Computer Science 320 Practice Final Exam
NAME: ______________________________

All questions to be answered on this exam. Please indicate if you are using the back of a page for your answer.
90 marks

Terminology:
Bal = the language over $\{), (\}$ of balanced strings of parentheses.
$A^nB^n = \{a^nb^n : n \geq 0\}$
$\text{Prime}_\Sigma = \{w \in \Sigma^* : |w| \text{ is prime }\}$
$H = \{< M, w >$: Turing Machine M halts on string $w\}$.
M_H is of a Turing Machine that semi-decides H. If M_H exists, its encoding is $< M_H >$.
VertexCover(G, k) is true iff G has a set of k **or fewer** vertices such that every edge in G has at least one end in the set.
Clique(G, k) is true iff G has a set of k **or more** pairwise-adjacent vertices (that is, a set $V' \subseteq V$ where $|V'| \geq k$ and for all $u \in V'$ and $v \in V'$ where $u \neq v$, (u, v) is an edge in G.)
$A_{TM} = \{< M, w > \mid M$ is a Turing Machine which accepts the string $w\}$.
$E_{TM} = \{< M > \mid M$ is a Turing Machine which accepts no strings$\}$.
$EQ_{TM} = \{< M_1, M_2 > \mid M_1$ and M_2 are Turing Machines that accept the same language$\}$.
The Pumping Lemma: If L is a regular language then there is a pumping constant p for L such that, for all strings w where $|w| \geq p$, w can be divided into three parts, $w = xyz$, where: (a) $|y| \geq 1$, (b) $|xy| \leq p$, and (c) for all $i \geq 0$, $xy^iz \in L$.

1. (9 marks) Answer True or False, by placing a T or F beside the statement. Answer on this sheet.

   T There is a deterministic TM that decides membership in the empty language

   F There is a deterministic TM that decides membership in $\{< M >$: M is a TM that accepts the empty language $\}$

   T There is a deterministic TM that decides membership in $NoBigClique = \{< G, k >$: G is a graph, and k is an integer, and G has **no** clique of size k **or more** $\}$.

   T There is a polynomial time deterministic TM that decides membership in $SmallClique = \{< G, k >$: G is a graph, and k is a positive integer, and G has a clique of size k **or less** $\}$.

   F There are an uncountably infinite number of regular languages

   countable, since reg. expressions are countable

   F If $L_1 \cup L_2$ is context-free then it must be the case that at least one of L_1 and L_2 are context-free.

   T If a PDA were modified so it had a queue in place of its stack, the result would be Turing-equivalent to a Turing Machine.

   T There are a countably infinite number of languages over any given (finite) alphabet.

|w| is odd, and

Covered in class

2. (5 marks) Give a PDA for the language $L = \{w \in \{a,b\}^* :$ if the middle symbol of w is a, then w must be a palindrome, i.e, the latter half of w is the reverse of the first half $\}$. Note that this language is the **union** of $\{waw^r : w \in \{a,b\}^*\}$ and $\{w : w$ has odd length and the middle symbol in w is $b\}$.

"

3. (5 marks) Give a Context-Free Grammar for the Kleene-star closure of the language in the previous question.

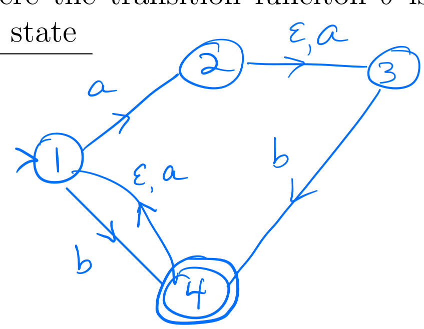
See end notes

4. (5 marks) Let $\bigoplus$ denote "exclusive or" for languages: that is, $L_1 \bigoplus L_2 = \{w : w \in L_1$ or $w \in L_2$ but not both $\}$. Prove **constructively** that the class of regular languages are closed under $\bigoplus$ by doing the following: given that L_1 and L_2 are regular languages with FAs M_1 and M_2, respectively, show that $L \bigoplus L_2$ is also regular by giving a construction for the FA for $L_1 \bigoplus L_2$.

5. (4 marks) Find the equivalent deterministic FA for the DFA below, using the method given in class. Include all useful states, and no states that are not useful (i.e., not reachable from the start state).

$M = (\{q_1, q_2, q_3, q_4\}, \delta, q_1, \{q_4\})$, where the transition funciton δ is given by the table below:

state	symbol	destination state
q_1	a	q_2
q_1	b	q_4
q_2	ϵ	q_3
q_2	a	q_3
q_3	b	q_4
q_4	ϵ	q_1
q_4	a	q_1



6. (8 marks) Prove that the language $\{w \in \{a,b\}^* :$ the number of a's in w is an integer multiple of the number of b's $\}$ is not regular. Zero is an integer multiple of all integers.

