Open book, open notes, closed neighbors, 170 minutes. The exam has six problems and totals 240 pts., subdivided as shown. Show your work—this may help for partial credit. Please write in the exam books only.

Notation: The names of complexity classes are either completely standard or explained in the questions themselves—although these two classes are not expressly referenced, it may still help you to clarify \( E = \text{DTIME}[2^{O(n)}] \) and \( \text{EXP} = \text{DTIME}[2^{\alpha(n)}] \). The alphabet \( \Sigma \) over which languages are encoded is immaterial; you are always welcome to consider \( \Sigma = \{0, 1\} \) or \( \Sigma = \{0, 1, \#\} \). The alphabet \( \Gamma \) used as the worktape alphabet of Turing machines may, however, be much larger. The tupling notation \( \langle x, y \rangle \) or \( \langle x, y, z \rangle \) may be considered either as representing the strings \( x\#y \) and \( x\#y\#z \) or as the application of pairing functions as in some other sources; in no problem does the difference really matter. As usual the notation “\( \#0(x) \)” means “the number of 0s in the string \( x \)” and so on. Always “\( x \leq y \)” refers to the standard “lex” order on strings, which is the same as the usual order on corresponding natural numbers.

You may cite any major relevant theorem or fact or definition covered in the course without needing to give a justification (unless one is specifically asked for). Some parts refer to “current knowledge” in complexity theory.

Time Guideline: Up to 60 minutes for Problem (1), 20 minutes for (2), 30 for (3), 15 minutes for (4), 20 minutes for (5), and 25 minutes for (6).

(1) (60 pts.)

Classify each of the following languages \( L_1, \ldots, L_{12} \) according to the classification on the next page. Please write your answers in this form: if \( L_{13} \) were the language of the Halting Problem, you would write “13. i” or “13. (i)”, perhaps adding the words “c.e. but undecidable” to guard against silly errors. The classes and languages are on the next page. No justifications are needed, but may help for partial credit. There is a unique best answer for each language, and some answer(s) may be unused. Unless otherwise specified, “\( M \)” or “\( M_1 \)” etc. refers to a deterministic Turing machine.
(a) regular;
(b) in deterministic logspace but not regular;
(c) in NL and not known to be—or believed not to be—in deterministic logspace;
(d) in P and not known to be—or believed not to be—in NL;
(e) in NP and strongly believed not to belong to co-NP;
(f) in co-NP and strongly believed not to belong to NP;
(g) in PSPACE and strongly believed not to belong to NP nor to co-NP;
(h) decidable but known to be not in PSPACE;
(i) c.e. but not decidable;
(j) not c.e.

The languages are:

1. $L_1 = \{ \langle M, 0^r \rangle : M \text{ does not accept } \langle M, 0^r \rangle, \ r \geq 0 \}$. 
2. $L_2 = \{ \langle M, 0^r \rangle : M \text{ does not accept } \langle M, 0^r \rangle \text{ using only } 2^r \text{ of its own tape cells} \}$. 
3. $L_3 = \{ \langle C_1, C_2 \rangle : C_1, C_2 \text{ are Boolean circuits with the same number } n \text{ of inputs and } C_1(0^n) = C_2(0^n) \}$. 
4. $L_4 = \{ \langle C_1, C_2 \rangle : C_1, C_2 \text{ are Boolean circuits with the same number } n \text{ of inputs and } (\forall x \in \{0, 1\}^n) C_1(x) = C_2(x) \}$. 
5. $L_5 = \{ M : M \text{ is a DFA that accepts the string } 0^n1^n \text{ where } M \text{ has } n \text{ states} \}$. 
6. $L_6 = \{ M : M \text{ is a DFA that has no path from any final state back to its start state} \}$. 
7. $L_7 = \{ 0^m1^n : m + n = 6 \}$. 
8. $L_8 = \{ 0^m1^n : m - n = 6 \}$. 
9. $L_9 = \{ \langle G \rangle : G \text{ is 4-colorable} \}$. 
10. $L_{10} = \{ \langle M_1, M_2 \rangle : L(M_1) \cup L(M_2) \neq \emptyset \}$. 
11. $L_{11} = \{ \langle M_1, M_2 \rangle : L(M_1) \cup L(M_2) = \Sigma^* \}$. 
12. $L_{12} = \{ M\#u : M \text{ is a DFA that accepts some string beginning with } u \}$. 
Recall the two-qubit quantum circuit used to illustrate entanglement in lecture: Hadamard gate on line 1 followed by CNOT with control on line 1 and target on line 2, with $x = 00$ as basis-state input.

