Faculty of Information Technology

Monash University

FIT2014 Theory of Computation
FINAL EXAM

2nd Semester 2016

Working Space

Question 1 **(4 marks)**

Suppose we have propositions B, C and T, with the following meanings.

B: Charles Babbage was the first computer scientist.
C: Alonzo Church was the first computer scientist.
T: Alan Turing was the first computer scientist.

Use B, C and T to write a proposition that is True if and only if the first computer scientist was *one, and only one,* of Charles Babbage, Alonzo Church and Alan Turing.

For full marks, your proposition should be in Conjunctive Normal Form.

Question 2 **(4 marks)**

Suppose you have predicates `automatic`, `generalPurpose`, `programmable` and `storedProgram` with the following meanings, where variable X represents an arbitrary device:

`automatic`(X):	the device X does not require human intervention, once its instructions are entered.
`computer`(X):	the device X is a computer.
`generalPurpose`(X):	the device X can compute any computable function.
`programmable`(X):	the device X can be programmed.
`storedProgram`(X):	the device X stores its instructions in its memory, in order to execute them.

(a) In the space below, write a statement in predicate logic with the meaning:

> A device is a computer if and only if it is automatic, programmable, stored program, and can compute any computable function.

To do this, you may only use: the above five predicates; quantifiers; logical connectives. (In particular, you may not use set theory symbols such as $\subseteq$, $\subset$, $\cap$, $\cup$, etc, and in fact they would not help.)

(b) What type of Turing machine best models the idea of a computer, according to the above definition?

Official use only
8

Question 3 **(6 marks)**

A traffic light displays a sequence of colours Red (`R`), Green (`G`), and Amber (`A`), in that order, then going back to Red and repeating the cycle again, and so on, for some period of time. The first colour displayed is always Red, but it can finish at any time, on any colour.

Consider the language of all finite nonempty strings over the alphabet {`R`,`G`,`A`} that represent a valid sequence of traffic light colours. No two consecutive letters can be the same. For example, the language includes the strings `R`, `RG`, `RGA`, `RGAR`, `RGARG`, `RGARGA`, ...

(a) Give a regular expression for this language.

(b) Give a Finite Automaton for this language.

Official use only
6

Question 4 **(5 marks)**

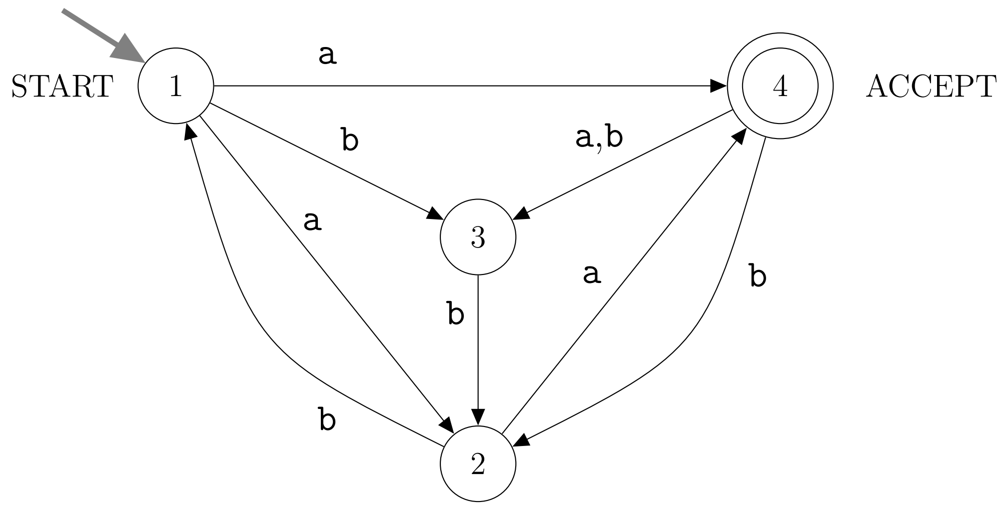
Draw a Nondeterministic Finite Automaton (NFA) that recognises the language of strings that match the following regular expression:

$$\mathtt{b}(\mathtt{ab} \cup \mathtt{ba})^*\mathtt{a}$$

Official use only
5

Question 5 **(7 marks)**

Consider the following Nondeterministic Finite Automaton (NFA).



Let L be the language of strings accepted by this NFA.

(a) What are the possible states that this NFA could be in, after reading the input string `abba`?

(b) Prove, by induction on n, that for all positive integers n, the string $(\texttt{abba})^n$ is accepted by this NFA. (This string is obtained by n repetitions of `abba`.)

