IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2023

BEng Honours Degree in Computing Part II
MEng Honours Degrees in Computing Part II
BEng Honours Degree in Mathematics and Computer Science Part II
MEng Honours Degree in Mathematics and Computer Science Part II
BEng Honours Degree in Mathematics and Computer Science Part III
MEng Honours Degree in Mathematics and Computer Science Part III
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 City and Guilds of London Institute

PAPER COMP50003

MODELS OF COMPUTATION

Wednesday 17th May 2023, 10:00
Duration: 90 minutes

Answer ALL TWO questions

Paper contains 2 questions
Calculators required

1 This question is about the operational semantics of programming languages.

Consider a small, imperative language, which is very similar to the imperative languages we discussed in the lectures and tutorials, and whose syntax is defined below:

$E \in \mathit{Exp} \quad ::= \quad x \mid \mathtt{n} \mid E{+}E \mid E{-}E \mid \ldots$
$B \in \mathit{Bool} \quad ::= \quad E{>}E \mid \ldots$
$C \in \mathit{Com} \quad ::= \quad x := E \mid C;C \mid \mathtt{skip} \mid \mathtt{while}\ B\ \mathtt{do}\ C$

Evaluation is in terms of large step semantics. Expression evaluation has the shape $\langle E, s\rangle \Downarrow n$, where $n \in \mathbb{Z}$. Boolean evaluation has the shape $\langle B, s\rangle \Downarrow \mathtt{b}$, where $\mathtt{b} \in \{\mathtt{true}, \mathtt{false}\}$. Command evaluation has the shape $\langle C, s\rangle \Downarrow s'$.

$$\text{E.1}\ \frac{}{\langle \mathtt{n}, s\rangle \Downarrow n} \qquad \text{E.2}\ \frac{}{\langle x, s\rangle \Downarrow n}\ n = s(x)$$

$$\text{E.3}\ \frac{\langle E_1, s\rangle \Downarrow n_1 \quad \langle E_2, s\rangle \Downarrow n_2}{\langle E_1{+}E_2, s\rangle \Downarrow n}\ n = n_1 + n_2 \qquad \text{E.4}\ \frac{\langle E_1, s\rangle \Downarrow n_1 \quad \langle E_2, s\rangle \Downarrow n_2}{\langle E_1{-}E_2, s\rangle \Downarrow n}\ n = n_1 - n_2$$

$$\text{B.1}\ \frac{\langle E_1, s\rangle \Downarrow n_1 \quad \langle E_2, s\rangle \Downarrow n_2}{\langle E_1{>}E_2, s\rangle \Downarrow \mathtt{true}}\ n_1 > n_2 \qquad \text{B.2}\ \frac{\langle E_1, s\rangle \Downarrow n_1 \quad \langle E_2, s\rangle \Downarrow n_2}{\langle E_1{>}E_2, s\rangle \Downarrow \mathtt{false}}\ n_1 \leq n_2$$

$$\text{C.1}\ \frac{}{\langle \mathtt{skip}, s\rangle \Downarrow s} \qquad \text{C.2}\ \frac{\langle E, s\rangle \Downarrow v}{\langle x := E, s\rangle \Downarrow s[x \mapsto v]} \qquad \text{C.3}\ \frac{\langle C_1, s\rangle \Downarrow s'' \quad \langle C_2, s''\rangle \Downarrow s'}{\langle C_1; C_2, s\rangle \Downarrow s'}$$

$$\text{C.4}\ \frac{\langle B, s\rangle \Downarrow \mathtt{false}}{\langle \mathtt{while}\ B\ \mathtt{do}\ C, s\rangle \Downarrow s} \qquad \text{C.5}\ \frac{\langle B, s\rangle \Downarrow \mathtt{true} \quad \langle C, s\rangle \Downarrow s'' \quad \langle \mathtt{while}\ B\ \mathtt{do}\ C, s''\rangle \Downarrow s'}{\langle \mathtt{while}\ B\ \mathtt{do}\ C, s\rangle \Downarrow s'}$$

a Prove that for all commands C_1, C_2, and C_3:

$(A)\ \forall s, s' \in \mathit{State}.[\ \langle (C_1; C_2); C_3, s\rangle \Downarrow s' \implies \langle C_1; (C_2; C_3), s\rangle \Downarrow s'\ ]$

Write what is assumed, what is to be shown, what is taken arbitrary, and justify each step.

