1. Provide two valid strings in the languages described by each of the following regular expressions, with alphabet $\Sigma = \{0,1,2\}$.

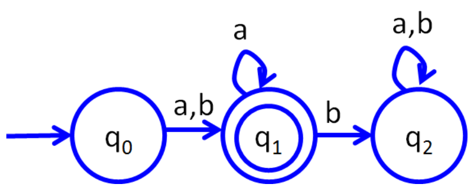
(a) $0(010)^*1$
Examples: 01, 00101, 00100101, 00100100100101

(b) $(21 \cup 10)^*0012^*$

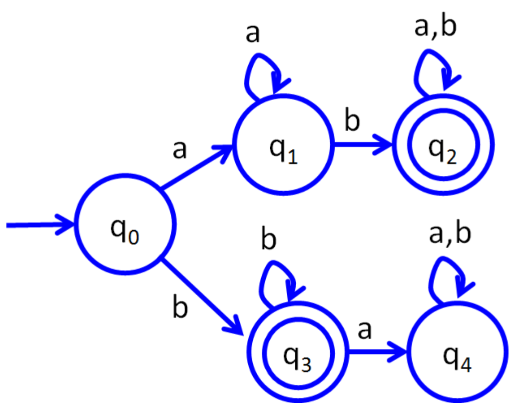
(c) $1^*(200)^* \cup 100^*01$

2. For each of the following DFAs, provide a Regular Expression to describe the language, with alphabet $\Sigma = \{a, b\}$.

(a) RED QUESTION

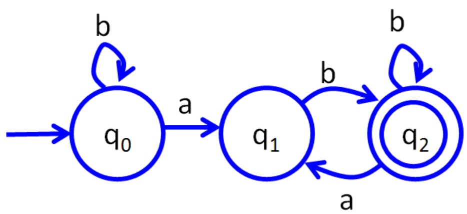


(b) BLUE QUESTION



L(B) = aa*b(aUb)* U bb*

(c) GREEN QUESTION

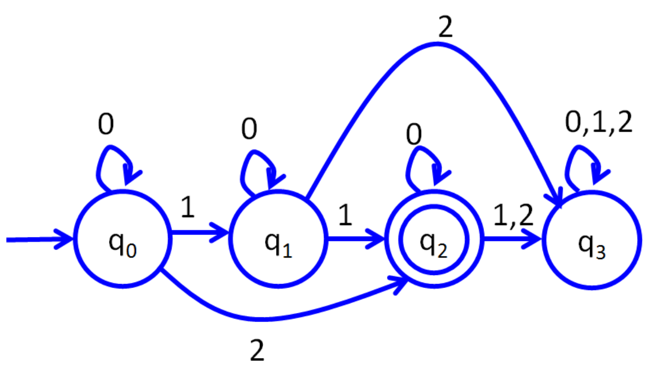


3. Create a DFA to accept each of the following languages.

A={w | last number in w is even} , given alphabet $\Sigma = \{0,1,2,3\}$

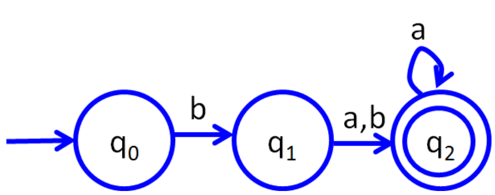
B={w | at least three symbols in w} , given alphabet $\Sigma = \{a, b, c\}$

C={w | sum of digits in w equals 2} , given alphabet $\Sigma = \{0,1,2\}$

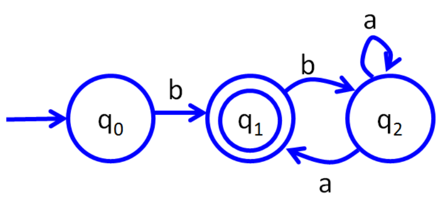


4. Convert each of the following NFAs to a DFA, with alphabet $\Sigma = \{a, b\}$.

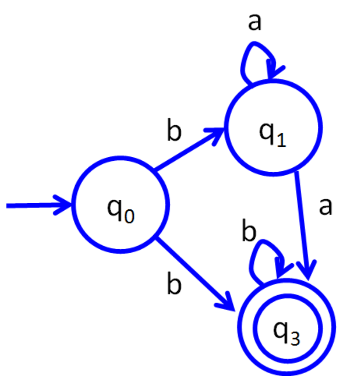
(a) RED QUESTION



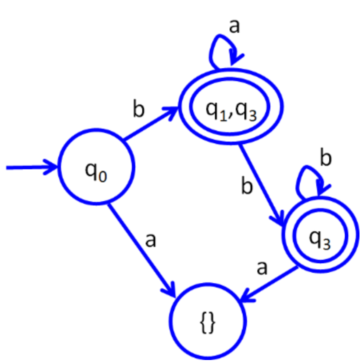
(b) GREEN QUESTION



(c) BLUE QUESTION



ANSWER:



5. Prove the following languages are **not** regular.

(a) $A=\{b^{k!}a \mid k>0\}$

Pumping lemma!

