PAPER CODE NO.	EXAMINER : Michael Fisher
COMP 313	DEPARTMENT : Computer Science Tel. No. 4-6701

THE UNIVERSITY
of LIVERPOOL

MAY 2003 EXAMINATIONS

Bachelor of Arts : Year 3
Bachelor of Science : Year 3

FORMAL METHODS

TIME ALLOWED : Two Hours and a Half

INSTRUCTIONS TO CANDIDATES

Answer **four** questions only.

If you attempt to answer more than the required number of questions (in any section), the marks awarded for the excess questions will be discarded (starting with your lowest mark).

1. This question concerns the basic structures used within Z specifications.

   (a) Using first-order logic, express the fact that there is not a largest Integer. **[4]**

   (b) If A and B are both sets of the same type, then under what circumstances (if any) will

   $$(A \setminus B) = (B \setminus A)$$

   be true? Give examples and counter examples to illustrate your answer. **[5]**

   (c) How can we represent both sequences and bags in terms of functions? **[5]**

   (d) Given

   $$S == \{a, b, c, d\}$$
   $$T == \{e, f, g\}$$
   $$h : S \nrightarrow T$$

   Then is $h == \{a \mapsto e, b \mapsto f, c \mapsto g, d \mapsto e\}$ a surjection? Explain your answer. **[5]**

   (e) Show why the following is true for any sequence σ and set S.

   $$(\mathrm{ran}\,\sigma \subseteq S) \Rightarrow (\sigma \,|\, S = \sigma)$$

   [6]

2. We wish to specify a system that records details about the phone numbers of people. Assuming that *NAME* is a set of names, and *PHONE* is a set of phone numbers, then we can define

$$
\begin{array}{|l}
\textit{PhoneBook} \\
\hline
known : \mathbb{P}\, NAME \\
tel : NAME \pfun PHONE \\
\hline
\mathrm{dom}\, tel = known \\
\hline
\end{array}
$$

(a) Explain, in English, what the declaration and predicate parts of this schema mean. **[3]**

(b) If we use $\Xi PhoneBook$ as part of a new operation schema, what would the relationship between *known* and $known'$ be in the predicate part of the operation schema? **[3]**

(c) Write a Z schema for the operation

$$Find(name? : NAME, phone! : PHONE)$$

where *name*? is the name to be found and *phone*! is the number associated with that name. **[5]**

(d) Write a Z schema for the operation

$$CountEntries(number! : \mathbb{N})$$

which returns the number of entries in the phone book. **[3]**

(e) Write a Z schema for the operation

$$AddName(name? : NAME, phone? : PHONE)$$

which adds a new *name*? to the phone book and records its number as *phone*? **[5]**

(f) How would you modify the *AddName* operation above so that it is robust (i.e. it will be defined even if the name already exists in the phone book)? Assume that a *REPORT* type exists for reporting errors. **[6]**

3. This question concerns temporal logic.

   (a) The formula $\Box\Diamond\varphi$ is commonly termed "infinitely often φ". Describe what property the formula $\Diamond\Box\psi$ characterises. **[3]**

   (b) We can conjoin together next-formulae such as

   $$p \wedge \bigcirc p \wedge \bigcirc\bigcirc p \wedge \bigcirc\bigcirc\bigcirc p \wedge \ldots$$

   How many such formulae would we have to conjoin together to give the same behaviour as $\Box p$? **[3]**

   (c) Is the following valid, i.e. true in all possible models?

   $$\bigcirc\Box\bigcirc\varphi \Rightarrow \Diamond\bigcirc\Diamond\varphi$$

   Justify your answer, either informally or by appealing to the formal semantics. **[7]**

   (d) How does temporal logic extend classical logic? In your answer give an example of a statement that is more naturally represented in temporal, rather than classical, logic. **[8]**

   (e) Given $\Box(\varphi \Rightarrow \Diamond\psi)$, then if we also know that $\Box\varphi$ is true, how many times will ψ be forced to occur? **[4]**

4. This question concerns the relationship between temporal logic and programs.

   (a) Give a temporal formula capturing a semantics of the statement in a standard imperative programming language:

   ```
   (if (x>2) then x:=1 else x:=x+2); end
   ```

   [8]

   (b) What temporal formula would we write to describe the property that, at some point in the above program's execution, the variable x is guaranteed to have a value less than 5? Is this a liveness property, a safety property, or neither? **[5]**

   (c) Given temporal specifications of two components, $Spec_X$ and $Spec_Y$, explain what the purpose of the *Comms* formula in

   $$Spec_A \wedge Spec_B \wedge Comms$$

   is. Give examples of three possible types of formulae that might be used as *Comms* and explain what constraints these three represent. **[12]**

5. This question concerns model checking.

   (a) Describe how you might check, using the relation '$\models$', whether the formula $\Box(a \Rightarrow \bigcirc b)$ is true on a model defined by

   $$\begin{array}{rcl} \pi(0) & = & \{c\} \\ \pi(1) & = & \{a, c\} \\ \pi(2) & = & \{b, c\} \\ \pi(3) & = & \{c\} \\ \pi(4) & = & \{a, b\} \\ \forall i \in \mathbb{N}.\ (i > 4) & \Rightarrow & (\pi(i) = \{b\}) \end{array}$$

   What answer should you get? **[8]**

   (b) Why is an automaton generated from a formula such as $\neg\ \varphi$ useful in model checking the property φ over a particular model? **[4]**

   (c) If we have a model structure derived from a program, what does model checking actually test? If model checking fails, what does this say about the original program? How would this aid the software development process? **[6]**

   (d) Describe two current problems with the model checking approach in general. **[7]**

THE UNIVERSITY
of LIVERPOOL

6. Consider the following Promela code describing a three process system.

```
proctype A (chan in, out, dump)
{
        int total;
        total = 0;                              /* initial state */
L1:     out!(total+3);
        in?total;
        if
        :: (total < 8) -> goto L1;
        :: (total >= 8) -> dump!total
        fi }

proctype B (chan in, out)
{
        int total;
S1:     in?total;
        assert(total <= 10);
        out!total;
        goto S1 }

proctype C (chan in)
{
        int dumped;
        in?dumped;
        assert(dumped == 9)
}

init {  chan a2b = [0] of { int };
        chan b2a = [0] of { int };
        chan dump = [0] of { int };
        atomic { run A(b2a, a2b, dump);
                 run B(a2b, b2a);
                 run C(dump) }
     }
```

(a) What is the communication structure between the three processes A, B, and C, i.e. which channels link the processes and in which direction does information flow between them? **[6]**

(b) Will the assertion in process B succeed? Explain your answer. **[6]**

(c) Will the assertion in process C succeed? Explain your answer. **[6]**

(d) If we wanted to verify that dumped in C eventually has value 8, what temporal formula would we wish to check? **[7]**