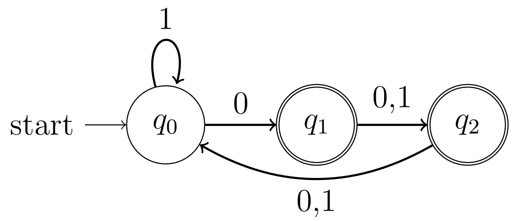
COMS 461 - Midterm 1 Review Solutions

1. (8 points) Consider the DFA shown below.



(a) What sequence of states will this DFA enter as it reads the string 110101?

Solution:

$$q_0 \to q_0 \to q_0 \to q_1 \to q_2 \to q_0 \to q_0$$

(b) Will this DFA accept the string 110101?

Solution: No, it will not accept that string because it ends in state q_0.

2. (16 points) The following statements are all false. For each one, explain why it is false.

(a) The cardinality of $\{0,1\}^*$ is uncountable.

Solution: This is false because $\{0,1\}^*$ is countably infinite. It is a countable union of finite sets.

(b) The NAND function is universal which means that any function $f : \{0,1\}^* \to \{0,1\}$ can be expressed using NAND functions.

Solution: This is false because NAND can only compute all Boolean valued functions defined on binary strings of a fixed finite length, not arbitrary length. Universal means that all function $f : \{0,1\}^n \to \{0,1\}$ can be expressed using NAND functions for any fixed n.

(c) The union of any two languages $A, B \subset \{0,1\}^*$ is a regular language.

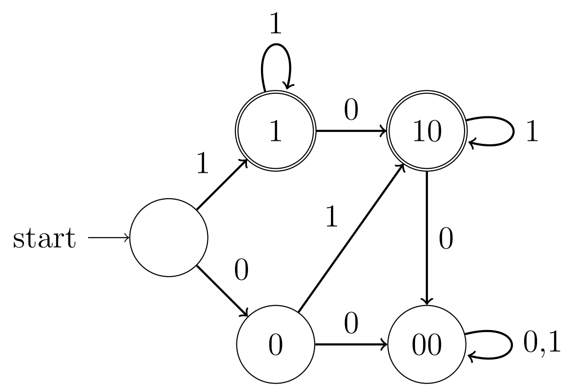
Solution: This is false unless you also assume that A and B are both regular languages.

(d) Boolean logic circuits, DFAs, NFAs, and regular expressions are all equivalent computationally. They are all able to recognize regular languages.

Solution: Every regular language can be checked with an NFA or DFA or expressed with a regular expression, but not every regular language can be checked by a Boolean logic circuit.

3. (12 points) Let $L \subset \{0,1\}^*$ be the language that contains all strings with at least one 1 and at most one 0. Construct a DFA that accepts L.

Solution:



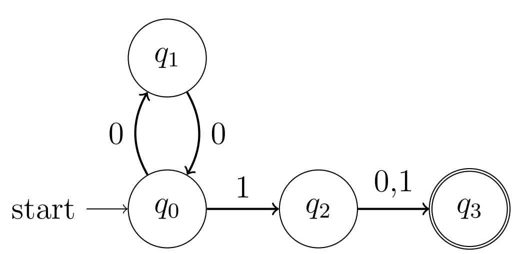
4. (16 points) Consider the regular expression $(00)^*1(0|1)$.

(a) Describe in words the set of strings that this regular expression will match.

Solution: Any string that combines an even number of zeros with either a 10 or 11 at the end.

(b) Construct an NFA (or DFA) that accepts exactly that set of strings.

Solution:



5. (8 points) In biology, strings of three DNA nucleotides (called *codons*) are known to encode 20 different amino acids. Here, the alphabet consists of the four DNA nucleotides $\Sigma = \{A, C, G, T\}$. Let $\mathcal{A}$ denote the set of 20 possible amino acids.

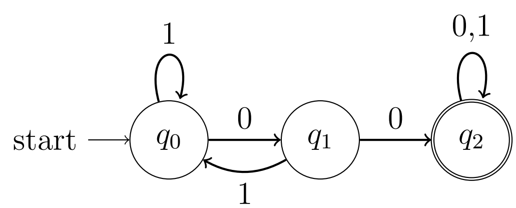
(a) How many possible strings of three nucleotides are there? In other words, find $|\Sigma^3|$.

Solution: There are $4^3 = 64$ possible strings of length three from $\{A, C, G, T\}$.

(b) How many possible functions are there from $\Sigma^3 \to \mathcal{A}$?

Solution: There are 20^{64} possible functions from Σ^3 to $\mathcal{A}$.

6. (8 points) Consider the DFA shown below.



(a) This DFA can be described by a quintuple $(Q, \Sigma, \delta, q, F)$ where $\Sigma = \{0, 1\}$. What are Q, q and F in this notation?

Solution: $Q = \{q_0, q_1, q_2\}$, $q = q_0$, and $F = \{q_2\}$.

(b) Find a regular expression that matches the same set of strings that this DFA accepts.

Solution:

$$(0|1)^*00(0|1)^*$$

7. (12 points) Let $L \subset \{0,1\}^*$ be the language consisting of all strings with more 0's than 1's. Use the pumping lemma to prove that L is not regular.

> **Solution:** If L is regular, then it has a pumping number p. Consider the string $1^p 0^{p+1}$. If you pump this string, then you will increase the number of 1's, which will give you a string not in L. That is a contradiction, so L is not regular.

8. (20 points) Let $\Sigma = \{0, 1, +, =\}$ and let

$$L = \{x = y + z : x, y, z \text{ are binary integers, and } x \text{ is the sum of } y \text{ and } z\}.$$

(a) Which of the following strings are in L? Circle all correct answers, there might be more than one.

A. `100=1+10`

B. `11=10+1`

C. `2=1+1`

D. `1000=111+1`

E. `x=11+z, if y=11`

(b) Is it possible to write a regular expression over Σ that represents all valid strings in L? Explain why or why not.

> **Solution:** No, there is no regular expression for this language because it is not regular. If it were regular, there would be a pumping length p, such that any string longer than p in L could be pumped. But a string that begins with p 1's could not be pumped since adding more ones to the binary number on the left side of the equality will make the equation not true.