CS Theory (Fall 25)

Midterm Review

Concepts to review

Be comfortable answering the following:

- What is an alphabet, string, and language?
- What is a DFA? What is its 5-tuple? What does the transition function tell us? When does a DFA accept a string? When does a DFA reject a string?
- What is a regular language? How do we know if a language is regular? What are the closure properties for regular languages?
- What is an NFA? What is its 5-tuple?
- How is an NFA different from a DFA?
- How do we convert between DFAs and NFAs?
- Do you know what a Regular Expression is and what is the language of a regular expression ?
- How do we know when two RegEx are equivalent, i.e. they describe the same language?
- What is big-O notation? What is big-Omega notation?
- What does it mean for a function to be poly(n) ? [1]
- What is a streaming algorithm? What is its 5-tuple representation?
- What is the space usage of a streaming algorithm?
- When is the language recognized by a streaming algorithm regular?
- What is a one-way communication protocol?
- What is the communication complexity of a one-way communication protocol?
- If a language is regular, what is its one-way communication complexity cost?
- Know how to prove that a language is not regular via a nonconstant 1-way communication lower bound.

[1] Definitions for big-O and big-Omega and poly(n) will be provided but it is better if you know them!

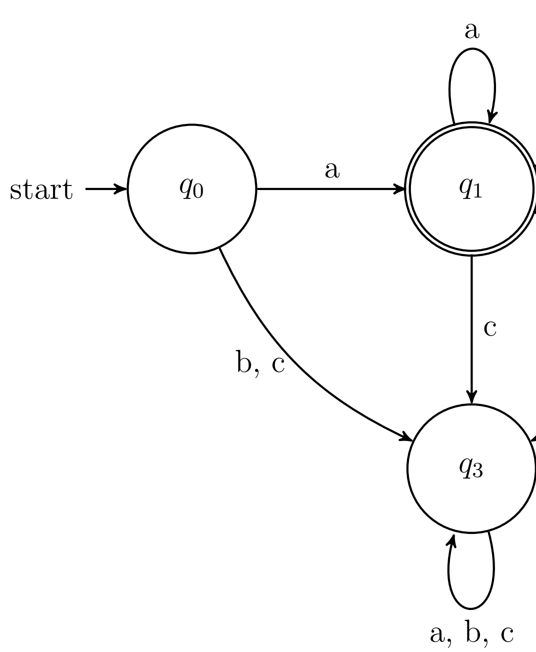
DFA/NFA

Construct a DFA (or NFA) for the following languages and explain why your DFA (or NFA) is correct.

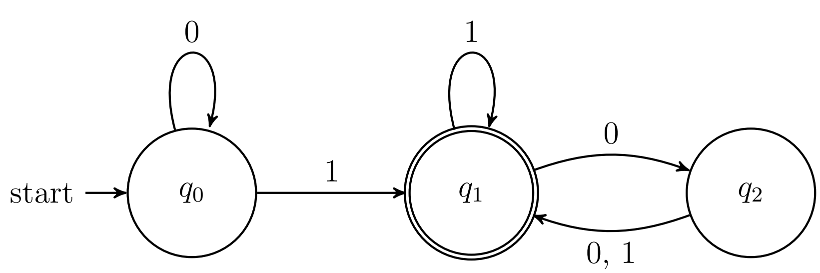
1. $L_1 = \{xy \mid x \in \{0,1\}^*$ has an even number of 1's and $y \in \{0,1\}^*$ has an odd number of 0's$\}$
2. $L_2 = \{w \in \{0,1\}^* \mid w \in L$ and the length of w is a multiple of 2 or 3$\}$, where L is a regular language recognized by the DFA $(Q, \Sigma, \delta, q_0, F)$.

Describe the corresponding regular language for the following DFAs or NFAs.

