CMPS 130 Computational Models Spring 2017

Midterm 2, May 17, 2017

NAME KEY ID#

Question 1	10 points	
Question 2	10 points	
Question 3	10 points	
Question 4	10 points	
Question 5	10 points	
Question 6	10 points	
Question 7	20 points	
Question 8	20 points	
Question 9	10 points	
Question 10	20 points	
Question 11	10 points	
Question 12	10 points	

1. (10pts) How is an NFA different from a DFA?

TRANSITIONS
DFA FOR EACH STATE AND LETTER THERE MUST BE ONE AND ONLY ONE ARROW.
NFA FOR EACH STATE AND LETTER THERE MAY BE $\emptyset$, 1, OR MORE ARROWS

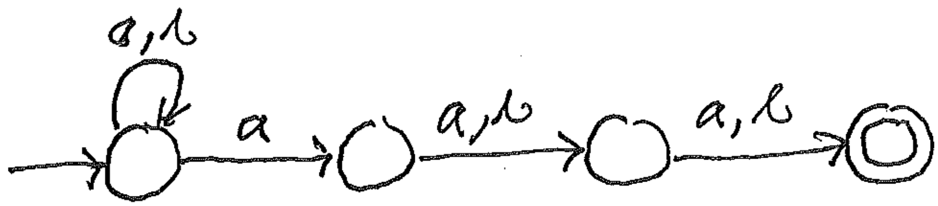
ACCEPTANCE
DFA ACCEPT IF STOPS IN AN ACCEPT STATE
NFA ACCEPT IF POSSIBLE TO END IN AN ACCEPT STATE

2. (10pts) What does the subset construction prove about the relative expressive power of NFAs vs DFAs?

THEY HAVE THE SAME EXPRESSIVE POWER

3. (10pts) Show the state diagram for an NFA with no more than four states that recognizes the language:

$$L_3 = \{x \in \{a, b\}^* \mid \text{ the third symbol from the right in } x \text{ is an } a.\}$$



4. (10pts) What is the minimum number of states required in a DFA that accepts the above language, L_3?

8

5. (10pts) Write the formal definition of regular expression as given in class.

ε IS A r.e.

$\emptyset$ IS A r.e.

a $\forall a \in \Sigma$ IS A r.e.

IF r_1 AND r_2 ARE r.e. THEN

$(r_1 + r_2)$ IS A r.e.

$(r_1 r_2)$ IS A r.e.

r_1^* IS A r.e.

6. (10pts) What does Kleene's Theorem state about the family of languages that can be defined with DFAs and the family of languages that can be denoted with regular expressions?

THEY ARE THE SAME FAMILY OF LANGUAGES

7. (20pts) Prove the first part of Kleene's Theorem. For all regular expressions there exists a DFA that accepts the language denoted by the regular expression. Use proof by induction on the length of the regular expression. You may assume that if an ϵ-NFA or NFA exists then an equivalent DFA exists.

KLEENE'S THEOREM

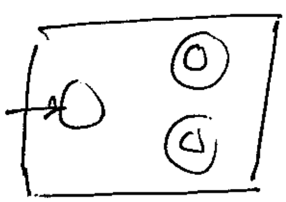
PART I

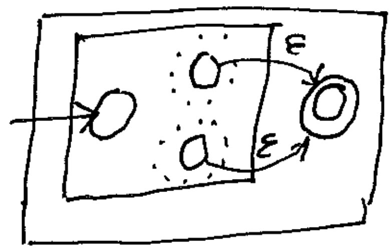
THM FOR EVERY REGULAR EXPRESSION THERE IS AN EQUIVALENT ε-NFA.

PROOF: USE FACT THAT R.E.S ARE RECURSIVELY DEFINED.

AND USE CONSTRUCTIVE INDUCTION TO GET AN EQUIVALENT ε-NFA.

