Roll No.: ______________________

CS 208 : Automata Theory and Logic Spring 2024

End-semester Exam

Time: 14:00 - 17:00 *Max Marks: 60*

Instructions:

- ***Please write your roll number on all pages in the space provided at the top.***
- *Be brief, complete, and stick to what has been asked.*
- ***You must write your answer for every question only in the space allocated for answering the question. Answers written outside the allocated space risk not being graded.***
- ***Extra sheets are provided at the end of your answer book for rough calculations. If you need rough sheets beyond these, please ask an invigilator. You must write your roll number on all rough calculation sheets.***
- ***You must submit this question+answer book in its entirey along with any extra answer book for rough calculations (if you used one).***
- *Untidy presentation of answers, and random ramblings will be penalized by negative marks.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***
- ***Do not copy solutions from others. All detected cases of copying will be reported to DADAC with names and roll nos. of all involved. The stakes are high if you get reported to DADAC, so you are strongly advised not to risk this.***

DO NOT TURN THIS PAGE UNTIL YOU ARE ASKED TO.

Marks:

1a	1b	1c	2a	2b	3a	3b	3c	4a	4b	5a	5b	6

1. **A dose of logic** 10

   Consider the following first order logic formula over the vocabulary $\{P, Q, f\}$, where P and Q are binary predicates and f is a unary function.

   $$\varphi \;\equiv\; \forall x\big(\exists y\big(P(x,z) \rightarrow \exists z\big(Q(x,z) \rightarrow P(y, f(z))\big)\big) \vee Q(f(y), x)\big)$$

   In the above formula, the scope of every quantifier is explicitly indicated by parentheses immediately after the quantifier. Thus, the scope of $\exists z$ in the above formula is the formula $Q(x,z) \rightarrow P(y, f(z))$ and this is indicated by $\exists z\big(Q(x,z) \rightarrow P(y, f(z))\big)$.

   (a) Find the free variables of φ. You must clearly show your reasoning; otherwise your solution will fetch no marks. 2

   (b) Write a formula in prenex conjunctive normal form that is *semantically equivalent* to φ. Remember that a semantically equivalent formula must necessarily be over the same vocabulary and must have the same free variables (if any) as φ. In addition, your formula must be in prenex normal form in which the matrix (or quantifier-free part) is in conjunctive normal form (over the predicates Q and P). You must clearly show your reasoning; otherwise your solution will fetch no marks. 3

(c) A student wants to construct a structure M over the vocabulary $\{P, Q, f\}$ and find two assignments α and β of free variables (if any) of φ, such that $M, \alpha \models \varphi$ and $M, \beta \models \neg\varphi$. Indicate whether this is possible. If your answer is in the positive, you must give M, α, β and clearly show your reasoning why $M, \alpha \models \varphi$ and $M, \beta \models \neg\varphi$. If your answer is in the negative, you must give clear reasoning why it is impossible to find M, α, β such that $M, \alpha \models \varphi$ and $M, \beta \models \neg\varphi$. $\boxed{5}$

2. **Something regular** $\boxed{10}$

   Let $\Sigma_n = \{0, 1, 2, 3, \ldots n\} \cup \{\#\}$, where $n \geq 1$. Define $L_n = \{w \in \Sigma_n^* \mid \text{every non } \# \text{ symbol that appears after the first } \# \text{ in } w \text{ also appears before the first } \# \text{ in } w\}$. For example, $012\#0\#21 \in L_2$, but $01\#021 \notin L_2$.

   (a) Construct a deterministic finite automaton (DFA) for the language L_1. $\boxed{4}$

(b) Show that any deterministic finite automaton for the language L_n must have at least 2^{n+1} states. You must provide clear justification for your arguments; otherwise your answer will fetch no marks. $\boxed{6}$

3. **Context free (or not?) languages** [10]

   For each of the following languages, you are required to determine if the language is context free or not.

   If you think the language is context-free, you have to provide a CFG/PDA along with complete justification of why your CFG generates/PDA accepts by empty stack all and only the strings in the language. Alternatively, if you think the language is not context-free, you have to prove why the language violates the Pumpling Lemma for context-free languages. Simply saying that the language is/is not context-free without justification will fetch no marks.

