Exam 1 November 2015, questions

Theory Of Computation (Monash University)

Faculty of Information Technology

Monash University

FIT2014 Theory of Computation
FINAL EXAM

2nd Semester 2015

Working Space

Question 1 **(3 marks)**

You are choosing a main course at a restaurant, from a menu containing three items: bibimbap, haggis and manakish.

Suppose we have propositions B, H and M, with the following meanings.

B: You choose bibimbap.
H: You choose haggis.
M: You choose manakish.

Use B, H and M to write a proposition, in Conjunctive Normal Form, that is True precisely when you choose *either* the bibimbap *or* both the other items.

Question 2 **(5 marks)**

Suppose you have predicates `belongsToP` and `belongsToNP` with the following meanings, where variable X represents an arbitrary language:

`belongsToP`(X): the language X belongs to the class P.
`belongsToNP`(X): the language X belongs to the class NP.

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

P is a proper subset of NP (i.e., P $\subseteq$ NP and P $\neq$ NP).

To do this, you may only use: the above two predicates; quantifiers; logical connectives; equality. (In particular, you may not use $\subseteq$ or $\subset$ or $\neq$, etc, and in fact they would not help.)

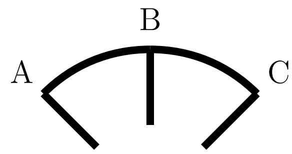
(b) For the statement "P is a proper subset of NP", which one of the following holds? Circle (A), (B), (C) or (D) to indicate your answer.

(A) The statement is True.
(B) The statement is False.
(C) It is not known whether it's True or False, but it's generally believed to be True.
(D) It is not known whether it's True or False, but it's generally believed to be False.

Official use only
8

Question 3 **(7 marks)**

A knob is used to select one of three positions A, B, C, as shown in the following diagram.



The position of the knob is recorded every second for some nonzero period of time.

The knob can never be turned from A to C, or from C to A, without spending at least one second in position B. The initial position of the knob can be any of A, B, C.

Consider the language of all possible strings of knob positions recorded in this way.

(a) Give a regular expression for this language.

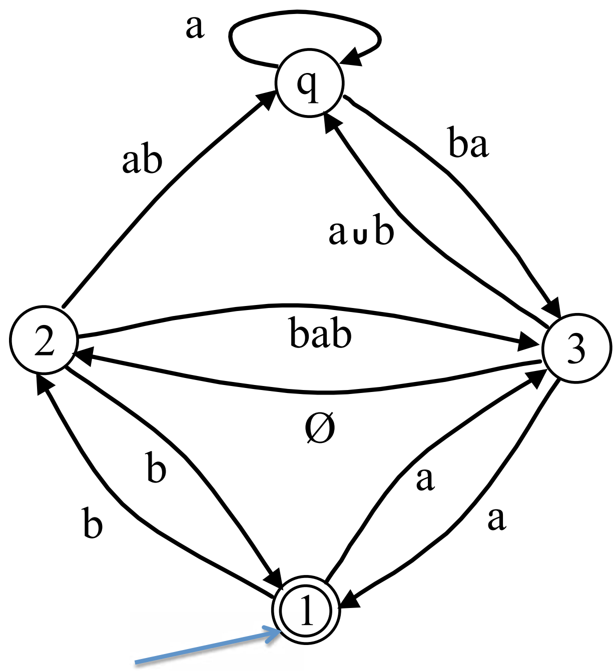
(b) Give a Finite Automaton for this language.