7. (5 marks) Give a CFG for the language $\{a^i b^j c^{i+j+k} d^k : i, j, k \geq 0\}$.

8. (5 marks) Give a PDA for the same language as in the previous question.

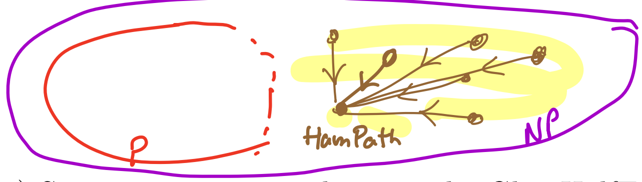
9. Recall: "integers" includes both positive and negative integers, and zero.

   (a) (3 marks) Show that the integers are countably infinite.

   (b) (3 marks) Show that the ordered pairs of integers (not necessarily positive) are countably infinite.

10. (5 marks) Consider the following language: **StateUse** $= \{< M_1, q, w > | M$ is a Turing Machine which contains state q, and and M enters state q when run on input w.$\}$

Is StateUse decidable? Prove your answer.



11. (10 marks) Suppose you are working on the GlassHalfEmpty problem. You have shown that it is in NP, and you want to see whether it there is a polynomial time algorithm to solve it. You can only make reductions to or from the problems GlassHalfEmpty, Clique ~~VertexCover~~, HamiltonPath, and GraphConnectivity, where GraphConnectivity is the problem of determining whether an input undirected graph G is connected, that is, for every pair of vertices a, b in the graph, there is a path between a and b. Recall that "P1 reduces to P2 by a polynomial-time mapping" is written "P1 $\le_P$ P2".

Below are all the candidate relations **(not all are true)**. Beside each relation,
...write **TRUE** if the statement is known to be true
...write **FALSE** if the statement is known to be false
...write **P=NP** if the statement implies that P=NP
...write **P $\neq$ NP** if the statement implies that P$\neq$NP
...write *GHE*$\in$ **P** if the statement implies that GlassHalfEmpty is in P
...write *GHE*$\notin$ **P** if the statement implies that GlassHalfEmpty is not in P
...write *GHE* $\in$ **NP-c** if the statement implies that GlassHalfEmpty is NP-complete
...write *GHE* $\notin$ **NP-c** if the statement implies that GlassHalfEmpty is not NP-complete

Clique ~~VertexCover~~ $\le_P$ HamiltonPath (NP-c) TRUE

Clique ~~VertexCover~~ $\le_P$ GraphConnectivity (∈P) P = NP, ⇒ GHE ∈ P

Clique ~~VertexCover~~ $\le_P$ GlassHalfEmpty GHE ∈ NP-c

GlassHalfEmpty $\le_P$ Clique ~~VertexCover~~ TRUE

GlassHalfEmpty $\le_P$ GraphConnectivity GHE ∈ P

GlassHalfEmpty $\le_P$ HamiltonPath TRUE

GraphConnectivity $\le_P$ Clique ~~VertexCover~~ TRUE

GraphConnectivity $\le_P$ HamiltonPath TRUE

GraphConnectivity $\le_P$ GlassHalfEmpty TRUE

HamiltonPath $\le_P$ VertexCover TRUE

HamiltonPath $\le_P$ GraphConnectivity P = NP ⇒ GHE ∈ P

HamiltonPath $\le_P$ GlassHalfEmpty GHE ∈ NP-c

12. Suppose L is a language for which there exists a Printer-TM.

(a) (3 marks) What can we conclude about L: is it polynomially decidable, decidable, recognizable ~~semidecidable~~, or undecidable? Breifly explain why.

(b) (5 marks) Suppose further that there is a Turing computable function that, for every string w, returns a number r such "if w is in L then the Printer-TM prints it out within the first r strings." What can we now conclude about L: is it polynomially decidable, decidable, semidecidable, or undecidable? Breifly explain why.

13. (5 marks) In this question, we look at two ways of constraining the problem SET PARTITION. Recall that SET PARTITION is the following problem:

    Problem SET PARTITION:
    Given: a multiset of integers $S = \{x_1, x_2, \ldots, x_n\}$
    Decide: Is there a way to partition S so that both partitions sum to the same amount?

    We showed this problem was NP-c, by reducing SUBSET SUM to it.

    NP-completeness results are made even stronger by adding constraints to the problem. For example, it has been proved that: "SUBSET SUM remains NP-c even if all the set elements are positive integers. I.e., POSITIVE SUBSET SUM is NP-complete." Certain constraints, however, will render the problem polynomial.

