THE UNIVERSITY OF BIRMINGHAM

THIS PAGE TO BE REPLACED BY OFFICE

BannerCode?????

ModuleName?????

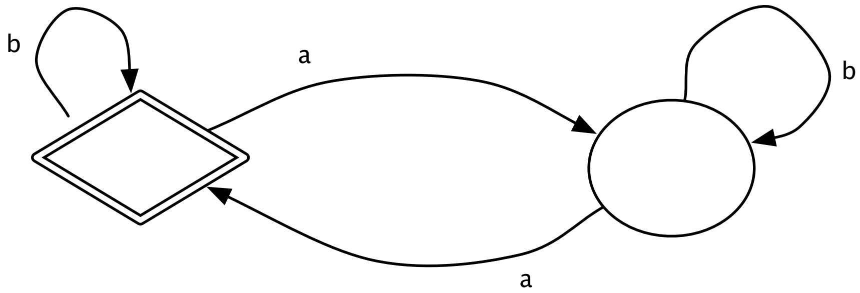
Date????? Duration?????

[Answer ?????? questions]

1. **[29%]** This question is about regular languages and finite-state automata.

   (a) Give a regular expression for the language over the alphabet $\{\mathtt{a}, \mathtt{b}\}$ consisting of all words in which a occurs an even number of times.
   Solution b*(ab*ab*)* **[7%]**

   (b) Give a deterministic finite state automaton for this language.

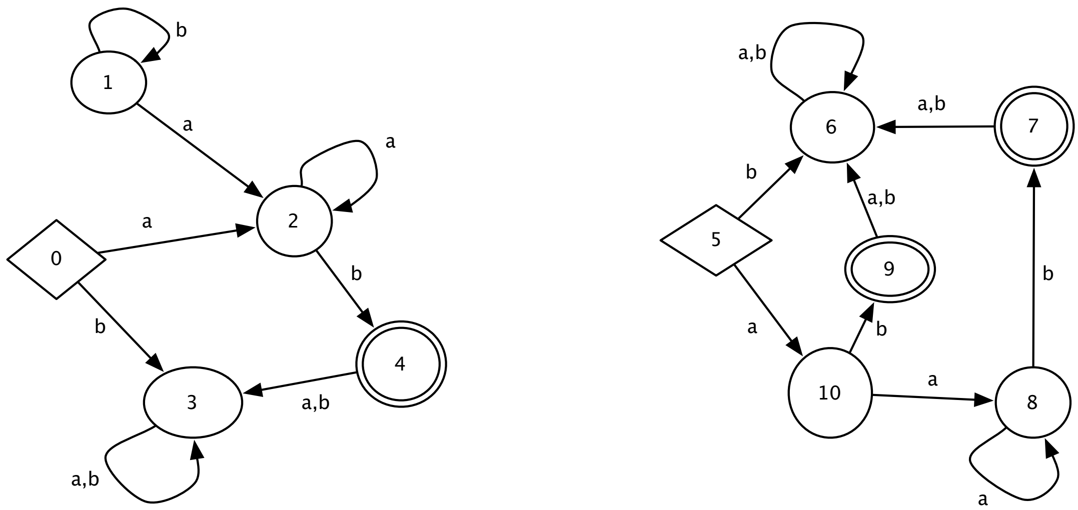


   Solution

   [7%]

   (c) Let L be the language over the alphabet $\{\mathtt{a}, \mathtt{b}\}$ consisting of all words in which a and b occur an equal number of times. Using the pumping lemma, show that L is not regular.
   Solution Suppose it is regular. By the pumping lemma, it has a pumping length n. Now $a^n b^n$ is a word of the language. The first n letters is a stretch of length n so it must contain a pumpable part, of length $k > 0$. Therefore $a^{n-k} b^n$ is a word of the language, contradiction. **[8%]**

   (d) Here are two deterministic finite state automata.



   Give a bisimulation to show that they recognize the same language.
   Solution

$$\{(0,5), (2,10), (2,8), (4,9), (4,7), (3,6)\}$$

   [7%]

2. **[22%]** This question is about decidability.

   You are an employee of Ellipse Ltd, which sells an integrated development environment (IDE) for Java programmers. The IDE is itself written in Java. A new version is being developed.

   (a) The manager says to you: "Create a button that determines whether, in the Java file selected by the user, there are two different declarations of variables with the same name." (For example, if there are two different lines declaring `int i`, then the button will print Yes.) Can this be accomplished? Explain your answer.

   Solution Yes, this is decidable via the following algorithm. Go through the file looking for each variable declaration, and in each case go through the file again to see if there is any other declaration of a variable with the same name. **[7%]**

   (b) The manager says to you: "Create a button that determines whether the Java method selected by the user may be called with some argument that will cause an exception to be thrown." Can this be accomplished? Explain your answer.

