Two hours - online hybrid

EXAM PAPER MUST NOT BE REMOVED FROM THE EXAM ROOM

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Computation

Date: Tuesday 4th June 2019
Time: 09:45 - 11:45

This is a hybrid examination with sections to be answered online and questions to be answered on paper

Please answer All Questions in Section A and Section B online and All Questions in Section C and Section D each in separate answerbooks

© The University of Manchester, 2019

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text

[PTO]
Sections A and B contain multiple choice questions and are, therefore, restricted
Section C

1. Consider the following:

Proposition: For any two regular expressions $p_1$ and $p_2$, the regular expression $(p_1|p_2)^*$ is equivalent to $(p_1^*|p_2^*)$, where equivalence means that the regular expressions define precisely the same language.

Give a proof of the above proposition or demonstrate that it is false. (3 marks)

2. Consider the language of words over \{a, b, c\} such that each word:

(a) finishes with a c; and

(b) either

i. starts with an a; or

ii. has no more than 1 b.

- Give a DFA that accepts precisely this language
- Give a grammar that will generate precisely this language

(6 marks)

3. Consider the following DFA over the alphabet \{a, b, c\}.

![DFA Diagram]

Give a regular expression that defines the same language as the DFA. (6 marks)
Section D

1. State with justification the Big-O run time of the following programs:

(a) \( y := 5 + (15 \times 20) \);

(b) \textbf{while} \ \ y > 0 \ \textbf{do} ( \\
    \hspace{1em} y := y - 1; \\
    )

(c) \( y := 5 + (15 \times 20); \ z := 0; \)
    \textbf{while} \ y < 350 \ \textbf{do} ( \\
    \hspace{1em} y := y + 10; \\
    \hspace{1em} z := z + y; \\
    )

(d) \( y := 0; \)
    \textbf{while} \ y < n \ \textbf{do} ( \\
    \hspace{1em} y := y + 1; \\
    )

(e) \( y := 2; \ p := 1; \)
    \textbf{while} \ y < n \ \textbf{do} ( \\
    \hspace{1em} x := n; \\
    \hspace{1em} \textbf{while} \ x > 0 \ \textbf{do} ( \\
    \hspace{2em} x := x - y; \\
    \hspace{1em} ); \\
    \hspace{1em} \textbf{if} \ x == 0 \ \textbf{then} \ p := p - 1; \\
    \hspace{1em} y := y + 1; \\
    )

(5 marks)

2. State the halting problem and explain why it is both undecidable and partially decidable.

(5 marks)
3. Given the while program Prog:

\[
\text{if } x > y \text{ then } z := y \text{ else } z := x
\]

(a) Write a specification

\[
\{P\} \text{ Prog } \{Q\}
\]

that captures the property that the program places the minimum of \(x\) and \(y\) in \(z\).

(b) Prove the program is correct.

(c) Explain in one sentence why the program is either partially or totally correct.

(5 marks)
Semantics of the WHILE language

[ass] \(< x := a, s > \Rightarrow s[x \mapsto A[a] s]\)

[skip] \(< \text{skip}, s > \Rightarrow s\)

[comp'] \(< S_1, s > \Rightarrow < S'_1, s' >\)
\(< S_1; S_2, s > \Rightarrow < S'_1; S_2, s' >\)

[comp'] \(< S_1, s > \Rightarrow s'\)
\(< S_1; S_2, s > \Rightarrow < S_2, s' >\)

[if] \(< \text{if } b \text{ then } S_1 \text{ else } S_2, s > \Rightarrow < S_1, s >\) if \(B[b] s = \text{tt}\)

[if] \(< \text{if } b \text{ then } S_1 \text{ else } S_2, s > \Rightarrow < S_2, s >\) if \(B[b] s = \text{ff}\)

[while] \(< \text{while } b \text{ do } S, s > \Rightarrow < \text{if } b \text{ then } (S; \text{while } b \text{ do } S) \text{ else } \text{skip}, s >\)

Axiomatic Rules for Partial Correctness of WHILE

\(\text{ass}_p\) \(\{ P[x \mapsto A[a]] \} x := a \{ P \}\)

\(\text{skip}_p\) \(\{ P \} \text{skip} \{ P \}\)

\(\text{comp}_p\) \(\{ P \} S_1 \{ Q \}, \{ Q \} S_2 \{ R \}\)
\(\{ P \} S_1; S_2 \{ R \}\)

\(\text{if}_p\) \(\{ P \land B[b] \} S_1 \{ Q \}, \{ P \land \neg B[b] \} S_2 \{ Q \}\)
\(\{ P \} \text{if } b \text{ then } S_1 \text{ else } S_2 \{ Q \}\)

\(\text{while}_p\) \(\{ P \land B[b] \} S \{ P \}\)
\(\{ P \} \text{while } b \text{ do } S \{ P \land \neg B[b] \}\)

\(\text{cons}_p\) \(\{ P' \} S \{ Q' \}\)
\(\{ P \} S \{ Q \}\) if \(P \Rightarrow P'\) and \(Q' \Rightarrow Q\)