    (a) (2 marks) Prove that ALL EQUAL SET PARTITION is polynomial, where the problem is:

        Problem ALL EQUAL SET PARTITION:
        Given: a multiset of integers $S = \{x_1, x_2, \ldots, x_n\}$ where all x_i are equal
        Decide: Is there a way to partition S so that both partitions sum to the same amount?

    (b) (5 marks) Prove that POSITIVE SET PARTITION is NP-c, where the problem is:
        Problem POSITIVE SET PARTITION:
        Given: a multiset of integers $S = \{x_1, x_2, \ldots, x_n\}$ where all x_is are positive
        Decide: Is there a way to partition S so that both partitions sum to the same amount?
        For full marks, you should include indications as to why the reduction works (that is, returns the correct answer) and why you believe it is poly-time.

14. **Problem:** HAMPATH
    Given: An undirected graph G with vertex set V and edge set E, where $|V| = n$ and $|E| = m$.
    Decide: Is there a path H in G that visits each vertex exactly once? (Such a path corresponds to a permutation $v_1, v_2, \ldots, v_n$ of the vertices such that $(v_i, v_{i+1}) \in E$ for all i where $1 \le i < n$.)
    Problem: HAMCYC
    Given: An undirected graph G with vertex set V and edge set E, where $|V| = n$ and $|E| = m$.
    Decide: Is there a cycle C in G that visits each vertex exactly once? (Such a cycle corresponds to a permutation $v_1, v_2, \ldots, v_n$ of the vertices such that $(v_i, v_{i+1}) \in E$ for all i where $1 \le i < n$; and such that $(v_n, v_1) \in E$.)

(a) (4 marks) Prove that HAMPATH $\leq_P$ HAMCYC. Briefly explain why the reduction is **correct**, and why it is polynomial time.

(b) (4 marks) Prove that HAMCYC $\leq_P$ HAMPATH. Briefly explain why the reduction is **correct**, and why it is polynomial time.

$\exists$ a printer-TM P_L for L,

and a F where $F(\omega) = n_\omega$

where - if $\omega \in L$ then P_L prints it at or before string# n_ω.

A decider X for L:

$X =$ " on input $\langle \omega \rangle$ where ω is a string:

1. Run $F(\omega)$, get n_ω.
2. Run P_L. Wait for n_ω strings to be printed. If any are ω, ACCEPT.
   If none are ω, REJECT. "

$$(x_1 \vee \overline{x}_2 \vee x_3 \vee \overline{x}_4 \vee x_5 \vee x_6)$$

$$(x_1 \vee \overline{x}_2 \vee A) \wedge (\overline{A} \vee x_3 \vee B) \wedge (\overline{B} \vee \overline{x}_4 \vee C) \wedge (\overline{C} \vee x_5 \vee x_6)$$

Q4. Prove Class of RL is closed under XOR.

Proof:

Let L_1 and L_2 be any two RLs, and let Σ be their (combined) alphabet.

Let M_1 and M_2 be DFAs for L_1 and L_2, respectively, where:

$$M_1 = (P, \Sigma, \delta_1, p_1, F_1)$$

$$M_2 = (Q, \Sigma, \delta_2, q_1, F_2)$$

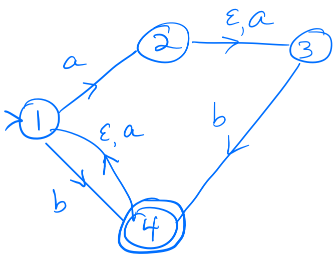
and WLOG assume $P \cap Q = \emptyset$.

Then a FA for $L = L_1 \otimes L_2$ is:

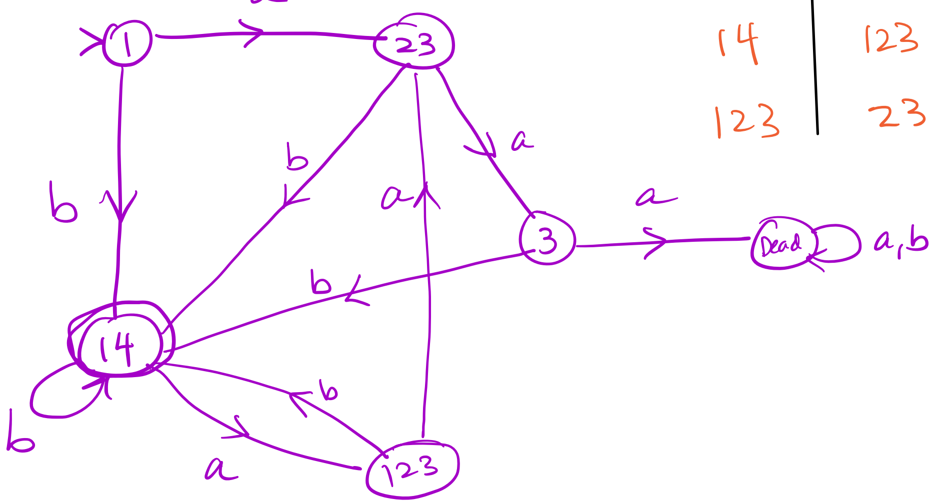
$$M = (P \times Q, \Sigma, \delta, (p_1, q_1), F_1 \times (Q \setminus F_2) \cup (P \setminus F_1) \times F_2 \}),$$

