**NAME:**

# Harvard University
## Computer Science 121 — Final Exam, Friday, December 13, 2013

This is a closed-book examination. You may use any result from lecture, Sipser, or the problem sets, as long as you quote it clearly. The alphabet is $\Sigma = \{a, b\}$ except where otherwise stated.

You have 180 minutes. The problems total to 180 points. Write your name above and on all bluebooks you use. Do the first problem on this page and turn it in along with your blue books. Good luck!

### PROBLEM 1 (19 points)

DO THIS PROBLEM ON THIS PAGE!

Complete the following table with `YES`, `NO`, or `?? (CURRENTLY UNKNOWN)`. No explanations needed. $M$ stands for a Turing machine, $G_1$ and $G_2$ are context-free grammars, and $\Sigma = \{a, b\}$.

| Language: | regular | CF | recursive | r.e. | co-r.e. | $\mathcal{P}$ | $\mathcal{NP}$ | co-$\mathcal{NP}$ |
|---|---|---|---|---|---|---|---|---|
| $\{a^n b^{2n} a^{3n} : n \geq 0\}$ | NO | NO | YES | YES | YES | YES | YES | YES |
| $\{a^n : n = 2^k \text{ for some } k \geq 0\}$ | NO | NO | YES | YES | YES | YES | YES | YES |
| $\overline{\text{3-SAT}}$ (the complement of 3-SAT) | NO | NO | YES | YES | YES | ?? | ?? | YES |
| $\{\langle M \rangle : M \text{ accepts } abab\}$ | NO | NO | NO | YES | NO | NO | NO | NO |
| $\{\langle G_1, G_2 \rangle : L(G_1) \cap L(G_2) = \emptyset\}$ | NO | NO | NO | NO | YES | NO | NO | NO |

### PROBLEM 2 (10 points)

Draw a DFA that accepts a string if and only if every occurrence of $a$ is immediately followed by at least two consecutive $b$'s.

### PROBLEM 3 (32 points)

(a) Prove that any $n$-state DFA that accepts at least one string accepts a string of length $< n$.

(b) Prove that any $n$-state NFA that accepts at least one string accepts a string of length $< 2^n$.

(c) Prove that the set of Turing machines that accept at least one string is recognizable (r.e.).

(d) Prove that the set of Turing machines that accept at least one string is undecidable.

(a) Solution: *Let $M$ be an $n$-state DFA. Suppose that the shortest string accepted by $M$ was of length $\geq n$. Then on the computational path followed by $M$ on input $w$ some state would be repeated. Excising the segment between the state repetitions would produce a shorter string accepted by $M$, contradicting the minimality hypothesis.*

(b) Solution: *Perform the subset construction on an $n$-state NFA, obtaining a $2^n$-state DFA, and appeal to the previous result.*

(c) Solution: *A nondeterministic recognizer for this set takes the code of a TM and checks if it accepts a non-deterministically chosen string in $\Sigma^*$. Since the languages recognized by NDTMs are exactly the r.e. sets, this suffices. Alternatively, this language can be enumerated by trying all TMs on all possible inputs for all possible number of steps and dovetailing.*

(d) Solution: *This holds by Rice's theorem, since the class of nonempty r.e. sets is a nonempty proper subset of the class of all r.e. sets.*

PROBLEM 4 (18 points)

Write regular expressions for these languages or explain why it can't be done.

(a) All strings of the form $a^ib^j$ where $i$ is a multiple of 3 and $j$ is a multiple of 5

(b) All strings without consecutive $b$'s.

(a) $(aaa)^*(bbbbb)^*$

(b) $a^*(baa^*)^*(b \cup \varepsilon)$

**PLEASE TURN OVER**

PROBLEM 5 (21 points)

Which of the following are necessarily true for any languages $A$ and $B$? True/False only, no explanations needed.

(a) If $A$ is cofinite then $A$ is regular.

(b) If $A$ is regular then $A$ has countably many regular subsets.

(c) If $A$ is not recursive then $A$ has countably many non-recursive subsets.

(d) $(A \cup B) - B = A$

(e) $A^*$ is nonempty.

(f) $A^*$ is infinite.

(g) $(A^*)^* = A^*$

**Solutions**

(a) True

(b) True

(c) False

(d) False

(e) True

(f) False

(g) True

PROBLEM 6 (20 points)

(a) Write a context-free grammar that generates all regular expressions over $\{a, b\}$. (It's OK if it generates redundant parentheses.)

(b) For any language $L$, let $S(L)$ be the language $\{xyx^R : y \in L, x \in \Sigma^*\}$. Show that the class of context-free languages is closed under $S$.

(c) Show that the class of regular languages is not closed under $S$.

(d) Find a regular language such that $S(L)$ is regular.

**Solutions**

(a) You can be pretty generous with where they stick in extra parentheses:

$$S \to \epsilon \mid a \mid b \mid (S) \mid (S \circ S) \mid (S^*) \mid (S \cup S) \mid \varnothing$$

(b) Given a context-free grammar $G$, let's show that $S(L(G))$ is context-free. Let $S_G$ be the start state of the grammar $G$. Then a grammar that generates $S(L(G))$ is:

$$\begin{aligned} S &\to aSa \mid bSb \mid S_G \\ S_G &\to \dots \end{aligned}$$

In $xyx^R$, $x, x^R$ can be generated by the $aSa, aBa$ rules. The $\epsilon$ case is also handled when $S$ generates only $S_G$.

