Name:

Formal Languages & Automata Theory

Summer 2017 Practice Midterm

Four basic concepts where presented in the introductory material: Two machines (DFAs and NFAs) and two language descriptors (regular expressions and regular grammars). DFAs and NFAs can be described by transition graphs or transition tables. An overarching theorem is that each of these representations define exactly the same language class: Regular languages. A basic skill is the ability to construct a reduction from any one of these representations to any other. In practice, the typical sequence of reductions is:

$$\text{Regular expression} \mapsto \text{NFA} \mapsto \text{DFA} \mapsto \text{minimal DFA}$$

Regular expression can describe numbers and names (identifiers, password, ...) and can be useful in many applications.

Another basic idea is that regular languages are closed under many common operations: Union, intersection, complement, concatenation, star, and reversal. And another basic idea: The pumping lemma can be used to show some languages are not regular. And, yet another basic idea: Decision problems about regular languages are usually decidable. That is, decision problems usually have terminating algorithms that always correctly answers the question.

1 Induction, Contradictions, & Relations

Score

1. (5 pts) In graph theory, a *complete graph* G has an edge from every node to *every other* node. But, a transition graph TG is *complete* only if there is an edge from every node to *every* node. Let $|\mathbb{V}| = n$ be the number of nodes in a graph. Prove that $|\mathbb{E}| = \binom{n}{2}$ for a complete graph G. Prove that $|\mathbb{E}| = n^2$ for a complete transition graph TG.

Score

2. (5 pts) One Gradiance question caused me to wonder: Is it possible to construct an NFA over $\Sigma = \{0, 1\}$ such that $n(k)$, the number of strings of length k, is F_{k+1}? I think it is. Prove me right or wrong.

Score

3. (5 pts) Pretend that a string w of length n is accepted by DFA M. Pretend $n \geq m$ where m is the cardinality of states Q. Use the pigeonhole principle to prove that the transition graph for M has a cycle of length 1 or more.

Score

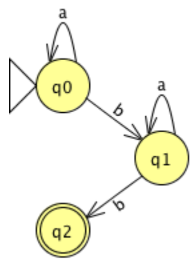
4. (5 pts) Let M be a DFA. Say strings x and y in Σ^* are *indistinguishable*, written $x \equiv y$ if and only if

$$\delta^*(q_0, x) = \delta^*(q_0, y)$$

Prove that the *indistinguishable* relation is reflexive, symmetric, and transitive.

2 Machines and Languages

Consider the transition graph below for machine M (drawn using JFLAP).



Score

1. (5 pts) Does graph describe a DFA or an NFA?

Score

2. (5 pts) Create a transition table for the machine

Score

3. (5 pts) Is $L = L(M)$ regular? If so, answer the questions below. Otherwise, why is L not regular?

   (a) Write an English description of the strings w accepted by L.

   (b) Write a regular expression that describes strings w accepted by L.

   (c) Construct a right-linear grammar for L.

3 Reduce NFA to DFA

Consider the transition table for NFA N:

	a	b	λ
q_0	$\{q_0, q_1\}$	$\emptyset$	$\{q_2\}$
q_1	$\emptyset$	$\{q_1, q_2\}$	$\emptyset$
q_2	$\{q_2\}$	$\emptyset$	$\{q_1\}$

Score

1. (5 pts) Draw a transition graph for machine N. Consider using JFLAP unless you have another tool you know for the problem. (I guess by hand will do, but that is so nineteenth century)

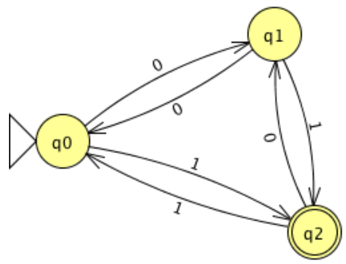
Score

2. (5 pts) Convert N into a DFA.

4 Reduce an NFA to a Regular Expression

Score

- (5 pts) Eliminate state q_1 from the transition graph below by:



  - Create an edge from q_0 to q_0 and appropriately label it with a regular expression r_{00} that describes all walks from q_0 to q_0.
  - Create an edge from q_0 to q_2 and appropriately label it with a regular expression r_{02}.
  - Create an edge from q_2 to q_0 and appropriately label it with a regular expression r_{20}.
  - Create an edge from q_2 to q_2 and appropriately label it with a regular expression r_{22}.

  1. Why can you eliminate state q_1 and all edges from or to it to create a reduced 2-state transition graph?
  2. Why does the regular expression

  $$r_{00}^*r_{02}(r_{22} + r_{20}r_{00}^*r_{02})^*$$

  describe the regular expression of

5 Reduce Regular Expression to NFA

Score

(5 pts) Construct an NFA that accepts the language $L((bb)^*a^* + b^*(aa)^*)$.

6 A NFA to a Regular Grammar

Score

1. (5 pts) Define: Right-Linear Grammar

Score

2. (15 pts) Consider the grammar $G = (S, T, V, P)$ with production rules

$$S \to aA$$
$$A \to aaA \mid \lambda$$

   (a) Construct a DFA that accepts/recognizes, strings derived from the grammar G with production rules

   (b) What is the language $L = L(G)$ defined by the above grammar G?

   (c) What is a regular expression describes the grammar G?

7 The Pumping Lemma

Score

1. (5 pts) Let L be a regular language. What are the conclusions of the pumping lemma for regular languages? Give simple English conclusions and more precise mathematical formulations.

Score

2. (5 pts) Use the pumping lemma to show that language

$$L = \left\{ww : w \in \{a, b\}^* \quad \text{is not regular.}\right\}$$

Score

3. (5 pts) Let L be a regular language with m states. Clearly, if its DFA M accepts some string of length less than m, then L is non-empty

   Prove that if L is non-empty, then M accepts a string w with $|w| < m$. (Hint: If L is non-empty, let $w \in L$ be as short as any word in L. If $|w| \geq m$, use the pumping lemma to derive a contradiction that w is among the shortest words accepted by M.

8 Applications

Score

1. (5 pts) Write a regular expression that describes fixed point rational numbers (an optionally signed arbitrarily long string of digits followed by an optional decimal point optionally followed by an arbitrarily lone string of digits). Use the productions below for atomic derivations.

$$\begin{aligned} \textbf{digit} &\to 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9 \\ \textbf{sign} &\to + \mid - \mid \lambda \\ \textbf{point} &\to . \end{aligned}$$

Score

2. (5 pts) Describe the following passwords rules using regular expressions.

   (a) A password must be 5 or more characters long.

   (b) A password must be contain at least one lower case and one upper case English letter.

   (c) A password must be contain at least one number.