Theories of Computation
Solutions

May 2016

Theories of Computation

Exam length: 1.5 hours. Answer all questions.

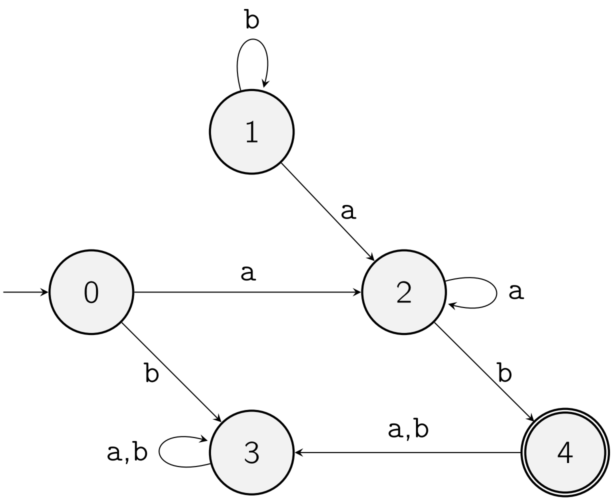
Question 1 : Regular Languages and Automata

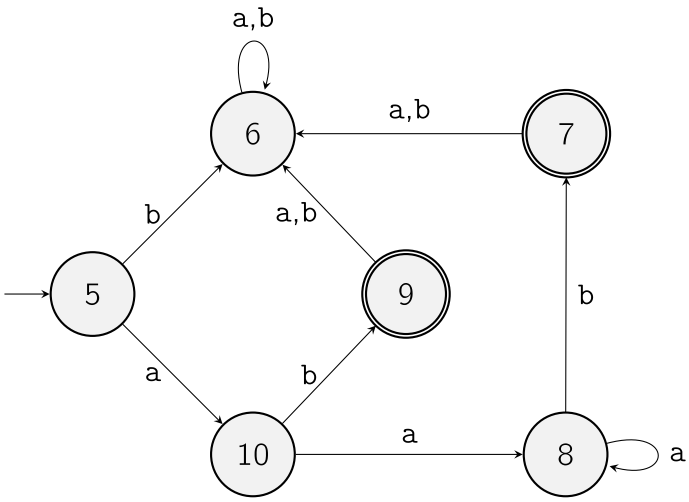
(a) Give a regular expression for the language over the alphabet $\{\mathtt{a}, \mathtt{b}\}$ consisting of all words in which a occurs an even number of times. **[7 marks]**

(b) Give a deterministic finite state automaton for this language. **[7 marks]**

(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. Show that L is not regular. **[8 marks]**

(d) Here are two deterministic finite state automata.



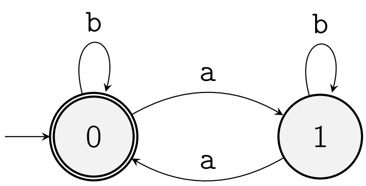


Show that they recognize the same language. **[7 marks]**

Solutions

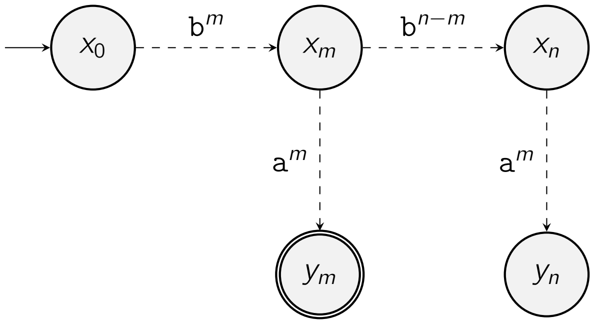
(a) $\mathtt{b}^*(\mathtt{a}\mathtt{b}^*\mathtt{a}\mathtt{b}^*)^*$

(b)



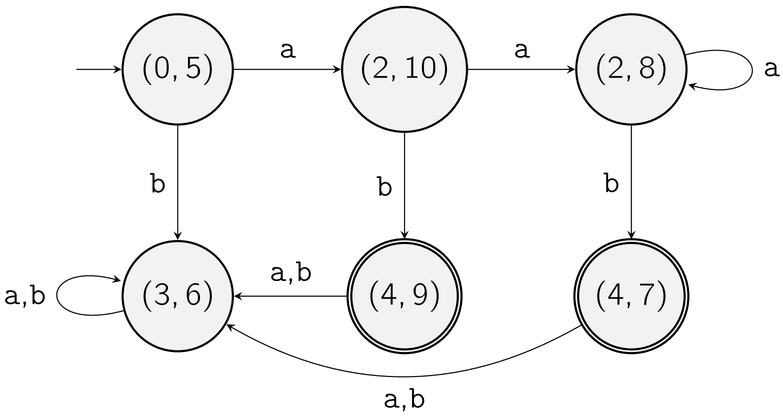
(c) Suppose that we can give a deterministic finite automaton D that recognizes L. For any $n \in \mathbb{N}$, let x_n denote the state of D reached after reading $\mathtt{b}^n$. A given state x_n

does not accept the word $\mathtt{a}^m$ for any m such that $1 \leq m < n$, while $\mathtt{a}^m$ would be accepted by the corresponding state x_m.



Hence any x_n is inequivalent to all x_p for $p < n$. This means that D would need to consist of infinitely many different states, which contradicts its assumed finiteness.

(d)



We see that each pair of states consists of either two accepting states or two rejecting states. So the two automata are language equivalent.