(c) Let $L$ be the regular language $\{\epsilon\}$. In $xyx^R$, it must be the case that $y = \epsilon$, in which case $\{xx^R, x \in \Sigma^*\}$ is the language of even palindromes, which we know not to be regular.

(d) Either $L = \emptyset$ or $L = \Sigma^*$ works.

PROBLEM 7 (30 points)

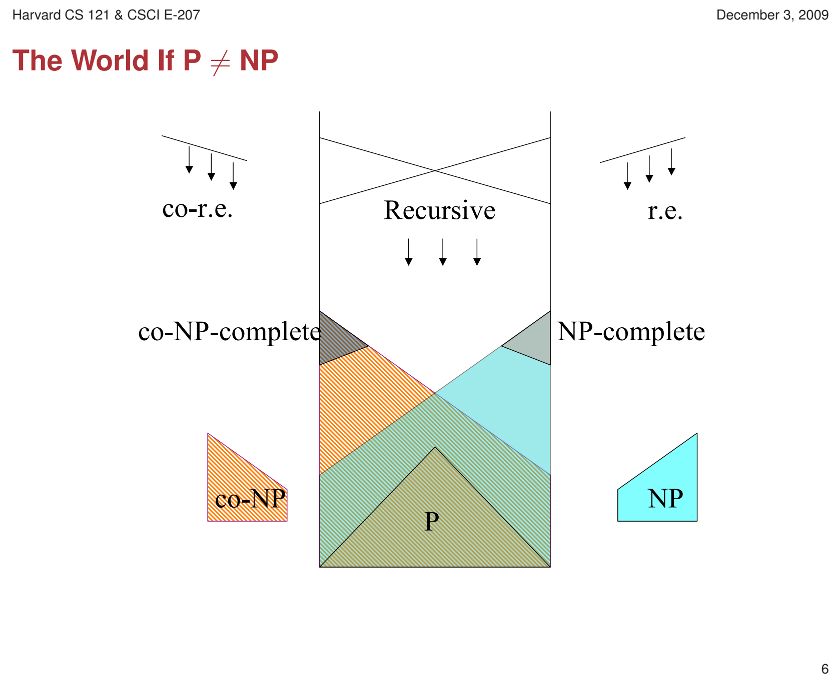
(a) Define NP-complete.

(b) Draw a diagram illustrating the likely relations between P, NP, co-NP, and the NP-complete, recursive, r.e., and co-r.e. sets on the hypothesis that P≠NP.

(c) Can an NP-complete set be polynomial-time reducible to a proper subset of itself? Explain.

**Solutions**

(a) A language $L$ is $\mathcal{NP}$-complete if it is in $\mathcal{NP}$ and for any $L' \in \mathcal{NP}$, $L' \leq_p L$.




(b)

(c) Yes. For example, 3SAT is $\mathcal{NP}$-complete, so SAT reduces to it, and of course 3SAT $\subseteq$ SAT.

PROBLEM 8 (30 points)

(a) Let SatFive be the set of Boolean formulas that are satisfied by a truth-assignment in which at most 5 of the Boolean variables are true. Explain why SatFive $\in$ P.

(b) Explain why, if P = NP, it would be possible not only to determine in polynomial time whether a formula is satisfiable but, if so, to find a satisfying truth-assignment.

(c) Let Independent Set be the set of all $\langle G, k \rangle$ for which undirected graph $G$ has a set $S$ of $k$ vertices such that no pair of vertices in $S$ is connected by an edge in $G$. Prove that Independent Set is NP-complete. (Hint: Reduce from Clique.)

(a) Solution: *To decide* SatFive *in polynomial time, one can just search through the set of satisfying assignments in which at most 5 of the Boolean variables are true, since there are only* $\binom{n}{5} + \binom{n}{4} + \binom{n}{3} + \binom{n}{2} + \binom{n}{1} + \binom{n}{0})$ *such assignments, which is quintic in* $n$.

(b) Solution: *If* $\mathcal{P} = \mathcal{NP}$, *there is a decider* $M$ *for* Sat *that runs in less than* $p(n)$*-time for some polynomial* $p$. *Suppose* $\varphi(v_1, \ldots, v_n)$ *is a satisfiable formula and* $a_1, \ldots, a_k$ *is an assignment such that* $\varphi(a_1, \ldots, a_k, v_{k+1}, \ldots, v_n)$ *has a satisfying assignment. Then by running* $M$ *on* $\varphi(a_1, \ldots, a_k, T, v_{k+2}, \ldots, v_n)$ *and* $\varphi(a_1, \ldots, a_k, F, v_{k+2}, \ldots, v_n)$, *we find in less that* $2p(|\varphi|)$*-time a value* $a_{k+1} \in \{T, F\}$ *such that* $\varphi(a_1, \ldots, a_k, a_{k+1}, v_{k+2}, \ldots, v_n)$ *has a satisfying assingment. Therefore in* $O(np(|\varphi|))$*-time one can find a satisfying assignment to* $\varphi(v_1, \ldots, v_n)$.

(c) Solution: $G$ *has a clique of size* $k$ *if only if the complement* $\overline{G}$ *of* $G$ *has an independent set of size* $k$. *Hence the map* $\langle G, k \rangle \mapsto \langle \overline{G}, k \rangle$ *is a polynomial reduction from* Clique *to* Independent Set *and vice versa. It follows that* Clique $\equiv_p$ Independent Set *and hence* Independent Set *is NP-complete.*

**THE END**