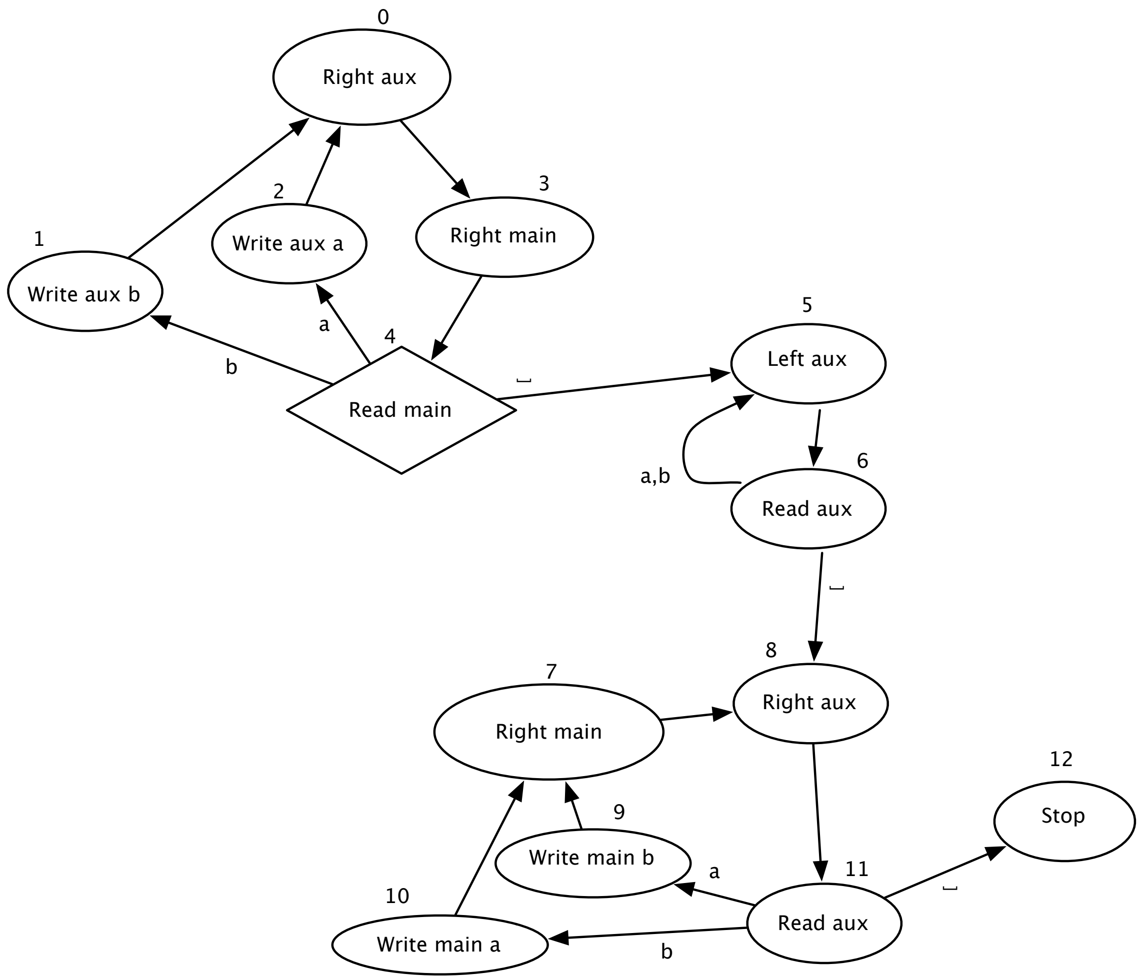
   Solution This property of the code is semantic—it depends only on the observable behaviour for each input. It is non-trivial since there are methods that have this property and methods that do not. So by Rice's theorem it is undecidable. **[7%]**

   (c) The manager has decided that the new IDE will be written not in Java but in a different programming language. Does that make a difference to your answers? Explain.

   Solution No. For the first part, any reasonable programming language could express this. For the second part, this answer is no by Church's thesis: the new language will have no more power than Java. **[8%]**

3. **[21%]** This question is about two-tape Turing machines.

   Look at the two-tape machine below.



Initially, the main tape contains a nonempty block of a's and b's with the rest of the tape blank, and the head over the leftmost character. The auxiliary tape is initially blank.

