

May 2019

Coventry University

Faculty of Engineering, Environment and Computing

380CT

Theoretical Aspects of Computer Science

Instructions to candidates

Time allowed: **2** hours

This is **a Closed Book** Examination

Answer: All Questions

The total number of questions in this paper: 4

All questions carry equal marks (25 marks each)

Write your answers in this questions paper, in the provided framed boxes labelled Solution.

If more space is needed then use the extra pages provided at the end of this paper, and write a note in the appropriate box.

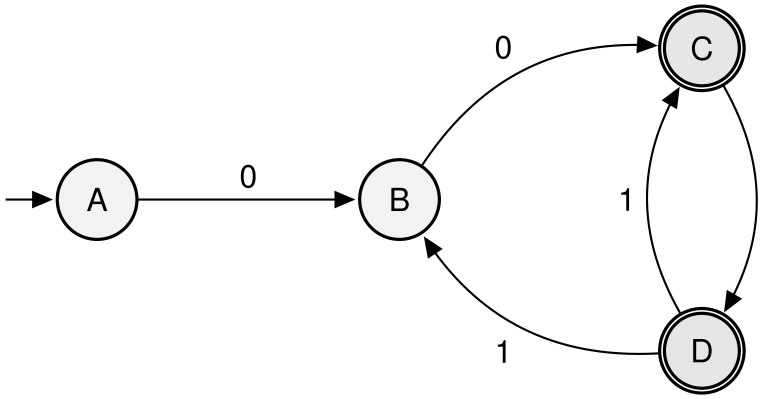
These pages have Extra page .../2 in their header.

These pages can also be used for rough work. (Please cross it out.)

You must hand this question paper in at the end of the examination**.**

---

❶ Consider the Non-deterministic Finite Automaton (NFA) $\mathcal{M}$ defined by the following transition diagram:



6 marks

a) Produce the formal specification of $\mathcal{M}$.

Solution

$\mathcal{M} = (Q, \Sigma, \delta, q_{\text{start}}, F)$ where:

- $Q =$
- $\Sigma =$
- $q_{\text{start}} =$
- $F =$
- Transitions table for δ: (Complete the transition table below)

	0	1

("→" denotes the start state, and "$*$" denotes accepting states.)

Question ❶ is continued on the next page

5 marks

b) For each of the following strings, state if it will be *accepted* or *rejected* by $\mathcal{M}$. (Tick a box).

Solution

String	Accept	Reject
ε		
11		
000		
111		
00110		

6 marks

c) Use the *subset construction method* to produce the transition table for a Deterministic Finite Automaton (DFA) equivalent to $\mathcal{M}$.

Solution

	0	1

(Use "$\rightarrow$" to denote the start state, and "$*$" to denote the accepting states.)

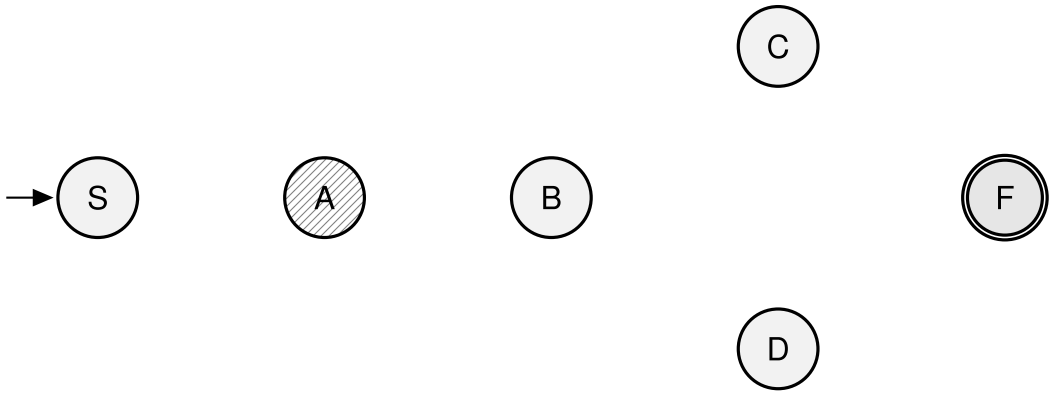
Question ❶ is continued on the next page

