UNIVERSITY OF PORTSMOUTH

SCHOOL OF COMPUTING

IN CLASS TEST 2017/18

LEVEL 6

U21276 – THEOCS – THEORETICAL COMPUTER SCIENCE

Duration:	1 Hour 30 Minutes
Instructions:	Section A is compulsory; choose any TWO of the THREE questions in Section B
Additional Information:	This is a CLOSED book examination
Permitted:	No materials permitted
Calculator:	Calculators ARE NOT permitted
Provided:	Nothing

Section A

This section is compulsory

Question 1:

(a) Consider the languages $L = \{ab, ac, ccc\}$ over the alphabet $\Sigma = \{a, b, c\}$.

(i) List all strings from $L^1 \cup L^2 \cup L^3$ with exactly three c-s.

[*4 marks*]

(ii) Find a language K over the alphabet Σ such that

$$L \cdot K = \{acab, acb, cccb, abab, abb, cccab\}.$$

Give a reason if such a language does not exist.

[*3 marks*]

(iii) Give an example of two strings of length four from $L^* \cap \Sigma^*$. If such a string does not exist give a reason.

[*3 marks*]

(b) Consider the following grammar over the alphabet $\{a, b, c\}$ with non-terminals S (the initial symbol), C and the production rules:

$S \to aSb \mid bSb \mid cC \mid \Lambda$

$C \to cC \mid \Lambda$

(i) List all strings of length four that can be generated by this grammar?

[*4 marks*]

(ii) Describe the structure of all strings generated by this grammar (using mathematical symbols or in your own words).

[*5 marks*]

(iii) Can the language generated by the grammar be recognised by a Turing machine. Justify your answer in at most three sentences.

[*3 marks*]

[Question 1 continued over the page]

(c) Give an example of an infinite regular language K over the alphabet $\Sigma = \{a, b, c\}$ and explain why the language K is regular.

[*4 marks*]

(d) Describe briefly the main differences between finite and pushdown automata. Give an example of a language which can be recognised by a pushdown, but not a finite automaton.

[*5 marks*]

(e) Formulate the pumping lemma for regular languages and explain its importance. What are the most important ideas of its proof?

[*6 marks*]

(f) Explain briefly the main differences between non-deterministic and deterministic computational models.

[*3 marks*]

[Total marks for the question: *40 marks*]

Section B

Choose any TWO questions from this section

Question 2:
Deterministic and non-deterministic finite automata (DFA/NFA)

(a) Consider the language L over the alphabet $\Sigma = \{a, b\}$ comprising of all strings of even length (at least 2) that begin and end with the same characters.

For example, the language contains the strings aa, $abaa$, $babb$ but not the strings Λ, ab, aba, $baba$.

(i) Find a regular expression that describes the language L.

[*6 marks*]

(ii) Draw a finite automaton (deterministic or non-deterministic) that recognises the language L (see also part (iv)).

[*6 marks*]

(iii) Draw a deterministic finite automaton that recognises the language L, if there exists one. (Just add a reference to part (ii), if your solution there is already a deterministic one.) If such a DFA does not exist explain why.

[*6 marks*]

(iv) Explain when two states of a DFA are not equivalent. Choose two non-final states in your solution (iii) and decide whether they are equivalent or not. Give a reason.

[*4 marks*]

(v) Write down a grammar that generates the language L.

[*5 marks*]

[Question 2 continued over the page]

(b) Is it true that for each regular language there exists an unique deterministic finite automaton with the minimum number of states (up to renaming the states)? Justify your answer.

[*3 marks*]

[Total marks for the question: *30 marks*]

Question 3: Pushdown Automata (PDA)

(a) Consider the language L over the alphabet $\Sigma = \{a, b, c\}$ comprising only those stings which have at least one c and exactly as many b-s as a-s. For example, the strings *cabbcbaa*, *c*, *caabbcc* are from the language L, but *aabb*, *ca*, *aabbccaa* are not.

Design a pushdown automaton that recognises the language L:

(i) First describe your planned algorithm in words.

[*4 marks*]

(ii) Following your plan in part (i) draw your PDA or describe the PDA by transitions.

[*11 marks*]

(iii) Is your pushdown automaton deterministic or non-deterministic? Give a reason.

[*3 marks*]

(iv) Can the language L be described by a context-free grammar? You are only asked to give the arguments why such a grammar exists or not.

[*2 marks*]

(b) Explain when a grammar is ambiguous. Are there non-ambiguous context-free grammars? Why concept of non-ambiguous grammar is important for programming languages?

[*6 marks*]

(c) Is it true that any context-free language can be recognised by a deterministic pushdown automaton? Explain.

[*4 marks*]

[Total marks for the question: *30 marks*]

Question 4: Turing machines (TM)

(a) Consider the alphabet $\Sigma = \{a, b, c\}$ and the language $L = \{a^n c^m b^n,\ n \geq 0, m \geq 0\}$ over Σ. Design a Turing machine that recognizes the language L. The initial position of the reading head is at the beginning of the string.

(i) First give a clear plan of how your Turing machine will perform this task.

[*4 marks*]

(ii) Either draw the Turing machine or write out its full set of instructions with comments that refer to your plan in part (i).

[*11 marks*]

(iii) Give the name of the smallest Chomsky class to which the language L belongs to. Justify your answer.

[*5 marks*]

(b) What is the difference between recursive and recursive enumerable languages. Give an example of a recursive language over $\Sigma = \{a, b\}$.

[*5 marks*]

(c) Is it true that all theoretical computational models are equivalent in terms of 'computational' power? Compare also TMs with today's computer in terms of 'computational' power.

[*5 marks*]

[Total marks for the question: *30 marks*]

END OF EXAM