(a) If the initial block is ab, show the first five execution steps. (Altogether 25 steps are performed before the machine halts.) After each step, you should show the current state, the contents of each tape and the position of each head. **[7%]**

Solution

State	Main	Auxiliary
4	$\overset{H}{a}b$	$\overset{H}{\sqcup}$
2	$\overset{H}{a}b$	$\overset{H}{\sqcup}$
0	$\overset{H}{a}b$	$\overset{H}{a}$
3	$\overset{H}{a}b$	$a\overset{H}{\sqcup}$
4	$a\overset{H}{b}$	$a\overset{H}{\sqcup}$
1	$a\overset{H}{b}$	$a\overset{H}{\sqcup}$

(b) What does this machine do in general?

Solution It places a copy of the original block, with a's and b's replaced by b's and a's respectively, to the right of the original block, and then ends on the blank to the right of the copy. **[7%]**

(c) What is the complexity of this machine, in terms of the length n of the block that is initially on the main tape? Explain your answer briefly.

Solution The first phase, copying to the auxiliary tape is linear because it does a fixed amount of work for each character. The second phase, moving leftwards through the block on the auxiliary tape is linear for the same reason. The third phase, copying back to the main tape with a's and b's interchanged is linear for the same reason. Therefore, the total time taken is linear in n. **[7%]**

4. **[14%]** This question is about $\mathcal{NP}$.

In the popular puzzle Sudoku, a square grid of $n^2 \times n^2$ small squares is displayed, divided into $n \times n$ square subgrids each consisting of $n \times n$ small squares, and partially filled with numbers in the range 1 to n^2 inclusive. The task is to fill all the remaining squares with numbers in the range 1 to n^2 inclusive, in such a way that in each column, each row and each subgrid, every number appears precisely once. Here is an example with $n = 3$.

Partially filled grid

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Solution

5	3	*4*	*6*	7	*8*	*9*	*1*	*2*
6	*7*	*2*	1	9	5	*3*	*4*	*8*
1	9	8	*3*	*4*	*2*	*5*	6	*7*
8	*5*	*9*	*7*	6	*1*	*4*	*2*	3
4	*2*	*6*	8	*5*	3	*7*	*9*	1
7	*1*	*3*	*9*	2	*4*	*8*	*5*	6
9	6	*1*	*5*	*3*	*7*	2	8	*4*
2	*8*	*7*	4	1	9	*6*	*3*	5
3	*4*	*5*	*2*	8	*6*	*1*	7	9

(a) Not every partially filled grid is solvable. Explain why solvability of a partially filled grid is in $\mathcal{NP}$.

Solution Here is a polytime algorithm to test whether a given filling is a solution. Within each row, there are $n^2 \times (n^2 - 1)$ distinct pairs of entries; check each pair to see that they contain distinct numbers. There are n^2 rows so this takes $n^2 \times n^2 \times (n^2 - 1)$ steps. Likewise for columns and subgrids. Altogether we have $O(n^6)$ running time.

A partially filled grid g is solvable if there exists a filling f such that f is a solution for g. Since the latter is in $\mathcal{P}$, and solvability is its existence problem, solvability is in $\mathcal{NP}$. **[7%]**

(b) It has been shown that this problem is $\mathcal{NP}$-complete. Explain what that means.
Solution It means that every problem in $\mathcal{NP}$ can be reduced to it in polytime.
[7%]

5. **[14%]** This is a question about λ-calculus with arithmetic. The syntax of terms is given by the following grammar.

$$M \;::=\; x \mid M\,M \mid \lambda x.\,M \mid n \mid M + M$$

where x ranges over variables and n over the integers.

(a) Here is a term:

$$(\lambda x.\,\lambda y.\,x\,y)\,(\lambda x.\,x+y)\,(3+1)$$

By applying β-reductions and δ-reductions, reduce this term to normal form. **[7%]**

Solution

$$\begin{array}{rl}
(\lambda x.\,\lambda y.\,x\,y)\,(\lambda x.\,x+y)\,(3+1) & \rightsquigarrow_\delta \\
(\lambda x.\,\lambda y.\,x\,y)\,(\lambda x.\,x+y)\,4 & \rightsquigarrow_\beta \\
(\lambda u.\,(\lambda x.\,x+y)u)\,4 & \rightsquigarrow_\beta \\
(\lambda x.\,x+y)\,4 & \rightsquigarrow_\beta \\
4+y &
\end{array}$$

(b) Let the syntax of types be given by

$$A \;::=\; \texttt{int} \mid A \to A$$

where `int` is the type of integers.
For each natural number n we define a type A_n as follows:

$$\begin{array}{rcl}
A_0 & = & \texttt{int} \\
A_{n+1} & = & A_n \to \texttt{int}
\end{array}$$

Define, for each natural number n, a closed term that has type A_n and no other type. **[7%]**
Solution

$$\begin{array}{rcl}
M_0 & = & 3 \\
M_1 & = & \lambda x.\,x+5 \\
M_{n+2} & = & \lambda x.\,(x\,M_n)+6
\end{array}$$