8 marks

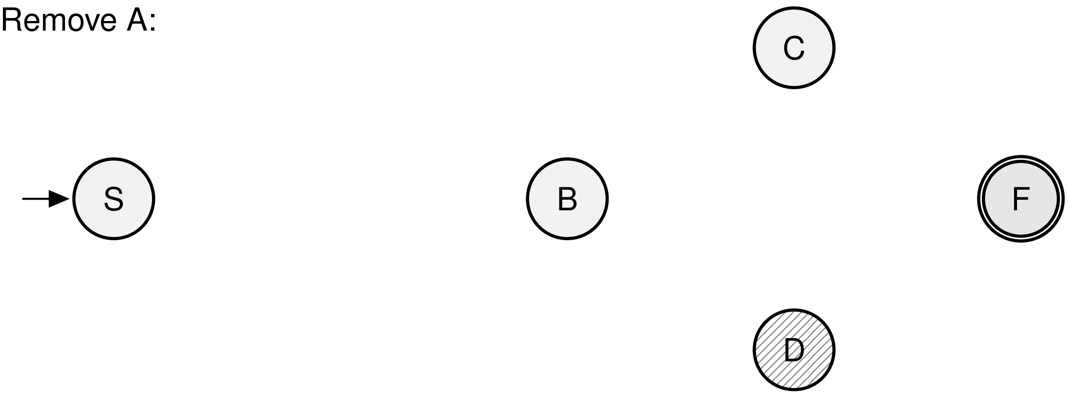
d) Use the *GNFA algorithm* to produce a regular expression for the language recognized by $\mathcal{M}$. (Add the missing transitions and labels.)

Solution

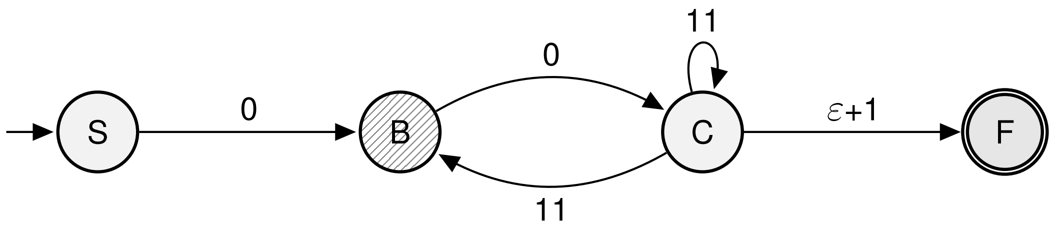
First convert the NFA into a Generalized NFA (GNFA)



Remove A:



Remove D: (This is completed fully – complete the next ones.)



Remove B:

Remove C:

Continue

❷ Consider the following context free grammar G with alphabet $\Sigma = \{\text{a}, \text{b}\}$ and production rules:

$$\begin{array}{rcl} S & \to & A \quad | \; B \; | \; AB \\ A & \to & AA \; | \; B \; | \; \text{a} \\ B & \to & BB \; | \; A \; | \; \text{b} \end{array}$$

3 marks

a) Use a regular expression to describe the pattern of strings generated by the rule $V \to V\,V \mid \text{a}$.

Solution

3 marks

b) Use a regular expression to describe the pattern generated by the last two rules:

$$\begin{array}{rcl} A & \to & AA \; | \; B \; | \; \text{a} \\ B & \to & BB \; | \; A \; | \; \text{b} \end{array}$$

Solution

3 marks

c) G actually generates every string in Σ^* except one string – what is it?

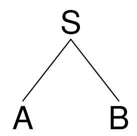
Solution

Question ❷ is continued on the next page

3 marks

d) Finish the below **parse tree** for the string bab:

Solution



3 marks

e) Finish the below **derivation** for the string bba:

Solution

$S \rightarrow AB \rightarrow$

Question ❷ is continued on the next page

4 marks

f) Design an NFA with 2 states whose language is exactly the one generated by G.

Solution

2 marks

g) Can a 1-state NFA do the same? Justify your answer.

Solution

h) The language generated by G can also be generated by a simpler grammar. Complete the set of rules given below to find it.

Write the missing *variables* or *terminals* in the given four boxes.

4 marks

Solution

$S \rightarrow$ a $\boxed{}$ | b $\boxed{}$ | $\boxed{}$ | $\boxed{}$

Continue

❸ a) Below is a list of claims about languages.

For each claim state if it is **correct** or **incorrect** giving a short **justification**.

The first one has been done as an example.

