HW10
Due: Dec 3, 2015

Attach a fully filled-in cover sheet to the front of your printed homework. Your name should not appear anywhere except on the cover sheet; each individual page of the homework should include your banner ID instead.

While collaboration is encouraged in this class, please remember not to take away notes from any collaboration sessions. Also please list the names and logins of any of your collaborators at the beginning of each homework.

Please monitor Piazza, as we will post clarifications of questions there. You should hand in your solutions by 12:55 to the CSCI 51 bin on the second floor of the CIT. Late homeworks are not accepted.

Problem 1

Determine whether P, NP, and PSPACE are closed under the complementation operation. Also determine whether PSPACE is closed under the Kleene star operation. Prove your answer, or relate it to an open problem.

Problem 2

Let $SATRange$ be a language defined as follows:

$SATRange = \{ \langle \phi, lo, hi \rangle \mid hi \geq lo \geq 0,$

$\phi$ is a Boolean formula with $lo \leq k \leq hi$ satisfying assignments, $lo$ and $hi$ are integers represented in binary\}

(a) Show that $SATRange$ is in PSPACE.
(b) Show that $SATRange$ is NP-hard.
(c) Show that $SATRange$ is coNP-hard.
(d) Show that if $SATRange$ is in NP, then NP=coNP.
(e) Give a polynomial-time algorithm that, on input a Boolean formula $\phi$, determines the number of satisfying assignments for $\phi$ using an oracle for the language SATRange.

Problem 3

In his days at Starfleet Academy, James T. Kirk was challenged with an instructional game LOSINGSUBSET played on a strange, holographic “board” representing a graph. The rules were as follows:

- At the beginning of the game, a certain node of the graph $s$ is lit up green.
- The two players take turns picking a neighboring unlit node to the green node. It becomes the new green node, and the previous green node is now colored red. (If the green is at node $u$, it can only move to node $v$ if $(u, v) \in E$ and if $v$ is not red).
- There is a subset of nodes $L \subset V(E)$. The first player to move onto a vertex that belongs to the subset $L$ loses, and the other player wins.
- If the current player cannot make a move (because all neighboring nodes have been visited), and no one has lost yet, the game ends in a tie.

Kirk never was very good at it, and never developed an intuition about whether or not he could force a win. Now, he wants to explain that it wasn’t his fault.

Show that determining whether the first player has a winning strategy in an arbitrary instance of the game LOSINGSUBSET is PSPACE-complete. A tie is not a win. Note that an instance of the game LOSINGSUBSET consists of a directed graph $G$, a starting vertex $s$, and a losing subset $L$.

*Hint*: Recall from class the PSPACE-Complete problem GENERALIZED-GEOGRAPHY. To remind you, GENERALIZEDGEOGRAPHY is played in the same manner as LOSINGSUBSET, with two players taking turns moving a pebble on a directed graph. A player loses if he cannot move the pebble anywhere (i.e. if all neighboring nodes have already been visited).

The following questions are lab problems.
Lab Problem 1

Now that you’ve learned about the classes P and NP, you must have tried to imagine what the world would be like if P = NP. There would be efficient ways to solve so many hard problems: scheduling classes, planning your tour of every baseball stadium in the country, moving studios, distributing guns—the list goes on.

However, while making hard problems easy would be helpful in some areas of society, there are other areas that depend on the existence of hard problems. One such area is the field of cryptography. You want it to be hard for people to read your secret messages, but if all of NP suddenly became easy, then deciphering your messages might become easy as well.

In this problem, we will give one example of a building block in cryptography that could not exist if P = NP. That building block is a one-way function. A one-way function is a function that is easy to compute, but hard to invert, which is why we call it “one-way”. (You can imagine how an operation that is easy to perform, but difficult to reverse would be useful in cryptography!) We will be interested in functions over binary strings, where the length of the output is the same as the length of the input.

**Definition:** A function \( f : \{0, 1\}^* \to \{0, 1\}^* \) is a one-way function if

- For all \( x \in \{0, 1\}^* \), \(|x| = |f(x)|\).
- \( f \) is a polynomial time computable function. Recall that this means there exists a polynomial time TM \( M \) that halts with just \( f(x) \) on its tape, when started on any input \( x \in \{0, 1\}^* \).
- There does not exist a polynomial time TM \( A \) that on any input \( y \), outputs some \( x' \) such that \( f(x') = y \) if it exists.

An example of such a function is multiplication: while there is a polynomial time procedure that takes as input two \( k \)-bit integers and computes their \( 2k \)-bit product, we do not know of a polynomial-time algorithm that takes as input a \( 2k \)-bit integer and, in polynomial time, finds its \( k \)-bit factors if they exist.

(a) Consider the following language: if \( f \) is a one-way function, then \( \text{INVERSE}_\text{SUFFIX} f = \{ (y, w) \text{ where } y = f(x) \text{ for some } x \in \{0, 1\}^* \text{, and } w \text{ is the suffix (not} \)

\footnote{This definition is a simplified version of the standard cryptographic definition.}
necessarily proper) of some inverse of \( y \). That is, for some \( w' \in \{0, 1\}^* \), 
\[ f(w' \circ w) = y \].

Prove that for any one-way function \( f \), \( \text{INVERSE\_SUFFIX}\ f \) is in NP.

(b) Prove that if \( P = \text{NP} \), then no one-way function can exist. That is, prove that if \( f: \{0, 1\}^* \to \{0, 1\}^* \) is a function such that \( |x| = |f(x)| \) for all \( x \) and \( f \) is polynomial time computable, then there exists a polynomial time TM \( A \) that on any input \( y \), outputs some \( x' \) such that \( f(x') = y \) if it exists, and fails otherwise.

**Lab Problem 2**

Recall that you showed the following languages are decidable:

\[
EQ_{\text{DFA}} = \{ \langle R, S \rangle \mid R \text{ and } S \text{ are descriptions of DFAs s.t. } L(R) = L(S) \}
\]

\[
EQ_{\text{NFA}} = \{ \langle R, S \rangle \mid R \text{ and } S \text{ are descriptions of NFAs s.t. } L(R) = L(S) \}
\]

The crew of the Enterprise are attempting to bridge the gap between their galaxy and one thought to be incredibly far away, so much so that it would take a long, long time even for light to reach. (This is the same galaxy that they got the message from in Homework 9.) However, a new type of warp system has been developed, and the navigation solutions can be expressed as DFAs and NFAs.\(^2\)

Show that \( EQ_{\text{DFA}} \in P \), and show that \( EQ_{\text{NFA}} \in \text{PSPACE} \), so that the Enterprise can navigate to the distant galaxy, and boldly go where no one from their universe has gone before.

**Lab Problem 3**

A **boolean polynomial** is a polynomial with coefficients in \( \{0, 1\} \). Like regular polynomials, boolean polynomials can be added and multiplied with each other, however, \( 1 + 1 = 0 \). Here are some examples of boolean polynomials:

\(^2\)Remember, space is weird.
A boolean polynomial $p$ is called **irreducible** if it cannot written as $p = f \ast g$ where $f$ and $g$ are two lower degree boolean polynomials.

For this problem, we will assume that we represent boolean polynomials as 0/1 strings where the $n$th digit represents the coefficient of $x^n$. So, these strings will be of length $k + 1$, where $k$ is the largest-degree term of the polynomial.

a. Show that the following polynomial is not irreducible:

$$x^3 + 1$$

b. Consider the language

$$\text{IrredPoly} = \{(p) \mid p \text{ is an irreducible boolean polynomial}\}$$

Show that $\text{IrredPoly} \in \text{PSPACE}$. 