Theory of Computation, CSCI 438, Spring 2021
Final, April 22nd

Name _____

Definitions (10 pts.)

1. Give the definition of a deterministic Turing Machine which we used in class and is given in the text. Be sure to define each of the elements. (5 pts.)

A deterministic Turing Machines is defined by
$M=(Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$

- Special character _ indicates blank. The tape begins with the input on the left portion of the tape and the remaining portion of the tape is blank
- $\_ \notin \Sigma$
- $\{\_\} \cup \Sigma \subseteq \Gamma$
- $q_0, q_{accept}, q_{reject} \in Q$ and $q_{accept} \neq q_{reject}$
- $\delta: Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

(Definition 3.3, page 168)

2. Give the definition of what it means for a context-free grammar to be in Chomsky Normal which we used in class and is given in the text. Be sure to define each of the elements. (5 pts.)

A context-free grammar is in Chomsky normal form if every rule is of the form

$A \rightarrow BC$
$A \rightarrow \alpha$

Where α is any terminal and A, B, and C are any variables – except that B and C may not be the start variable. In addition allow the rule $S \rightarrow \varepsilon$, where S is the start variable.

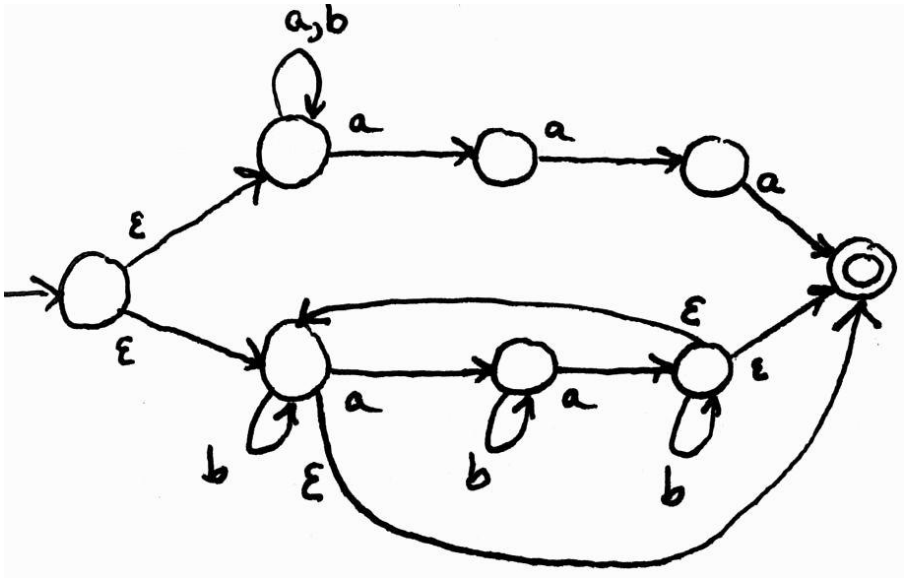
Short Answer (10 pts.)

3. Give a regular expression for all strings on the alphabet Σ={a,b} which end with three a's, or contain an even number of a's. (5 pts.)

[(a∪b)*aaa] ∪ [b*ab*ab*]*

4. Give an NFA for all strings on the alphabet Σ={a,b} which end with three a's, or contain an even number of a's. (5 pts.)

Sample answer:



Assimilation (20 pts.)

5. Describe what is meant by a Universal Turing machine. Include its inputs and outputs. Also describe the difference between deciders, recognizers, and a machine that is neither a decider nor a recognizer. Relate these concepts by telling if a Universal Turing machine is a decider, recognizer, or a machine that is neither a decider nor a recognizer.

   (20 pts.)

   A Universal Turing machine is a machine that takes as input the description of a Turing machine and a string in the language of that Turing machine. The Universal Turing machine will accept if the string is in the language of the Turing machine.

   A decider is a machine that always halts. It will halt in an accept state if the input is in the language described by the machine; otherwise it will reject.

   A recognizer is a machine which always accepts if the input is in the language described by the machine; however, if the input is not, this machine may either halt or loop forever.

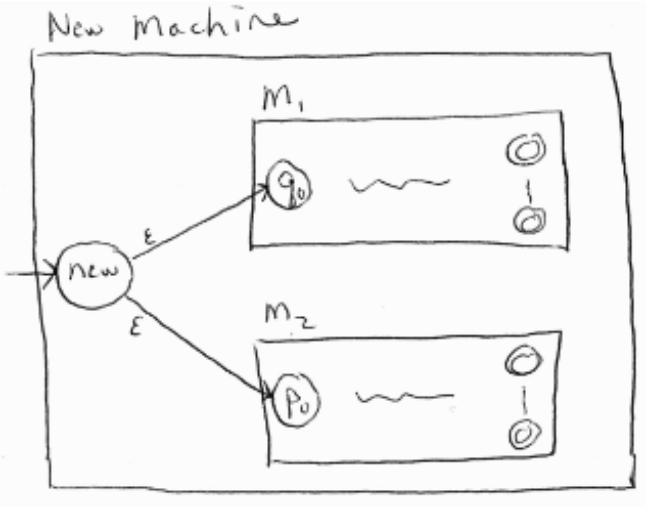
   A machine that is not either,

   We would like a Universal Turing machine to be a decider, but it is not. At least it is a recognizer.