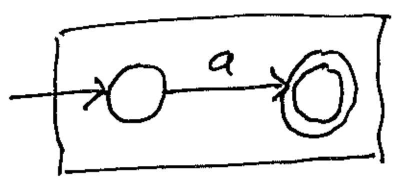
FOR CONVENIENCE ALL OUR FAs WILL HAVE A UNIQUE ACCEPTING STATE.
ANY ε-NFA CAN BE SO TRANSFORMED, SIMPLY ADD THE NEW UNIQUE ACCEPTING STATE AND ε MOVES FROM ALL THE OLD ACCEPTING STATE AND THEN MAKE THE OLD ACCEPTING STATES REJECTING STATES.



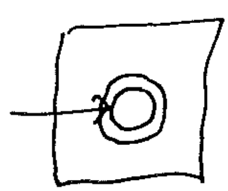


BASE M = 1 LENGTH OF R.E.

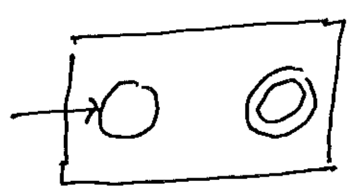
a WHERE $a \in \Sigma$



ε



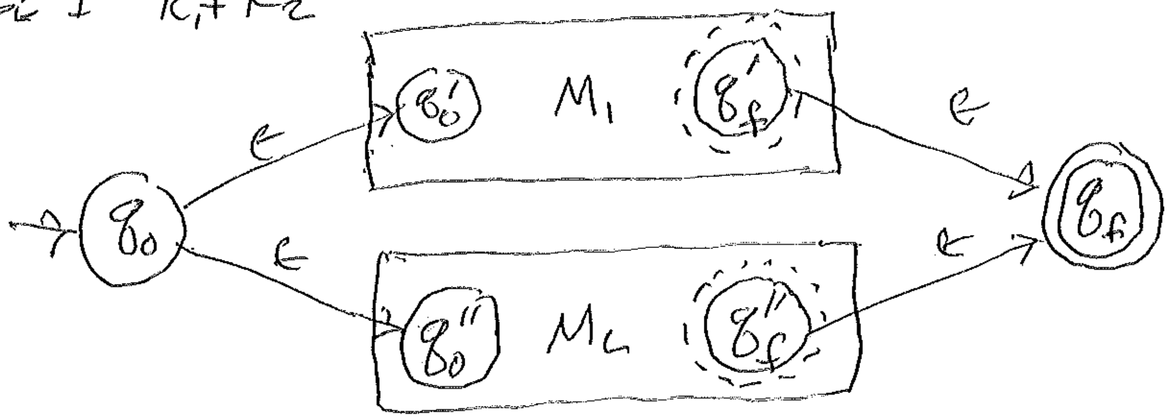
∅



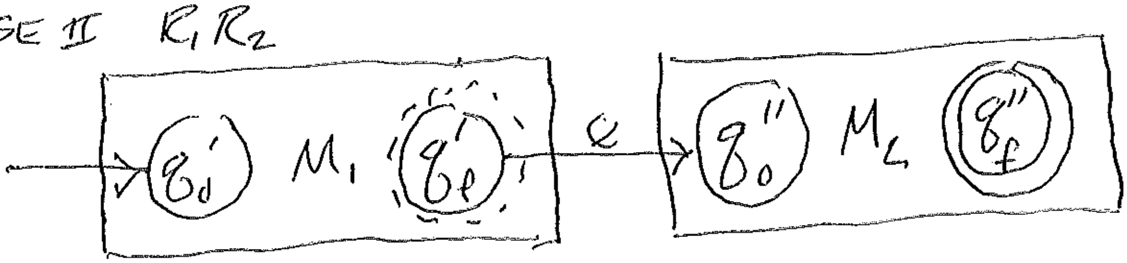
INDUCTIVE STEP IF R IS NOT A BASE CASE THEN R MUST BE $R_1 + R_2$ OR $R_1 R_2$ OR R_1^* WHERE R_1, R_2 ARE REG. EXPS WHOSE LENGTHS IS LESS THAN THE LENGTH OF R.

