UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

KNOWLEDGE BASES IN SOFTWARE ENGINEERING

Monday 5 June 2000

14:00 to 15:30

Examiners:

Prof. K. Turner (External)

Prof. G. Brebner (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) Suppose information about the syntax of some programming language involves the following statements.

$$\begin{array}{lll}
\forall x\ term(x) & \rightarrow & (var(x) \vee cst(x)) \\
\forall x\ var(x) & \rightarrow & term(x) \\
\forall x\ cst(x) & \rightarrow & term(x) \\
\forall x\ \neg(var(x) \wedge cst(x)) & & \\
\forall x\ \forall y\ decl(type(x,y)) & \rightarrow & (var(x) \wedge type(y)) \\
\forall x\ \forall y\ decl(type(x,y)) & \rightarrow & hasType(x,y) \\
\forall x\ \neg(hasType(x,int) \wedge hasType(x,string) & &
\end{array}$$

Convert this information into clausal form. [*5 marks*]

(b) Suppose we know in addition that $decl(type(a, nat))$ and $hasType(b, string)$, and want to show $\neg cst(a)$. What additional clauses need to be added to those obtained so far? [*3 marks*]

(c) Show that a derivation can be obtained that $\neg cst(a)$. [*8 marks*]

(d) Why is depth-first search inappropriate to use with the clauses obtained here? You should illustrate your answer with an example of probelmatic behaviour. [*4 marks*]

(e) Suppose we are analysing a syntactically *incorrect* program using these clauses. Explain how diagnostic information can be obtained from the mismatch between our description of the language and some syntactically flawed code. [*5 marks*]

2. The *Medusa* system is an example of a domain-specific synthesis system, in this case for putting together algorithms to compute geometrical properties.

   (a) Explain why it is easier to build domain-specific synthesis systems, rather than general-purpose ones. [*3 marks*]

   (b) What does it mean for subtasks in the design process to be *functionally independent*, and why is this property important when trying to complete a design task via top-down decomposition of the design task? [*3 marks*]

   (c) Explain how *efficiency constraints* are used in the Medusa system to ensure a specified level of efficiency in the synthesised algorithm. [*5 marks*]

   (d) Describe the four subtask generation procedures employed in this system, with a brief example to illustrate each one. [*10 marks*]

   (e) Describe two heuristics used by Medusa to perform automatic task decomposition. [*4 marks*]

3. (a) Describe the aims of the *Programmer's Apprentice* project. What sorts of knowledge about software are involved? [*5 marks*]

   (b) How are programming *cliches* represented in this approach, and what is their role in constructing software interactively? You should give a small illustration of how they are used. [*8 marks*]

   (c) Compare the notion of cliche here with that of an ML functor. Which features of the use of cliches can be captured in EML development by using functors instead of cliches? Which features cannot easily be so treated? [*5 marks*]

   (d) Compare the use of cliches and plans to some other approach that you know of for putting together software. What are the strengths and weaknesses of each approach? [*7 marks*]