Official use only
7

Question 6 **(4 marks)**

Consider the five-state Finite Automaton represented by the following table.

	state	`a`	`b`
Start	1	5	5
	2	3	5
	3	4	5
	4	2	5
Final	5	2	3

Convert this into an equivalent FA with the minimum possible number of states.

Write your answer in the following table. You may not need all the rows available.

state	`a`	`b`

Official use only
4

Working Space

Question 7 **(9 marks)**

The language DOG consists of all strings of the form

$$\mathtt{gr}^n(\mathtt{woof})^n$$

where n is any positive integer. For example, the strings `grwoof` and `grrwoofwoof` both belong to DOG, but `grrwoof` does not.

(a) Use the Pumping Lemma for Regular Languages to prove that DOG is not regular.

(b) Give a Context-Free Grammar for DOG.

Official use only
9

Question 8 **(13 marks)**

Consider the following Context-Free Grammar:

$$
\begin{aligned}
S &\rightarrow SS && (1)\\
S &\rightarrow \mathtt{h}S\mathtt{m} && (2)\\
S &\rightarrow \varepsilon && (3)
\end{aligned}
$$

(a) Give a derivation for the string `hhhmmhmm`.

Each step in your derivation must be labelled, on its right, by the number of the rule used.

(b) Give a parse tree for the same string, `hhhmmhmm`.

(c) Prove by induction on n, that for all $n \geq 0$, the string $\mathtt{h}^n\mathtt{m}^n$ has a derivation in this grammar of $n+1$ steps.

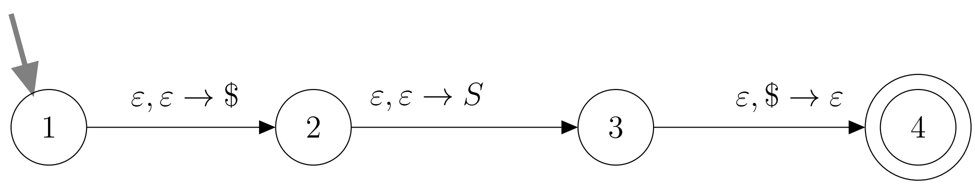
Official use only
13

Question 9 **(6 marks)**

This question uses the same Context-Free Grammar as the previous question. Here it is again for convenience:

$$S \rightarrow SS \quad (1)$$
$$S \rightarrow \mathtt{h}S\mathtt{m} \quad (2)$$
$$S \rightarrow \varepsilon \quad (3)$$

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



Official use only
6

Question 10 **(6 marks)**

This question is about doing the *last part* of the Cocke-Younger-Kasami (CYK) algorithm, to determine whether the string `abab` can be generated by the following grammar.

$$\begin{aligned} S &\rightarrow AA \mid AB \\ A &\rightarrow BB \mid \mathtt{a} \\ B &\rightarrow AB \mid BA \mid \mathtt{b} \end{aligned}$$

Suppose you have dealt with all substrings of length ≤ 3, and that, for each of these strings, you have worked out all the nonterminals that can generate it. This information is summarised in the following table.

string	Nonterminals that can generate the string
`a`	A
`b`	B
`ab`	S, B
`ba`	B
`aba`	S, B
`bab`	A

(a) Determine all *pairs* of nonterminals that can generate the string `abab`.

(b) Determine all *single* nonterminals that can generate this same string, `abab`.

(c) State whether or not `abab` can be generated by the above grammar, by circling the appropriate word YES or NO below:

YES, `abab` does belong to this language.

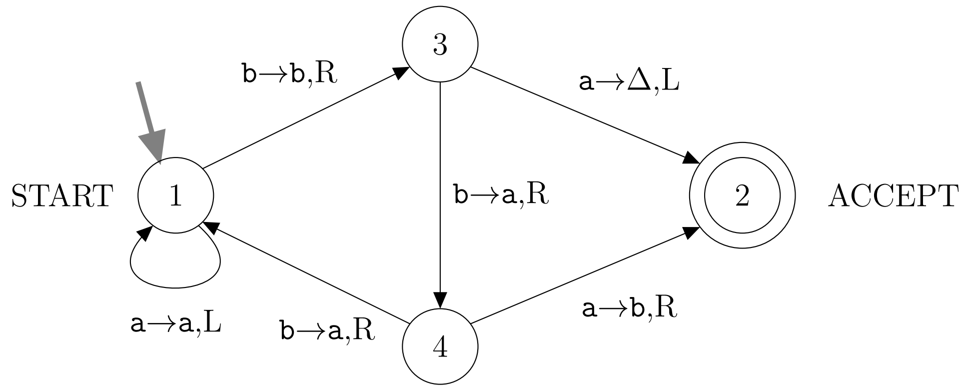
NO, `abab` does **not** belong to this language.