8 marks

Solution

$\{\text{a}^{\ell}\text{b}^{m}\text{c}^{n} \mid \ell, m, n \geq 0\}$ is not regular.

Incorrect. *It is generated by the regular expression:* a*b*c*.

i. The language of all the month names in a year is regular.

ii. The language generated by the grammar $S \to \text{a} \mid SS$ is regular.

iii. The language of all syntactically valid Python code is context free.

iv. It is possible to have a language L and its complement $\overline{L}$ both recognizable, yet L is not decidable.

Question ❸ is continued on the next page

b) Use O-notation to express the order of growth of the following expressions.

6 marks

Solution

Expression	O-notation
2019	
$2019 + n$	
$2019n + n^2$	
$n^{2019} + 2019^{2019}$	
$n + \log n + n \log n$	
$n^{2019} + n^{\log n}$	

c) You are given that

$$f(n) = O(\log n), \quad \text{and} \quad g(n) = O(2^n).$$

What is the order of the following functions:

3 marks

Solution

- $f(n) + g(n) =$
- $f(n) \times g(n) =$
- $f\Big(g(n)\Big) =$

Question ❸ is continued on the next page

d) Below is pseudocode for an algorithm that tries to decide if a given integer n is prime or not.

```
for d ← 1, ..., n do
    if d divides n then
        return false
    end if
end for
return true
```

4 marks

Why will this algorithm fail to decide the primality of n?

Solution

4 marks

e) How would you change the above pseudocode to correct this algorithm?

Solution

Continue

❹ The *Shortest Common Superstring (SCS)* problem is defined as follows:

Shortest Common Superstring (SCS)

Given a finite set of strings $\{s_1, \ldots, s_n\}$ over an alphabet Σ, find a string that is as short as possible and contains each string s_i as a substring.

SCS is **NP-hard** given that $|\Sigma| \geq 2$. It has application in DNA sequencing.

Examples:

- $\{cab, ba, abc\}$ admits $cababc$ as the shortest superstring.
- $\{ab, cd\}$ admits $abcd$ as the shortest superstring (no overlap possible).

2 marks

a) Classify the above problem formulation as a **decision**, **search** or **optimization** problem. Justify your answer.

Solution

2 marks

b) Find a solution to $\{abc, a, ca, b\}$

Solution

2 marks

c) Find a solution to $\{bc, bca, abc, cab\}$

Solution

Question ❹ is continued on the next page

3 marks

d) Complete the following **exhaustive search** algorithm to solve this problem.

Write in the provided three rectangles.

Solution

1: $best \leftarrow$ Concatenation of all the strings $s_1, \ldots, s_n$.
2: **for all** permutations $(t_1, \ldots, t_n)$ of $(s_1, \ldots, s_n)$ **do**
3: $\quad candidate \leftarrow t_1$
4: $\quad$ **for** $i \leftarrow 2, \ldots, n$ **do**
5: $\quad\quad$ Merge t_i into $candidate$ while maximizing their overlap.
6: $\quad$ **end for**
7: $\quad$ **if** $candidate$ is ______ than $best$ **then**
8: $\quad\quad best \leftarrow$ ______
9: $\quad$ **end if**
10: **end for**
11: **return** ______

3 marks

e) Estimate its worst-case cost using O-notation. Justify your answer.

Assume that the merging step costs $O(1)$ only.

(Refer to the line numbers in the pseudocode above.)

Solution

Question ❹ is continued on the next page

f) If $|\Sigma| = 1$ then SCS can be solved in polynomial time. Outline an algorithm that achieves this over $\Sigma = \{1\}$.

Hint: What is the solution to $\{11, 1111, 1\}$?

2 marks

Solution

g) Consider the following greedy solution method for this version:

Greedy

1: Select a random permutation $(t_1, \dots, t_n)$ of $(s_1, \dots, s_n)$
2: $solution \leftarrow t_1$
3: **for** $i \leftarrow 2, \dots, n$ **do**
4: Merge t_i into $solution$ to maximize their overlap.
5: **end for**
6: **return** $solution$

Simulate the greedy method on $S = \{01, 11, 1011, 0110\}$.

5 marks

Solution

Question ❹ is continued on the next page

6 marks

h) Specify a meta-heuristic which can be used to improve the greedy method.

Write **pseudocode** and **be specific** to SCS.

Solution

The End