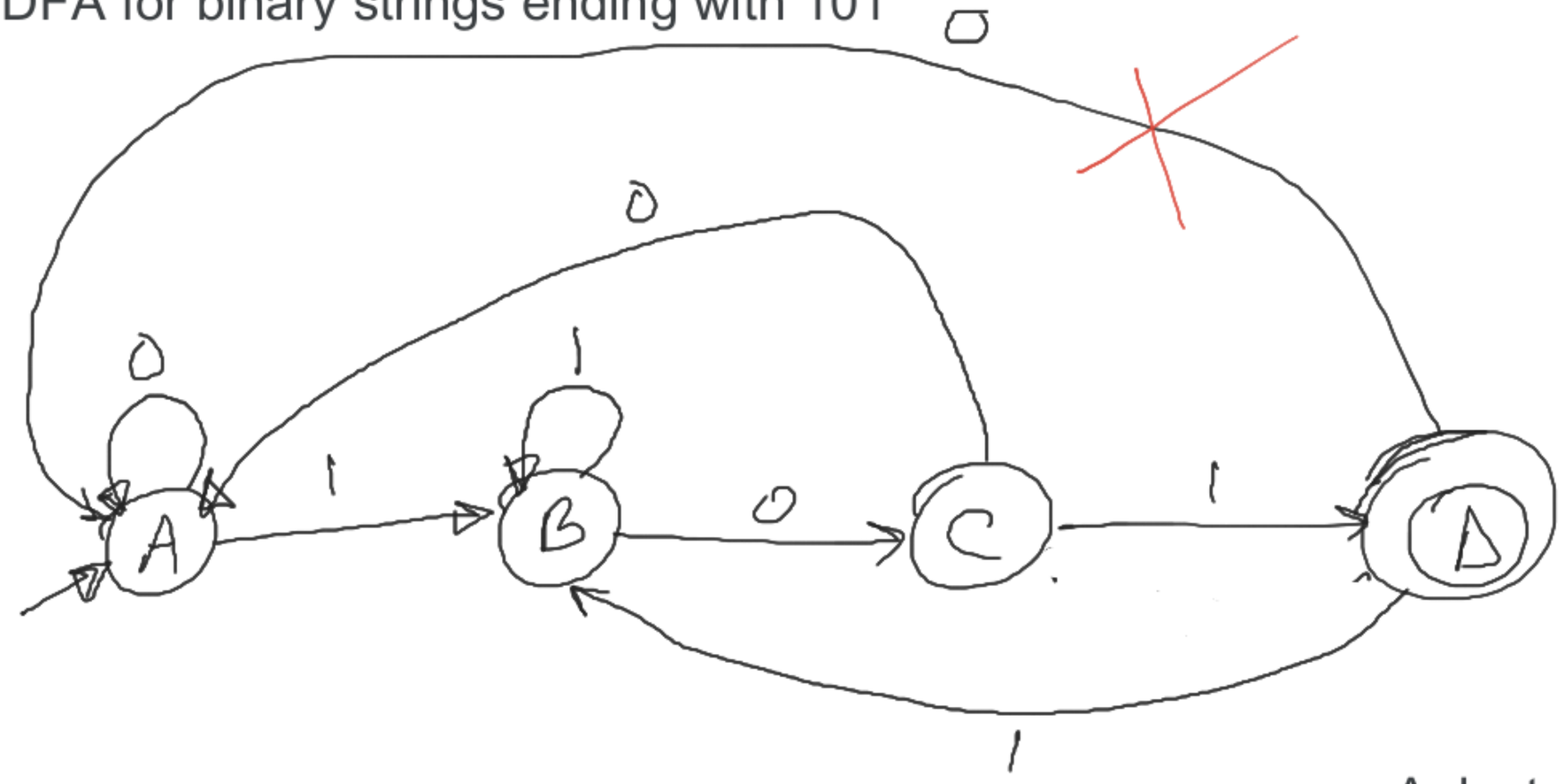
Incorrect DFA for binary strings ending with 101



