CSE 350 – Honors Theory of Computation **Midterm Exam**
Michael A. Bender **April 5, 2018**

Name: ______________________ ID #: ______________________

INSTRUCTIONS:

- Except for your proofs, your answers should be at most 1 or 2 sentences (excluding work.)
- This is a closed book, closed notes exam.
- Check to see that you have 12 pages including this cover and scratch pages.
- Read all the problems before starting work.
- Think before you write.
- If you leave a question blank or write just "I do not know," you get 25
- Good luck!!

Academic integrity is expected of all students at all times, whether in the presence or absence of members of the faculty.

Understanding this, I declare that I shall not give, use, or receive unauthorized aid in this examination. I have been warned that if I am caught cheating (either receiving or giving unauthorized aid) I will get a "Q" grade for this course, and a letter will be sent to the Committee on Academic Standing and Appeals (CASA) requesting that an academic dishonesty notation be placed on my transcript. Further action against me may also be taken.

Signature[1] ______________________

Problem	Score	Maximum
your signature		2
1		32
2		16
3		22
4		14
5		14
extra credit		10
Total		100

[1] No "I dunno" points for leaving this blank. ☺

Pumping Lemmas[2]

Pumping Lemma for Regular Languages (Weak version):
If L is infinite and regular, then there exist strings x, y, and z, where $y \neq \varepsilon$, such that for each $n \geq 0$, $xy^nz \in L$.

Pumping Lemma for Regular Languages (Strong version):
If L is infinite and regular, then there exists a constant k, such that for any string $\omega \in L$ with $|\omega| \geq k$, $\omega = xyz$ where:

1. $|xy| \leq k$,
2. $|y| > 0$, and
3. for all $n \geq 0$, $xy^nz \in L$.

[2] Wasn't it nice of your prof and TAs to add this page? ☺

Problem 1. (32 points)

Please give a one-sentence justification for each question revealing your thinking.

(a) **T** **F** If Σ is a finite set, then Σ^* is countable.

True. If we let each element in Σ map to an integer, every word in Σ^* will be a concatenation of integers, mapping to an element in $\mathbb{N}$.

(b) **T** **F** If L and L_1 are regular, and $L = L_1 \cup L_2$, then L_2 is regular.

False. If L_2 is the union of the complement of L_1 and a nonregular language, $L_1 \cup L_2 = \Sigma^*$, which is regular.

(c) **T** **F** If n is the length of a string and m is the length of the pattern, then the KMP string matching algorithm runs in $O(n + m)$ steps.

True. Every iteration of KMP increments either the progress reading through the input string by 1 or the prograss reading through the pattern by 1.

(d) **T** **F** The intersection of two uncountably infinite languages could be countably infinite.

True. We can consider the languages representing real numbers in the ranges $\mathbb{N} \cup [0, 1]$ and $\mathbb{N} \cup [1, 2]$. The intersection would be the language of strings representing natural numbers, which map to their numeric forms.

(e) **T** **F** The language of strings in which the substrings “ab” and “ba” appear the same number of times is regular.

True.There are a finite number of equivalence classes since the number of "ab", "ba" only differ by at most 1 assuming $\Sigma = \{a, b\}$ since any consecutive repetition of one string will generate an instance of the other.

(f) **T** **F** If there exists $w = xyz \in L$, such that for all $n \geq 0$, $xy^n z \in L$, then L is regular.

False. Let $L = xy^*z \cup a^n b^n$ - since $a^n b^n$ is not regular, the subset of the language recognized by that pattern is not representable by a DFA, meaning L is not regular.

(g) **T** **F** There are countably many regular languages on a fixed alphabet.

True. A regular language can be represented by a regular expression, and since the number of characters in a regular expression is finite and the size of the expression is finite, the number of regular expressions using an alphabet and the number of regular languages on it are countably infinite.

(h) **T** **F** In class we saw that if you take a homing sequence and tack extra characters onto the end, it stays a homing sequence. Suppose that you tack them onto the beginning instead. You still have a homing sequence.

True. Homing sequences work regardless of where one starts - it only matches a unique output to some final state and prepending characters to the beginning doesn't disrupt the mapping of unique outputs to final states.

Problem 2. (16 points)

Given below is a proof to show that all languages are regular. State very clearly and explicitly what is wrong with the following proof and why.

The proof is by induction on the size of the language.

Base cases: $n \in \{0, 1\}$

The empty set is a regular language, and any one-element set is a regular language. Therefore every language of size 0 or 1 is regular.

