Enrolment number (student number):

The University of Melbourne
Practice Exam Paper

School of Computing and Information Systems

COMP30026 Models of Computation

Reading Time: 15 minutes
Exam Duration: 3 hours

This paper has 14 pages, including this front page.

Authorised Materials:
This is a closed book exam. Electronic devices, including calculators and laptop computers are **not** permitted.

Calculators:
No calculators are permitted.

Instructions to Invigilators:
Students will provide answers in the exam paper itself. The exam paper must remain in the exam venue and must be returned to the examiner.

Instructions to Students:
This is not an actual exam paper. It is a practice paper which has been put together to show you the format that you can expect in the exam. Many aspects of this paper's contents do not necessarily reflect the contents of the actual exam paper: The selection of topics, the number of questions or sub-questions, the perceived difficulty of individual questions, and the distribution of weights are all aspects that may be different. Hence, when preparing for the exam, you should cover the entire syllabus and not focus only on topics or question types used in this practice paper.
There are 9 questions. As in the exam, you should attempt them all. Of course your answers must be *readable*. Any unreadable parts will be considered wrong. You will find some questions easier than others; in the actual exam you should allocate your time accordingly. Marks are indicated for each question, adding to a total of 70.
The actual exam paper will be printed single-sided, so you will have plenty of space for rough work on the flip sides. Only what you write inside the allocated boxes will be marked. Page 14 is overflow space, in case you need more writing space for some question.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	
Examiners' use:										

Question 1 **(6 marks)**

The MacGuffin movie theatre has six showtimes per week. They prefer to show as many different films as possible. For the coming week they must choose amongst four films to show, namely p, q, r, and s. The distributors, however, pose many restrictions. The following conditions must be satisfied:

- Either both of r and s must be shown, or neither can be shown.
- If neither r nor s is shown then p cannot be shown either.
- If q is shown then one, but not both, of r and s must be shown.
- If r and s are both shown then q must be shown.

Tick the correct statement:

- ☐ MacGuffin can show several different films that week
- ☐ MacGuffin must show the same film all week, but has choice of which film to show
- ☐ MacGuffin must show the same film all week, with no choice of which film to show
- ☐ MacGuffin cannot show films that week
- ☐ The conditions that have been posed are unsatisfiable

Question 2 **(8 marks)**

Consider the closed first-order predicate logic formulas F, G, and H:

$$\begin{array}{lcl} F & : & \forall x\ P(x,x) \\ G & : & \forall x\ \forall y\ (P(x,y) \Rightarrow P(y,x)) \\ H & : & \forall x\ (P(x,x) \vee \exists y\ (\neg P(y,x))) \end{array}$$

A. Show that $F \wedge G$ is satisfiable but not valid.

B. Determine whether $F \vee G$ is valid. Justify your answer.

C. Recall that $\varphi \models \psi$ says that ψ is a logical consequence of φ. Tick the most appropriate statement from the list on the right:

- $G \models H$ ☐
- $H \models G$ ☐
- $G \equiv H$ ☐
- None of the above ☐

Question 3 **(10 marks)**

Consider the following predicates:

- $C(x)$, which stands for "x is a cat";
- $D(x)$, which stands for "x is a dog";
- $M(x)$, which stands for "x is a mouse";
- $P(x)$, which stands for "x is a pasta dish";
- $E(x, y)$, which stands for "x eats y";
- $L(x, y)$, which stands for "x likes y";
- $F(x, y)$, which stands for "x is a friend of y";

A. Express, *as a formula in first-order predicate logic* (not clausal form), the statement "No mouse likes a cat who likes mice". (Here "likes mice" means "likes every mouse".)

B. Turn the following closed formula into clausal form:

$$\forall x\ \forall y\ \Big[\Big(M(x) \wedge \forall z\ (D(z) \Rightarrow L(x, z))\Big) \Rightarrow \Big(M(y) \Rightarrow \neg L(y, x)\Big)\Big]$$

C. Using c for "Garfield" and b for "Harold", we can express various statements about cats, mice and men in clausal form, as follows:

Garfield is a cat who likes pasta dishes:	$\{C(c)\}, \{\neg P(x), L(c,x)\}$
Garfield is a friend of Harold:	$\{F(c,b)\}$
Harold likes anyone who likes Garfield:	$\{L(b,x), \neg L(x,c)\}$
Whatever Garfield likes, he eats:	$\{\neg L(c,x), E(c,x)\}$
Cats like mice:	$\{L(x,y), \neg C(x), \neg M(y)\}$
Friendship is mutual:	$\{\neg F(x,y), F(y,x)\}$
If you are a friend of somebody, you like them:	$\{\neg F(x,y), L(x,y)\}$

Provide a proof by resolution to show that Harold likes himself, given the assumptions expressed in the table.

Question 4 **(8 marks)**

A. For each of the following six strings, indicate (with a tick in the box) if the string is an element of the language $(\texttt{ab})^*(\texttt{ba})^*$:

☐	abba	☐	abbbba	☐	ababba
☐	abaab	☐	bababa	☐	baab

B. Draw a DFA which recognises $(\texttt{ab})^*(\texttt{ba})^*$. Make sure your automaton is complete and deterministic.

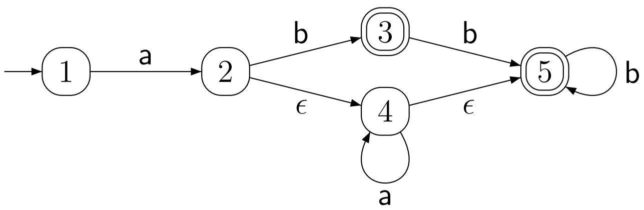
C. The class of regular languages is closed under intersection, so the language $L = \texttt{a}^*\texttt{b}^* \cap \texttt{b}^*\texttt{a}^*$ is regular. Write a regular expression for L, making it as simple as you can.

a*b* = a*, b*, (ab)*
b*a* = a*, b*, (ba)*
a* U b*
(a U b)*

Question 5

(8 marks)

Consider this NFA N:



A. Assuming N's alphabet is $\{\texttt{a}, \texttt{b}\}$, use the subset construction method to transform N to an equivalent DFA. Label the DFA's states so that it is clear how you obtained the DFA from the NFA.

