Clayton School of Information Technology
Faculty of Information Technology
Monash University

Introduction to Theory of Computation
FINAL EXAM

2nd Semester, 2014

Instructions:

10 minutes reading time.

3 hours writing time.

No books, calculators or devices.

Total marks on the exam = 120.

Question 1 **(4 marks)**

Use De Morgan's Laws to prove that $\neg(P \wedge Q) \vee (\neg P \wedge \neg Q)$ is a tautology.

You may use other principles of propositional logic too, if you wish. But your proof must make significant use of De Morgan's Laws.

Official use only
4

Question 2 **(6 marks)**

Suppose you have predicates `algorithm` and `TuringMachine` with the following meanings, where variable X represents an arbitrary function:

`algorithm`(X): there is an algorithm for X.
`TuringMachine`(X): there is a Turing machine for computing X.

(a) Write a universal quantification logic with the meaning:

"Every function for which there is an algorithm can be computed by a Turing machine."

(b) By what name is this assertion usually known?

(c) What reasons are there for believing that this assertion is true?

Question 3 **(3 marks)**

Give a regular expression for the language of all strings over alphabet {`a`,`b`} whose first and last letters are different.

Official use only
9

Question 4 **(4 marks)**

Let L be the language of nonempty strings over {a,b} that can start and finish with any letter but all other letters must be a. Draw a Finite Automaton to recognise L.

Question 5 **(3 marks)**

When converting a Nondeterministic Finite Automaton (NFA) to an equivalent Finite Automaton (FA), how do you determine which states in the NFA are combined to form the single Start state in the FA?

Official use only
7

Question 6 **(4 marks)**

The following table describes a Finite Automaton with three states.

Find another FA that is equivalent to this one and has only two states.

Write your two-state table below.

state	a	b
1	2	3
2	1	3
3	1	2

YOUR ANSWER:

state	a	b

Official use only
4

Question 7 (4 marks)

Explain why the class of regular languages over the alphabet {a,b} is closed under interchange of a and b.

Definition:

If L is any lang bet {a,b}, then the *interchange of* a *and* b forms a new language as foll ng in L, and replace every a by b and every b by a, simultaneously. So, ing abb becomes baa.

Official use only
4

Question 8 **(7 marks)**

Consider the following Context-Free Grammar. Its non-terminals are S and L.

$$S \rightarrow L \quad (1)$$
$$L \rightarrow \texttt{ha}L \quad (2)$$
$$L \rightarrow \texttt{a}L \quad (3)$$
$$L \rightarrow \texttt{h} \quad (4)$$

Give

(a) a derivation, and

(b) a parse tree,

for the string `ahahaah` .

Label each step in the derivation on its right by the rule used. Use the spaces below for your answers.

(a)

(b)

Official use only
7

Question 9 **(3 marks)**

Give a regular grammar for the language of positive integers represented in binary with leading bit 1. (The *leading* bit is the most significant bit, i.e., the leftmost bit.)

Question 10 **(6 marks)**

Given the following Context-Free Grammar:

$$S \rightarrow \varepsilon \quad (1)$$

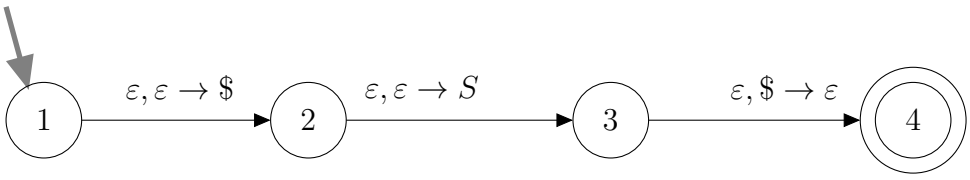
$$S \rightarrow T \quad (2)$$

$$T \rightarrow aTb \quad (3)$$

$$T \rightarrow aT \quad (4)$$

$$T \rightarrow b \quad (5)$$

complete the following diagram to give a Pushdown Automaton for the language generated by the grammar.



Official use only
9

Question 11 **(10 marks)**

This question uses the same grammar as the previous question.

Let L be the language generated by this grammar.

(a) Prove by induction that every string of the form $\mathtt{a}^m\mathtt{b}^n$, where $m \geq n-1$ and $n \geq 1$, belongs to L.