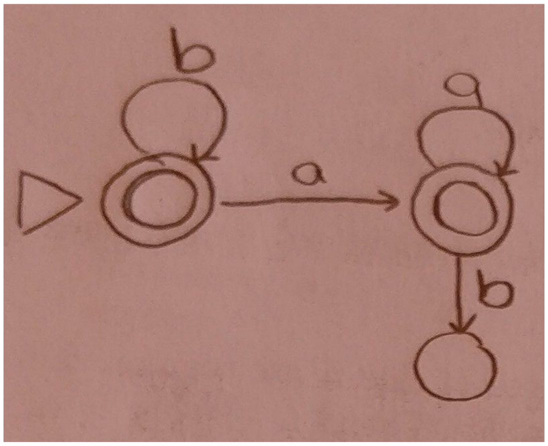
Inductive step: Assume that all languages of size n are regular. Let L be any language of size $n + 1$, and choose some string $w \in L$. Then $|L - \{w\}| = n$, so by the inductive hypothesis, $L - \{w\}$ is regular. Since $L = \{w\} \cup (L - \{w\})$ and regular languages are closed under union, L is regular. Since we assumed nothing about L other than its size, any language of size $n + 1$ is regular.

Therefore, by induction, all languages are regular.

The problem with this proof is that it does not take into account countably infinite languages where we can't simply assume a size $(\infty - 1)$ language is regular to prove a size ∞ language is also regular since both are essentially equivalent.

Problem 3. (22 points)

Let $L = b^*a^*$. Please draw a DFA for L below.



How many equivalence classes for L are there? _____3_____

Please list them below.

_________$[b^*], [bb * a], [\Sigma^* a \Sigma^* b \Sigma^*]$_________

Use your findings above to prove that any DFA for L must have at least two accepting states.

If we assume for sake of contradiction that L has fewer than 2 accepting states:

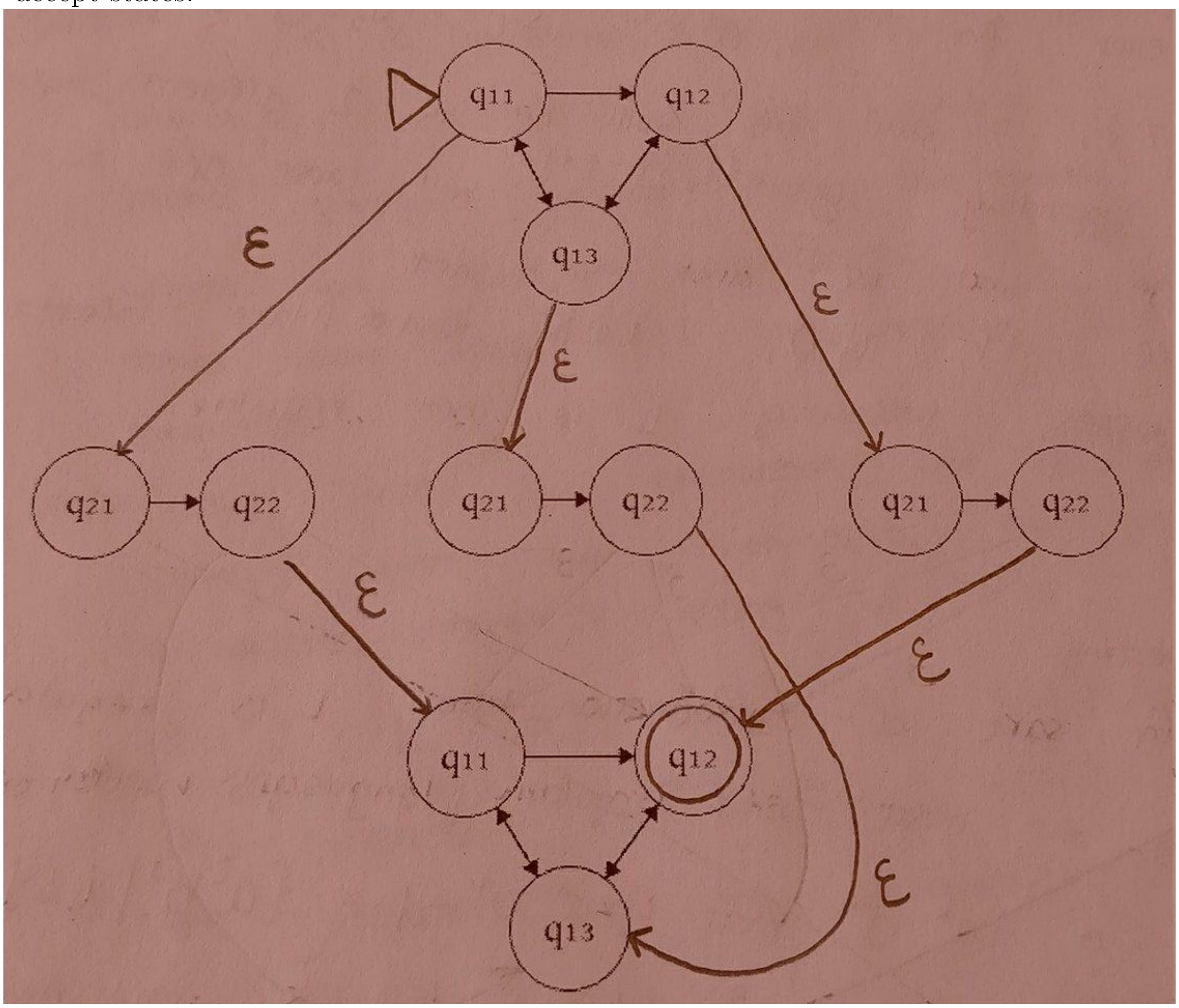
- there can't be 0 accepting states because at least one string (ba) is in the language
- If there is 1 state, the for any string accepted by the DFA, the same transition will bring both to the same state. However, 'b' and 'ba' are both in the language and should be part of the same state - appending 'b' to both yields $'bb' \in L$ and $'bab' \notin L$, a contradiction that proves the assumption false.

