithm on this graph by

(a) labelling the nodes 1, 2, 3, ..., in the order they are removed by the algorithm, and

(b) indicating the color (a number from $1, \ldots, k$) assigned to each node by the algorithm.

13. [2 marks] Label the following tree with its Ershov numbers.



14. [2 marks] Draw a T-diagram illustrating how to compile gcc (a full-featured optimizing C compiler written in C that generates i386 code) using the Solaris cc compiler (a crappy non-optimizing C compiler that runs on the i386 and generates i386 code) and then using the resulting compiler to compile gcc again. What can you say about the resulting C compiler

15. [2 marks] What do you plan on doing with your summer?

(a) labelling the nodes 1, 2, 3, $\ldots$, in the order they are removed by the algorithm, and

(b) indicating the color (a number from $1, \ldots, k$) assigned to each node by the algorithm.

13. [2 marks] Label the following tree with its Ershov numbers.

14. [2 marks] Draw a T-diagram illustrating how to compile gcc (a full-featured optimizing C compiler written in C that generates i386 code) using the Solaris cc compiler (a crappy non-optimizing C compiler that runs on the i386 and generates i386 code) and then using the resulting compiler to compile gcc again. What can you say about the resulting C compiler

15. [2 marks] What do you plan on doing with your summer?

(a) labelling the nodes 1, 2, 3, ..., in the order they are removed by the algorithm, and

(b) indicating the color (a number from $1, \ldots, k$) assigned to each node by the algorithm.

13. [2 marks] Label the following tree with its Ershov numbers.

14. [2 marks] Draw a T-diagram illustrating how to compile gcc (a full-featured optimizing C compiler written in C that generates i386 code) using the Solaris cc compiler (a crappy non-optimizing C compiler that runs on the i386 and generates i386 code) and then using the resulting compiler to compile gcc again. What can you say about the resulting C compiler

15. [2 marks] What do you plan on doing with your summer?