UNIVERSITY OF LONDON
IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 1998

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BSc Honours Degree in Mathematics and Computer Science Part II
MSci Honours Degree in Mathematics and Computer Science Part II
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the Royal College of Science Associateship of the City and Guilds of London Institute

PAPER 2.12 / MC2.12

SEMANTICS - OPERATIONAL
Wednesday, April 29th 1998, 2.00 - 3.30

Answer THREE questions

For admin. only: paper contains 4 questions

1 The following is the abstract syntax of a Knight control language:

$$\begin{array}{rcl} p & \in & \text{Program} \\ d & ::= & \mathtt{N} \mid \mathtt{E} \mid \mathtt{S} \mid \mathtt{W} \\ p & ::= & \mathtt{jump}(d) \mid p_1;p_2 \end{array}$$

The Knight navigates its way around a two dimensional space, composed of fields with coordinates that range from $(0, 0)$ to $(11, 7)$.

While it moves around, if it reaches an unmarked position, it will mark it; if it reaches a marked position, it will unmark it. It can move from its current position to the relative co-ordinates indicated by the direction d: if $d = \mathtt{N}$, then the relative co-ordinates are $(1, 2)$, if $d = \mathtt{E}$, they are $(2, -1)$, if $d = \mathtt{S}$, they are $(-1, -2)$, and if $d = \mathtt{W}$, they are $(-2, 1)$. However, if the `jump` would bring the Knight outside the board, the instruction is to be ignored (so if the Knight is at $(1, 1)$, performing `jump(W)` would leave the Knight at $(1, 1)$).

You can ignore the differences between the syntactic and the semantic representation of numbers: you do not need to write $\mathcal{N}[\![n]\!]$, but can use n instead.

a The state of the Knight is represented by a triple: $(x, y, \mathtt{F})$. The first two elements give the Knight's current position (cartesian coordinates); the third element is a a double array of booleans indicating whether any position in the $\mathtt{F}$ is marked or not.
Give the Natural Semantics of programs.

b The Knight has configurations $\langle c, e, s\rangle \in \text{Code} \times \text{Stack} \times \text{State}$, where

$$\begin{array}{rcl} c & \in & \text{Code} \\ i & \in & \text{Instruction} \\ c & ::= & \varepsilon \mid i : c \\ i & ::= & \mathtt{NORTH} \mid \mathtt{EAST} \mid \mathtt{SOUTH} \mid \mathtt{WEST} \mid \mathtt{MOVE} \end{array}$$

Define an operational semantics for the Knight.

c Define suitable translation functions to translate control programs into Knight code.

d Assuming that:

$$\text{if } \langle c_1, e_1, s\rangle \rhd^k \langle c', e', s'\rangle \text{ then } \langle c_1 : c_2, e_1 : e_2, s\rangle \rhd^k \langle c' : c_2, e' : e_2, s'\rangle,$$

show that the translation function for programs is correct. Show at least one base case and one inductive case.

The four parts carry 20%, 20%, 10%, and 50% of the marks, respectively.

2 The abstract syntax for the language Exif-Loop is given by:

$$
\begin{array}{rcl}
x & \in & \text{Variable} \\
a & \in & \text{Arithmetic} - \text{expression} \\
b & \in & \text{Boolean} - \text{expression} \\
S & \in & \text{Statement} \\
a & ::= & n \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 \times a_2 \\
S & ::= & x := a \mid S_1 ; S_2 \mid \texttt{if } b \texttt{ then } S_1 \texttt{ else } S_2 \mid \texttt{skip} \\
 & & \texttt{loop } S_1 ; \texttt{ exif } b ; S_2 \texttt{ endl}
\end{array}
$$

(The idea of the `loop` - `endl` construct is that, other than in a `while` loop, before testing the boolean, first a number of statements will be executed. If the boolean tests `tt`, then execution of the loop is terminated.) The syntax of boolean expressions is unspecified.

a Assuming the existence of the function $b \mapsto \mathcal{B}[\![b]\!]s$ that defines the semantics of boolean expressions, define the Natural Semantics for Exif-Loop.

b Formulate what it would mean for the Natural Semantics for Exif-Loop to be terminating. Will it be possible to show termination for Exif-Loop? Explain your answer.

c Show that the Natural Semantics for Exif-Loop is deterministic.

d Give a possible extension of Exif-Loop such that determinism is lost. Give the Natural Semantics for your extension. Give one well-chosen counter example that shows that indeed determinism is lost.

The four parts carry 30%, 15%, 40%, and 15% of the marks, respectively.

Turn over ...

3 The abstract syntax for the language While is given by:

$$
\begin{array}{lll}
x & \in & \text{Variable} \\
a & \in & \text{Arithmetic} - \text{expression} \\
b & \in & \text{Boolean} - \text{expression} \\
S & \in & \text{Statement} \\
a & ::= & n \mid a_1 + a_2 \mid a_1 - a_2 \mid a_1 \times a_2 \\
S & ::= & x := a \mid S_1 ; S_2 \mid \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2 \mid \texttt{skip} \mid \texttt{while}\ b\ \texttt{do}\ S
\end{array}
$$

The syntax of boolean expressions is unspecified.

a Assuming the existence of suitable functions $a \mapsto \mathcal{A}[\![a]\!]$ and $b \mapsto \mathcal{B}[\![b]\!]s$ for the semantics of both arithmetic and boolean expressions, define the Structural Operational Semantics for While.

b Assuming that:

$$\text{if } \langle S_1, s_1 \rangle \Rightarrow^* s_2 \text{ then } \langle S_1; S_2, s_1 \rangle \Rightarrow^* \langle S_2, s_2 \rangle,$$

show that, for the language While, structural operational and natural semantics coincide. The proof will follow an inductive reasoning; it suffices to show a base case and one (non-trivial) inductive step in each direction.

c Extend the syntax of While with a `repeat` statement:

$$\texttt{repeat}\ S\ \texttt{until}\ b$$

Extend the Natural Semantics of While to cover this extension. Show that '$\texttt{repeat}\ S\ \texttt{until}\ b$' is semantically equivalent to '$S;\ \texttt{while}\ \neg b\ \texttt{do}\ S$'.

The three parts carry, respectively, 30%, 40%, and 30% of the marks.

4 a The abstract syntax for the language Repeat is given by:

$$\begin{array}{lcl} x & \in & \text{Variable} \\ a & \in & \text{Arithmetic} - \text{expression} \\ b & \in & \text{Boolean} - \text{expression} \\ S & \in & \text{Statement} \\ S & ::= & x := a \mid S1;S2 \mid \texttt{if}\ b\ \texttt{then}\ S1\ \texttt{else}\ S2 \mid \texttt{skip} \mid \texttt{repeat}\ S\ \texttt{until}\ b \end{array}$$

The syntax of expressions (both arithmetic and boolean) is unspecified.
Assuming the existence of suitable functions $a \mapsto \mathcal{A}[\![a]\!]$ and $b \mapsto \mathcal{B}[\![b]\!]s$ for the semantics of both arithmetic and boolean expressions, write down the denotational semantics of statements.

b
- How is the partial order relation $\sqsubseteq$ defined on functions?
- Give the definition of a ccpo.
- What is a continuous function?

c Let $f : D \to D$ be a continuous function on the ccpo $(D, \sqsubseteq)$ and let $d = \sqcup\{f^n \perp \mid n \geq 0\}$.

- Show that d is a fixed-point of f.
- Show that d is the least fixed-point of f.

The three parts carry, respectively, 30%, 30%, and 40% of the marks.

End of paper