22C:131 Midterm Exam
Tuesday, 10/9

Notes: (a) Undergraduate students are required to solve the first 3 problems in the exam. Graduate students are required to solve all problems. (b) This is an open book/notes exam. (c) For graduate students, each problem is worth 50 points. For undergraduate students, Problem 1 is worth 70 points and Problems 2 and 3 are worth 65 points each.

1. Below I provide 10 “claims.” For each given “claim” state whether it is “True”, “False,” or has unknown status. If your answer is “True” or “False,” provide a 1-2 sentence justification for your answer.

(a) If $NP = coNP$ then $P = NP$.

UNKNOWN.

(b) Either $P \subseteq NP$ or $NP \subseteq EXP$.

TRUE

By Deterministic Time Hierarchy Theorem $P \subsetneq EXP$. Since $P \subseteq NP \subseteq EXP$ (Claim 2.4), either $P \subsetneq NP$ or $NP \subsetneq EXP$ (or both).

(c) $Connectivity \leq_p 3SAT$.

$Connectivity \in P \subseteq NP$ and $3SAT$ is NP-hard. Therefore $Connectivity \leq_p 3SAT$. 
22C:131 Midterm Exam
Tuesday, 10/9

Notes: (a) Undergraduate students are required to solve the first 3 problems in the exam. Graduate students are required to solve all problems. (b) This is an open book/notes exam. (c) For graduate students, each problem is worth 50 points. For undergraduate students, Problem 1 is worth 70 points and Problems 2 and 3 are worth 65 points each.

1. Below I provide 10 “claims.” For each given “claim” state whether it is “True”, “False,” or has unknown status. If your answer is “True” or “False,” provide a 1-2 sentence justification for your answer.

(a) If $NP = coNP$ then $P = NP$.

\textit{UNKNOWN.}

(b) Either $P \subseteq NP$ or $NP \subseteq EXP$.

\textit{TRUE}

By Deterministic Time Hierarchy Theorem $P \not\subseteq EXP$. Since $P \subseteq NP \subseteq EXP$ (Claim 2.4), either $P \not\subseteq NP$ or $NP \not\subseteq EXP$ (or both).

(c) Connectivity $\leq_P 3SAT$.

Connectivity $\in P \subseteq NP$ and 3SAT is NP-hard. Therefore $Connectivity \leq_P 3SAT$. 

(d) $NP \text{-hard} \setminus NP \neq \emptyset$. TRUE.

$HALT$ is an example of a language in $NP \text{-hard}$ that is not in $NP$.

(e) $3SAT \in DTIME(2^{\sqrt{n}})$. UNKNOWN
(Though unlikely)

(f) $DTIME(2^{\sqrt{n}}) \subseteq DTIME(n \cdot 2^{\sqrt{n}})$. TRUE.

By the Deterministic Time Hierarchy Theorem, $DTIME(2^{\sqrt{n}}) \not\subseteq DTIME(g(n))$ for any $g(n)$ such that $\sqrt{n} \cdot 2^{\sqrt{n}} = o(g(n))$.

Since $\sqrt{n} \cdot 2^{\sqrt{n}} = o(n \cdot 2^{\sqrt{n}})$, we obtain the above claim.

(g) $3SAT \leq_P SAT$. TRUE.

$SAT$ is $NP$-hard and therefore $\forall L \in NP$, $L \leq_P SAT$. We know that $3SAT \in NP$ and hence $3SAT \leq_P SAT$. 
(h) \((NP \cap coNP) \setminus P \neq \emptyset.\) **UNKNOWN.**
(Though quite likely)

(i) \(coEXP = EXP.\) **TRUE.**

For any \(L \in EXP,\) \(L\) can be decided by a DTM in exponential time. Therefore for any \(L \in EXP, \neg L\) can be decided by a DTM in exponential time. Since \(coEXP = \{\neg L | L \in EXP\},\) every language \(L \in coEXP\) can also be decided in exponential time.

\(\therefore coEXP \subseteq EXP\) & similarly \(EXP \subseteq coEXP,\) implying that \(coEXP = EXP.\)

(j) If \(TAUTOLOGY \in P\) then \(P = NP.\) **TRUE.**

\(TAUTOLOGY\) is \(coNP\)-complete. Therefore, if \(TAUTOLOGY \in P,\)

\(coNP = P.\) Thus every \(L \in coNP\) can be decided by a DTM in poly-time. Thus every \(L \in coNP, \neg L\) can be decided by a DTM in poly-time. Thus every \(L \in NP\) can be decided by a DTM in poly-time. Thus \(NP \subseteq P \Rightarrow NP = P = coNP.\)
2. Define a function \( \text{three} : \{0, 1\}^* \to \{0, 1\} \) as follows:

\[
\text{three}(\alpha) = \begin{cases} 
1 & \text{if} |L(M_\alpha)| = 3 \\
0 & \text{otherwise}
\end{cases}
\]

Show that \( \text{three} \) is uncomputable.

\textbf{Hint:} Use reduction from ACCEPT.

\textbf{Proof:} Suppose that \( \text{three} \) is computable and let \( T \) be a TM that computes \( \text{three} \). We now construct a TM for computing ACCEPT.