(a) First flip the CNOT gate “upside down” so that its control is on line 2 and target is on line 1. Are the qubits still entangled? Show your work using either matrices or “maze diagrams.”

(b) Instead of flipping the CNOT upside down, append a third gate: Hadamard on line 2 at right. Now are the qubits entangled? Show the amplitudes for each of the four possible outcomes.

(3) (36 pts. total)
Show that the following decision problem is \textbf{NP}-complete, by reduction from 3SAT and also using fewer nodes than the reduction to Graph 3-Coloring from class:

\textbf{ECONOMICAL 3-COLORING}

\textbf{INSTANCE}: An undirected graph $G$, an integer $k \geq 1$.

\textbf{QUESTION}: Is it possible to assign colors \textcolor{red}{R}, \textcolor{green}{G}, \textcolor{blue}{B} (red, green, blue) to the nodes of $G$ so that no adjacent nodes have the same color and B (blue) is used at most $k$ times?

(4) (36 pts.) \textit{True-False with reasons}.

Please write out the words \textbf{true} and \textbf{false} in full, for 3 points, and for the other 3 points, write a relevant justification (need not be a full proof, and should be brief).

(a) \textbf{RE} has a complete language under polynomial-time many-one reductions.

(b) On current knowledge, \textbf{NSPACE}[O(n)] is a proper subclass of \textbf{DSPACE}[n^3].

(c) On current knowledge, \textbf{NSPACE}[O(n)] is a proper subclass of \textbf{P}.

(d) The difference of two regular languages is always regular.

(e) The language TQBF is hard for \textbf{NP} under logspace many-one reductions.

(f) If BQP = NL then $P = NL$. 
(5) (18 + 12 = 30 pts.)

Given any \( k \), consider the following decision problem:

**Nonempty Acyclic NFA Intersection**

**Instance:** NFAs \( N_1, \ldots, N_k \) where each \( N_j \) has no cycles in its directed graph of possible transitions.

**Question:** Is \( L(N_1) \cap L(N_2) \cap \cdots \cap L(N_k) \neq \emptyset \)?

You may, if you wish, suppose that all the NFAs have the same number \( m \) of states which are broken into the same number \( \ell + 1 \) of levels, with the start state as the only state in level 0, accepting states only in level \( \ell \), and all arcs from states in one level going to states in the next level. Last, you may assume they have no \( \epsilon \)-arcs, so that \( L(N_j) \) is a subset of \( \{0, 1\}^\ell \). But none of this is essential.

(a) First let \( k \) be fixed—it is OK to think of \( k = 3 \) here. Give a polynomial-time algorithm to solve the problem. (**Hint:** Even though the machines are nondeterministic, it may still help to think of \( k \)-tuples of states and graphs with those tuples as nodes.)

(b) Now let \( k \) vary—i.e., the given instance \( \langle N_1, \ldots, N_k \rangle \) can have any number of NFAs. Does the problem still belong to \( \text{NP} \)? (**Note:** we are asking about \( \text{NP} \) not \( \text{P} \) so it’s not like part (a).)

(6) (42 pts.)

One of the following two decision problems is decidable, the other undecidable. Say which is which, and prove your answers. Here the Turing machines \( M \) not only have read-only input tape, but also the input head cannot move left (it can stay).

(a) **Instance:** A Turing machine \( M \).

**Question:** Does there exist an integer \( c > 0 \) such that for all inputs \( x \), \( M(x) \) uses no more than \( c \) worktape cells?

(b) **Instance:** A Turing machine \( M \), and an integer \( c > 0 \).

**Question:** Is it the case that for all inputs \( x \), \( M(x) \) uses no more than \( c \) worktape cells?

End of Exam.