Problem Solving /Proofs (60 pts.)

6. Prove that regular languages are closed under union by beginning with a DFA for each of the regular languages, and constructing an NFA for the union of the languages.

Begin by drawing a picture of what you plan to do. (5 pts.)



Write a formal definition of the construction. (5 pts.)

Let $L_1 = \mathcal{L}(M_1)$ where $M_1 = (Q, \sum, \delta_1, q_0, F_1)$ is a DFA and
$L_2 = \mathcal{L}(M_2)$ where $M_2 = (P, \sum, \delta_2, p_0, F_2)$ is a DFA.

Define the NFA

$M' = (Q \cup P \cup \{new\}, \sum, \delta', new, F_1 \cup F_2)$ where

$\delta'(q,a) =$

$\{\delta_1(q,a)\}$ for $q \in Q$ and $a \in \sum$

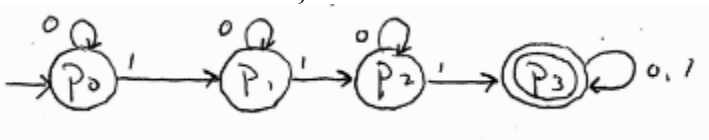
$\{\delta_2(p,a)\}$ for $p \in P$ and $a \in \sum$

$\{q_0, p_0\}$ for $q = new$ and $a = \varepsilon$

7. DFA's are defined below for the languages L_1 and L_2 on $\Sigma = \{0,1\}$.

   L_1 = {w | every other symbol in w, beginning with the second symbol, is a 1}

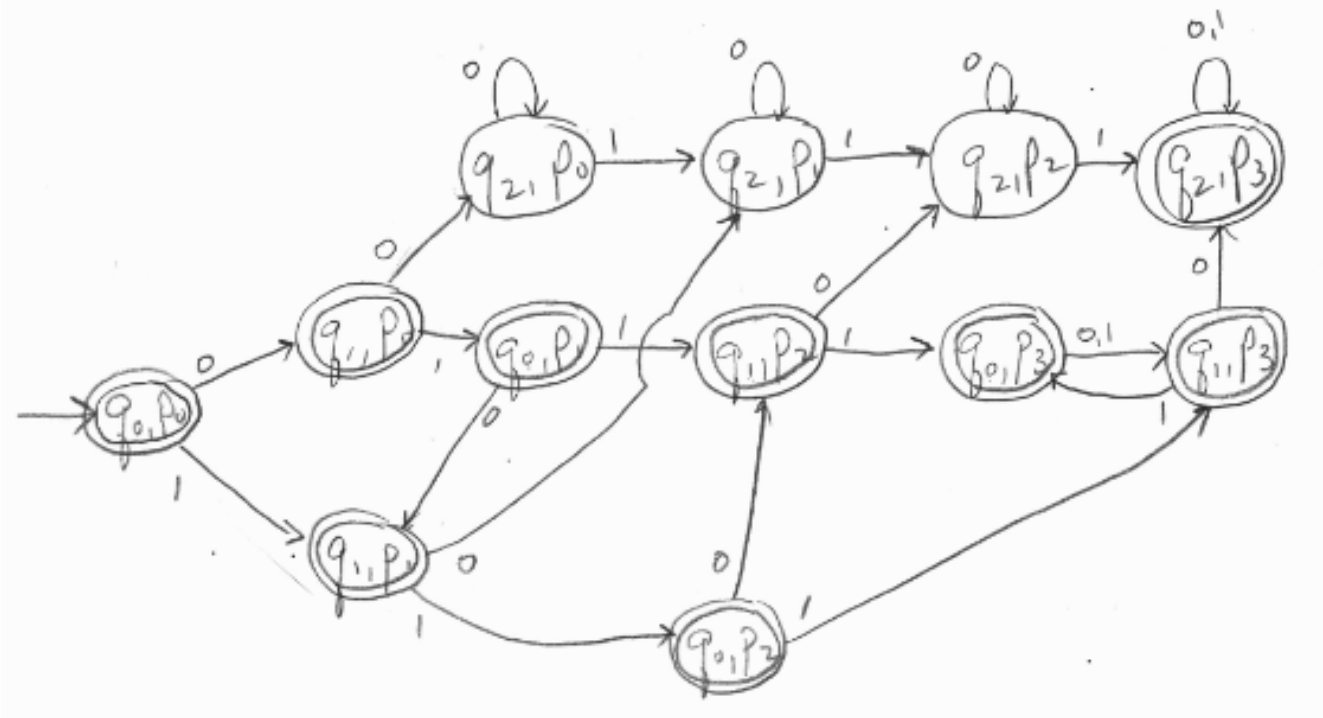
   L_2 = {w | w contains at least three 1's}



   Using the method shown in class and in Theorem 1.25, create a DFA for the union of L_1 and L_2.

