Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Computation

Date: Friday 26th May 2017

Time: 14:00 - 16:00

Please answer ALL Questions provided.

Use a SEPARATE answerbook for each SECTION.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

Section A

Answer ALL questions

1. Consider the language $\mathcal{L}$ of all words over the alphabet $\{a, b\}$ that

   (a) start with an a; **and**

   (b) either

      i. finish with a b; **or**

      ii. contain at least two bs

   Note that this is not an *exclusive* or – it is possible for both conditions (i) and (ii) to hold.

   Give a description of $\mathcal{L}$ via the following means:

   a) using a DFA (3 marks)

   b) using a grammar (3 marks)

   c) using a regular expression (2 marks)

2. Give a DFA for the language of all words over the alphabet $\{a, b, c\}$ that match the regular expression $(abc^* \mid a^*bc)$.

   (6 marks)

3. Regular expressions obey distributive laws. For expressions p_1, p_2, p_3, any word matching the regular expression

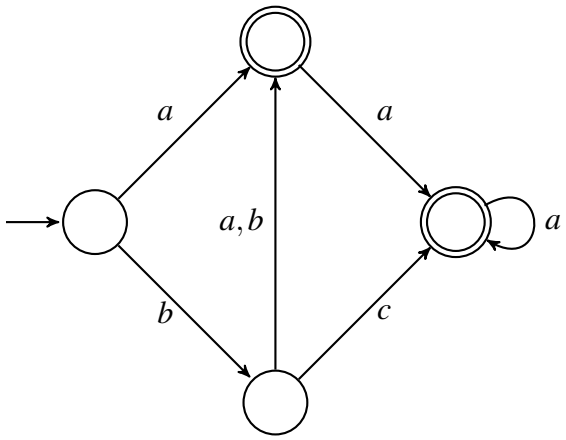
$$(p_1(p_2|p_3))$$

   also matches the regular expression

$$((p_1p_2)|(p_1p_3))$$

   Give a proof of this. (4 marks)

4. Consider the DFA X shown below.



a) X has 4 states. Provide a DFA which has at most 3 states and is equivalent to X, where equivalence of two automata means that they accept exactly the same language. (2 marks)

b) Demonstrate that the automaton you have given **is** equivalent to X. (3 marks)

5. Consider the following grammar. The underlying alphabet is $\{a,b,c\}$, there are two non-terminal symbols S and T, the start symbol is S and the production rules are:

$$\begin{array}{rcl} S & \rightarrow & TaTbT \\ T & \rightarrow & aT \mid bT \mid cT \mid \varepsilon \end{array}$$

a) Show that this grammar is ambiguous. (2 marks)

b) Describe the language generated by this grammar (2 marks)

c) Give an unambiguous grammar for the same language (3 marks)

Section B

Answer ALL questions

6. For each of the following statements, state whether it is true or false and briefly explain your answer. No marks will be given without an explanation.

   a) There are uncomputable functions of type `Int32` $\rightarrow$ `Int32` where `Int32` is the set of all 32-bit integers. (2 marks)

   b) The Halting Problem tells us that there are no programs that we can prove will always halt. (2 marks)

   c) Let $p(x)$ be a partially decidable predicate such that $p(5)$ is true and $p(6)$ is false.

   A program P exists that takes x as an input and halts with the correct value for $p(5)$.

   No such program can exist that halts with the correct value for $p(6)$.

   (Note that this consists of two statements and you should give an answer for each statement.) (2 marks)

   d) We can find a program P to make $\{true\}\ P\ \{false\}$ true. (2 marks)

   e) There is no relationship between a problem's *time complexity* and *space complexity*. (2 marks)

7. You are in charge of a student society and you have asked 5 students to each write a program to help *sort* the student records.

   a) You are told that the programs have the following average-case complexities where n is the number of student records. Order the functions in terms of their asymptotic complexity (i.e. Big-O) from smallest to largest, indicating, if relevant, where two functions have the same complexity.

$$100n \quad 2^n \quad n \log n \quad (2n)^2 \quad \left(\frac{n}{10}\right)^2$$

   (2 marks)

   b) This year you have 500 students in your society but next year you will have 1000. Which program should you pick for each year? Justify your answer. (2 marks)