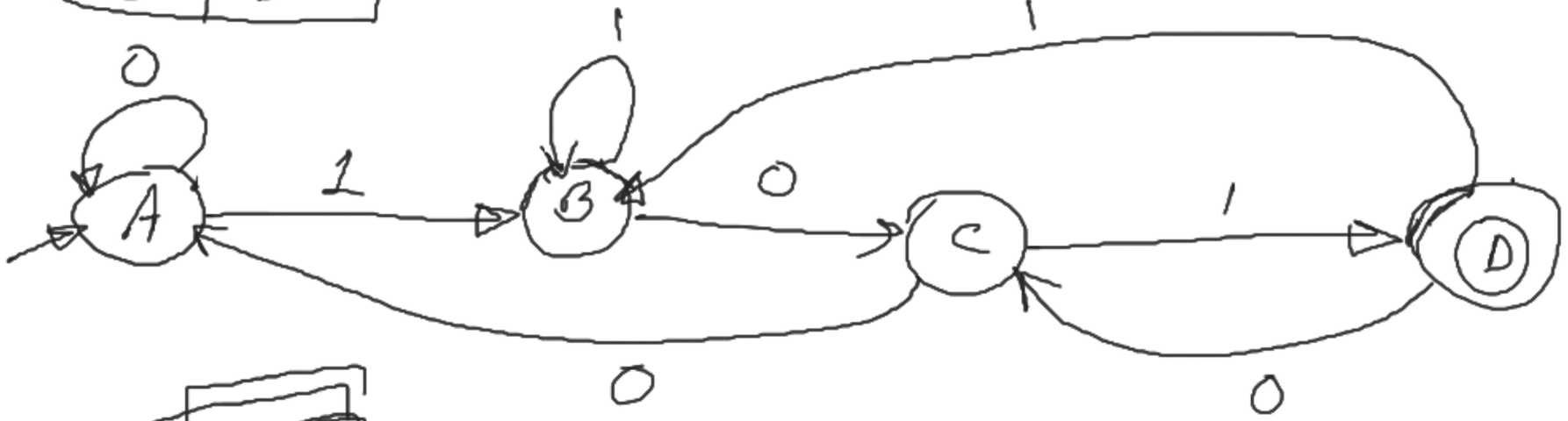
A: last char != 1
B: last 1 char = 1
C: last 2 chars = 10
D: last 3 chars = 101