b The function $\mathcal{S}(E)$ simplifies expressions by removing superfluous additions or subtractions of 0, and is defined below:

$$\begin{array}{rcll}
\mathcal{S}(\mathtt{n}) & = & \mathtt{n} & \text{for all numbers n} \\
\mathcal{S}(\mathtt{x}) & = & \mathtt{x} & \text{for all variables x} \\
\mathcal{S}(E_1 + 0) & = & \mathcal{S}(E_1) & \\
\mathcal{S}(0 + E_2) & = & \mathcal{S}(E_2) & \\
\mathcal{S}(E_1 + E_2) & = & \mathcal{S}(E_1) + \mathcal{S}(E_2) & \text{if } E_1 \neq 0, \text{ and } E_2 \neq 0 \\
\mathcal{S}(E_1 - 0) & = & \mathcal{S}(E_1) & \\
\mathcal{S}(0 - E_2) & = & 0 - \mathcal{S}(E_2) & \\
\mathcal{S}(E_1 - E_2) & = & \mathcal{S}(E_1) - \mathcal{S}(E_2) & \text{if } E_1 \neq 0, \text{ and } E_2 \neq 0
\end{array}$$

i) Write out the value of $\mathcal{S}(\texttt{((0+x)+(y-0))+(z-(0-x))})$. You do not need to write any calculations or intermediate steps.

ii) Assertion (B) given below can be proven by **induction on the definition** of $\mathcal{S}(\_)$:

$(B)\quad \forall E \in Exp.\forall s \in State.\forall m \in \mathbb{Z}.[\quad \langle E, s\rangle \Downarrow m \implies \langle \mathcal{S}(E), s\rangle \Downarrow m \quad]$

Write out the proof for **one base case**, and **one inductive step** – you may chose which one. For each of these (the base case and the inductive step you chose), write what is assumed, what is to be shown, what is taken arbitrary, and justify each proof step.

c Consider the command C_w defined below

$$C_w \triangleq \texttt{while (N - cnt > 0) do (acc:=acc+M; cnt:=cnt+1)}$$

i) Assume that $\texttt{M} = 8$, and $\texttt{N} = 4$. Take states s, s', such that $s(\texttt{acc}) = 0$, and $\langle C_w, s\rangle \Downarrow s'$.

a) If $s(\texttt{cnt}) = 0$, then what is the value of $s'(\texttt{cnt})$, and of $s'(\texttt{acc})$.

b) If $s(\texttt{cnt}) = 10$, then what is the value $s'(\texttt{cnt})$, and of $s'(\texttt{acc})$?

(You do not need to show how you obtained your answer.)

ii) We want to prove

(C) $\forall M, N \in \mathbb{N}.\forall s, s' \in State.$
$[\, s(\texttt{cnt}) = 0 = s(\texttt{acc}) \wedge \langle C_w, s\rangle \Downarrow s' \implies s'(\texttt{acc}) = N * M \,]$

a) (C) is too weak to be proven directly. Write a stronger assertion, (D), which implies (C), and which *can* be proven by induction.

b) Over what entity will be the induction in the proof of (D)?
(No more than 5 words needed. Also, notice that $M, N \in \mathbb{N}$, and therefore M, N are not negative.)

d Consider a new command,

$$\texttt{repeat}\ C\ \texttt{until}\ B$$

which repeats executing C, until B becomes true. In particular, it executes C at least once.

Give large step operational semantics for the repeat command.

The four parts carry, respectively, 20%, 40%, 25%, and 15% of the marks.

2a Consider the following list of (natural) numbers:

$$[54, 20, 16, 2808, 7, 69600, 30688, 68, 0]$$

as a (partial) encoding of a Register Machine (RM) R, i.e. each number is the encoding of a single instruction of R.

Note that the binary representations of these numbers are: 110110, 10100, 10000, 101011111000, 111, 10000111111100000, 111011111100000,1000100, and 0. What is the length of the (complete) binary encoding of all of R?

Decode every instruction in order to get the program for R. Give a graphical representation of R and describe what it computes if started with $R_0 = n$ and all other registers zero, for example for $n = 1, 2, 3, 4, 5$.

b Consider the following Turing Machine $T = (Q, \Sigma, q_1, \delta)$ with $Q = \{q_1, q_2, q_3, q_4, q_5, r, a\}$, $\Sigma = \{\text{␣}, 0, 1\}$ and

δ	␣	0	1
q_1	$(r, \text{␣}, R)$	$(q_2, \text{␣}, R)$	$(r, 1, R)$
q_2	$(a, \text{␣}, R)$	$(q_2, 1, R)$	$(q_3, 1, R)$
q_3	$(q_5, \text{␣}, L)$	$(q_4, 0, R)$	$(q_5, 1, R)$
q_4	$(r, \text{␣}, R)$	$(q_3, 1, R)$	$(q_4, 1, R)$
q_5	$(q_2, \text{␣}, R)$	$(q_5, 0, L)$	$(q_5, 1, L)$
r			
a			

Describe the execution of T (e.g. in the style used in the tutorials) with the initial tape 000 and the tape 0000 (otherwise empty, i.e. containing only '␣'s).

For which kind of tapes containing initially only 0's (otherwise empty) will T halt in state a, i.e. what tapes does it (a)ccept?

c Consider the λ-calculus: What is a grounded term? Does every grounded term have a (β-)normal form? Explain why or why not?

Using Church numerals evaluate the following expression:

$$\text{plus } \underline{1}\ \underline{1}$$

Expand the term and execute each β reduction step until you reach a normal form.

The three parts carry, respectively, 40%, 40%, and 20% of the marks.