

## 2010 Paper 3 Question 3

### Compiler Construction

- (a) What are the principal features of a regular language and of a context-free language and their respective parsers? [4 marks]
- (b) When implementing a compiler, why is a regular language commonly used for lexical analysis and why are context-free grammars commonly used for the main syntax analysis phase? [3 marks]
- (c) Give **two** main features that distinguish a recursive-descent parser from an SLR(k) parser (or similar). [2 marks]
- (d) The C++ language uses the “>>” character sequence to denote the right-shift operator. This character sequence can also appear when a template (generic) type takes another template as its argument, as in “`stack<list<int>>`” that is supposed to denote a stack of integer lists. What problem can this cause? [1 mark]

Comment on how you might solve this problem

- (i) in the syntax analyser; and [2 marks]
- (ii) in the lexical analyser. [2 marks]

Comment on the relative elegance of the two solutions. [1 mark]

- (e) A common language construct is “`T v;`” to declare a variable called “`v`” or type “`T`”. What parsing problem can arise if “`T`” is a user-defined type? Explain how the lexical analyser could benefit from feedback from the syntax analyser in this situation. [5 marks]