A06472 No calculator permitted in this examination

THE UNIVERSITY OF BIRMINGHAM

THIS PAGE TO BE REPLACED BY OFFICE

06 05934

Models of Computation

May 2012 1½ hours

[Answer THREE questions out of the four.
The marks available total 102, but will be capped at 100.]

1. (a) (i) Write out a regular expression that is matched by exactly those strings over an alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$ in which after any b there can be no more occurrences of a. **[4%]**

   (ii) Consider the two strings ccaccbcc, which has the given property, and ccbccacc, which does not. Explain why ccaccbcc matches your pattern, but ccbccacc does not. **[8%]**

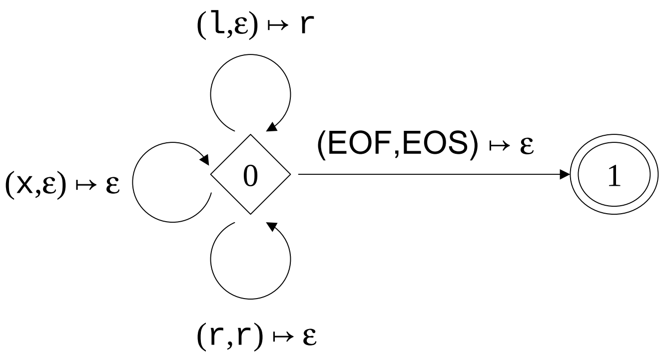
   (b) Design a *deterministic* finite automaton that accepts exactly those strings that match the regular expression $(\mathtt{ab}* \mid (\mathtt{ab})*)$. **[12%]**

   (c) Consider the language L consisting of all those strings over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ that contain more as than bs. Use the Pumping Lemma to show that L is not regular. **[10%]**

2. In this question we consider a grammar given, over an alphabet $\Sigma = \{\mathtt{l}, \mathtt{r}, \mathtt{x}\}$ of terminals, by

$$S ::= \varepsilon \mid TS$$
$$T ::= \mathtt{x} \mid \mathtt{l}S\mathtt{r}$$

We also consider a pushdown automaton whose state transition diagram is as follows.



(a) Of the two strings $\mathtt{xlrxrxlx}$ and $\mathtt{xlxlxrrxx}$, one is derivable (from S) in the grammar and one is not. Which is which? Justify your answer by giving in each case either a derivation tree or a careful argument to show that every attempt to find a derivation fails. **[14%]**

(b) Of the following three strings, which are accepted by the pushdown automaton?

$$\mathtt{xlr},\ \mathtt{lxl},\ \mathtt{rlx}$$

For each one, show the operation of the machine when that string is presented as input. (At each stage of the operation, show the state, the remaining input and the stack contents.) **[6%]**

In the rest of the question we shall use the notion of *bracket count*. (Think of $\mathtt{l}$ and $\mathtt{r}$ as left and right brackets.) Imagine counting along from the left of a string, starting 0 and adding $+1$ for each $\mathtt{l}$, -1 for each $\mathtt{r}$ and 0 for each $\mathtt{x}$. At each point of the string, the total so far is called the bracket count there. For example, in the string $\mathtt{xxlxlrrrx}$ the bracket counts are shown as subscripts as

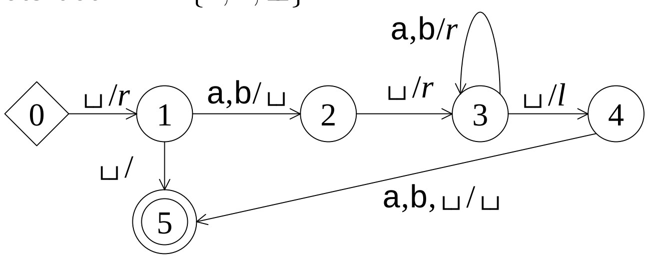
$${}_0\mathtt{x}_0\mathtt{x}_0\mathtt{l}_1\mathtt{x}_1\mathtt{l}_2\mathtt{r}_1\mathtt{r}_0\mathtt{r}_{-1}\mathtt{x}_{-1}.$$

We say that a string is *well bracketed* if its bracket count finishes at 0 and never goes negative at any intermediate point.

(c) Explain why the strings accepted by the automaton are *exactly* the well bracketed ones. **[7%]**

(d) Show by induction that every string derived from S is well bracketed. **[7%]**

3. (a) The Turing machine with the following transition diagram has input and tape alphabets both $\Sigma = \{\mathtt{a}, \mathtt{b}, \sqcup\}$.



   (i) Write out the transition table for this machine. **[6%]**

   (ii) Write out the configurations for the execution of this machine if it starts with input $\mathtt{abb}$ and the read/write head immediately to its left: so the initial configuration is $0\sqcup\mathtt{abb}\sqcup$. **[6%]**

(b) What does it mean to say that a Turing machine executes *in polynomial time?* (You may use the "O" notation without defining it.) **[4%]**

(c) (i) Outline how a Turing machine M with two tapes, each with its own read/write head, can be simulated by a Turing machine N with one tape. **[5%]**

   (ii) Explain why if M executes in polynomial time then so does N. **[5%]**

   (Your outline in part (i) can be very brief, but it should include enough detail to support your explanation in part (ii).)

(d) Which of the following operations are known to be executable in polynomial time on a Turing machine?

   (i) Determining for two integers n and m whether m is a factor of n. **[2%]**

   (ii) Determining for an integer n whether it has a factor other than 1 and n. **[2%]**

   (iii) Finding the smallest factor (other than 1) of n. **[2%]**

   How would your answers change if the "$\mathcal{P} = \mathcal{NP}$?" problem were solved, either way? Give reasons. **[2%]**

4. (a) We consider Java programs that take their input from a file and print their output to `System.out` when the program terminates. What does it mean to say that a property of such programs is *semantic?* What does *Rice's Theorem* tell us about semantic properties? **[5%]**

   (b) The following two properties of programs are both undecidable.

      (i) The output of the program can differ, depending on the first character of the input.

      (ii) The output of the program can differ, depending on the name of the input file.

   Which one is a semantic property? Outline how you would prove undecidability of each property, using Rice's Theorem where appropriate. **[12%]**

   (c) In the λ-calculus, what is the bracketing convention for xyz? Explain how this allows us to think of xyz as applying a function with *two* parameters. **[5%]**

   (d) Draw the syntax tree for the λ-term T, defined as $\lambda x.\lambda y.y(xxy)$. **[6%]**

   (e) If Y is defined to be TT, where T is as defined in part (d), show the β-reduction of Yf far enough to show the behaviour of Y as a *fixpoint combinator.* Explain what this means. **[6%]**