Roll No.: ____________________

CS 208 : Automata Theory and Logic Spring 2024

Mid-semester Exam

Time: 8:30 - 10:30 *Max Marks:* 55

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.***
- ***You can use an extra answer book for rough calculations. You must write your roll number on the extra answer book if you are using one.***
- ***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	4a	4b	5

THIS PAGE IS INTENTIONALLY LEFT BLANK

1. **A bit of warmup** [10]

   Consider the formula $\varphi = \bigwedge_{i=1}^{5} \left(x_i \rightarrow (x_{i+1} \wedge x_{i+2})\right)$ on propositional variables $x_1, \ldots x_7$.

   (a) Convert φ to a semantically equivalent conjunctive normal form (CNF) formula with at most 10 clauses. You must clearly indicate what the clauses of your CNF formula are, along with justification of the semantic equivalence of your formula with φ. [3]

(b) What is the count of satisfying assignments of φ? You must give clear justification for your answer. Answers without justification will fetch no marks, even if the answer is correct. $\boxed{4}$

(c) Taking cue from your answer to the previous subquestion, convert φ to a semantically equivalent disjunctive normal form (DNF) formula with at most 10 cubes. You must clearly indicate what the cubes of your DNF formula are, along with justification of the semantic equivalence of your formula with φ. $\boxed{3}$

2. **Horn Ok Please!** [10]

Consider the following clauses on propositional variables $\{x_1, \ldots x_{10}\}$.

Clause id	Clause	Clause id	Clause
C_1	$\neg x_7 \vee \neg x_8 \vee \neg x_9 \vee x_{10}$	C_2	$\neg x_2 \vee \neg x_6 \vee x_7$
C_3	$\neg x_1 \vee \neg x_2 \vee \neg x_{10} \vee x_7$	C_4	$\neg x_2 \vee \neg x_{10} \vee x_3$
C_5	$\neg x_3 \vee \neg x_4 \vee \neg x_5 \vee x_6$	C_6	$\neg x_3 \vee \neg x_{10} \vee x_2$
C_7	$\neg x_3 \vee \neg x_4 \vee \neg x_7$	C_8	x_6
C_9	$\neg x_6 \vee \neg x_{10}$	C_{10}	$\neg x_6 \vee x_2$

The formula $\varphi = \bigwedge_{i=1}^{10} C_i$ is a Horn formula.

(a) Using the algorithm for checking satisfiability of a Horn formula studied in class, find a satisfying assignment of φ. You must clearly show which variables are assigned what values in each step of the algorithm along with the reason for the assignment (i.e. "implication with true LHS" or "set all remaining variables to ...") for each step, in the form of a table as shown below. Feel free to extend the table as much as is required. Simply giving the final satisfying assignment without showing any steps and reasons will fetch no marks. [5]

Final assignment:

Step No.	Var assigned	Value assigned	Reason

(b) Suppose we apply the DPLL algorithm to φ, where DPLL involves checking for unit clauses, then for pure literals and finally branching and backtracking, as studied in class. Suppose that whenever we need to choose a variable for branching, variable x_i is chosen before variable x_j for $i, j \in \{1, \ldots 10\}$ and $i < j$. Finally, suppose that whenever we need to choose a value for a variable on which we are branching, the value 1 is chosen before the value 0. Find the satisfying assignment of φ identified by DPLL under the above assumptions. You must clearly show which variable is assigned what value in each step, along with the reason (i.e. one of "unit clause elimination", "pure literal elimination", "branching" or "backtracking to last branch level") for each step in the form of a table as shown below. Feel free to extend the table as much as is required. Simply giving the final satisfying assignment without showing any steps and reasons will fetch no marks. $\boxed{5}$

Final assignment:

Step No.	Var assigned	Value assigned	Reason

3. **Being regularly expressive** [10]

Let $\Sigma = \{0, 1\}$. For each pair of regular expressions shown below, either show that they represent the same language or show that they represent different languages. To show the former, you have to argue why every string in the language of one expression is also in the language of the other expression and vice versa. To show the latter, you have to show one string in the language of one regular expression that is not in the language of the other expression. Simply saying that the expressions represent the same/different languages without giving any justification will fetch no marks, even if your answer is correct.

