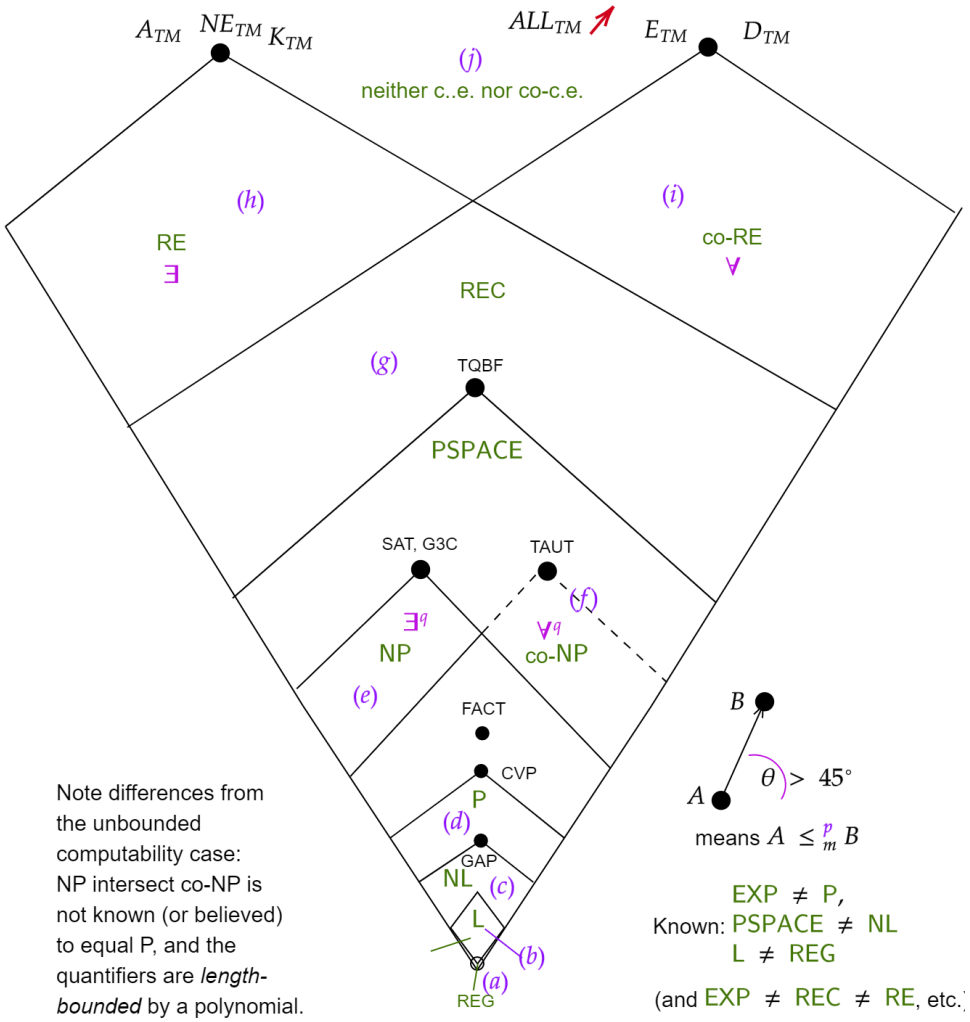
CSE491/596 **Final Exam** **Dec. 16, 2022**

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 $\Sigma = \{0, 1\}$. The alphabet Γ used as the work-tape 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 problem (1):



(1) ($12 \times 4 = 48$ 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, "G" to an undirected graph, and $x, y, \ldots$ to ordinary strings. Multiples of 3 can be zero or negative.

(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 PSPACE and strongly believed not to belong to NP

(g) decidable but known to be not in PSPACE;

(h) c.e. but not decidable;

(i) co-c.e. but not c.e.

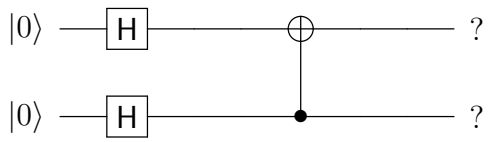
(j) neither c.e. nor co-c.e.

The languages are:

1. $L_1 = \{ \langle M, x, y \rangle : M \text{ accepts both } x \text{ and } y \}$.
2. $L_2 = \{ \langle M, x, y \rangle : M \text{ accepts at least one of } x \text{ or } y \}$.
3. $L_3 = \{ \langle M, x, y \rangle : M \text{ accepts } x \text{ but does not accept } y \}$.
4. $L_4 = \{ \langle M, x, y \rangle : M \text{ accepts neither } x \text{ nor } y \}$.
5. $L_5 = \{\langle M, x, y \rangle : M \text{ is a DFA that accepts at least one of } x \text{ or } y \}$.
6. $L_6 = \{\langle N, x, y \rangle : N \text{ is an NFA that accepts at least one of } x \text{ or } y \}$.
7. $L_7 = \{\langle C, x, y \rangle : C \text{ is a circuit with } n = |x| = |y| \text{ inputs that accepts both } x \text{ and } y \}$.
8. $L_8 = \{\langle C, x, y \rangle : C \text{ is a circuit with } n = |x| = |y| \text{ inputs that accepts some string besides } x \text{ and } y \}$.
9. $L_9 = \{ \langle G \rangle : G \text{ is not 2-colorable} \}$.
10. $L_{10} = \{ \langle G \rangle : G \text{ is not 4-colorable} \}$.
11. $L_{11} = \{0^m 1^n : m + n \text{ is a multiple of 3}\}$.
12. $L_{12} = \{0^m 1^n : m - n \text{ is a multiple of 3}\}$.