Official use only
7

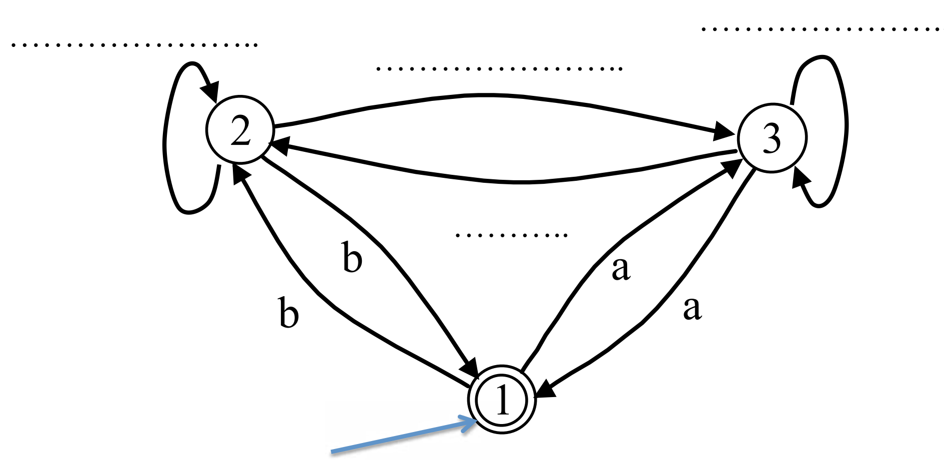
Question 4 **(5 marks)**

Consider the following Generalised Nondeterministic Finite Automaton (GNFA). Construct an equivalent GNFA with the top state, q, removed.

Show your answer by writing in the four missing transition labels, on the dotted lines, in the second diagram below.



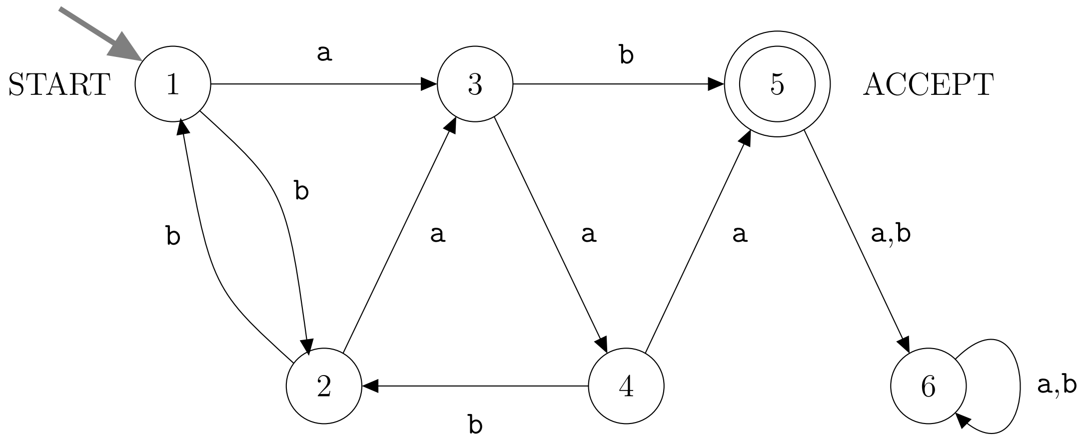
YOUR ANSWER:



Official use only
5

Question 5 **(6 marks)**

Consider the following Finite Automaton.



(a) Which, if any, of the following strings are accepted by this FA? Circle the ones that are accepted.

`ab` `aba` `aabb` `aabab`

(b) Consider the string `aabab`. Show how to split it up into three parts xyz such that, for all $i \geq 0$, the string xy^iz is also accepted by the FA.

Do this by drawing vertical lines between the parts of the string given on the next line:

`a a b a b`

Official use only
6

Question 6 **(4 marks)**

The following table describes part of a Finite Automaton with three states. The second row, for state 2, has not been filled in.

What could the state-2 row be, if you are given that the number of states of this FA is **not minimum**?

Write all possible state-2 rows in the second table below. Use as many rows as you need, one for each possible state-2 row. You may not need all the rows available.

	state	a	b
Start	1	2	3
	2		
Final	3	1	2

YOUR ANSWER:

state	a	b
2		
2		
2		
2		

Official use only
4

Question 7 **(6 marks)**

This question uses the following **Definitions** and **Facts**.

Definitions:

- If L is any language, the *reversal* of L is the set of all reversals of strings in L. In other words, you obtain the reversal of L by taking every string in L and writing it backwards. So, for example, the string `abb` becomes `bba`.
- If L is any language over the alphabet $\{\mathtt{a},\mathtt{b}\}$, then the *interchange of* `a` *and* `b` forms a new language as follows: take every string in L, and replace every `a` by `b` and every `b` by `a`, simultaneously. So, for example, the string `abb` becomes `baa`.

Facts:

- The class of regular languages is closed under reversal.
- The class of regular languages is closed under interchange of `a` and `b`.
- The language $\mathrm{AB} := \{ \mathtt{a}^n\mathtt{b}^n \ : \ n \in \mathbb{N} \}$ is not regular.