Question 2 : 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. **[7 marks]**

(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. **[7 marks]**

(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. **[8 marks]**

Solutions

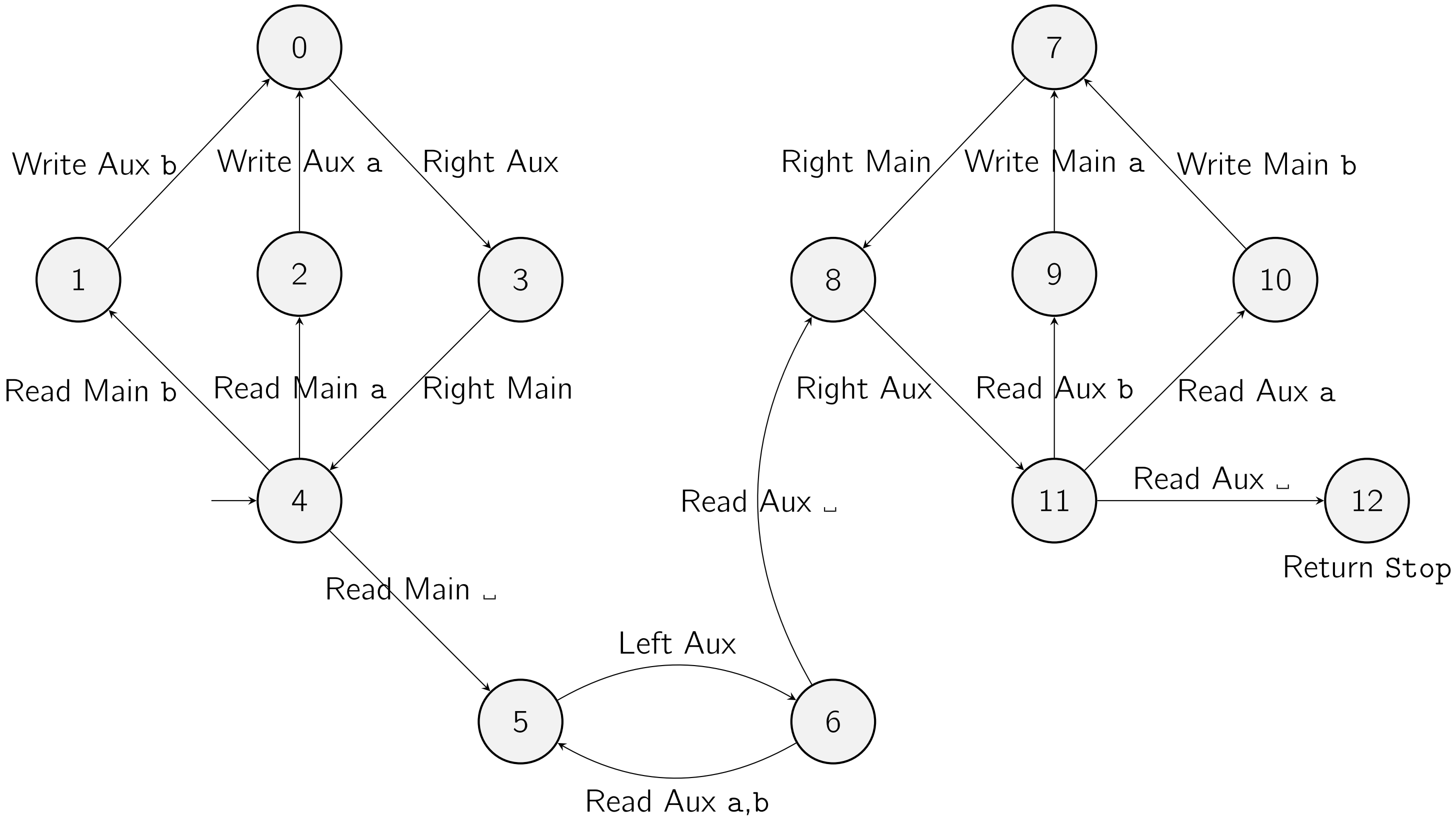
(a) Yes, this is decidable via the following algorithm. Go through the file looking for each variable declaration, and in each case go through the file again to see if there is any other declaration of a variable with the same name.

(b) 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.

(c) 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.

Question 3 : 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 marks]**

(b) What does this machine do in general? **[7 marks]**

(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. **[7 marks]**

Solutions

(a)

Main:	␣	•a	b	␣	␣	␣	4	Read Main a
Aux:	␣	␣	•␣	␣	␣	␣		
Main:	␣	•a	b	␣	␣	␣	2	Write Aux a
Aux:	␣	␣	•␣	␣	␣	␣		
Main:	␣	•a	b	␣	␣	␣	0	Right Aux
Aux:	␣	␣	•a	␣	␣	␣		
Main:	␣	•a	b	␣	␣	␣	3	Right Main
Aux:	␣	␣	a	•␣	␣	␣		
Main:	␣	a	•b	␣	␣	␣	4	Read b
Aux:	␣	␣	a	•␣	␣	␣		

(b) 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.

(c) 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*.

Question 4 : Complexity

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 $\mathcal{NP}$. **[7 marks]**

(b) It has been shown that this problem is $\mathcal{NP}$-complete. Explain what that means. **[7 marks]**

Solutions

(a) 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 $\mathcal{NP}$.

(b) It means that every problem in NP can be reduced to it in polytime.

Question 5 : Lambda calculus

Removed. Not on the syllabus this year. **[14 marks]**