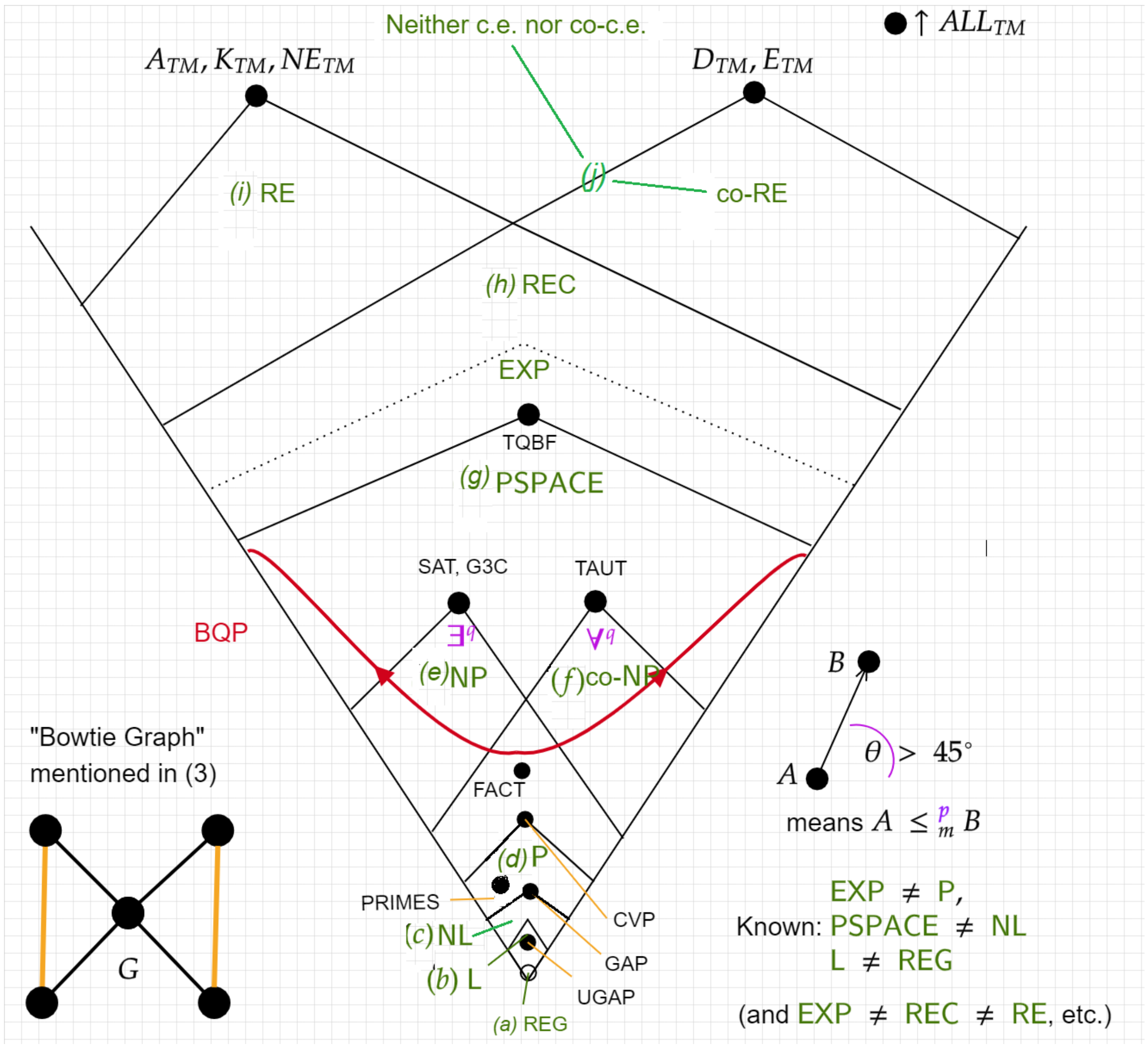
CSE491/596 **Final Exam** **Dec. 17, 2021**

Open book, open notes, closed neighbors, 170 minutes. The exam has six problems and totals 240 pts., subdivided as shown. Problem (6) involves a *choice*: (6a) XOR (6b). *Show your work*—this may help for partial credit.

Notation: As always $\mathsf{E} = \mathsf{DTIME}[2^{O(n)}]$, while $\mathsf{EXP} = \mathsf{DTIME}[2^{n^{O(1)}}]$ and $\mathsf{NEXP} = \mathsf{NTIME}[2^{n^{O(1)}}]$, and the names of other complexity classes are either completely standard or explained in the questions themselves. The alphabet Σ over which languages are encoded is immaterial; you are always welcome to consider $\Sigma = \{0,1\}$ or $\Sigma = \{0,1,\#\}$. The alphabet Γ 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 giving the strings $x\#y$ and $x\#y\#z$ or as the application of pairing functions. The choice of tupling scheme is not intended to matter, and any language using $\langle\cdot,\cdot\rangle$ notation may be assumed nonregular unless it is $\emptyset$ or Σ^*.

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). The following diagram conveys "current knowledge" in complexity theory and is referenced by problems (1) and (3):



(1) (12 $\times$ 5 = 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 could write "13. i" or "13. (i)"—but *even better*, add 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 specified as a DFA, "M" or "M_1" etc. refers to a deterministic Turing machine, "C" to a Boolean circuit, and "G" to an undirected graph.