B. Give the simplest possible regular expression for $L(N)$, the language recognised by N:

C. Let G be the context-free grammar $(\{S, T\}, \{\texttt{a}, \texttt{b}\}, R, S)$ with set R of rules

$$
\begin{array}{lcl}
S & \to & \texttt{a}\ S\ \texttt{a} \\
S & \to & \texttt{b}\ S\ \texttt{b} \\
S & \to & T
\end{array}
\qquad
\begin{array}{lcl}
T & \to & \texttt{a}\ T \\
T & \to & \texttt{b}\ T \\
T & \to & \epsilon
\end{array}
$$

(aUb)*

and let G' be the context-free grammar $(\{S'\}, \{\texttt{a}, \texttt{b}\}, R', S')$ with set R' of rules

$$
\begin{array}{lcl}
S' & \to & \texttt{a}\ S'\ \texttt{b} \\
S' & \to & \epsilon
\end{array}
$$

(ab)*

Give a regular expression for $L(G) \cup L(G')$.

(aUb)*

Question 6 **(8 marks)**

A. Use generalised induction to show that every integer greater than 17 can be written as a sum of 4s and 7s. That is, for every $n > 17$, there exist non-negative integers i and j such that $n = 4i + 7j$.

B. Let G be the following *ambiguous* context-free grammar:

$$S \rightarrow \epsilon \mid S\ \mathsf{a\ a\ a\ a} \mid S\ \mathsf{a\ a\ a\ a\ a\ a\ a}$$

Describe a string that demonstrates the ambiguity of G, that is, a string which has two different parse trees.

C. Find an unambiguous context-free grammar equivalent to G. You may use the result in part **A** even if you didn't answer that part.

Question 7 **(8 marks)**

A. Let $\mathcal{F}$ and $\mathcal{G}$ be sets of sets. Using the membership predicate $\in$ together with quantifiers, we can express set relation in logical form. For example, $\bigcup \mathcal{F} \subseteq \bigcap \mathcal{G}$ becomes

$$\forall x \ (\exists y \ (y \in \mathcal{F} \land x \in y) \Rightarrow \forall z \ (z \in \mathcal{G} \Rightarrow x \in z))$$

Give a logical translation of $\bigcap \mathcal{F} \subseteq \bigcup \mathcal{G}$.

B. Show that, for all languages L and M, $(L \setminus M)^* \not\subseteq (L^* \setminus M^*)$.

C. Give an example of languages L and M for which $(L^* \setminus M^*) \subseteq (L \setminus M)^*$ fails to hold.

Question 8 **(8 marks)**

Let $\mathbb{N}_n = \{0, 1, 2, \ldots, n\}$. Assume that functions are given as binary relations (or, when we represent functions in Haskell, as lists of pairs). The following are two example functions from $\mathbb{N}_6$ to $\mathbb{N}_6$:

$$\begin{array}{rcl} g_1 & = & \{(5,5),(2,3),(4,5),(0,0),(1,0)\} \\ g_2 & = & \{(5,5),(2,3),(4,5),(3,4),(0,0),(1,0),(6,0)\} \end{array}$$

A function $f : X \to X$ is *idempotent* iff $f(x) = f(f(x))$ for all $x \in X$. Note that g_1 is not total, and g_2, while a total function, is not idempotent.

For this question you can make use of functions from Haskell's Prelude, as well as functions from the `List` library, including, if needed, `sort` and `nub` (the latter removes duplicates from a list).

A. Write a Haskell function

```
isTotalFct :: Int -> [(Int,Int)] -> Bool
```

so that '`isTotalFct n r`' decides whether the binary relation `r` represents a total function from $\mathbb{N}_{\mathtt{n}}$ to $\mathbb{N}_{\mathtt{n}}$.

B. Write a Haskell function

```
isIdempotent :: Int -> [(Int,Int)] -> Bool
```

so that '`isIdempotent n r`' decides whether `r` is idempotent. For this part you can assume that `r` is known to be a total function from $\mathbb{N}_{\mathtt{n}}$ to $\mathbb{N}_{\mathtt{n}}$.

Question 9 **(6 marks)**

Construct a Turing machine M (over alphabet $\{\mathsf{a}, \mathsf{b}\}$) which will decide the language A consisting of all strings of length 4 or greater, having a as their fourth last symbol. More formally,

$$A = \left\{ w \;\middle|\; \begin{array}{l} w \in \{\mathsf{a}, \mathsf{b}\}^* \text{ has length 4 or more,} \\ \text{and the fourth last symbol in } w \text{ is } \mathsf{a} \end{array} \right\}$$

For example, abba and bbaaab are in A, but baba and aaa are not. You should present the Turing machine as a state diagram. You can leave out its reject state, with the understanding that missing transitions are transitions to the reject state. However, indicate clearly the initial state q_0 and the accept state q_a.

Overflow space

Use this page if you ran out of writing space in some question. Make sure to leave a pointer to this page from the relevant question.