Theories of Computation

Solutions

Main Summer Examinations 2025

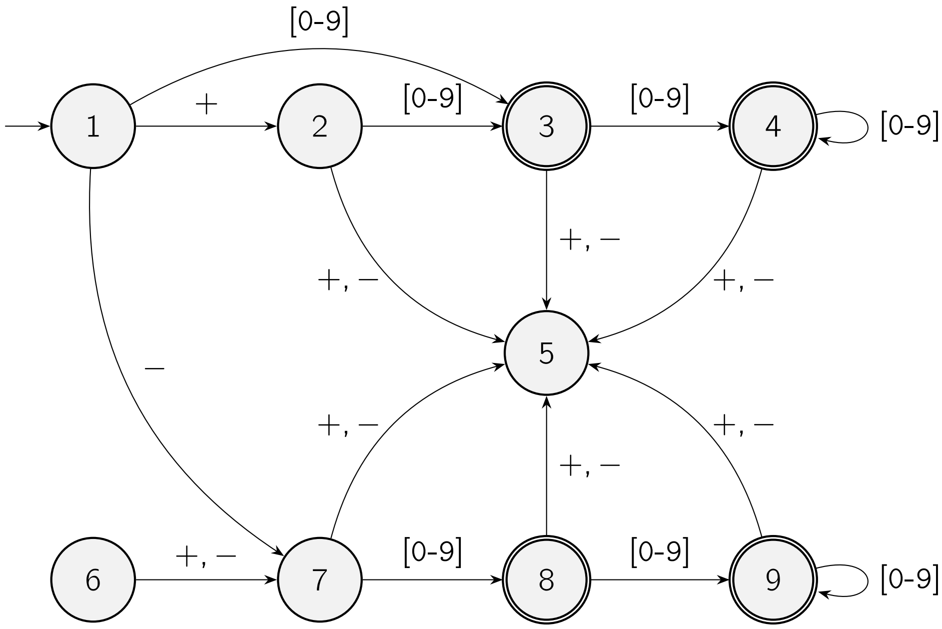
Note

Answer ALL questions.

Question 1 – Languages and Automata [34 marks]

(a) Write a regular expression over the alphabet $\Sigma = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ that recognises all even natural numbers (with leading zeros allowed). For example, 0, 000, 42 and 0585926 should be accepted; while 3, 101 and 074983249 should not. **[8 marks]**

(b) The following partial DFA recognises any integer over the alphabet $\Sigma \cup \{+, -\}$. Note that the [0-9]-transitions accept any digit.



(i) Give every reason why this is *not* a minimal partial DFA. **[8 marks]**

(ii) Draw an equivalent minimal partial DFA. To show that it is equivalent, label each state as the set of all corresponding states of the partial DFA above. **[9 marks]**

(c) Consider the following context-free grammar, which generates a language of simple mathematical expressions:

$$\Rightarrow S \quad ::= \quad S + S - 1 \quad | \quad 1 \quad | \quad 0 + S$$

For example, it can generate the expression $1 + 0 + 1 - 1$, which evaluates to 1.

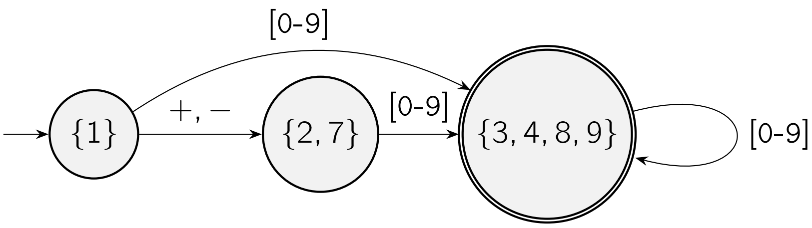
Prove, using structural induction, that the expressions generated by this language never evaluate to a number less than zero. **[9 marks]**

Model answers:

(a) (0|1|2|3|4|5|6|7|8|9)*(0|2|4|6|8) Any equivalent expression gets full marks. Partial marks are given for expressions with small mistakes, such as excluding leading zeros, or use of $+$ instead of $*$.

(b) (i) State 6 is unreachable,
State 5 is hopeless (does not lead to an accepting state),
States 2 and 7 are equivalent,
States 3, 4, 8 and 9 are equivalent (this may be written as two or more pairs of states).

(ii)



(c) In order to prove that every expression in the language *does not* evaluate to a *negative* number (i.e. that every expression S is such that $S \geq 0$), we will use structual induction to instead prove that every expression in the language *does* evaluate to a *positive* number (i.e. every expression S is such that $S \geq 1$); the conclusion will then be immediate because $S \geq 1 \geq 0$.