10101 ⇒ rejected, but should be accepted

A B C D A B

a) DFA for binary strings ending with 101

	0	1
→A	A	B
B	C	B
C	A	D
*D	C	B

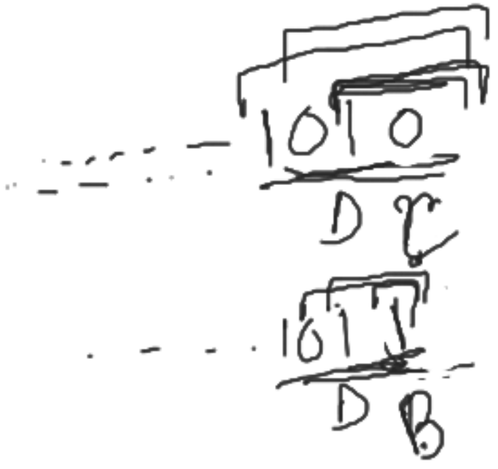


A: last char read is not 1

B: last char = 1

C: " 2 " = 10

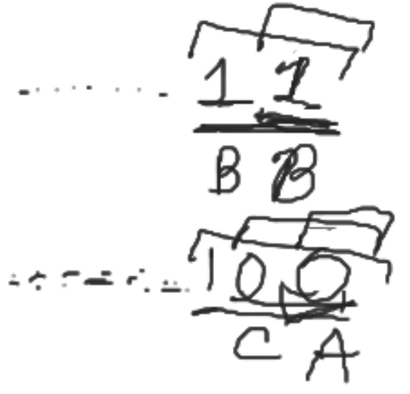
D: " 3 " = 101



10101 ⇒ Accept

A, B C D B

C D



A) DFA for binary strings, ending with 101