Your task:

Using these facts, and any other closure properties of regular languages you like, *prove by contradiction* that the language

$$\{ \mathtt{a}^m\mathtt{b}^n \ : \ m \leq n \}$$

is not regular.

Question 8 **(3 marks)**

Explain how to derive, for any Finite Automaton, a regular grammar for the language recognised by the FA.

Official use only

9

Question 9 **(12 marks)**

Consider the following Context-Free Grammar:

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

(a) Give a derivation for the string `ssshhh`.
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, `ssshhh`.

(c) Prove by induction on n, that for all $n \geq 0$, the string $\mathtt{s}^n\mathtt{h}^n$ has a derivation in this grammar of $n+2$ steps.

Official use only
12

Question 10 **(6 marks)**

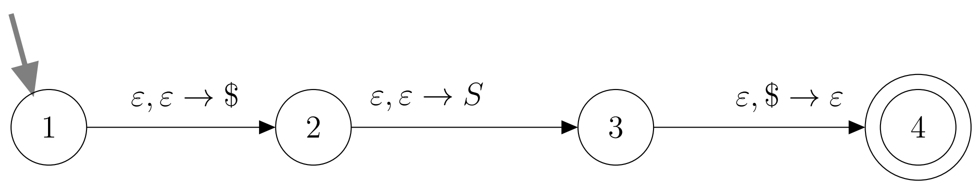
This question uses the same Context-Free Grammar as the previous question. Here it is again for convenience:

$$S \rightarrow X \tag{1}$$

$$X \rightarrow \texttt{s}X\texttt{h} \tag{2}$$

$$X \rightarrow \varepsilon \tag{3}$$

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



Official use only
6

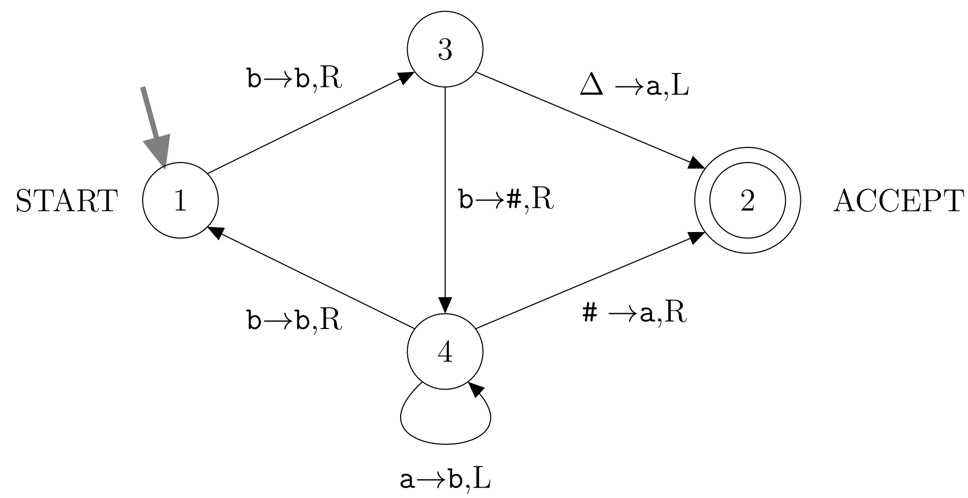
Question 11 **(5 marks)**

State (without proof) the Pumping Lemma for Context-Free Languages, and briefly describe its main purpose.

Official use only
5

Question 12 **(6 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:	<u>b</u>	b	a	Δ		
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
6

Working Space

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. Question: Does there exist a string w that is accepted by M in at most 7 steps?	☐ Decidable	☐ Undecidable
Input: a Turing machine M. Question: Does there exist a string w that is accepted by M?	☐ Decidable	☐ Undecidable
Input: a string w. Question: Does there exist a Turing machine that accepts w?	☐ Decidable	☐ Undecidable
Input: a Turing machine M, and a string w. Question: Is w the encoding, in CWL, of M?	☐ Decidable	☐ Undecidable

Official use only
4

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 set of all palindromes of even length.

(b) The set of all positive integers, in binary, whose number of bits is odd.

(c) The set of all strings of correctly matched parentheses.

