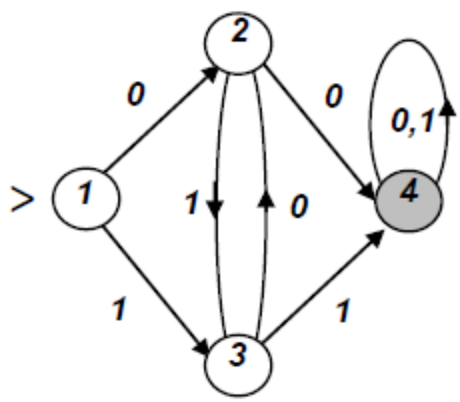
Homework #3 *due November 15, 2022, Tuesday before recitation*

(1) Consider the regular expression $\boldsymbol{E = (1+(0+101)^*)^*}$. Draw an ε-*NFA* accepting the language corresponding to $\boldsymbol{E}$ above using as little number of states as possible; compute and sketch the equivalent *NFA* without ε-transitions ; and finally compute the equivalent *DFA* accepting the language corresponding to $\boldsymbol{E}$ above.

(2) Convert the following *DFA* to ***RE*** using the state elimination technique. Try to simplify the regular expression using the equivalence relations stated in class.



(3) For following languages,prove or disprove the statement that the language is ***regular***.

(a) $\{ww^R \mid w \in (0+1)^*\}$, where w^R stands for the string w written in reverse (backwards)

(b) $\{w \mid w$ *has same number of occurrences of* ***01*** *and* ***10*** *as substrings*$\}$

(4) Consider the *Deterministic Finite Automata*,

$A = (Q_A, \Sigma_A, \delta_A, q_{0A}, F_A)$ *and* $B = (Q_B, \Sigma_B, \delta_B, q_{0B}, F_B)$

where

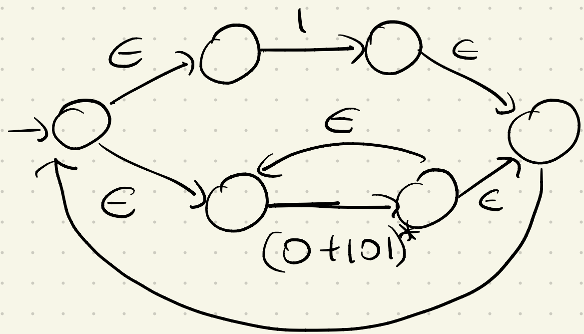
$Q_A \cap Q_B = \varnothing$, $\Sigma_A \cap \Sigma_B = \varnothing$ where $\varnothing$ stands for the null set.

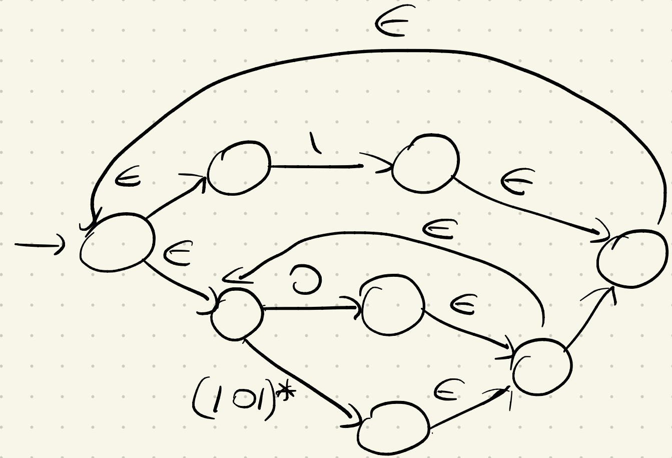
Let $L_A \subseteq \Sigma_A^*$ and $L_B \subseteq \Sigma_B^*$ be the languages accepted by $\boldsymbol{A}$ and $\boldsymbol{B}$ respectively and define the interleaved language:

