CS 208 : Automata Theory and Logic Spring 2024

Homework 2

Due: 2^{nd} Feb, 2024 *Max Marks:* 40 (+bonus 10)

Instructions:

- *Please start writing your solution to each homework problem on a fresh page.*
- *For each homework problem, you must scan your solution and upload a separate PDF file on Moodle. Please check Moodle for detailed instructions on file naming and uploading instructions.*
- *Be brief, complete, and stick to what has been asked.*
- *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.***

---

1. The Case of Dr. Equisemantic and Mr. Irredundant 15 points

Assume we have a countably infinite list of propositional variables $p_1, p_2, \ldots$. For this problem, by "formula", we always mean a finite string representing "syntactically-correct formula". Let Σ be a set of formulae. For any formula φ, we say $\Sigma \models \varphi$ (read as Σ semantically entails φ) if for any assignment α of the propositional variables that makes all the formulae contained in Σ true, α also makes φ true.

Let us call two sets of formulae Σ_1 and Σ_2 *equisemantic* if for every formula φ, we have $\Sigma_1 \models \varphi$ if and only if $\Sigma_2 \models \varphi$. Furthermore, let us call a non-empty set of formulae Σ *irredundant* if no formula σ in Σ is semantically entailed by $\Sigma \setminus \{\sigma\}$.

1. *[5 points]* Show that any set of formulae Σ must always be countable. This implies that we can enumerate the elements of Σ. Assume from now on that $\Sigma = \{\sigma_1, \sigma_2, \sigma_3, \ldots\}$.
2. Suppose we define Σ' as follows:

$$\begin{aligned}\Sigma' = \{&\sigma_1,\\ &(\sigma_1) \implies \sigma_2,\\ &(\sigma_1 \wedge \sigma_2) \implies \sigma_3,\\ &(\sigma_1 \wedge \sigma_2 \wedge \sigma_3) \implies \sigma_4,\\ &\vdots\\ \}&\end{aligned}$$

   Suppose we remove all the tautologies from Σ' and call this reduced set Σ''. Prove that Σ'' is irredundant and equisemantic to Σ. You can proceed as follows:

   (a) *[5 points]* Show that a non-empty satisfiable set Γ with $|\Gamma| \geq 2$ is irredundant if and only if there is no $\gamma \in \Gamma$ such that $(\Gamma \setminus \{\gamma\}) \cup \{\neg\gamma\}$ is satisfiable.

   (b) *[5 points]* Use the above result to show that Σ'' is irredundant and equisemantic to Σ.

2. A follow-up of the take-away question of Tutorial 2 25 points

To recap from the take-away question of Tutorial 2, we will view the set of assignments satisfying a set of propositional formulae as a language and examine some properties of such languages.

Let $\mathbf{P}$ denote a countably infinite set of propositional variables $p_0, p_1, p_2, \ldots$. Let us call these variables positional variables. Let Σ be a countable set of formulae over these positional variables. Every assignment $\alpha : \mathbf{P} \to \{0, 1\}$ to the positional variable can be uniquely associated with an infinite bitstring w, where the i^{th} bit $w_i = \alpha(p_i)$. The language defined by Σ, also called $L(\Sigma)$, is the set of bitstrings w for which the corresponding assignment α, that has $\alpha(p_i) = w_i$ for each i, satisfies Σ, that is, for each formula $F \in \Sigma$, $\alpha \models F$. In this case, we say that $\alpha \models \Sigma$. Let us call the languages definable in this manner as PL-definable languages.

Example:

Let $\Sigma = \{p_0 \to p_1, p_1 \to p_2, p_2 \to p_3, \ldots\}$.

Then $L(\Sigma) = \{1111\ldots, 0111\ldots, 0011\ldots, 0001\ldots, \ldots, 0000\ldots\}$, or, to be precise, if we denote the infinite bitstring containing only 1s by 1^ω and the infinite bitstring containing only 0s by 0^ω, and the finite bitstring consisting of k 0s by 0^k, then $L(\Sigma) = \{0^k 1^\omega : k \in \mathbb{N}\} \cup \{0^\omega\}$.

(a) *[5 marks]* Show that the language L consisting of all infinite bitstrings except $000\ldots$ (the bitstring consisting only of zeroes) is **not** PL-definable. You may want to prove the following lemma in order to solve this question:

Lemma:

For every PL-definable language L and bitstring $x \notin L$ there exists a finite prefix y of x such that for any infinite bitstring w, $yw \notin L$ (yw refers to the concatenation of y and w).

(b) *[5 + 5 points]* Show that PL-definable languages are closed neither under countable union nor under complementation.

Hint: Try using the result proven in part (a)

(c) *[10 points]* Show that a PL-definable language either contains every bitstring or does not contain uncountably many bitstrings.

Hint: Try using the lemma proven in part (a)

(d) *[Bonus 10 points]* A student tries to extend the definition of PL-languages by allowing the use of "dummy" variables.

Let $\mathbf{X} = \{x_0, x_1, \ldots\}$ denote a countably infinite set of "dummy" variables and let Σ denote a countable set of formulae over both positional and dummy variables. An infinite bitstring w is in the language defined by Σ if and only if there exists an assignment $\alpha : \mathbf{P} \cup \mathbf{X} \to \{0, 1\}$ such that $\alpha \models \Sigma$ and $w_i = \alpha(p_i)$ for each i. Note that the assignment of "dummy" variables in $\mathbf{X}$ are not represented in w. Let us call the languages definable this way extended PL-definable languages, or EPL-definable languages.

Show that EPL and PL are equally expressive, ie every EPL-definable language is a PL definable language and vice versa. This means our attempt to strengthen PL this way has failed. You can use the following theorem without proof:

Theorem:

Let $S_0, S_1, S_2, \ldots$ denote an infinite sequence of non-empty sets of finite bitstrings such that for every $i > 0$ and for every bitstring $x \in S_i$ and every $j \leq i$, there exists a prefix y of x in S_j. Then there exists an infinite bitstring z such that every S_i contains a prefix of z.