(d) The set of all encodings of Turing machines that have at least three states.

(e) The set of all encodings of Turing machines that have at most three states.

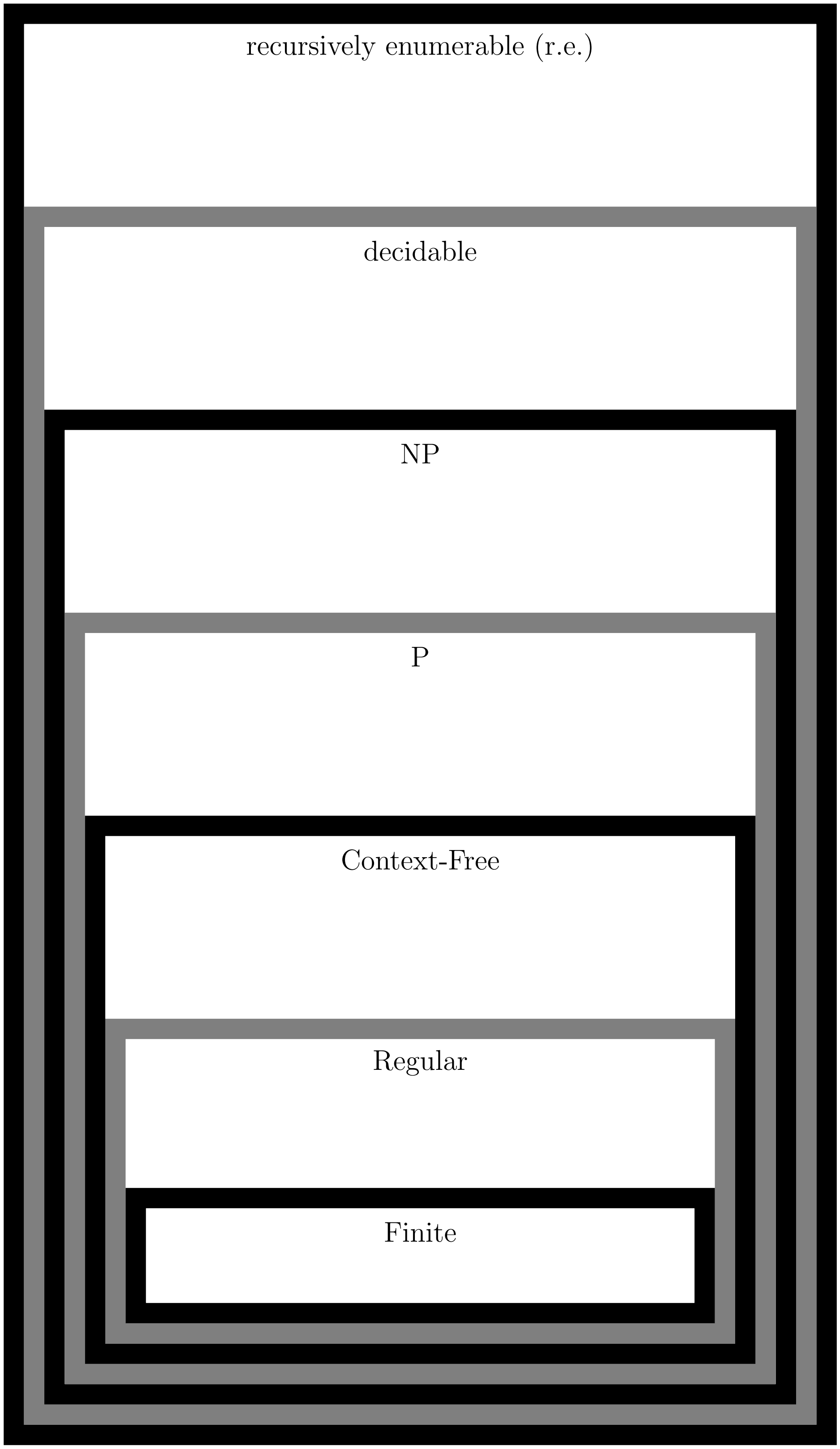
(f) The set of all encodings of Turing machines that halt for some input.

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

(h) The set of all satisfiable Boolean expressions.