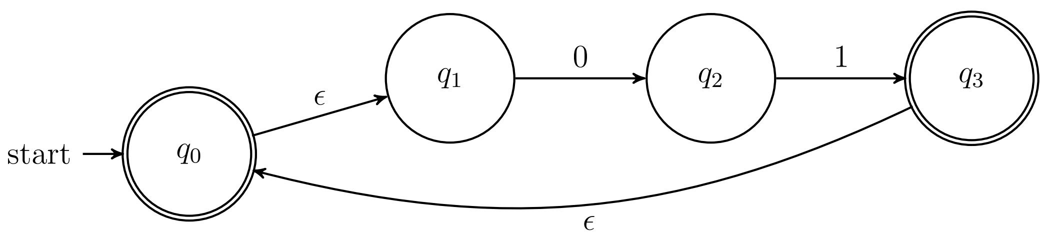
1.



2.



3.



If we were to turn the NFA given above into an equivalent DFA (whose states now correspond to subsets of $\{q_0, q_1, q_2, q_3\}$, what state would be the start state ? Which states would be accept states ? What's the transition for state $\{q_1, q_2\}$ when reading a 1 ?

RegEx

Construct regular expressions for the following languages.

1. $L_2 = \{w \in \{a, b, c\}^* \mid w \text{ does contain the substring } abc\}$
2. $L_3 = \{w \in \{a, b\}^* \mid \text{ does not contain the substring } ab\}$.

Prove or disprove the following equivalences between regular expressions. Assume r, s, and t are regular expressions.

1. $s^*(rs^*r)^*s^* = (r \cup s)^*$
2. $r(s \cup t)^* = (rs)^* \cup (rt)^*$
3. $(r^* \cup s^*)^* = (r \cup s)^*$
4. $r^*(s \cup t)^* = (r \cup s \cup t)^*$

Asymptotic Notation

For each of the following functions, circle *all* of the letters that are true.

1. $f(n) = (\log_2 n)^2$
   (a) $f(n) = O(\log_{10} n)$
   (b) $f(n) = O((\log_{10} n)^2)$
   (c) $f(n) = O(n^{1/10})$
2. $f(n) = 2^n$
   (a) $f(n) = \Omega(2^{\sqrt{n}})$
   (b) $f(n) = \Omega(n^2)$
   (c) $f(n) = \Omega((\log_2 n)^n)$

Streaming

1. Give a streaming algorithm for recognizing the language

$$L = \{w \in \{0, 1, 2\}^* \mid \text{the number of 2's in } w \text{ is at least } 2\times(\text{number of 1's in } w)\}$$

(For example, $00121222 \in L$ because there are four 2's and two 1's, and $4 \geq 2 \times 2$. $112022 \notin L$ because there are three 2's and two 1's, and $3 \not\geq 2 \times 2$.)

Also, prove that your streaming algorithm uses $O(\log n)$ space.

2. Give a streaming algorithm for recognizing the language

$$L = \{w \in \{0,1,2\}^* \mid \text{the most common symbol in } w \text{ appears } k \text{ times and the least common appears } k-1 \text{ times}\}.$$

Ex: $w = 001112 \notin L$ since the most common element is 1 and it appears 3 times and the least common element 2 appears only 1 time. But $w = 0011122$ is in L (most common appears 3 times, least common appears 2 times).

Also, prove that your streaming algorithm uses $O(\log n)$ space.

Proving Non-Regularity

Show that the following languages are *not* regular. Do this by giving a lower bound on the one-way communication complexity of language, or by using a direct argument. You should practice proving the lower bound by both a direct argument, and by a one-way communication complexity lower bound argument. Do not prove your lower bound by a reduction from another language.

1. $L = \{w \in \{0,1\}^* \mid w = xy$ and the first 1 in x does not appear after the first 1 in $y\}$.

   Note: this is the example "First" given in the class notes, so you already have the answer. Practice solving it again without looking.