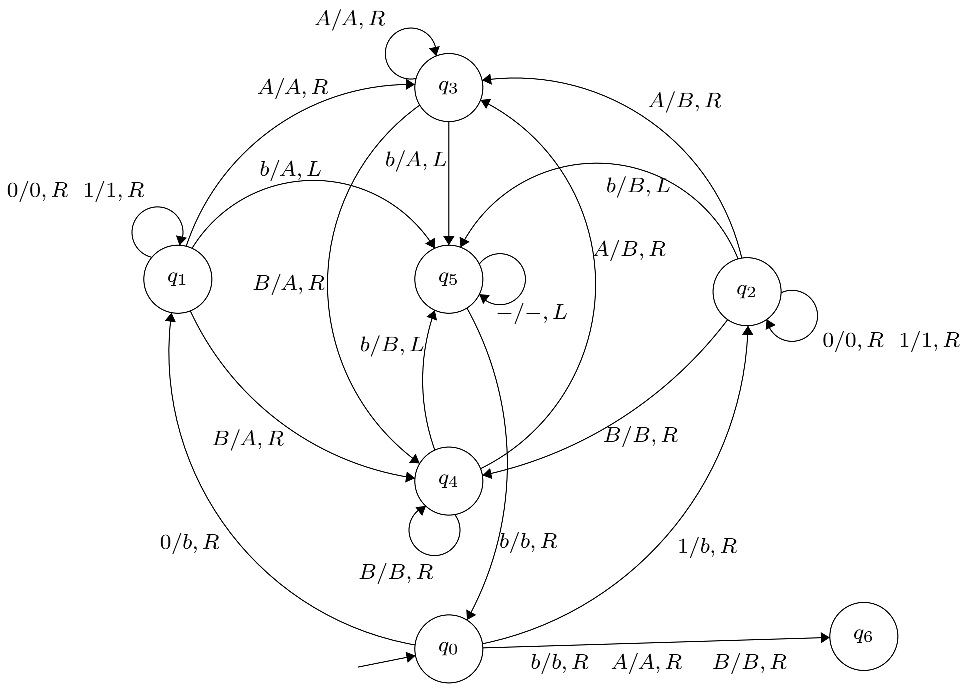
   (a) $L_1 = \{0^m 1^n 2^q \mid m, n, q \geq 1 \text{ and } n = 2m + 2q\}$ [3]

(b) $L_2 = \{0^m 1^n \mid m, n \geq 1 \text{ and } n \geq m^2\}$. $\boxed{3}$

(c) $L_3 = \{0^m 1^n 0^p 1^q \mid m, n, p, q \geq 1 \text{ and } m + p = n + q\}$. You may not assume that $n > m$ or $p > q$. $\boxed{4}$

4. **Turing machines as function evaluators** [10]

Consider the single-tape single-head Turing machine shown below. The tape alphabet of this machine is $\Gamma = \{0, 1, b, A, B\}$, where b is the blank symbol. The tape is assumed to be infinite on both sides. The starting state is q_0. The annotation $-/-, L$ on the transition from q_5 to q_5 is shorthand for $0/0, L \;\; 1/1, L \;\; A/A, L \;\; B/B, L$. For transitions annotated by multiple labels, a space has been used between labels. For example, the transition from q_2 to q_2 is labeled by $0/0, R$ and $1/1, R$. Similarly, for the transitions from q_1 to q_1 and from q_0 to q_6.



A student erroneously claims that this Turing machine has the following behaviour:

> If the machine is started with any string $w \in \{0, 1\}^*$ on its tape and with the head positioned at the leftmost letter of w (if $w \neq \varepsilon$), it halts with a string $u \in \{A, B\}^*$ on its tape (ignoring infinite sequences of blanks before and after w and u), where $u = f(w)$. Furthermore, the function $f : \{0, 1\}^* \to \{A, B\}^*$ is defined as follows:
> - $f(\varepsilon) = \varepsilon$
> - $f(0.v) = A.f(v)$, where $v \in \{0, 1\}^*$
> - $f(1.v) = B.f(v)$, where $v \in \{0, 1\}^*$
>
> It is assumed that the empty string ε is represented by a tape with only blank symbols.

