UNIVERSITY OF TEXAS AT EL PASO

Department of Computer Science

CS 3350 Automata, Computability, and Formal Languages

Midterm Exam 1

Spring 2020

Name: _____

- This test is closed book and closed notes. Total time limit 1 1/3 hour (80 minutes).
- You can answer each question on the exam sheets. Use the back of the pages if needed.
- The main goal of most questions is to show that you know the corresponding algorithms; so if you are running out of time, just follow the few first steps of the corresponding algorithm.

Question	Points	Score
1	14	
2	10	
3	20	
4	20	
5	20	
Total:	84	

1. Consider the following finite automaton:

   $\text{FA}_1 = (Q, \Sigma, \delta, q_0, F)$ where

   1. $Q = \{q_1, q_2\}$
   2. $\Sigma = \{0, 1\}$
   3. $\delta =$

	0	1
q_1	q_1	q_2
q_2	q_2	q_2

   4. q_1 is the start state
   5. $F = \{q_2\}$

   (a) (5 points) Draw the diagram for this finite automaton.

   Solution:

   (b) (2 points) Trace step by step (indicate the result of each step) what this automaton will do on input 010.

   Solution: Start at state q_1.
   On input 0, stay at state q_1.
   On input 1, go to state q_2.
   On input 0, stay at state q_2.
   Accept string.

   (c) (2 points) Trace step by step what this automaton will do on input 000.

   Solution: Start at state q_1.
   On input 0, stay at state q_1.
   On input 0, stay at state q_1.
   On input 0, stay at state q_1.
   Reject string.

   (d) (5 points) What is the language accepted by this automaton?

   Solution: All strings that contains at least one 1.

2. Give an example of each of the following. If no example exists, answer "impossible" and give a short justification.

   (a) (2 points) A regular subset of $\{\varepsilon, ab, aabb, aaabbb, aaaabbbb\}$.

   Solution: Any subset of the language is regular. For example, $\{ab\}$.

   (b) (2 points) A string of length 9 that is not in the language described by the regular expression $(aaa^* \cup abb^* \cup bba^*)^*$.

   Solution: Any string of length 9 starting by ba would work. There are many other correct answers.

   (c) (2 points) A subset of the language 0^*1^* that is not regular.

   Solution: The expected answer is $\{0^n1^n \mid n \geq 0\}$.

   (d) (2 points) Two languages L_1 and L_2 such that $L_1 \cdot L_2$ has exactly 4 strings.

   Solution: Example: $L_1 = \{\varepsilon, a\}$, $L_2 = \{\varepsilon, b\}$.

   (e) (2 points) Two strings w_1 and w_2 such that $w_1w_2 = w_1$.

   Solution: $w_1 = a$, $w_2 = \varepsilon$. (Or any string for w_1, as long as w_2 is empty string.)

3. Let Y be a language defined on $\Sigma\{a, b\}$. Define the operator $\widehat{Y}$ to be the set of all strings obtained by removing the last symbol of strings of Y.

$$\widehat{Y} = \{w \mid \text{ there is a string } s \text{ and a symbol } a \in \Sigma \text{ such that } s \cdot a \in Y\}$$

So, if $aaba \in Y$, then $aab \in \widehat{Y}$. For example, if $Y = \{\varepsilon, ab, bbbaa, bbbaaa\}$, then $\widehat{Y} = \{a, bbba, bbbaa\}$. (There is no string issued by removing the last symbol of ε.)

For each language below, give what language we obtain by applying the $\widehat{\ }$ operator them.

(a) (2 points) $L_1 = \{\varepsilon, a, b, aa, aaa, aab, bbba\}$

Solution: $\{\varepsilon, a, aa, bbb\}$

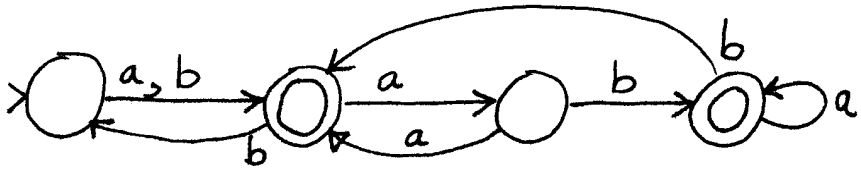
(b) (2 points) $L_2 = a^*bbbb^*$

Solution: a^*bbb^*

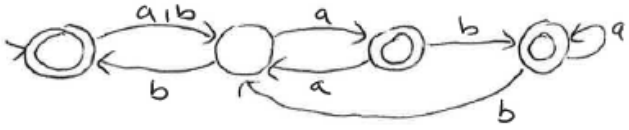
(c) (2 points) (more tricky) $L_3 = (a \cup b)^*aba(a \cup b)^*$

Solution: $(a \cup b)^*aba(a \cup b)^* \cup (a \cup b)^*ab$

(d) (6 points) If the following DFA accepts a language L, transform it into a DFA or NFA that accepts $\widehat{L}$. (Hint, consider changing which states are accepting and which states are rejecting.)