Problem 4. (14 points)
Let L_1 and L_2 be languages. Let $INSERT(L_1, L_2) = \{w = xyz \mid xz \in L_1 \text{ and } y \in L_2\}$.

Suppose that L_1 and L_2 are regular and have the following DFA's:

L_1 L_2

Below is a partial construction of an NDFA that recognizes $INSERT(L_1, L_2)$. Complete the construction by marking the start state, adding the missing ε-transitions, and circling any accept states. [3]



[3]It turns out you can generalize this procedure to work for any two DFAs. This means that regular languages are closed under the INSERT operation!

Problem 5. (14 points)

In class, we learned several methods to prove that a language is non-regular. Using **one of the three** methods given below, show that the language $L = \{a^n b^t | n \geq t\}$ is NOT regular. Indicate your choice by writing a checkmark on the line provided.

For each additional proof you provide, you will get 5 **points of extra credit**. Clearly indicate which proof method(s) you are using for extra credit by writing "EC" on the line provided.

_____ a) Distinguishability / Myhill-Nerode Theorem

By the Myhill Nerode Theorem, a language is regular if and only if there are finite equivalence classes.
However, for $w_1 = a^i b^t, i \geq t, w_2 = a^j b^t, j \geq t$, if $i \neq j$, w_1 and w_2 will be in 2 different equivalence classes since appending $b^{max\{i,j\}-t+1}$ will cause one to be in the language and the other to not be in the language.
Since for $t = 0, i, j \in \mathbb{N}$, there are infinite equivalence classes meaning L is not regular.

_____ b) Closure Properties

_____ c) Pumping Lemma (indicate which version you are using)

This will be a proof by contradiction.
We will assume for sake of contradiction that $L = \{a^n b^t | n \geq t\}$ is regular.
By the strong pumping lemma, there exists a value k for all strings w such that $|w| \geq k$, $w = xyz$, $|xy| \leq k$, $|y| > 0$, and $\forall n \geq 0, xy^n z \in L$ where $n \geq 0$.
Since k is a constant, we pick a string in the language, $w = a^k b^k$ Since $|w| > k$, by the strong pumping lemma, some nonempty substring occurring within the first k characters can be repeated n times where $n \geq 0$ and the string will be in the language.
Since the first k characters are 'a' and the pumping string must be nonzero, the pumping string is a nonzero, length-p string of 'a's. By the lemma, $xy^n z \in L$ for $n \geq 0$. So $xy^0 z \in L$. This means $a^{k-p} b^k \in L, p > 0$. This contradicts the definition of the language which states $n \geq t$.
So we have roven though contradiction L is not regular.

Scratch Paper

Scratch Paper

Scratch Paper