(2) (9 + 3 + 18 + 6 = 36 pts. total)

Consider the following quantum circuit C, noting that the CNOT gate is "upside down":



(a) Calculate the 4×4 unitary matrix U that gives the action of this circuit.

(b) Does the vector $u = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$ stand for $|00\rangle$, $|01\rangle$, $|10\rangle$, $|11\rangle$, or some combination of these?

(c) Let v be output of U on the input vector u. Is v entangled? If you say yes, prove it; if you say no, give a representation of v as a tensor product of single-qubit states.

(d) Where could you add one more Hadamard gate H to C to make a circuit C' such that $C'u$ is entangled? Note that if you answer "entangled" for (c), you want to add H so as not to disturb that property.

(3) (12 + 30 = 42 pts. total)

Consider the following decision problem

EDGE-JOINT PATHS
INSTANCE: A directed graph $G = (V, E)$ and nodes $s_1, s_2, t_1, t_2 \in V$.
QUESTION: Do there exist a path $(s_1 = u_0, u_1, u_2, \ldots, u_{q-1}, u_q = t_1)$ and a path $(s_2 = v_0, v_1, v_2, \ldots, v_{r-1}, v_r = t_2)$ such that every node v_j in the second path has an edge (u_i, v_j) from some node u_i in the first path? (The paths need not have the same length, i.e., $q \neq r$ is allowed.)

(a) Explain why the following prose sketch **fails** to demonstrate that the problem belongs to NL: "Design an NTM N with the same log-sized worktapes holding s_1, s_2, t_1, t_2 and the current nodes u_i and v_j of two paths that it guesses as before. Picture N as moving a queen along the first path and a king along the second path. Initially the king starts on $v_0 = s_2$ and the we need to find a node u_i along the first path so that (u_i, v_0) is an edge. So N guesses the nodes of the first path and moves the queen on each guess until there is an edge (u_i, v_0) (if none, the computation rejects), which intuitively means the queen checks the king. Then the king moves to a new guessed vertex v_1 along the edge (v_0, v_1) and we repeat the process: each time we move the king to a new v_j, we move the queen back or forth along the guessed path until it reaches a path vertex u_i such that $(u_i, v_j) \in E$. Finally, if the king reaches t_2 and (after checking the king there) the queen can move to t_1, N accepts."

(b) Prove that this problem is NP-complete, using a reduction from 3SAT .

(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) If f many-one reduces a language B to a language C, and $A \subseteq B$ and $C \subseteq D$, then f many-one reduces A to D as well.

(b) On current knowledge, it is unknown whether $\mathsf{P} \neq \mathsf{NL}$ or $\mathsf{P} \neq \mathsf{PSPACE}$, but it is currently known that at least one of those two inequalities must hold.

(c) If a computation uses quadratic time, then it must use at least a linear amount of space.

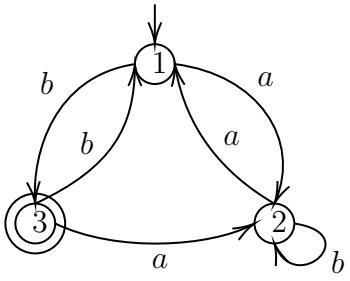
(d) The difference of two decidable languages is always decidable.

(e) On current knowledge, $\mathsf{DSPACE}[O(n)]$ is closed downward under polynomial time many-one reductions.

(f) The language $\{ \langle M, x \rangle : M$ is a deterministic Turing machine and M accepts $x \}$ is complete for the class of r.e. languages under *log-space* many-one reductions.

(5) (9 + 12 + 3 + 3 + 6 + 12 = 45 pts.)

Consider the following DFA $M = (Q, \Sigma, \delta, s, F)$ with $Q = \{1, 2, 3\}$, $\Sigma = \{a, b\}$, $s = 1$, $F = \{3\}$, and $\delta = \{(1, a, 2), (1, b, 3), (2, a, 1), (2, b, 2), (3, a, 2), (3, b, 1)$. It is pictured below:



(a) Design an NFA N' so that $L(N') = L(M)^R$, which is the reversal of $L(M)$ as defined on assignments 3–5. Be sure to show the start and final state(s) of your N' clearly.

(b) Convert N' into an equivalent DFA M'.

(c) Is there a string $u \in \Sigma^*$ that N' cannot cannot process? Give a shortest u if so.

(d) Is there a $v \in \Sigma^*$ so that for every $w \in \Sigma^*$, N' accepts vw? Give a shortest v if so.

(e) Is there a $v' \in \Sigma^*$ so that for every w', N' can process $v'w'$? Give a shortesrt v' if so.

(f) Write a regular expression either for $L(M)$ or for $L(M)^R$—your choice.

(6) (24 + 9 = 33 pts.) Do EXACTLY ONE of the following two problems.

(6a) Prove that the language $\{ \langle M \rangle : L(M) \cap \text{PAL} = \emptyset \}$ is not c.e. Here PAL stands for the language of palindromes over $\Sigma = \{0, 1\}$ and M stands for a deterministic Turing machine. Also answer and justify briefly: is the language co-c.e.?

XOR

(6b) Over $\Sigma = \{0, 1\}$, define A to be the set of strings w that can be broken as $w = uv$ such that $\#0(u) = \#0(v)$ AND $\#1(u) = \#1(v)$. For example, 0011100111 belongs to A because breaking in the middle balances both counts, but 0011001111 does not belong because the possible balance points for the count of 0s differ from the only balance point that works for 1s. Also answer and explain briefly: why does the language A' defined by replacing AND with OR become regular?

END OF EXAM.