(a) $E_1 = \big((\boldsymbol{\varepsilon} + \mathbf{0}) \cdot (\mathbf{1} + \boldsymbol{\varepsilon})\big)^*$ and $E_2 = \big(\mathbf{0} + \boldsymbol{\varepsilon}\big) \cdot \big(\mathbf{0} + \mathbf{1}\big)^* \cdot \big(\mathbf{1} + \boldsymbol{\varepsilon}\big)$ [5]

(b) $E_3 = \left((\mathbf{0}+\mathbf{1})^* \cdot \mathbf{11} \cdot (\mathbf{0}+\mathbf{1})^*\right)^+$ and $E_4 = \left((\mathbf{0}+\mathbf{1})^* \cdot \mathbf{1} \cdot (\mathbf{0}+\mathbf{1})^*\right) \cdot \left((\mathbf{0}+\mathbf{1})^* \cdot \mathbf{1} \cdot (\mathbf{0}+\mathbf{1})^*\right)^+$ $\boxed{5}$

4. **To be regular or irregular** $\boxed{10}$

Let $\Sigma = \{0, 1\}$. For each of the languages below, indicate whether the language is regular or not. In each case, you must give a proof of regularity or non-regularity in order to score marks. Simply indicating whether the language is regular or non-regular will fetch no marks even if your answer is correct.

(a) $L_1 = \{w \in \Sigma^* \mid \text{every } 0 \text{ in } w \text{ is followed by at least three } 1s\}$. Thus, $0110111 \notin L_1$, but $011110111011111 \in L_1$. $\boxed{5}$

(b) $L_2 = \{w \in \Sigma^* \mid |w| \text{ in base 10 has an even number of digits}\}$. Here, $|w|$ denotes the length of w. For example, $00 \notin L_2$ since $|00| = 2$ in base 10, and 2 has an odd number of digits. However, $001100110011 \in L_2$ since $|001100110011| = 12$ in base 10, and 12 has an even number of digits. $\boxed{5}$

5. **Robust Automata** [15]

In this question, we consider non-deterministic finite automata (NFA) without ε-transitions, but with a modification of the acceptance condition studied in class. This modification is motivated by the notion of *robust acceptance* of words by an NFA.

Formally, let $A = (Q, \Sigma, Q_0, \delta, F)$ be an NFA, where $Q_0, F \subseteq Q$ and $\delta : Q \times \Sigma \to 2^Q$. We define a *run* of A on a word $w = a_1 a_2 \ldots a_k$, where each $a_i \in \Sigma$, as a sequence of states $(q_0, q_1, \ldots q_k)$ such that $q_0 \in Q_0$ and $q_i \in \delta(q_{i-1}, a_i)$ for all $i \in \{1, \ldots k\}$. If $w = \varepsilon$, a run of A on w is (q_0), where $q_0 \in Q_0$. A run of A on w is said to be *accepting*, if the last state in the corresponding sequence of states, i.e q_k in the above notation, is in F. We say that a word $w \in \Sigma^*$ is *k-robustly accepted* by A iff there are at least k *distinct* accepting runs of A on w. Here, two runs $(q_0, \ldots q_k)$ and $(q'_0, \ldots q'_k)$ of A on a word w of length k (≥ 0) are said to be *distinct* iff there is some i $(0 \leq i \leq k)$ such that $q_i \neq q'_i$. Note that the usual notion of acceptance by an NFA coincides with 1-robust acceptance.

For $k \geq 1$, define $L_k(A) = \{w \in \Sigma^* \mid w \text{ is } k-\text{robustly accepted by } A\}$.

(a) Show that for every NFA A and every $k \geq 1$, $L_k(A)$ is regular. You must show your reasoning clearly. Answers without reasoning will fetch no marks. [10]

(b) Give an example of an NFA A with at most 5 states such that $L_{2i}(A)$ is a strict subset of $L_i(A)$, i.e $L_i(A) \cap \left(\Sigma^* \setminus L_{2i}(A)\right) \neq \emptyset$, for all $i \geq 1$. You must explain clearly why your NFA satisfies the above property. Answers without explanation or involving more than 5 states will fetch no marks. $\boxed{5}$