**Algorithm/TM for ACCEPT**

\textbf{Input:} \( \langle \alpha, x \rangle \in \{0, 1\}^* \times \{0, 1\}^* \)

1. Construct a TM \( M' \) that behaves as follows:

   - On input \( w \in \{0, 1\}^* \),
   - (a) \( M' \) runs \( M_\alpha \) on \( x \).
   - (b) If \( M_\alpha \) halts & accepts \( x \) then
     - If \( w \in \{0, 00, 000\} \), \( M' \) accepts \( w \)
     - Else \( M' \) rejects \( w \)
   - (c) If \( M_\alpha \) halts & rejects \( x \) then
     - \( M' \) rejects \( w \)

2. Compute \( T(\text{three}(LM')) \) and output 1 (i.e., accept) if

\( T(\text{three}(LM')) = 1 \); output 0 (i.e., reject) otherwise.

Correctness of reduction follows from the fact that if \( M_\alpha \) accepts \( x \), \( L(M') \supseteq \{0, 00, 000\} \) and if \( M_\alpha \) does not accept \( x \), then \( L(M') = \emptyset \). \( \square \)
3. These questions refer to the proof of the Cook-Levin Theorem (Lemma 2.11). Suppose that $L \in NP$. Then there is a (deterministic) TM $M$ and a polynomial $p(\cdot)$ such that for all $x \in \{0, 1\}^*$, $x \in L$ iff there exists $u \in \{0, 1\}^{p(|x|)}$ such that $M(x, u) = 1$. Suppose that for this particular $L$, $p(n) = n$ and $M$ runs in time $n^2$ where $|x| = n$. Furthermore, suppose that the alphabet for $M$ (denote $\Gamma$) is $\{0, 1, \triangleright, \square\}$ and that $M$ has 5 states (i.e., $|Q| = 5$). As in the proof, assume that $M$ uses two tapes (an input take and a work/output tape) and that $M$ is oblivious.

\[ [15] \] (a) Let $c$ denote the fewest number of bits needed to represent a snapshot of $M$. Calculate the value of $c$. Show you work for partial credit.

A snapshot is an element in $Q \times \Gamma \times \Gamma$ and therefore can take on $5 \times 4 \times 4 = 80$ possible values. It takes $\lceil \log_2 80 \rceil = 7$ bits to represent a snapshot.

\[ [20] \] (b) For a string $x \in \{0, 1\}^n$, calculate the number of variables that the boolean formula $\varphi_x$ has (as a function of $n$). Show your work for partial credit.

It takes $n + p(n) = 2n$ variables to represent the input. It takes 7 variables to represent each snapshot. There are $T(n) + 1 = n^2 + 1$ snapshots. Thus $\varphi_x$ contains $7n^2 + 2n + 7$ variables.

\[ [16] \] (c) Let $x \in \{0, 1\}^n$ and let $\varphi_x$ be the boolean formula constructed from $x$. Let $C$ be the smallest number such that every clause in $\varphi_x$ has at most $C$ literals. Calculate $C$. Show your work for partial credit.

The condition $z_i = F(z_{i-1}, y_{\text{inputpos}(i)}, z_{\text{prev(i)}})$ requires the largest clauses. Note that $F : \{0, 1\}^{2c+2} \rightarrow \{0, 1\}^c$ because $z_{i-1}$ & $z_{\text{prev(i)}}$ take $c$ bits each and $y_{\text{inputpos}(i)}$ takes 2 bits. The function $F : \{0, 1\}^{2c+2} \rightarrow \{0, 1\}^c$ can be represented using clauses of size $2c+2 = 16$. \[ \therefore C = 16. \]
4. These questions refer to the proof of the Non-deterministic Time Hierarchy Theorem (Theorem 3.2) in Arora-Barak.

(a) The theorem requires that \( f(n+1) = o(g(n)) \). Where exactly in the proof is the requirement used? (Another way of asking the same question is, where exactly would the proof break down if we replaced the condition \( f(n+1) = o(g(n)) \) by the weaker condition \( f(n) = o(g(n)) \)?)

It is used in step (1) of D's definition, where D's output on input \( 1^n \) is obtained by computing \( M_i \)'s output on \( 1^{n+1} \). On input \( 1^{n+1} \), \( M_i \) runs in time \( f(n+1) \) and for \( D \) to be able to simulate \( M_i \) to completion in time \( g(n) \), it has to be the case that \( f(n+1) = o(g(n)) \). If \( D \) were unable to simulate \( M_i \) to completion we could not claim that \( D(1^n) = M_i(1^{n+1}) \).

(b) Consider the description of the NDTM \( D \). In Case (1) (where \( f(i) < n < f(i+1) \)) \( D \) simulates \( M_i \) "using nondeterminism." Suppose that we change this step and make \( D \) simulate \( M_i \) deterministically, while retaining everything else in the description of \( D \). What would break down in the proof?

If \( D \) attempted to simulate \( M_i \) deterministically using a budget of \( n'' \) time (as in the textbook on Pg 70) then then \( D \) would not be able to simulate \( M_i \) to completion & we would not be able to claim that \( D(1^n) = M_i(1^{n+1}) \) for \( f(i) < n < f(i+1) \).

(b) Suppose we define the function \( f : \mathbb{N} \to \mathbb{N} \) as follows: \( f(1) = 2 \) and \( f(i+1) = 2 \cdot f(i) \) for all \( i > 1 \). Explain where the proof would break down if \( f \) were defined in this manner.

If \( f(i+1) = 2f(i) \) then in step (2) of D's definition at least \( (f(i)+1)^{11} \) time, which would be far more than \( f(i+1) \). The proof in the textbook relies on step 2 finishing in \( O(n^{15}) \) time but \( 2((f(i)+1)_{11}^{11}) \) asymptotically is \( 2^{((n+1)_{11}^{11})} \), which is of course much larger than \( n^{15} \).