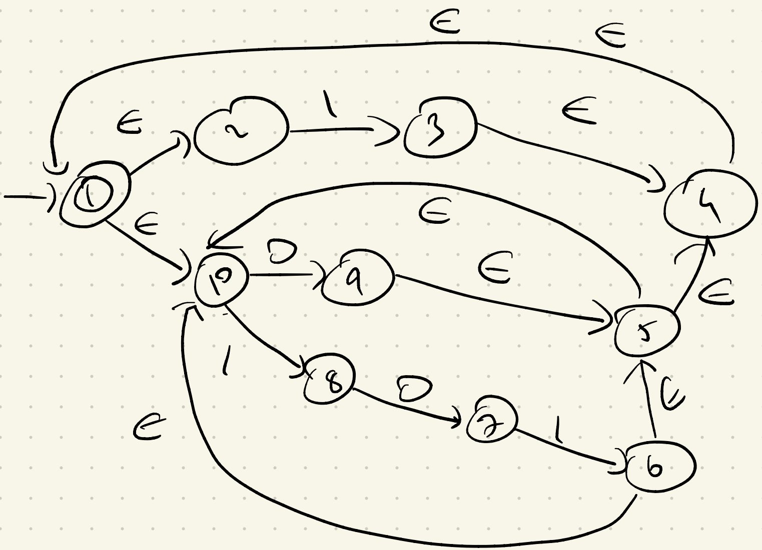
$L_A \parallel L_B := \{s \in (\Sigma_A \cup \Sigma_B)^* \mid s\uparrow_A \in L_A$ *and* $s\uparrow_B \in L_B\}$

Question 1i

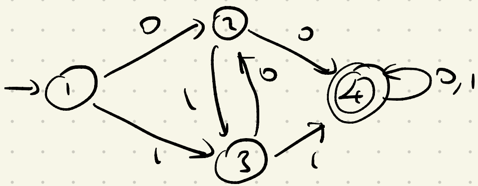
$E = (1 + (0+101)^*)^*$



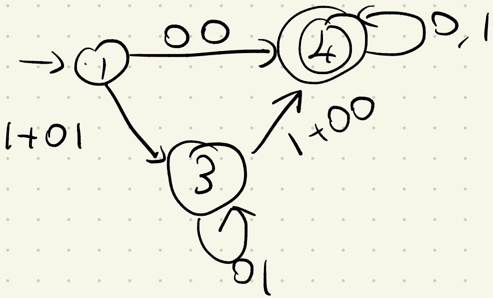




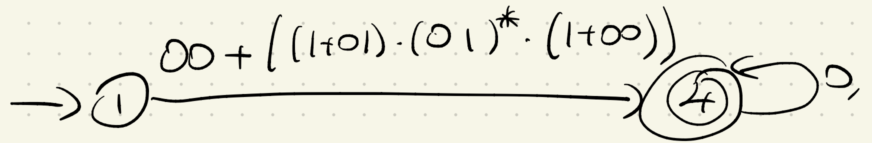
Question 2 i



eliminate 2



eliminate 3



Finally, we have

$$(00+((1+01)\cdot(01)^*\cdot(1+00)))\cdot(0+1)^*$$

as a regular expression.

$(0+1)^* = P$

Question 3:

a) Let p be the pumping length and $s = a^p bba^p$ $x = \epsilon$, $y = a^p$, $z = bba^p$

$$\Downarrow$$

$$s = \epsilon a^p bba^p = a^p bba^p$$

$|xy| \leq p$

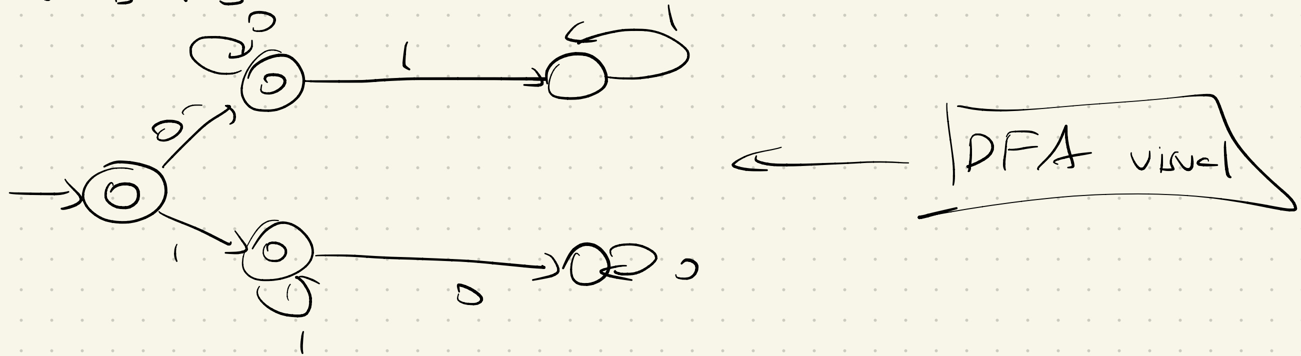
$|y| > 0$ If we take $i = 2$ $w = x.y^i.z$

$xyyz = \epsilon a^p a^p bba^p \notin L$, so L is not regular!