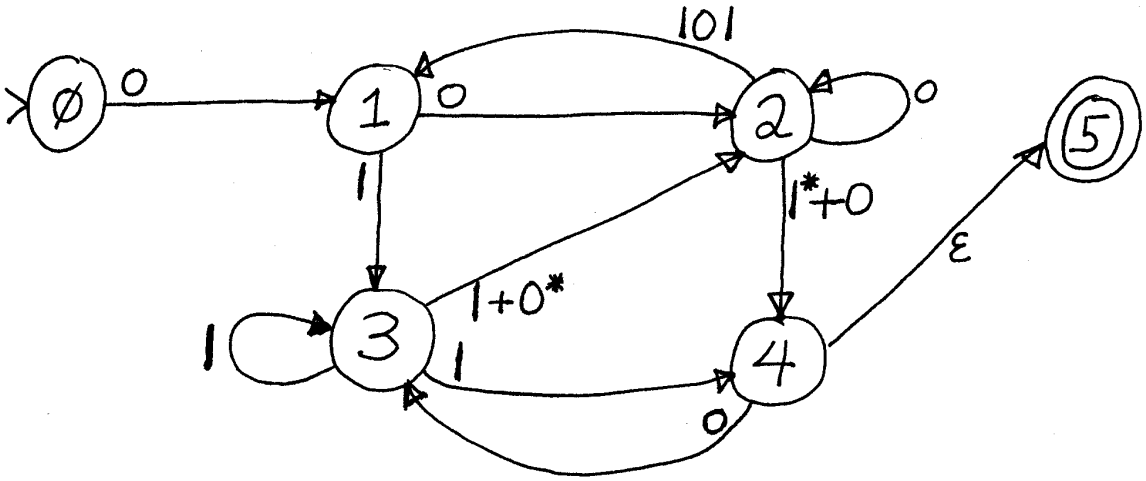
Solution:



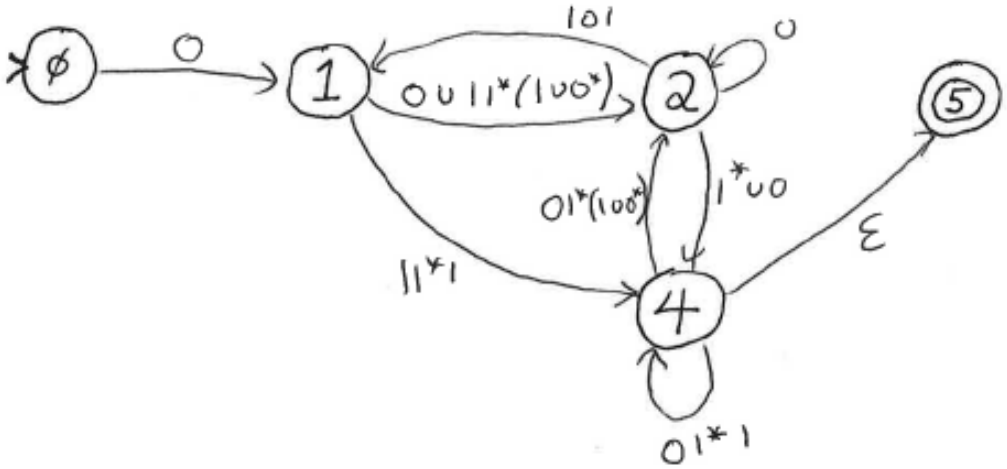
(e) (8 points) (10 pts) We want to show that regular languages are closed under the "$\widehat{\ }$" operator. To do this, we need a general construction to show that for any DFA M where $L(M) = Y$, we can construct an NFA accepting $\widehat{Y}$. Explain how to construct such a NFA starting from M.

Solution: Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA for Y. Construct $M' = (Q, \Sigma, \delta, q_0, F')$, where $q \in F'$ if and only if there is a letter $\sigma \in \Sigma$ such that $\delta(q, \sigma) \in F$. If M accepts string $w \cdot \sigma$ for $\sigma \in \Sigma$, in the computation, it will reach a state in F' just before reaching a state in F, and M' will accept w.

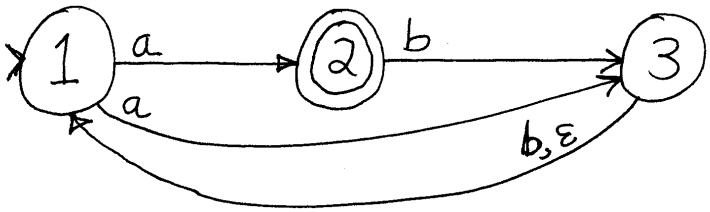
4. (20 points) During the algorithm for transforming a transition graph to a regular expression, we have reached the following GNFA. Perform one more step of the algorithm by removing state number 3 and updating the appropriate transitions. In this NFA, the + operator means ∪.



Solution:



5. (20 points) Below is an NFA for a language L. Construct a DFA for L.



Solution: