Open book, open notes, closed neighbors, 48 minutes. The exam totals 80 pts., subdivided as shown. Please work only on these sheets—note that the blank page 3 can be used for problems 1 and 2. Show your work—this may help for partial credit. Do Problems (1), (2), and your choice of (3a) XOR (3b). You must indicate clearly which of the latter two problems you are attempting.

Notation: Unless otherwise specified, all machines and languages use alphabet $\Sigma = \{0, 1\}$. TAUT denotes the language of Boolean formulas that are tautologies, which is pretty much the same as the complement of SAT. The complement of a language $A$ is denoted by $\overline{A}$ or by $\sim A$. One technical definition of an LBA (deterministic or nondeterministic) is a single-tape TM that receives its input $x$ in the form $\mathcal{A}x\mathcal{S}$ where $\mathcal{A}, \mathcal{S}$ are special endmarker characters, and every instruction that reads $\mathcal{A}$ (respectively, $\mathcal{S}$) leaves it alone and moves right (respectively, left). The problems begin overleaf.
Define $A = \{ (x, N) : N$ is an NFA with alphabet $\Sigma = \{0, 1\}$ and $x \in L(N) \}$. Show that $A$ belongs to the complexity class $\text{NL}$. A sufficient demonstration takes $n = |x|$ and $k = $ the number of instructions in $N$ and sketches an NTM $N'$ that accepts $A$ in space $O(\log(n + k))$ like so: Draw the input tape and worktapes (in this case, at least two worktapes) and what they hold, then give a few sentences of prose and/or pseudocode in the manner of lectures to say how $N'$ works.

Notes: The standard name for $A$ is $A_{\text{NFA}}$, which is the language of the acceptance problem for NFAs. As a warmup or for partial credit, you can try the easier language $\text{NE}_{\text{NFA}} = \{ (N) : N$ is an NFA and $L(N) \neq \emptyset \}$. For up to 5 points exam extra credit, state and justify the polynomial exponent of the algorithm that results from applying the proof of $\text{NL} \subseteq \text{P}$ to your $N'$. 
(2) (5 × 5 = 25 pts.) True/False/Unknown with justifications. Each is worth 3 pts. plus 2 for the justification.

(a) For every \( A \subseteq \Sigma^* \) there is a computable function \( f \) such that for all \( x \in \Sigma^* \), \( x \in A \iff f(x) \in \overline{A} \).

(b) Every language in NP is decidable.

(c) There are languages in NP that cannot be decided in deterministic \( O(\log n) \) space.

(d) There are languages in DSPACE[\( O(n) \)] that cannot be decided in deterministic \( O(\log n) \) space.

(e) If TAUT \( \leq^p_m \) SAT then NP = co-NP.
Do **Exactly One** of the following two problems:

**(3a) (28 pts. total)**

Define $B$ to be the language \{ $M : M$ is a DLBA and $L(M) \neq \emptyset$ \}. Prove that $B$ is undecidable, using a many-one reduction from a problem shown to be undecidable in the course. Also answer: is $B$ computably enumerable?

**(3b) (28 pts. total)**

Define a **serial** regular expression to be a concatenation of $n$ terms $t_1, \ldots, t_i, \ldots, t_n$, where each term $t_i$ is either 0, 1, or $(0 + 1)$. The last means that we don’t care what the $i$-th bit is. Prove by reduction from 3SAT that the following problem is \textbf{NP}-complete:

**Instance:** A collection $r_1, \ldots, r_m$ of serial regular expressions, each formed from the same number $n$ of terms.

**Question:** Is there a string $x \in \{0, 1\}^n$ that fails to match every $r_j$, $1 \leq j \leq m$?

**End of Exam.**