Question 3:

b) {w | w has same number of occurences of 01 and 10 as substrings}

→ It is regular.



Question 4.3.3

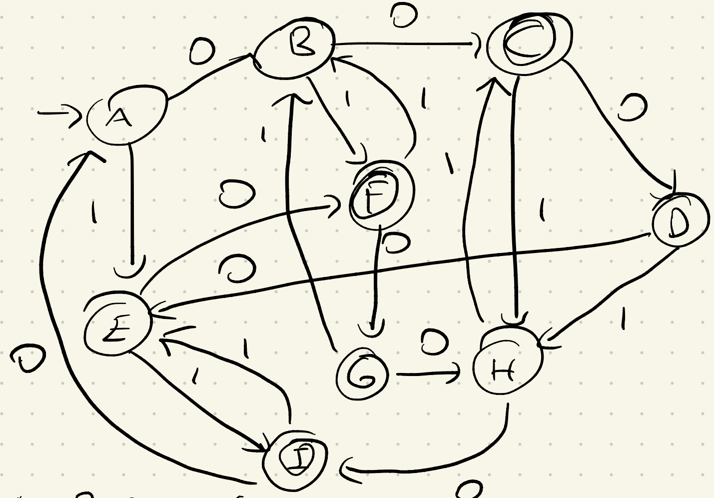
Observe that the question is equivalent to asking whether the complement language L^c is empty. If we switch the accepting and non-accepting states of the DFA A, we get the DFA (say A') that accepts L^c. In this new DFA, i.e., A', use a graph reachability algorithm to check whether any final state is reachable from the start state in A', it follows that L^c must be empty, which implies that $L = \Sigma^*$.

Question 4.3.4

Assume that the DFAs of the two languages, D_1 and D_2, are given to us. If not, we can always construct the DFAs from the regexes corresponding to the respective languages. Since L_1 and L_2 is regular we know that $L_3 = L_1 \cap L_2$ is also a regular language; indeed, we can use the procedures to construct the DFA for L_3. Now, all that we need to do is to check whether there exists a path from start state of D_3 to a final state of D_3. If there exists a path, then L_1 and L_2 have at least one string in common; otherwise, they have no common strings.

Question 4.4.2

	0	1
→A	B	E
B	C	F
(C)	D	H
D	E	H
E	F	I
(F)	G	B
G	H	B
H	I	C
(I)	A	E



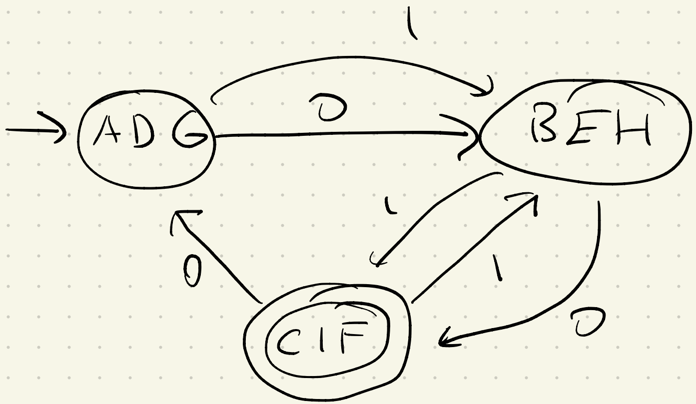
	A	B	D	E	G	H	C	F	I
A	X	1		1		1	0	0	0
B	X	X	1		1		0	0	0
D	X	X	X	1		1	0	0	0
E	X	X	X	X	1		0	0	0
G	X	X	X	X	X	1	0	0	0
H	X	X	X	X	X	X	0	0	0
C	X	X	X	X	X	X	X		
F	X	X	X	X	X	X	X	X	
I	X	X	X	X	X	X	X	X	X

(A, D)
(A, G)
(B, E)
(B, H)
(D, G)
(E, H)
(C, F)
(C, I)
(F, I)

Equivalences

→(A, D, G)
(B, E, H)
(C, I, F)

Final state



Minimized version of DFA?

Question 4.4.3

Let Q be the states, $|Q| = n$. Consider the equivalence relation $E_i \subseteq Q \times Q$.
Let k be the tightest upper bound. p and q are distinguishable states which the shortest distinguishing string w has length k_1. The shortest distinguishing word has length $k-1$.

→ By the chain of strict inclusions, E_1 has at least 3 parts, E_2 has at least 4 parts, and E_{n-2} has n parts which means that E_{n-2} is the identity relation. This implies that the worst-case bound is $k = n-2$