(i) The set of all satisfiable Boolean expressions in Conjunctive Normal Form in which every variable appears at most once (so each variable has only one literal).

(j) The set of all *nondecreasing strings* of digits, i.e., strings over the digits 0,1,. . .,9 such that no digit is ever followed by a lower digit. (So 03348 is ok, but 03328 is not ok.)



Official use only

10

Question 15 **(6 marks)**

If L is a language and s is a string, then $L\Delta s$ denotes the language formed by *removing* s from L if it is there already, and *adding* s to L otherwise. In other words,

$$L\Delta s := \begin{cases} L \setminus \{s\}, & \text{if } s \in L, \\ L \cup \{s\}, & \text{if } s \notin L. \end{cases}$$

Prove that L is decidable if and only if $L\Delta s$ is decidable.

Official use only
6

Question 16 **(5 marks)**

Below is a proof that the Halting Problem is undecidable. But some parts are missing; these are shown as blank spaces, underlined.

Your task is to fill in the underlined spaces to complete the proof. In some cases, options are given in square brackets.

Proof. Assume that the Halting Problem is actually decidable.
Then the Halting Problem has a decider D.
Using D, we can construct a Turing machine E that does the following:

1. Input: encoding of a Turing machine M.
2. Run decider D to determine whether or not M halts when given itself as input.
3. **IF** the answer from D is **Yes**, then

   ____

4. **IF** the answer from D is **No**, then

   ____

Now consider what happens if E is given, as input, an encoding of *itself*.

If E halts on input E, then running D in line 2 gives the answer ______ [Yes/No].

So, using statement ______ [3 or 4], we see that D actually ______________________________.

On the other hand,

if E does not halt on input E, then running D in line 2 gives the answer ______ [Yes/No].

So, using statement ______ [3 or 4], we see that D actually ______________________________.

This is a contradiction.

So the Halting Problem must be undecidable.

Official use only
5

Question 17 **(6 marks)**

Suppose you have an enumerator M_1 for a language L and another enumerator M_2 for its complement $\overline{L}$.

(a) Explain how to construct a decider for L that uses the enumerators M_1 and M_2.

(b) What does this tell you about recursively enumerable (r.e.) languages whose complements are also recursively enumerable? (Only one short sentence is required here.)

Official use only
6

Working Space

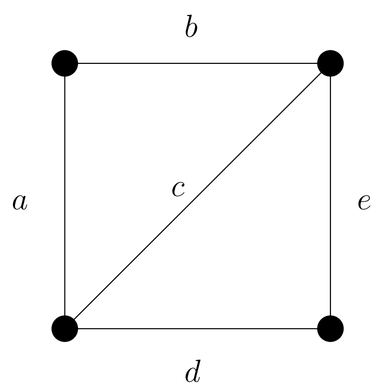
Question 18 **(13 marks)**

A **perfect matching** in a graph G is a subset X of the edge set of G that meets each vertex exactly once. In other words, no two edges in X share a vertex, and each vertex of G is incident with exactly one edge in X.

The PERFECT MATCHING decision problem is as follows.

PERFECT MATCHING
Input: Graph G.
Question: Does G have a perfect matching?

For example, in the following graph, the edge set $\{a, e\}$ is a perfect matching. But $\{a, b, e\}$ is not a perfect matching (since, for example, a and b share a vertex), and $\{a\}$ is not a perfect matching (since some vertices are not incident with the edge in this set).



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 perfect matchings in the above graph W.

(b) Give a polynomial-time reduction from PERFECT MATCHING to SATISFIABILITY.

Official use only
13

Question 19 **(8 marks)**

Prove that the GRAPH 4-COLOURABILITY problem is NP-complete, by reduction from GRAPH 3-COLOURABILITY. You may assume that GRAPH 3-COLOURABILITY is NP-complete.

Definitions:

For any positive integer k, a k-**colouring** in a graph G is an assignment of "colours" from the set $\{1, 2, \ldots, k\}$ to the vertices of G such that (a) each vertex gets exactly one colour from the set, and (b) adjacent vertices get different colours.

GRAPH 3-COLOURABILITY
Input: Graph G.
Question: Does G have a 3-colouring?

GRAPH 4-COLOURABILITY
Input: Graph G.
Question: Does G have a 4-colouring?

Official use only
8

END OF EXAMINATION