CSCI 1010 - Fall'26 Theory of Computation Lorenzo De Stefani

HW7

November 5, 2026 at 11:59pm

Reminder: Submit your assignment on Gradescope by the due date. Submissions must be typeset. Each page should include work for only one problem (i.e., make a new page/new pages for each problem). See the course syllabus for the late policy.

While collaboration is encouraged, please remember not to take away notes from any labs or collaboration sessions. Your work should be your own. Use of other third-party resources is strictly forbidden.

Please monitor Ed discussion, as we will post clarifications of questions there.

Problem 1

Boolean variables can take on values of TRUE or FALSE. Boolean operators are $\wedge$ (and), $\vee$ (or) and $\neg$ (not). A Boolean formula is an expression with Boolean variables and operators A Boolean formula is satisfiable if some assignment of 0s and 1s to the variables makes the formula evaluate to TRUE. For example, the formula $(\neg x \vee y) \wedge (x \vee \neg z)$ is satisfiable by several assignments including $x = FALSE$, $y = TRUE$, $z = FALSE$.

In the Boolean Satisfiability problem (SAT), given a Boolean formula ϕ, we aim to decide whether ϕ has any satisfying assignments. That is, given the encoding $\langle \phi \rangle$ of the Boolean formula ϕ we aim to decide whether $\langle \phi \rangle$ is a member of the language

$$SAT = \{\langle \phi \rangle | \phi \text{ is a satisfiable Boolean formula}\}.$$

As we will see in class, SAT is a very *HARD* problem. However, we can study variations of the problem that are considerably *easier*.

A *clause* is a type of Boolean formula in which literals (Boolean variables or their negation) are connected by only "or" operators. For example $(x_1 \vee x_2 \vee \neg x_1)$ is a clause. We say that a Boolean formula is in Conjunctive Normal Form (CNF) or clausal normal form if it is a conjunction (and) of one or more clauses. For example $(x_1 \vee x_2 \vee \neg x_1) \wedge (x_1 \vee x_3 \vee \neg x_2) \wedge x_1 \wedge (x_2 \vee \neg x_1)$ is a CNF Boolean formula.

We will consider CNF Boolean Formulas such that each variable appears at most twice (including the times that it appears as negated). For example,

in $(x_1 \vee x_2 \vee \neg x_1) \wedge (x_3 \vee x_3 \vee \neg x_2) \wedge x_4 \wedge (x_6 \vee \neg x_7)$ each variable occurs at most twice, while in $(x_1 \vee x_2 \vee \neg x_1) \wedge (x_1 \vee x_3 \vee \neg x_2) \wedge x_1 \wedge (x_2 \vee \neg x_1)$ that is not the case (x_1 appears 5 times).

Consider the language. $2RCNFSAT = \{\langle\phi\rangle | \phi$ is a satisfiable CNF Boolean formula in which each variable occurs at most twice $\}$.

Prove that $2RCNFSAT \in TIME(n^k)$, where n is the size of the encoding of a Boolean formula and k is some constant value independent of n. That is, provide an algorithm that decides whether an input $\langle\phi\rangle$ is a member of 2RCNFSAT in polynomial time with respect to the size of the input. Here you can assume that the size of the input is the number of literals in ϕ (i.e., the sum of occurrences of variables in ϕ). For example, if $\phi = (x_1 \vee x_2 \vee \neg x_1) \wedge (x_1 \vee x_3 \vee \neg x_2) \wedge x_1 \wedge (x_2 \vee \neg x_1)$, $|\phi| = n = 9$.

Problem 2

Let A be a language such that $A \in$ NP.

1. Let B be a language such that $B \in$ NP. Prove that $AB = \{x \cdot y \mid x \in A \text{ and } y \in B\} \in$ NP, so that NP is closed under concatenation.
2. Prove that $A^* = \{x_1 x_2 \dots x_n \mid n \geq 0 \text{ and } x_i \in A\} \in$ NP, so that NP is closed under the Kleene star operation.

Problem 3

Let $QUADRUPLE - SAT = \{\langle\phi\rangle | \phi$ has at least four satisfying assignments$\}$. Show that $QUADRUPLE - SAT$ is in NP.