UNIVERSITY OF LONDON

IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1998

MSc Degree in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Diploma of Membership of Imperial College

PAPER A4.15

PLANNING AND TEMPORAL REASONING

Monday, May 11th 1998, 10.00 - 12.00

Answer THREE questions

For admin. only: paper contains 4 questions

1 **Translation** (always write a dictionary with your predicates and their intended meaning)

a Translate statement (1) into two sorted predicate logic. Use the predicates in the table below.

b Translate statement (1) into temporal logic with connectives. Use the same predicates as in the previous item without $*$ and the argument for time.

predicate	meaning
$go^*(t, x, y)$	x goes to y at time t
$say^*(t, x, y, \varphi)$	x tells φ to y at time t
$meet^*(t, x, y,)$	x meets y at time t

"Whenever I go to a party, there is always someone there who says he had met me before" (1)

c Translate statement (2) below into two sorted predicate logic. Use the predicates in the table below, but add the time component and $*$.

d Translate statement (2) below into English

$$\boxplus \forall x[L(I, x) \rightarrow \Diamond \exists z F(I, x, z)] \qquad (2)$$

where the predicates in (2) have the following meaning:

predicate	meaning
$L(x, y)$	x looks for y
$F(x, y, z)$	x finds y at location z

e Rewrite the formulae below in an equivalent form where there is no until ($\mathcal{U}^+$) within since ($\mathcal{S}^-$). In other words, future and past should be separated in the formula.

Formulae to rewrite: $c\mathcal{S}^-(a\mathcal{U}^+b)$

2 **Specification**

Write a specification in two sorted predicate logic that regulates the use of a single printer according to the requirements given below. Make a dictionary of the predicates used and their meanings. Assume time is given by the set of natural numbers $\{0, 1, 2, \ldots\}$ and that there are only two users a and b:

a A user x can be in the state of using the printer ($U(t, x)$) or requesting the printer ($R(t, x)$), but not both.

b If both users have requested the printer and the printer is free, the one who placed his request first can use it.

c If both requested it at the same time, than the one who has not used it last gets access. If neither of them has ever used it, then access is granted to user a.

d Once access is granted, the user can use the printer for 3 time ticks, but has to stop using it after that. If he wants to continue using it, he must place another request first.

e Once a request is placed, it becomes active until access to the printer is granted.

Turn over ...

3 **Proof Rules for Planning Systems**

Using the proof rules for planning systems prove or disprove the following:

a $\boxplus A \wedge \diamondplus(B \wedge \boxminus C) \vdash \diamondplus(A \wedge B) \wedge C$

b $\vdash \diamondminus \exists x A(x) \leftrightarrow \exists x \diamondminus A(x)$ (assuming actions do not create or destroy objects).

c Can you prove (b) in the case when actions might create or destroy objects ? Explain.

4 Executable temporal logic

Consider the blocks world. We have blocks x, y, z, etc., which can be on the table or stacked on top of each other. Each block can have at most one block on top of it. The table can have many blocks on it. Use the following predicates:

predicate	meaning
$block(x)$	x is a block
$move(x, y)$	move block x to location y
$on(x, y)$	block x is on top of block y
$free(x)$	block x has nothing on top of it

a Explain the logical status of the action predicate *move* in comparison with the state predicates *on* and *free*.

b Using temporal logic with $\odot$, express *move* in terms of *on*.

c Write the specification for *free* and of preconditions for *move*, namely:

- A block can only be moved if it is free, and
- the argument of the predicate *move* representing the destination of the action can only be *table* or another free block.

d Write a program in executable temporal logic which generates a final state where all individual blocks are on the table, but such that sequences of actions which put two blocks on the table in two consecutives moments of time are not allowed. That is, if block A is put on the table at time t, no other block can be put on the table at time $t + 1$.

e Can you perform the action $move(x, y)$ when x is on top of y ? Why ?

Items (a)–(c) carry 20% *of the marks, item (d) carries* 30% *of the marks and item (e) carries* 10% *of the marks*

End of Paper