   (15 pts.)

   Answer:



8. Give an algorithm that shows that the acceptance problem for regular expressions is Turing-decidable.

A_{RegExp} = {<E, w> | E is a regular expression, w is a string in the language of E, and w $\in \mathcal{L}$ (E)}

(10 pts.)

M= "On input <E, w> where E encodes a regular expression and w is a string from the alphabet of E.

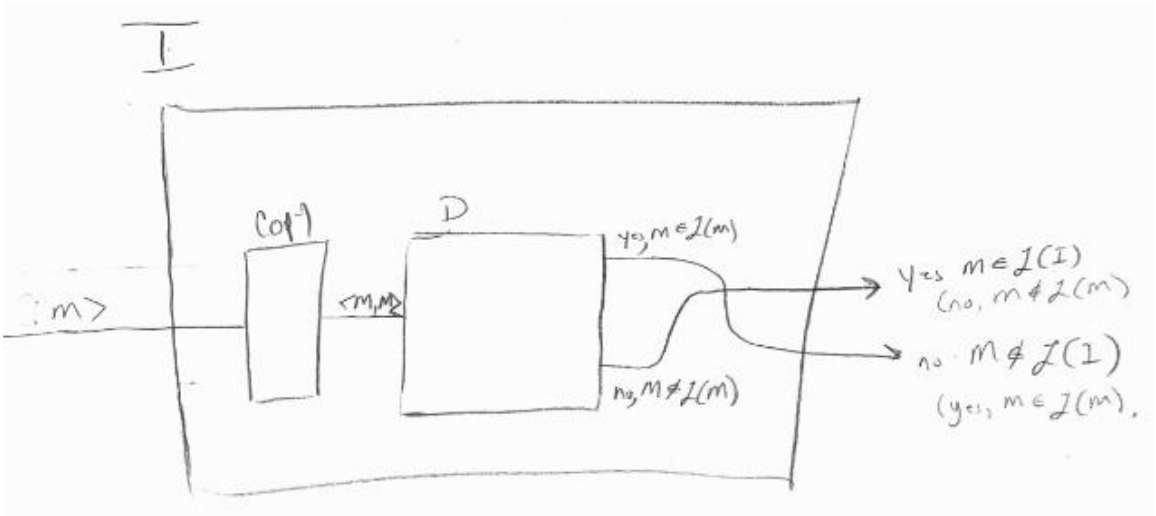
1. Convert the regular expression into an NFA as we did in class.
2. Convert the NFA to a DFA as we did in class.
3. Simulate the DFA on w. If the simulation ends in an accept state, accept; otherwise reject."

A mechanical procedure is known for each step in this algorithm. Furthermore, each of these mechanical procedures halts on all inputs (note that the string w will always be finite, otherwise it could not be encoded. Thus, this algorithm is well defined and halts on all inputs. Therefore, the algorithm defines a decider.

9. Prove that A_{TM} is not decidable. (15 pts.)

A_{TM} = {<M,w> | M is a TM, w is a string in the language of the TM and w∈ L(M)}

Suppose, by way of contradiction, that A_{TM} is decidable. Then there is some decider that decides it. Call this decider D. Consider the following TM I, for impossible.



Since TM I takes the description of a TM as input and I is a TM, one can input the description of I into I.

Is I∈ ℒ(I)?

If yes, I∈ℒ(x), then I∉ ℒ(I).

Is I∉ ℒ(I)?

If yes, I∉ℒ(x), then I∈ ℒ(I).

This is the absurdity that tells us that our original assumption must have been wrong. Thus, A_{TM} must not be decidable.

10. Prove the "if" direction (⇐) of the following theorem.

A language is decidable iff it is Turing-recognizable and co-Turing-recognizable. (10 pts.)

⇐ (if) Say that a language, L, is both Turing-recognizable and co-Turing-recognizable. This means that there are Turing recognizers, M_L and $M_{complement(L)}$, that recognize L and the complement of L, respectively. Create a machine, D, that takes an input, copies it onto another tape, and runs 1 step of M_L, followed by 1 step of $M_{complement(L)}$, repeatedly until one of the machines accepts. Tapes will also be required to sim simulate M_L and $M_{complement(L)}$. One of the machines will eventually accept because any string, w, must either be in L or the complement of L. Thus, the new machine is a decider, and L must be decidable.

Here is a definition of Turing machine D that decides the language L
D = "On input w

1. Run M_L and $M_{complement(L)}$ in parallel, performing one step on the first machine, followed by one step on the second, back and forth. If M_L accepts the string, accept. If $M_{complement(L)}$ accepts the string, reject."

D defines a Turing machine which decides the language, thus, the language is decidable.