R_1 AND R_2 HAVE EQUIVALENT ϵ-NFAs BY THE I.H., SAY THEY ARE M_1 AND M_2

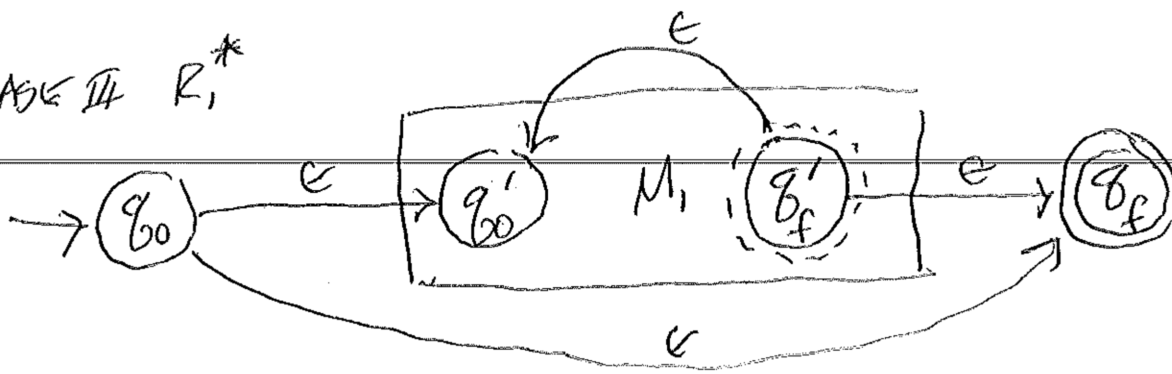
CASE I $R_1 + R_2$



CASE II $R_1 R_2$



CASE III R_1^*



SO WE HAVE BUILT ϵ-NFAs FOR ALL ~~FIVE~~ THREE CASES. AND THE INDUCTIVE STEP PROOF IS COMPLETED.

Q.E.D.

8. (20pts) Given: $\Sigma = \{a, b\}$, write clear regular expressions for the following languages:

Begins with a and has a length of at most three.

$a + a(a+b) + a(a+b)(a+b)$

Does not end with bb.

$\epsilon + a + b + (a+b)^* aa + (a+b)^* ab + (a+b)^* ba$

9. (10pts) (Fill in the blanks.) The basis of the pumping lemma is that for any REGULAR language, all strings in the language which are SUFFICIENTLY long must contain some non-void substring that can be pumped, where pumped means that if the string is modified by by removing the substring or arbitrarily repeating it the modified string will still be IN THE LANGUAGE

10. (20pts) Prove the language, $L = \{a^n b^n \mid n \geq 0\}$, is non-regular by showing you can win the daemon adversary game regardless of what legal choices the daemon makes.

p

$s = a^p b^p$ $s \in L$ AND $|s| = 2p \geq p$

$xyz = a^p b^p$

$\left.\begin{array}{l} |xy| \leq p \\ |y| \geq 1 \end{array}\right\} \Rightarrow y$ IS ONE OR MORE as

$i = 2$

$xy^{i}z \notin L$

BECAUSE MORE as THAN bs.

11. (10pts) We have previously proved by induction, that the *finite* union of regular languages is regular. In other words that for all $n \geq 2$

$$L_1 \cup L_2 \cup L_3 \cup \cdots \cup L_n$$

where all the L_i are regular languages must be regular.

Prove that the *infinite* union of regular languages is not necessarily regular. (Hint: a language with a finite number of strings, e.g., just one string is regular)

PROOF BY COUNTER EXAMPLE

$\{0^m 1^m \mid m \geq 0\}$ IS KNOWN TO BE NOT REGULAR.

AND IT IS THE INFINITE UNION OF

ONE STRING LANGUAGES, I.E., REGULAR LANGUAGES.

Q.E.D.

12. (10pts) Prove that the reverse of a non-regular language must be non-regular.

ASSUME TOWARD A CONTRADICTION THAT

L IS NOT REGULAR AND L^R IS REGULAR

L^R REGULAR $\Rightarrow (L^R)^R$ IS REGULAR

CLOSURE OF REVERSAL FOR REG. LANG.

$(L^R)^R = L$ SO WE HAVE

L^R REGULAR $\Rightarrow$ L IS REGULAR

A CONTRADICTION. Q.E.D.