(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:

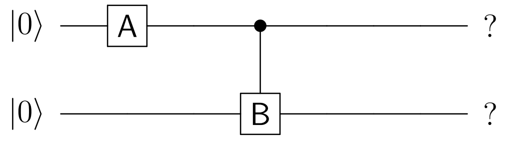
1. $L_1 = \{\, \langle M, 0^r \rangle : M$ does not accept $\langle M, 0^r \rangle,\ \ r \geq 0 \,\}$.
2. $L_2 = \{\, \langle M, 0^r \rangle : M$ does not accept $\langle M, 0^r \rangle$ using only 2^r of its own tape cells$\,\}$.
3. $L_3 = \{\, \langle C, x \rangle : n = |x| \wedge C$ is a Boolean circuit with n inputs $\wedge\ C(x) = C(0^n) \,\}$.
4. $L_4 = \{\, \langle C, x \rangle : n = |x| \wedge C$ is a Boolean circuit with n inputs $\wedge\ (\exists y) C(x) = C(y) \,\}$.
5. $L_5 = \{M : M$ is a DFA that accepts the string $0^n 1^n$ where M has n states$\}$.
6. $L_6 = \{M : M$ is a DFA that has no path from any final state back to its start state$\}$.
7. $L_7 = \{\, 0^m 1^n : m + n = 6 \,\}$.
8. $L_8 = \{\, 0^m 1^n : m - n = 6 \,\}$.
9. $L_9 = \{\, \langle G \rangle : G$ is not 3-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. $L_{12} = \{M\#u : M$ is a DFA that accepts some string beginning with the prefix string $u\}$.

(2) (36 pts. total)

(a) Define $\mathsf{A} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$. Show that A is unitary.

(b) Now calculate $\mathsf{B} = \mathbf{A}^2$. Then write controlled-$\mathsf{B} = \begin{bmatrix} \mathsf{I} & 0 \\ 0 & \mathsf{B} \end{bmatrix}$ as a 4×4 matrix.

(c) Now consider the following quantum circuit:



Here the second gate is controlled-B. Calculate the output of the circuit (on input $|00\rangle$ as shown) and say whether the resulting two-qubit quantum state is entangled.

(3) (36 pts.)

Consider the following decision problem "ARROWS":

INSTANCE: An undirected graph $G = (V, E)$ in which some edges are called "heavy."

QUESTION: Can each edge be given an arrow so that:

- In a heavy edge (u, v), if its arrow points at v then all other edges involving v have arrows pointing out of v and all other edges involving u have arrows pointing into u.
- In a heavy edge (u, v), if its arrow points at u then all other edges involving u have arrows pointing out of u and all other edges involving v have arrows pointing into v.
- All (other) nodes have at least one arrow pointing in to them.

For instance, if the orange edges in the bowtie graph on the front page are "heavy," then the answer is an easy "yes" no matter how you orient the arrows on the heavy edges: the other node will always have two arrows in and two arrows out. (Well, that node is like the doubly-trivial 4CNF clause $(x_1 \vee \bar{x}_1 \vee x_2 \vee \bar{x}_2)$—this is an oblique hint to get you started.)

(a) Show that the problem is NP-complete. For hardness, you must use a mapping reduction from 3SAT. (36 pts.)

(b) For optional *offsetting credit* (not extra credit), say what happens if we simplify the problem so that there is no "heavy edge" distinction and the QUESTION is simply this:

- Can each edge be given an arrow so that each node has either exactly one arrow coming in or exactly one arrow going out?

Can you show this simpler problem to be NP-hard using a variant form of (3)SAT? Does the proof of correctness really work? (Note that the bowtie graph, now with no orange edges, becomes an instance where the answer is *no*.)

(4) (36 pts.) *True-False with reasons.*

Please write out the words `true` and `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) RE has a complete language under polynomial-time many-one reductions.

(b) On current knowledge, $\mathsf{NSPACE}[O(n)]$ is a proper subclass of $\mathsf{DSPACE}[n^4]$.

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

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

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

(f) If $\mathsf{BQP} = \mathsf{NL}$ then $\mathsf{P} = \mathsf{NL}$.

(5) (42 pts. total)

One of these two languages L is regular; the other is not. For the non-regular one, give a proof via the Myhill-Nerode Theorem. For the regular one, design a DFA M such that $L(M) = L \cdot (01)^*$.

$$\begin{aligned} L_1 &= \{\, x0y : \#0(x) > \#0(y) \,\} \\ L_2 &= \{\, x1y : \#0(x) > \#0(y) \,\} \end{aligned}$$

(6) (30 pts.) Do EXACTLY ONE of the following two problems.

(6a) Give in pseudocode a *polynomial-time* decision procedure for the following problem:

INSTANCE: Three DFAs M_A, M_B, M_C, each with the same number n of states.

QUESTION: Is $L(M_A) \subseteq L(M_B) \subseteq L(M_C)$?

Your pseudocode must be detailed enough to estimate the running time of your procedure as $\tilde{O}(n^k)$ for some constant k, where the tilde means you may ignore factors of $\log n$. You need not optimize k, but it must be clear from your algorithm.

XOR

(6b) Define $B = \{\, \langle M \rangle : M \text{ does not accept any palindrome} \,\}$. Show by reduction from D_{TM} that B is not c.e. (If you wish, you may work with the complements of B and D_{TM} in your reduction instead.) *Also answer and briefly justify*: is B co-c.e.?

END OF EXAM.