UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

KNOWLEDGE BASED SOFTWARE ENGINEERING

Saturday 9 September 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) Convert the following three predicate calculus formulas into clausal form.

$$\begin{aligned}
&\forall v1\,\forall v2\,\forall v3\,\forall v4\ (type(v1,v2) \wedge type(v3,v4) \\
&\qquad \rightarrow type(pair(v1,v3), prod(v2,v4))) \\
&type(0, int) \\
&type(a, real)
\end{aligned}$$

[*5 marks*]

(b) Suppose we want to test whether the following query can be derived:

$$\exists v5\ (type(b, real) \quad \rightarrow \quad type(pair(0, pair(a,b)), v5))$$

What clauses should be added to those found above? [*4 marks*]

(c) Show that the query in part (b) does in fact have a derivation. [*6 marks*]

(d) In such a query, we often want to find out a value corresponding to the quantified variable ($\exists v5$ here) that satisfies the query. What value can be found from your derivation in part (c)? [*4 marks*]

(e) What restriction to search is imposed by *unit resolution*? Show by example that, even when unit resolution is used, it is possible for search to "loop" when using the clauses of part (a). [*6 marks*]

2. The technique of folding and unfolding definitions in a functional programming language has been proposed as a way of coding that is both efficient and correct.

(a) Explain how this technique in principle can ensure *correctness* of code that has been produced in this way. [*5 marks*]

(b) Describe the transformation of *unfolding*. What is the result of unfolding the call to app in the function g below?

```
fun app(nil, y) = y
  | app(h::t, y) = h::app(t,y)

fun g (y,z) = app(app(nil,y),z)
```

[*4 marks*]

(c) Apply unfolding, as above, as often as possible to the function

```
fun h (x,y,z) = app(app(x,y),z)
```

Is the resulting definition more efficient? [*6 marks*]

(d) Describe the transformations of *folding* and use of *laws*. When several transformations are available, the question arises of when it is appropriate to apply a given transformation. Outline one heuristic that can be used for this purpose, and say how this helps control the use of transformations. [*5 marks*]

(e) What are the strengths and weaknesses of this approach to producing correct and efficient code? Justify your answer with comparisons to other approaches. [*5 marks*]

3. (a) Outline the stages involved in knowledge acquisition for a KBSE system. [*8 marks*]

   (b) It is often desirable to get an *explanation* from a system, alongside the solution to a query. Why is it important to have this capability? [*4 marks*]

   (c) In a logic-based approach, a derivation is found when a query is solved. Explain how an explanation for the solution can be based upon such a derivation. What sort of information is *not* available in this way? What problems are there in presenting such explanations? [*8 marks*]

   (d) Contrast the use of a derivation for explanatory purposes with the use of a trace of the computation. Which is likely to give more useful information? [*5 marks*]