(b) Prove or disprove: some string in L has a derivation, using this grammar, that is neither a leftmost derivation nor a rightmost derivation.

Official use only
10

Question 12 **(7 marks)**

Recall that the **Fibonacci numbers**, F_n, are defined recursively as follows:

$$F_1 = 1,$$

$$F_n = F_{n-1} + F_{n-2} \quad \text{for } n \geq 3.$$

The first few numbers are

$$1, 1, 2, 3, 5, 8, 13, 21, 34, \ldots$$

Note that the Fibonacci numbers are (after the first term) an *increasing* sequence of positive integers.

The language FIBONACCI is defined to be the set of all strings of the letter `a` whose length is a Fibonacci number. So,

$$\text{FIBONACCI} = \{\, \mathtt{a}^{F_n} \;:\; n \in \mathbb{N} \,\}.$$

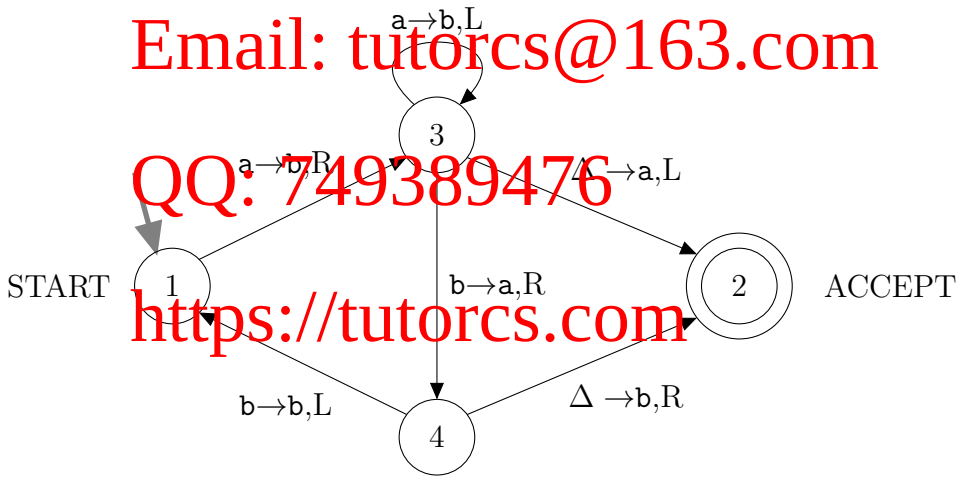
(a) Prove that the difference, $F_n - F_{n-1}$, between two consecutive Fibonacci numbers increases as n increases, i.e., $F_n - F_{n-1} > F_{n-1} - F_{n-2}$ for all $n \geq 5$.

(b) Using (a), prove that the language FIBONACCI is not context-free.

Official use only
7

Question 13 **(9 marks)**

Consider the following Turing machine.



Trace the execution of this Turing machine, writing your answer in the spaces provided on the next page.

The lines show the configuration of the Turing machine at the start of each step. For each line, fill in the state and the contents of the tape. On the tape, you should indicate the currently-scanned character by underlining it, and you should show the first blank character as Δ (but there is no need to show subsequent blank characters).

To get you started, the first line has been filled in already.

At start of step 1: State: 1 Tape: ␣ a a b ␣

At start of step 2: State: _____ Tape:

At start of step 3: State: _____ Tape:

At start of step 4: State: _____ Tape:

At start of step 5: State: _____ Tape:

At start of step 6: State: _____ Tape:

At start of step 7: State: _____ Tape:

At start of step 8: State: _____ Tape:

At start of step 9: State: _____ Tape:

At start of step 10: State: _____ Tape:

At start of step 11: State: _____ Tape:

At start of step 12: State: _____ Tape:

Official use only
9

Question 14 **(5 marks)**

For each of the following decision problems, indicate whether or not it is decidable.

You may assume that, when Turing machines are encoded as strings, this is done using the Code-Word Language (CWL).