The structural inductive proof has three cases:

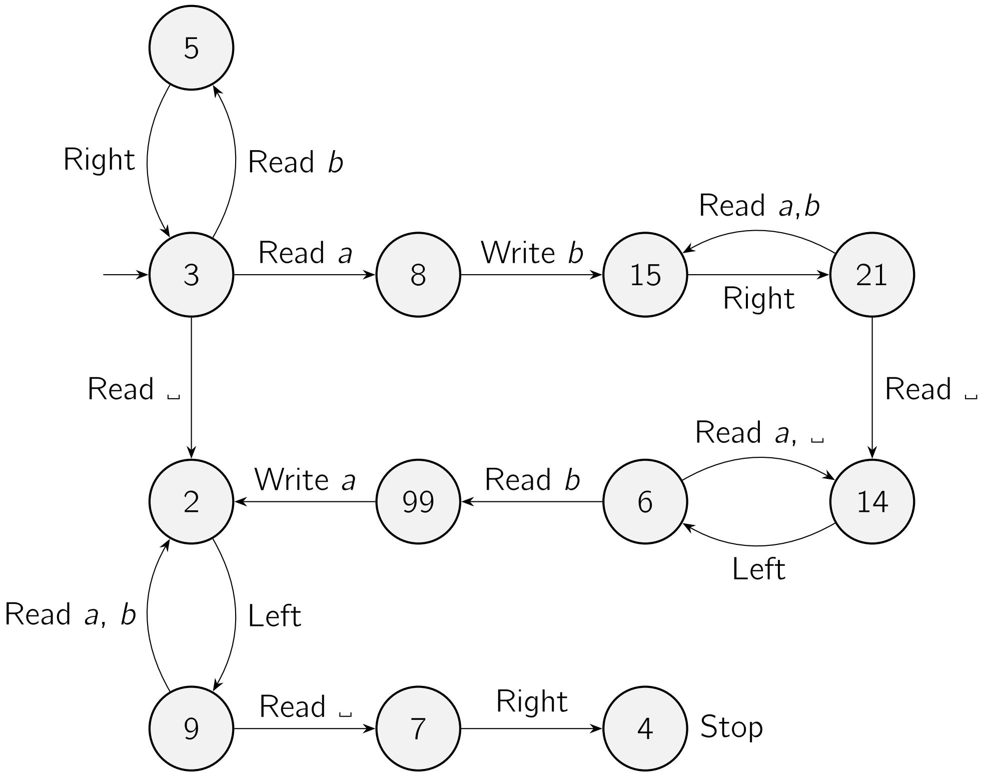
(i) First, the base case is that the expression 1 evaluates to a positive number; which is obviously true by $1 \geq 1$.

(ii) Next, the first inductive case is that $0 + S \geq 1$; by the inductive hypothesis $S \geq 1$ and therefore $0 + S = S \geq 1$.

(iii) Finally, the second inductive case is that $S + S' - 1 \geq 1$; by the inductive hypothesis $S \geq 1$ and $S' \geq 1$, meaning that $S + S' \geq 2$ and thus $S + S' + 1 \geq 1$.

Question 2 – Models and Time [33 marks]

(a) (i) Here is a Turing machine, for the tape alphabet $\{a, b, ␣\}$.



The machine starts on the leftmost cell of a nonempty block of a's and b's on an otherwise blank tape. In every case, it ends in the same place, on the leftmost cell of a nonempty block of a's and b's.

Trace the behaviour of the machine with initial tape contents $\dot{a}b$, where the dot indicates the head position. At each stage you should show the tape contents, head position, state, instruction and result in case of a read instruction. (The number of steps is no more than 15.) **[8 marks]**

(ii) In general, describe the machine's output in terms of the given input. That is, depending on the word on the tape initially, you should say what word is on the tape after execution stops. Take care to consider each possible case of the input. **[8 marks]**

(b) Rob wants to convert a fancy Turing machine with tape alphabet $\{a, b, c, \sqcup\}$ to a simple one with tape alphabet $\{a, b, \sqcup\}$. He decides to represent the fancy tape contents by a simple tape contents as follows:

Fancy	Simple	Fancy	Simple
a	a	$\dot{a}$	$\dot{a}$
b	bb	$\dot{b}$	$\dot{b}b$
c	ba	$\dot{c}$	$\dot{b}a$
$\sqcup$	$\sqcup$	$\dot{\sqcup}$	$\dot{\sqcup}$

For example, if the fancy tape contents is $a\dot{b}ac\sqcup b$, then the corresponding simple tape contents is $a\dot{b}baba\sqcup bb$.