where $\delta((p,q),\sigma) = (\delta_1(p,\sigma), \delta_2(q,\sigma))$

Q5:



	a	b
> 1	23	14
2	3	∅
3	∅	14
4	1	∅
23	3	14
14	123	14
123	23	14



Q6 $\#_a$'s is integer multiple of $\#_b$'s.

Not regular.

Proof: Suppose L is regular; let its pumping constant be denoted p.

Consider the string $w = a^{p+1} b^{p+1}$

$w \in L$ and $|w| \geq p$, so by P.L.

$a^{p+1} b^{p+1} = xyz$ where

① $|xy| \leq p$

② $|y| > 0$

③ $xy^i z \in L \quad \forall i \in \mathbb{W}$

Consequently (from ① + ②) $y = a^t$ for some t, $1 \leq t \leq p$

Hence (from the above and ③)

$a^{p+1-t} \, b^{p+1} \in L$.

But $p+1-t$ is an integer between 1 and p, and is therefore not an integer multiple of $p+1$ $\Rightarrow\Leftarrow$

$\therefore$ L is not regular.

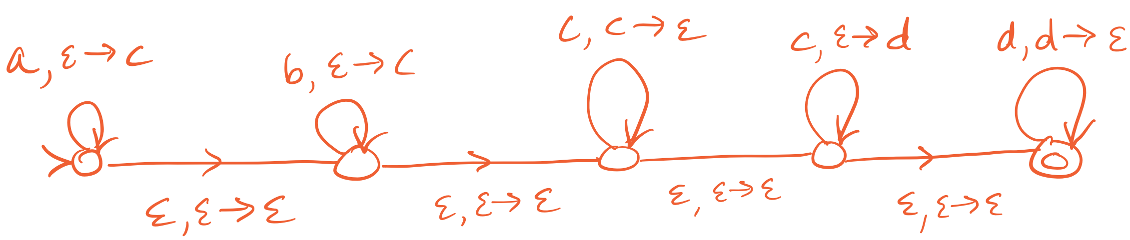
Q7. $\{ a^i b^j c^{i+j+k} d^k \mid i, j, k \in \mathbb{W} \}$.

Q8.

$S \to AD$ $\quad$ $D \to cDd \mid \varepsilon$

$A \to aAc \mid B$

$B \to bBc \mid \varepsilon$



$\text{StateUse} = \{ \langle M, q, w \rangle \mid M$ is a TM that enters state q during computation on input $w \}$.

Claim: $A_{TM} \leq_{TM}$ StateUse.

Proof: Suppose SU is a decider for State Use.
Then we can construct a decider X for A_{TM} as follows:

$X =$ " on input $\langle M, w \rangle$ where M is a TM:

1. Run SU on input $\langle M, h_a, w \rangle$, where h_a is M's "halt-accept" state.
   - if SU accepts, ACCEPT.
   - if SU rejects, REJECT. "

Since SU is a decider, so is X.

X decides A_{TM}, because:

M accepts w iff M enters state h_a at some point when run on w, iff SU accepts $\langle M, h_a, w \rangle$. $\blacksquare$

Q12 a) L is recognizable, because if we have an enumerator E for L, we can construct a recognizer R for L as follows:

R = "on input ⟨w⟩ :
1. Run E, and if E ever prints out w, ACCEPT."

b) If ∃ a TM Rank(w) that computes the function given, and E is the enumerator of L.
we can construct a decider R for L as follows:

R = "on input ⟨w⟩ :
1. Let r = Rank(w)
2. Run E.
   - if E prints out w, ACCEPT.
   - if E prints out r strings without printing w, REJECT."

R decides L.

Q13. We did not cover Set-Partition, but I could ask:

"Prove that SetPartition is $\in NP$."

$$\text{SetPartition} = \{ S \mid S = \{x_1, x_2, \dots, x_n\},\ x_i \in \mathbb{W}\ \forall i,\ \text{and}\ \exists X \subseteq S \text{ such that } \sum_{x_i \in X} x_i = \sum_{x_i \in S \setminus X} x_i \}.$$

To show SetPartition $\in NP$:

The certificate is the set X

In poly-time we:

- Sum elements of X
- Sum elements of $S \setminus X$
- Compare the two to determine they are equal
- ACCEPT if equal REJECT otherwise.

Q14 a, b — we have done these in class.