Official use only

6

Question 11 **(8 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).

You should not need all the lines provided.

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

At start of step 1:	State: 1	Tape:	b	b	b	a	$\triangle$	
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
8

Working Space

Question 12 **(3 marks)**

State the Church-Turing thesis, and give two reasons why it is believed to be true.

Question 13 **(4 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 machine M with at most ten states. Question: Does M eventually halt, when given itself as input?	☐ Decidable	☐ Undecidable
Input: a Turing machine M with at most ten transitions. Question: Does M eventually halt, when given itself as input?	☐ Decidable	☐ Undecidable
Input: a Turing machine M with a tape alphabet of at most ten letters. Question: Does M eventually halt, when given itself as input?	☐ Decidable	☐ Undecidable
Input: a Turing machine M that can only move to the Right. Question: Does M eventually halt, when given itself as input?	☐ Decidable	☐ Undecidable

Official use only
7

Question 14 **(10 marks)**

The Venn diagram on the right shows several classes of languages. For each language (a)–(j) 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 does not belong to any of these classes, then place its letter above the top of the diagram.

You may assume that, when Turing machines are encoded as strings, this is done using the Code-Word Language (CWL), with input alphabet {`a`,`b`} and tape alphabet {`a`,`b`,`#`,Δ}.

(a) The empty language.

(b) The set of all strings in which `a` and `b` appear the same number of times.

(c) The set of all adjacency matrices of graphs.
(Such a matrix is represented as a string of n^2 bits, where n is the number of vertices.)

(d) The set of adjacency matrices of 3-colourable graphs.

(e) The set of all Boolean expressions that are not satisfiable.

(f) The set of all encodings of Turing machines.

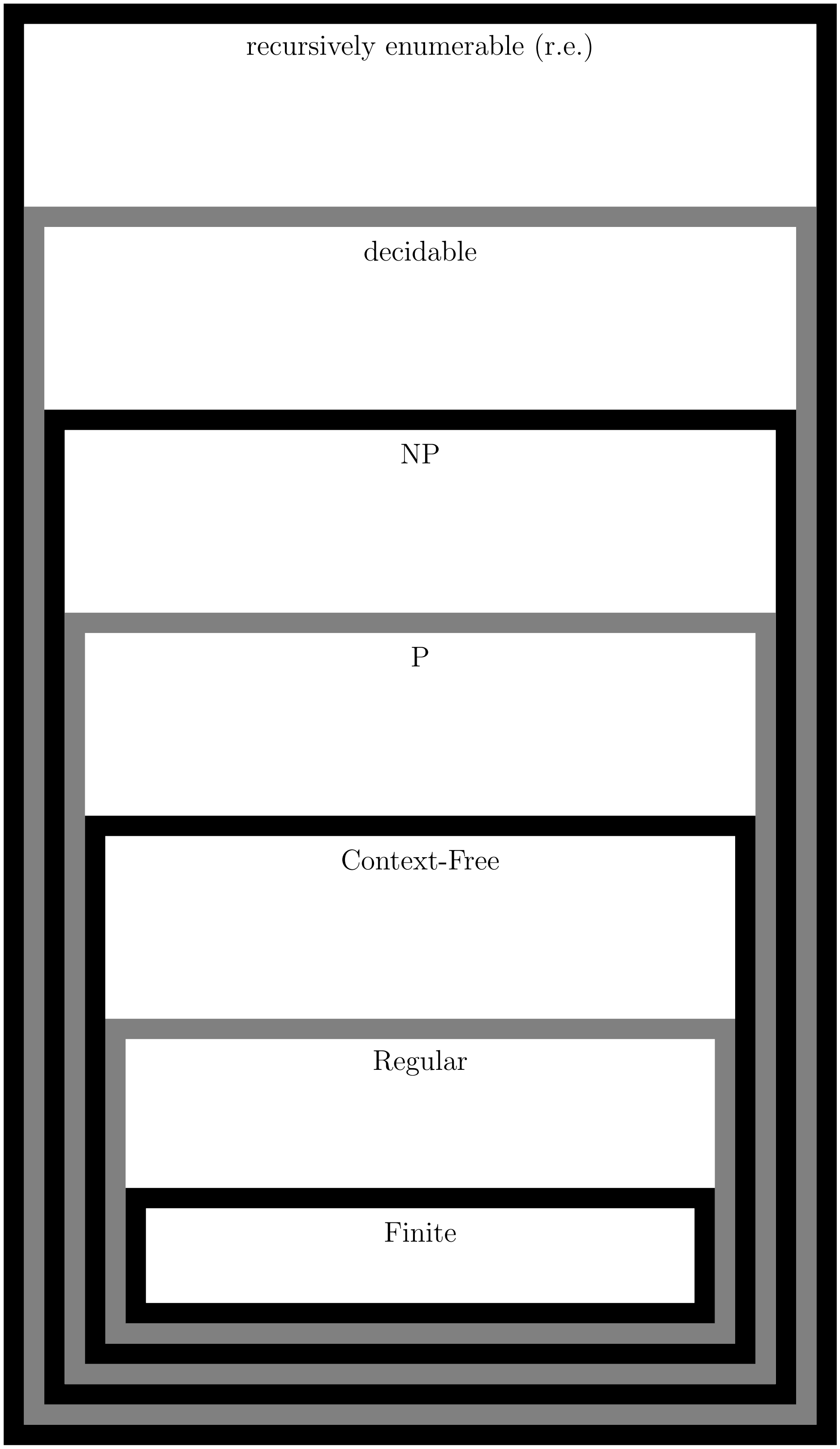
(g) The set of all encodings of Turing machines that loop forever for all inputs.