Give a simple machine that simulates the Right instruction of the fancy machine. **[8 marks]**

(c) Emma makes a Turing machine with tape alphabet $\{a, b, \sqcup\}$. For an input of length n starting with a, the running time is $5000n^2$ if $n < 100$ and $3n + 700$ otherwise. For an input of length n starting with b, the running time is $n + 17$ if $n < 30$ and $4n + 1$ otherwise. Prove that the worst case running time is $O(n)$. **[9 marks]**

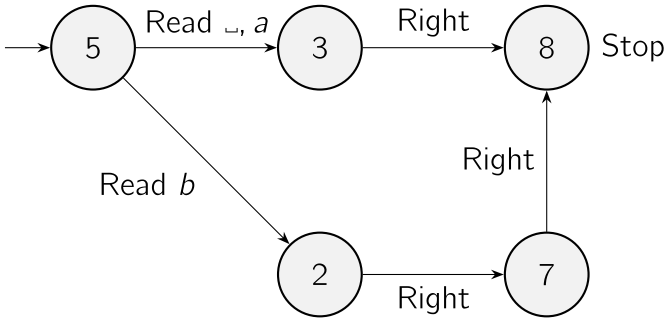
Model answers:

(a) (i)

Contents	State	Next instruction
$\sqcup\dot{a}b$	3	Read a
$\sqcup\dot{a}b$	8	Write b
$\sqcup\dot{b}b$	15	Right
$\sqcup b\dot{b}$	21	Read b
$\sqcup b\dot{b}$	15	Right
$\sqcup bb\dot{\sqcup}$	21	Read $\sqcup$
$\sqcup bb\dot{\sqcup}$	14	Left
$\sqcup b\dot{b}$	6	Read b
$\sqcup b\dot{b}$	99	Write a
$\sqcup b\dot{a}$	2	Left
$\sqcup\dot{b}a$	9	Read b
$\sqcup\dot{b}a$	2	Left
$\dot{\sqcup}ba$	9	Read $\sqcup$
$\dot{\sqcup}ba$	7	Right
$\sqcup\dot{b}a$	4	Stop

(ii) If the input word has the form $b^m a^n$, the output is the same as the input. Otherwise, the word has the form $b^m awba^n$, and the output is $b^{m+1} w a^{n+1}$.

(b)



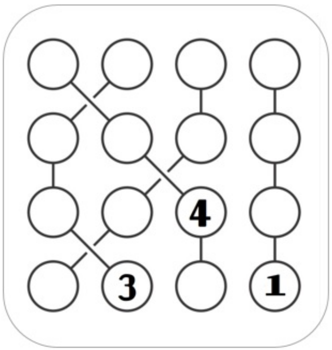
(c) Suppose $n \geqslant 100$. If the input starts with a, the running time is $\leqslant 3n+700 \leqslant 703n$. If the input starts with b, the running time is $\leqslant 4n+1 \leqslant 703n$. So in each case the running time is $\leqslant 703n$.

Question 3 – Hard and Impossible Problems [33 marks]

(a) (i) *Strimko* is a puzzle based on a $n \times n$ grid to be filled with numbers from 1 to n (inclusive). Cells in the grid are organized into n streams represented by continuous lines between them. Some numbers are given at the start and some must be added following three basic rules:

- Each row must contain each of the numbers 1 to n.
- Each column must contain each of the numbers 1 to n.
- Each stream must contain different numbers 1 to n.

Below left is an example of a *Strimko* grid of size 4 as given at the start and right is the same grid once filled in.



Explain why the problem of checking a candidate solution for a Strimko puzzle can be accomplished in polynomial time. **[8 marks]**

(ii) It follows that a SAT solver can be used to decide whether a Strimko puzzle is solvable. Explain why. **[8 marks]**

(b) State Rice's theorem. **[8 marks]**

(c) GameGalore is a company that produces video games. Jemima has been asked by the manager to ensure that each game displays the company's logo in every frame. Can she write a program to determine whether this is so? Explain your answer. **[9 marks]**

Model answers:

(a) (i) To check a candidate solution, we have to check whether each line in the puzzle (i.e., row, column, or stream) contains distinct numbers. Each line has size n. To check a line, one needs to check each number agains every other one in the line. That is, n^2 comparisons in the worst case. There are n rows, n columns and n streams so that is $3n^3$ comparisons in total. The time taken for each comparison is proportional to the length of the numbers, so it is $O(\log n)$ steps. Hence, checking a candidate solution for a Strimko grid is overall $O(n^4)$ and so polynomial.

(ii) SAT is NP-complete. This means that any problem in NP can be polynomially reduced to SAT. Since the Strimko checking solution can be achieved in polynomial time, Strimko solvability is in NP. We know that SAT is NP-complete, i.e. any problem in NP can be reduced to it in polynomial time, in particular Strimko can be.

(b) Any semantic property of code that holds in some case and fails in some case is undecidable.

Partial marks can be obtained for answers that show some understanding, or that omit the nontriviality requirements.

(c) This is a semantic property of game programs, since it depends only on the output. It holds for a game that never produces any frame, and fails for a game that displays a single blank frame. So, by Rice's theorem, it is undecidable. Therefore Jemima cannot write a program to decide it.