No: 509.xxx

THE CITY UNIVERSITY

LONDON

B.Eng. Computing
Professional Pathway 3 - Computing
Professional Pathway 3 - Computing (DISC)

PART II EXAMINATION

Language Processors

May 9th 2003 9.00-10.30

Answer TWO questions
All questions carry equal marks

1. (a) The following regular expression recognises certain strings consisting of the letters a, b and c:

$$a((ab)|(ac)) * c$$

Indicate which of these five strings are recognised by the above regular expression:

$aacc$, $abac$, ac, $ababababacac$, $aabacc$

Also, show three more strings that are recognised by the above expression. Finally, show two more strings consisting of the letters a, b and c that are *not* recognised by the above regular expression. [30]

(b) Most programming languages allow the same symbol to denote both the subtraction operator (in $x - y$) and the unary negation operator (in $-x$). Make clear the difficulty this causes for a parser-generator (for example, CUP) and explain how it may be overcome. [20]

(c) Explain what it means for a context-free grammar to be ambiguous. Write down an ambiguous grammar and show why it is ambiguous. [20]

(d) Consider the following Tiger function:

```
function f(a:string, b:int, c:int)=
      (print_int(b+c);
       let var c := "hi"
           var a := b
           var b := "hello"
       in print(b); print_int(a)
       end;
       print_int(c); print_int(b);
      )
```

Given an initial environment $\sigma_0 = \{\text{a} \rightarrow \text{int}, \text{b} \rightarrow \text{string}\}$, derive the type binding environments for the function at each use of an identifier and indicate where type lookups will occur. [30]

2. The reference manual for a Tiger-like programming language contains the following definition for a kind of expression:

   The if-expression

   $$\textbf{if } exp_1 \textbf{ then } exp_2 \textbf{ else } exp_3$$

   evaluates the expression exp_1. If the result is non-zero the if-expression yields the result of evaluating exp_2; otherwise it yields the result of evaluating exp_3.

   (a) Write down a BNF concrete syntax for the if-expression. [10]

   (b) Sketch a possible abstract syntax for the if-expression. [10]

   (c) Show how semantic actions in a grammar for a parser-generator such as CUP can be used to produce abstract syntax trees for the if-expression. [20]

   (d) Informally describe an appropriate typecheck for the if-expression. [20]

   (e) Suppose a Tiger compiler translates all expressions and subexpressions into intermediate code (eg expression trees). Outline the intermediate code that might be generated in translation of the if-expression:

   ```
   if a < b then c := a else c := b
   ```

   You can assume that the expression tree for any variable `v` is simply `TEMP v`. [40]

3. (a) Choose a programming language you know well and describe how run-time storage is organised and managed during program execution. Clearly associate any storage structures you mention with the implementation of particular language features. [25]

   (b) What is a stack frame? Outline a typical layout for a stack frame and describe each element of a frame. Comment on how local variables, arguments and non-local variables are addressed by the code generated for a procedure or method. [25]

   (c) Explain why registers might be used for parameter passing and suggest situations where passing in registers is particularly appropriate. Outline situations where it is necessary for the code generated for a procedure or method to write registers to the stack. [30]

   (d) Explain the difference between *caller-save* and *callee-save* registers. Why might caller-save registers sometimes not be saved?

   [20]

External examiners: Professor M.E.C. Hull
Professor M. Moulding

Examiners: D. Bolton