(a) Give an input string $w \in \{0, 1\}^*$ of length no larger than 3 and the corresponding sequence of configurations/instantaneous descriptions of the Turing machine (using the notation studied in class) that demonstrate the incorrectness of the student's claim. You must give both the string w and the sequence of configurations/instantaneous descriptions to get marks. Answers with any of these parts missing will fetch 0 marks. [5]

(b) Rectify the student's definition of f such that the her claim about the behaviour of the machine holds. You must give the new definition of f and also provide justification why your rectification makes the student's claim correct. Answers without justification will fetch no marks. $\boxed{5}$

5. **Closure properties (or not?) of languages** 10

   Recall from class discussions that to show that a language is not recursively enumerable (resp. not recursive), it is ***not sufficient*** to argue that you cannot think of a Turing machine that accepts the language (resp. the language and its complement) by halting. Instead, you must show that there ***does not exist*** any Turing machine that accepts the language (resp. the language and its complement) by halting.

   Let L be a language over an alphabet Σ and let $\mathit{pref}(L) = \{u | u \in \Sigma^* \text{ and } \exists v \in \Sigma^* \, u.v \in L\}$.

   (a) If L is recursively enumerable, is $\mathit{pref}(L)$ necessarily recursively enumerable? Either give a proof or provide a counterexample. In case you are giving a counterexample, you must provide Σ and L, and prove that L is recursively enumerable, but $\mathit{pref}(L)$ is not. 5

(b) If L is recursive, is $\mathit{pref}(L)$ necessarily recursive? Either give a proof or provide a counterexample. In case you are giving a counterexample, you must provide Σ and L, and prove that L is recursive, but $\mathit{pref}(L)$ is not. $\boxed{5}$

6. **Logic and computability** $\boxed{10}$

Let $\mathcal{V}$ be the vocabulary $\{R, s, =\}$, where R is a binary predicate, s is a unary function and $=$ is the usual equality predicate. We will say a $\mathcal{V}$-structure K, given by (U^K, R^K, s^K), is "good" if (a) the universe U^K is $\mathbb{N}$, i.e. the set of natural numbers, (b) the interpretation $s^K : \mathbb{N} \to \mathbb{N}$ maps every natural number to its successor, i.e. $s^k(x) = x + 1$ for all $x \in \mathbb{N}$, and (c) the interpretation $R^K : \mathbb{N} \times \mathbb{N} \to \{\mathsf{True}, \mathsf{False}\}$ is given by a Turing machine M_K that always halts on an input string $u\#v$, where u, v are (encodings of) natural numbers and $\#$ is a demarcator symbol, not used in the encoding of u or v. Specifically, if M_K halts in an accepting state on input string $u\#v$ for $u, v \in \mathbb{N}$, we say $R^K(u, v)$ evaluates to True; otherwise, $R^K(u, v)$ evaluates to False. Effectively, machine M_K gives a way of evaluating $R^K(u, v)$ for every $u, v \in \mathbb{N}$.

Given the above description, a "good" $\mathcal{V}$-structure K is completely specified by the Turing machine M_K referred to above. Let $L^K_{\mathcal{V}}$ denote the set of all syntactically correct $\mathcal{V}$-sentences (i.e. $\mathcal{V}$-formulas with no free variables) that evaluate to True on a "good" structure K. Clearly $L^K_{\mathcal{V}}$ is a language and we can talk about properties of this language in the same way we do for other languages.

Give an example of a "good" structure K such that $L^K_{\mathcal{V}}$ is neither recursively enumerable nor co-recursively enumerable. Note that the universe and interpretation of s in a "good" structure K are already fixed, and you cannot change them.

Your answer must consist of the following parts:

- Desciption of Turing machine M_K (this can be an English language description) specifying what M_K does on being provided an input $u\#v$, where $u, v \in \mathbb{N}$.
- Justification of why M_K halts on every input $u\#v$, where $u, v \in \mathbb{N}$.
- Description of the interpretation $R^k : \mathbb{N} \times \mathbb{N} \to \{\mathsf{True}, \mathsf{False}\}$ implemented by M_k.
- Justification of why $L^K_{\mathcal{V}}$ is neither recursively enumerable nor co-recursively enumerable for the chosen "good" structure K.

Answers missing any of the above parts will fetch no marks.