Decision Problem	your answer (tick **one** box in each row)	
Input: a Turing ma… …sitive integer k. Question: Does M … string of at most k letters?	☐ Decidable	☐ Undecidable
Input: a Turing machine M, and a positive integer k. Question: Does the encoding of M have at least k letters?	☐ Decidable	☐ Undecidable
Input: a Turing machine M, and a positive integer k. Question: When M is given an encoding of itself as input, does the computation go for at least k steps?	☐ Decidable	☐ Undecidable
Input: a Turing machine M. Question: Does M eventually print an encoding of M on the tape and then halt?	☐ Decidable	☐ Undecidable
Input: a Turing machine M, and a string w. Question: Does M reject w?	☐ Decidable	☐ Undecidable

Official use only
5

Question 15 **(12 marks)**

The Venn diagram on the right shows several classes of languages. For each language (a)–(l) in the list below, indicate which classes it belongs to, and which it doesn't belong to, by placing its corresponding letter in the correct region of the diagram.

If a language do[illegible] of these classes, then place its letter above the top of the diagram.

You may assum[illegible] machines are encoded as strings, this is done using the Code-Word Lan[illegible]

(a) The set of all [illegible] adjacency matrices.

(b) The set of all 2-colourable graphs. (A graph is *2-colourable* if each vertex can be coloured Black or White in such a way that adjacent vertices receive different colours.)

(c) The set of all 3-colourable graphs. (Definition is like 2-colourable, except now the available colours are Red, White and Black.)

(d) The set of all satisfiable Boolean expressions in Conjunctive Normal Form with at least two literals in each clause.

(e) EQUAL, the set of all strings over alphabet {a,b} with an equal number of a's and b's.

(f) The set of all strings of parentheses such that the parentheses are correctly matched.

(g) The set of all strings in which every letter is next to an identical letter.

(h) The set of all Finite Automata that define the empty language.

(i) The set of all Turing Machines with polynomial time complexity.

(j) The set of all Turing machines which, when given themselves as input, eventually either reject or loop forever.

(k) The set of all Turing machines whose language of accepted strings is regular.

(l) The set of all Turing machines that have ever been written.

recursively enumerable (r.e.)

decidable

NP

P

Context-Free

Regular

Finite

Official use only
12

Question 16 **(7 marks)**

Prove that the following problem is undecidable.

Input: a Turing machine M and a positive integer t.
Question: Is there a string σ which, if given as input to M, causes it to eventually erase everything on the ta[illegible]very letter by Blank), after which it never writes any other symbol?

You may use th[illegible]ng problem is undecidable.

Question 17 **(3 marks)**

Explain how to obtain, from any language that is recursively enumerable but not decidable, another closely related language that is not recursively enumerable. (Proof is not required.)

Official use only
10

Question 18 **(15 marks)**

Consider the language 3-EDGE-COLOURABILITY, which consists of all graphs G such that we can assign colours to the edges of the graph so that (i) each edge is Red, White or Black, and (ii) any two incident at a common vertex must get different colours.

Let H be the fo

(a) Construct a Boolean expression E_H in Conjunctive Normal Form such that the satisfying truth assignments for E_H correspond to solutions to the 3-EDGE-COLOURABILITY problem on the above graph H (i.e., they correspond to colourings of the edges of H, using at most three colours and which give different colours to incident edges).

To do this, use variable names $a_R, a_W, a_B, b_R, b_W, b_B, c_R, c_W, c_B, d_R, d_W, d_B$, where variable e_X is True if and only if edge e gets colour X (where $e \in \{a, b, c, d\}$ and $X \in \{R, W, B\}$).

(b) Give a polynomial-time reduction from 3-EDGE-COLOURABILITY to SATISFIABILITY.

Official use only
15

Question 19 **(8 marks)**

Prove that the HAMILTONIAN CIRCUIT problem is NP-complete, by reduction from HAMILTONIAN PATH. You may assume that HAMILTONIAN PATH is NP-complete.

Definitions:

A **Hamiltonian** G is a path that includes every vertex of G. All the vertices on the path

A **Hamiltonian** G is a circuit that includes every vertex of G. All the vertices on the circuit must be distinct.

HAMILTONIAN PATH
Input: Graph G.
Question: Does G have a Hamiltonian path?

HAMILTONIAN CIRCUIT
Input: Graph G.
Question: Does G have a Hamiltonian circuit?

Official use only
8

END OF EXAMINATION