$w = b^{p!}a = xyz$

$x=b^m \quad y=b^n \quad z=b^{p!-(m+n)}a \qquad p>=n>0$

If $w \in A$, we also need $xy^2z \in A$ --- check if this is true!

$xy^2z = b^{p!+n}a$

to be in language a, p!+n must be (p+q)! where q>0

(p+1)! = (p+1)xp! = pxp! + p!   Compare with p!+n

n = p x p! >> p   This violates the rules of n, which must be less than p

So xy^2z is NOT in language A, which means $b^{k!}a$ cannot be pumped, which means it is not regular!

(b) $B=\{0^k1^{2k}0^k \mid k>0\}$

7. Provide two valid strings for each of the following CFGs.

(a) G1:

S -> A | B
A -> DC | C
B -> EF | F
C -> dog | cat | mouse
D -> big | small | red | white
E -> quickly | slowly
F -> runs | swims | jumps | barks

(b) G2:

S -> BA | B
B -> xBx | ε
A -> c | de | f

(c) G3:

S -> CaC | C

C -> yCy | y

CaC -> **yay**

C -> yCy -> **yyy**

CaC -> yCyay -> yyCyyay -> **yyyyyay**

CaC -> yayCy -> **yayyy**

8. Convert the following CFGs to CNF (same as Q7).

(a) G1: (for G1, each word is a terminal)

S -> A | B

A -> DC | C

B -> EF | F

C -> dog | cat | mouse

D -> big | small | red | white

E -> quickly | slowly

F -> runs | swims | jumps | barks

S -> DC | C | EF | F *replace A and B*

C -> dog | cat | mouse

D -> big | small | red | white

E -> quickly | slowly

F -> runs | swims | jumps | barks

S -> DC | dog | cat | mouse | EF | runs | swims | jumps | barks

C -> dog | cat | mouse *replace C and F in S rule*

D -> big | small | red | white

E -> quickly | slowly

F -> runs | swims | jumps | barks

(b) G2:

S -> BA | B

B -> xBx | ε
A -> c | de | f

(c) G3:
S -> CaC | C
C -> yBy | y

9. Express each of the following languages as a **CFG**.

(a) A = $\{x^k y^{2k} z\}$

(b) B = {w | w is described by (ab)*ba }

(c) C = $\{ 010^k 101^{k+2} \mid k > 0 \}$

S -> 010A111
A -> 0A1 | 10

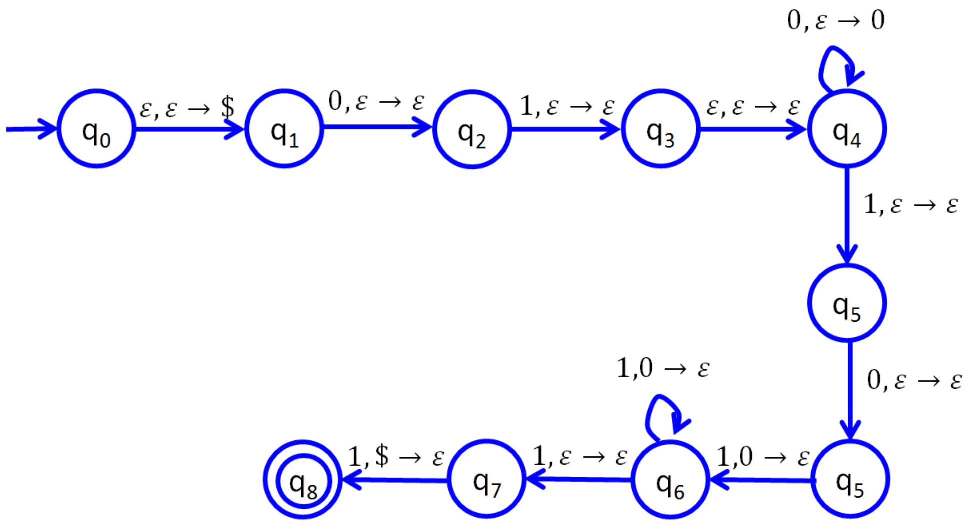
10. Describe the PDA to accept each of the following languages (languages from Q9).