8. Your friend complains that it does not make sense to study the `while` programming language when discussing computability as Java can compute anything.

   a) Give a proof sketch showing that there exist functions that cannot be computed by any Java program (it is not necessary to consider the full range of Java features). (4 marks)

   b) What are the *Church-Turing Thesis* and the *universal program*? By reference to these concepts, or otherwise, argue that for any Java program there exists a `while` program that computes the same function. (3 marks)

9. Consider the following `while` program which we will refer to as S.

   while x>0 **do** (y=y−1; x=x−1)

   a) Use the semantic rules given on the next page to compute the result of running the program on the initial state $[x \mapsto 1, y \mapsto 6]$. (2 marks)

   b) Complete the following specification by giving a predicate Q that captures the behaviour that $y - x$ is placed in y and a predicate P that makes the precondition strong enough to prove the resulting postconditon. Hint: what values of x and y will ensure $y > 0$.

   $$\{\ x \geq 0 \wedge P \wedge y = a \wedge x = b\ \}\ S\ \{\ y > 0 \wedge Q\ \}$$

   (1 mark)

   c) What is a *loop invariant*? Explain why $(a - x = b - y \wedge x < y)$ is a loop invariant for the above `while` program. (1 mark)

   d) Use the axiomatic rules on the next page to prove the partial correctness specification you completed in (b) using the loop invariant from (c). (4 marks)

   e) Give a *loop variant* for the program, what are loop variants used for? (1 mark)

Semantics of the `while` language

[ass] $< x := a,\ s > \ \Rightarrow\ s[x \mapsto \mathcal{A}[\![a]\!]\ s]$

[skip] $< \texttt{skip},\ s > \ \Rightarrow\ s$

[comp^1] $$\frac{< S_1,\ s > \ \Rightarrow\ < S_1',\ s' >}{< S_1; S_2,\ s > \ \Rightarrow\ < S_1'; S_2,\ s' >}$$

[comp^2] $$\frac{< S_1,\ s > \ \Rightarrow\ s'}{< S_1; S_2,\ s > \ \Rightarrow\ < S_2,\ s' >}$$

[if^{tt}] $< \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ s > \ \Rightarrow\ < S_1,\ s >$ if $\mathcal{B}[\![b]\!]\ s = \mathbf{tt}$

[if^{ff}] $< \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2,\ s > \ \Rightarrow\ < S_2,\ s >$ if $\mathcal{B}[\![b]\!]\ s = \mathbf{ff}$

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

Axiomatic Rules for Partial Correctness of `while`

ass$_p$ $\{\ P[x \mapsto \mathcal{A}[\![a]\!]]\ \}\ x := a\ \{\ P\ \}$

skip$_p$ $\{\ P\ \}\ \texttt{skip}\ \{\ P\ \}$

comp$_p$ $$\frac{\{\ P\ \}\ S_1\ \{\ Q\ \},\ \{\ Q\ \}\ S_2\ \{\ R\ \}}{\{\ P\ \}\ S_1;\ S_2\ \{\ R\ \}}$$

if$_p$ $$\frac{\{\ P \wedge \mathcal{B}[\![b]\!]\ \}\ S_1\ \{\ Q\ \},\ \{\ P \wedge \neg\mathcal{B}[\![b]\!]\ \}\ S_2\ \{\ Q\ \}}{\{\ P\ \}\ \texttt{if}\ b\ \texttt{then}\ S_1\ \texttt{else}\ S_2\ \{\ Q\ \}}$$

while$_p$ $$\frac{\{\ P \wedge \mathcal{B}[\![b]\!]\ \}\ S\ \{\ P\ \}}{\{\ P\ \}\ \texttt{while}\ b\ \texttt{do}\ S\ \{\ P \wedge \neg\mathcal{B}[\![b]\!]\ \}}$$

cons$_p$ $$\frac{\{\ P'\ \}\ S\ \{\ Q'\ \}}{\{\ P\ \}\ S\ \{\ Q\ \}}$$ if $P \Rightarrow P'$ and $Q' \Rightarrow Q$

END OF EXAMINATION