2. Index $= \{w \in \{0,1\}^* \mid w = xy$ and $x_i = 1$ where i is the first index in y with $y_i = 1\}$.

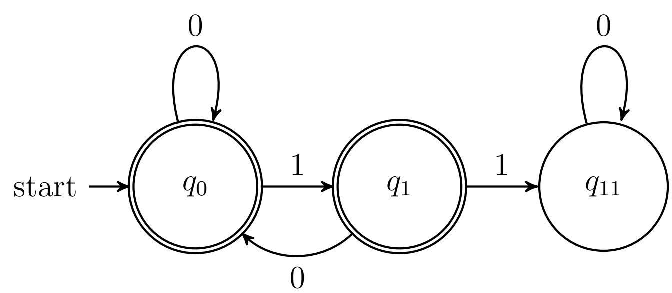
   Note: this is an example from the lecture notes. But we didn't cover the lower bound in class.

3. Practice proving one-way communication lower bounds for other examples that were proven in class or in the notes (for example: EQ, MAJ, and variants of MAJ such as the language Sum from HW3.)

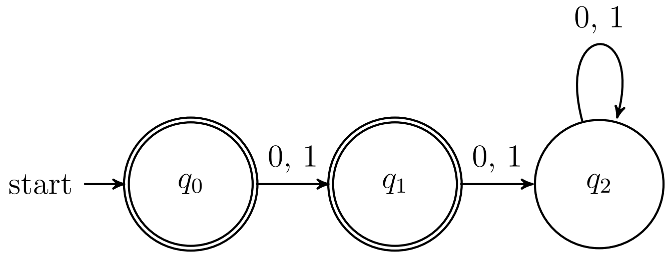
True/False Section

Determine whether each of the following statements is true or false:

1. (T/F) The following is a valid DFA for a language over $\{0,1\}$:



2. (T/F) The language recognized by the following DFA is $\{0, 1\}$:



3. (T/F) Suppose the languages L_1 and L_2 are recognized by the DFAs D_1 and D_2, respectively. If D_1 has 3 states and D_2 has 4 states, then there exists a 15 state DFA that recognizes $L_1 \backslash L_2 = \{w \in L_1 \text{ and } w \notin L_2\}$.

4. (T/F) If L_1 and L_2 are languages with $(L_1)^* = L_2$, and L_2 is regular, then L_1 is regular.

5. (T/F) If L_1 and L_2 are non-regular languages, then $L_1 \cup L_2$ is a non-regular language as well.

6. (T/F) The concatenation of a language with itself, $L^2 = LL$, can be written in set-theoretic notation as $\{ww \mid w \in L\}$.

7. (T/F) Suppose there exists an NFA $N = (Q, \Sigma, \delta, q_0, F)$ which recognizes the language L. Let $N' = (Q, \Sigma, \delta, q_0, Q \backslash F)$ be the NFA where the accept and reject states of N are switched. Then N' recognizes the complement $\overline{L}$.

8. (T/F) If L is a regular language recognized by an NFA with $k \in \mathbb{Z}_{>0}$ states, then there exists an DFA recognizing L with $2k - 1$ states.

9. (T/F) If the language L is recognized by an NFA with 4 states, then there exists a DFA recognizing L with less than 20 states.

10. (T/F) The language $L = \{w \in \{0, 1\}^* \mid w \text{ has even length and contains the substring } 1010\}$ can be described with a regular expression.

11. (T/F) The language described by the expression $(01)^*(0 \cup 1)^*\varnothing$ contains more strings than the language described by the expression $((0 \cup 10) \cup \epsilon)(0 \cup 1)110$.

12. Quick asymptotic notation:

    (a) (T/F) $n \log^2 n = O(n^2)$
    (b) (T/F) $n = \Omega(\log n)$
    (c) (T/F) $1/n = \Omega(1/\log n)$
    (d) (T/F) $2^n = o(3^n)$
    (e) (T/F) $3^n = 2^{O(n)}$
    (f) (T/F) $n^{-1} = \text{poly}(n)$

13. More challenging asymptotic notation:

    (a) (T/F) $\sum_{i=1}^{n} 3^i = O(2^n)$
    (b) (T/F) $n! = \text{poly}(n)$