(a) A = $\{x^k y^{2k} z\}$

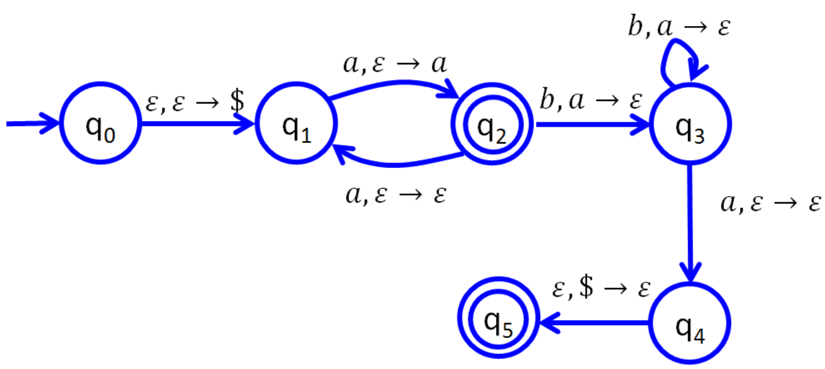
(b) B = {w | w is described by (ab)*ba }

(c) C = { 010^k101^{k+2} | k >0 }

NOTE: The answer below is slightly off: it is for k≥0, not k>0



11. What is the response of PDA P1 to each input: **i.e., does it reach an accept state?**



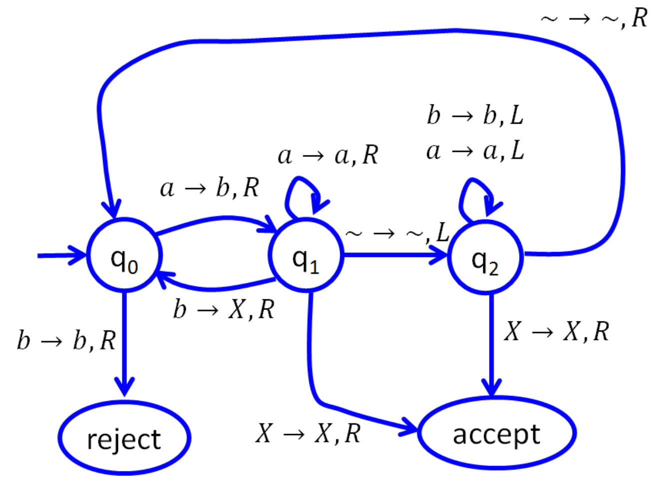
Input 1: bbaa

Input 2: aaa

Input 3: abb
Does not reach accept state!

Input 4: aaaaabbba

12. Describe the configurations resulting from each of the input tapes specified below for the following Turing Machine.



(a) aabb

(b) abaaa

q_0 abaaa
q_1 bbaaa
q_0 aXaaa
q_1 aXbaa
q_1 aXbaa
q_1 aXbaa~
q_2 aXbaa
q_2 aXbaa

q_2 aX<u>b</u>aa
q_2 a<u>X</u>baa
accept

(c) aaaba

13. Express the following problems as languages.

(a) Determine if two specified CFG's accept complementary inputs – every accepted input for the first CFG is rejected by the second CFG and vice versa.

L={<G1,G2> | L(G1) = (L(G2)')}

(b) Determine if a specified DFA accepts a specified string repeated zero or more times.

(c) Determine if a specified Turing machine accepts the same language as a specified PDA.

14. Prove the follow languages are decidable.

(a) Determine if a specified DFA accepts a specified string repeated zero or more times.

(b) Determine if a specified CFG is in Chomsky Normal Form.

Each CFG has a finite number of rules. For each rule, simply test if it has one terminal or two variables. If ever find a rule that fails these criteria, reject. **Looping through the rules takes a finite number of steps, so the algorithm to determine this question will halt with "accept" or "reject" decision for every grammar.**

(c) Determine if a specified CFG does not accept a specified word.

15. Provide a big-O and a little-o complexity for each function.

(a) $f(n) = 20\ n \log n + 5n + 2$

(b) $f(n) = 30\ n^3 + 6\ n^5 + \log n$

(c) $f(n) = 5\ n^2 + n^3 \log n + 4^n + 8$
Smallest: $O(4^n)$; alternatively $O(n\ 4^n)$, $O(4^n \log n)$
Small: $o(4^n \log n)$, $o(n\ 4^n)$ … anything bigger than $o(4^n)$

16. Compute the complexity for each algorithm described below.

(a) Algorithm 1: (State the complexity based on *r* and *c*)
Start with a table of *r* rows and *c* columns

1. Sum the elements in each row
    - Use a running sum with a loop across all columns
2. Find the row with the maximum sum
    - Loop through all rows, saving biggest sum and its row in two separate variables

Step 1: r x c Step 2: r
In total: **O(r c)**

(b) Algorithm 2: (State the complexity based on *n*)

Start with a list of *n* elements

1. While list is longer than 1 element long
   - Replace each pair of elements with the product of the two elements (elements 1 and 2 replaced by single product, elements 3 and 4 replaced by single product, elements 5 and 6 replaced by single product, etc.)

17. Determine if the following problems are in P and/or NP.

(a) Given a directed graph and two nodes a and b, determine if there are at least two different paths to get from node a to node b. Paths are "different" if they differ by at least one edge.

(b) In an undirected graph, determine if every node is attached to every other node.

(c) Determine if the language of a DFA is empty.
Algorithm involves marking states in DFA until no new states marked. This take $O(n^3)$ time, where *n* is the number of nodes (go through $O(n^2)$ edges at most n times (given n nodes)). **Thus, DFA is in P, and also in NP (all of P is in NP).**