(h) The set of strings of the form $\mathtt{x}^n\mathtt{y}^{2n}\mathtt{z}^{3n}$, where n is any positive integer.

(i) The set of legal positions in One-Dimensional Go.

(j) The set of illegal positions in One-Dimensional Go.

Reminder: a legal position in 1D-Go is a string over the alphabet {`b`,`w`,`u`} such that every letter `b` is part of a string of consecutive `b`s with `u` at one end (or both ends), and every letter `w` is part of a string of consecutive `w`s with `u` at one end (or both ends). An illegal position in 1D-Go is a string over the same alphabet that is not a legal position.



Official use only

10

Question 15 **(6 marks)**

Prove by contradiction that, if L_1 is undecidable, $L_2 \subseteq L_1$, and L_2 is decidable, then $L_1 \setminus L_2$ is undecidable.

(Here, $L_1 \setminus L_2$ is the set of strings that belong to L_1 but not L_2.)

Official use only
6

Question 16 **(5 marks)**

Suppose you have an enumerator M for a language L.

Give an algorithm that accepts precisely the strings in L.

Your algorithm may use M.
For strings not in L, the algorithm must either reject or loop forever.

Official use only
5

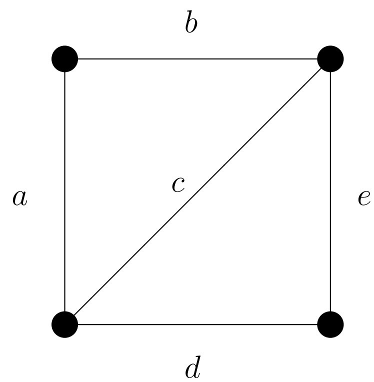
Question 17 **(12 marks)**

An **edge cover** in a graph G is a set X of edges that meets every vertex of G. So, every vertex is incident with at least one edge in X.

The EDGE COVER decision problem is as follows.

EDGE COVER
Input: Graph G.
Question: Does G have an edge cover?

For example, in the following graph, the edge set $\{a, b, c, d\}$ is an edge cover, and so is $\{a, e\}$. But $\{a, b, c\}$ is not an edge cover, since it does not meet every vertex. (Specifically, it misses the bottom right vertex.)



Let W be the above graph.

(a) Construct a Boolean expression E_W in Conjunctive Normal Form such that the satisfying truth assignments for E_W correspond to edge covers in the above graph W.

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

Official use only
12

Question 18 **(8 marks)**

Prove that the problem BIG INDEPENDENT SET problem is NP-complete, by reduction from HUGE INDEPENDENT SET. You may assume that HUGE INDEPENDENT SET is NP-complete.

Definitions:

For any positive integer k, an **independent set** in a graph G is a subset X of the vertex set of G such that no two vertices in X are adjacent.

HUGE INDEPENDENT SET
Input: Graph G, with an even number of vertices.
Question: Does G have an independent set of size $\geq n/\mathbf{2}$?

BIG INDEPENDENT SET
Input: Graph G.
Question: Does G have an independent set of size $\geq n/\mathbf{3}$?

In each of these definitions, n is the number of vertices in the graph G.

Official use only
8

Working Space

END OF EXAMINATION