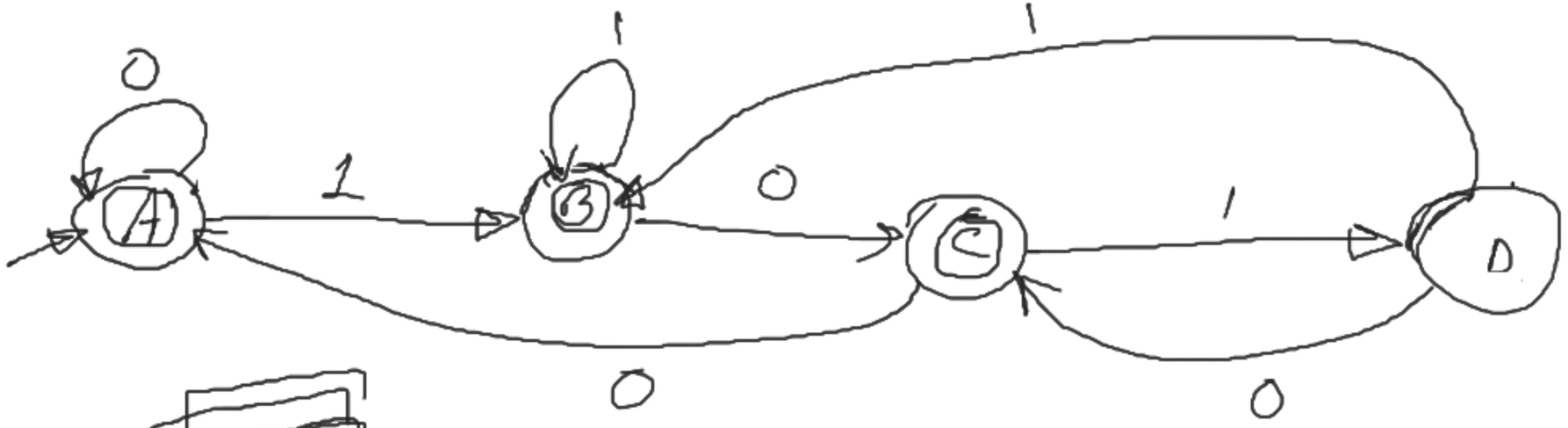
NOT

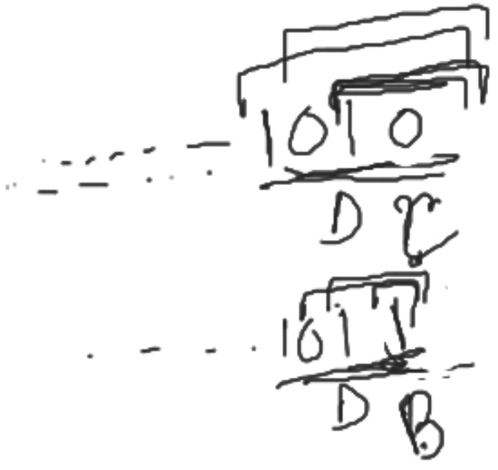
A: last char read is not 1

B: last char = 1

C: " 2 " = 10

D: " 3 " = 101

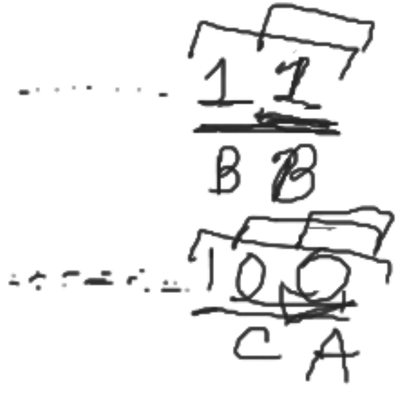




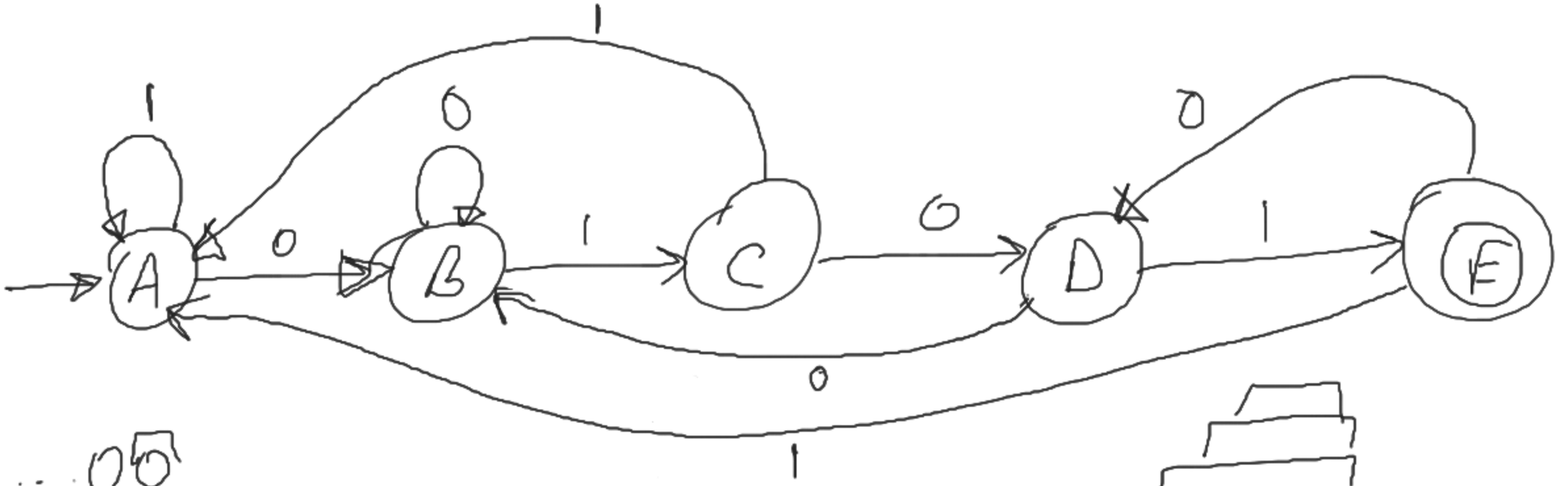
10101 ⇒ Accept

A, B C D B

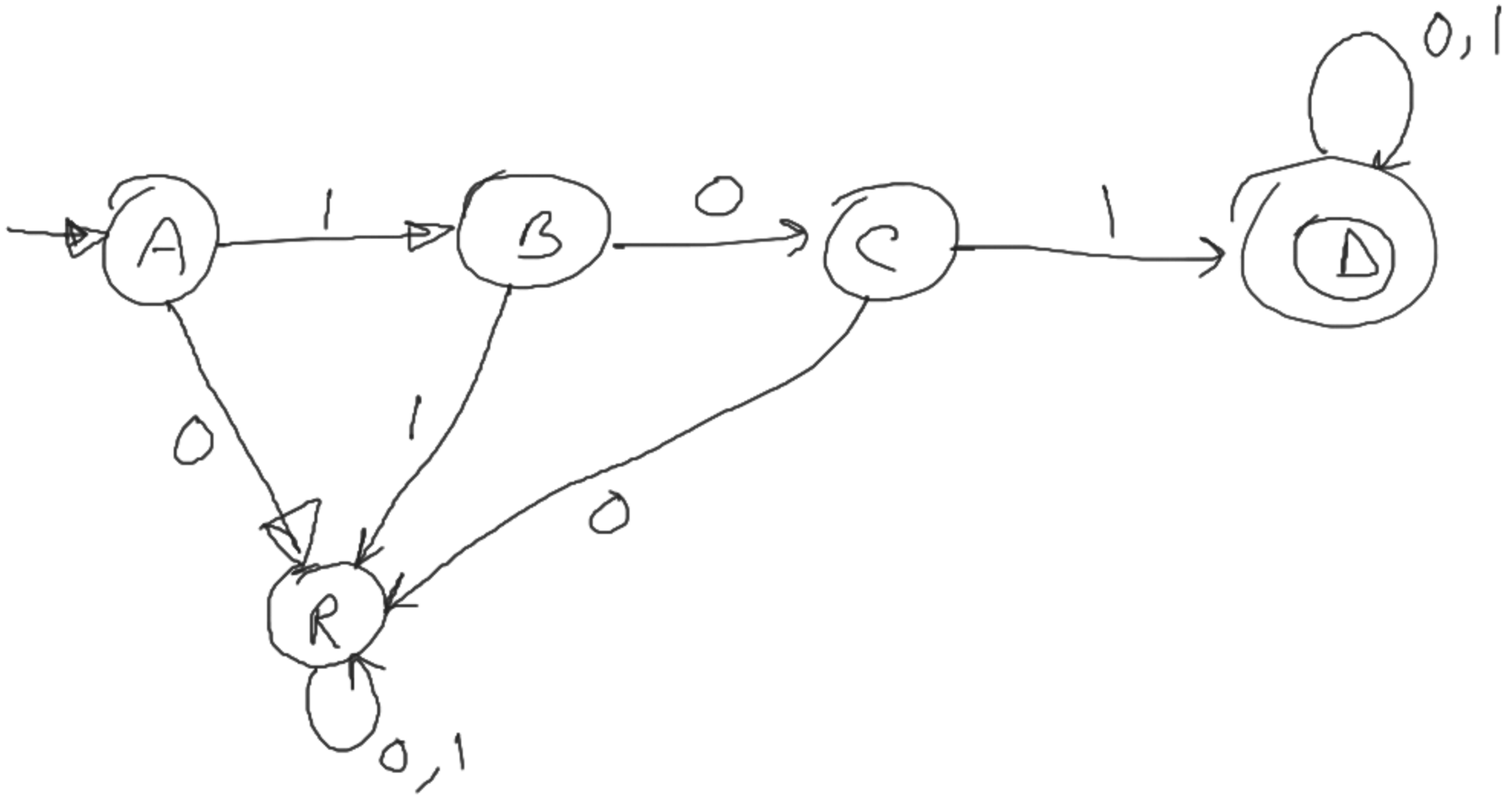
C D



Draw DFA for binary strings ending with 0101

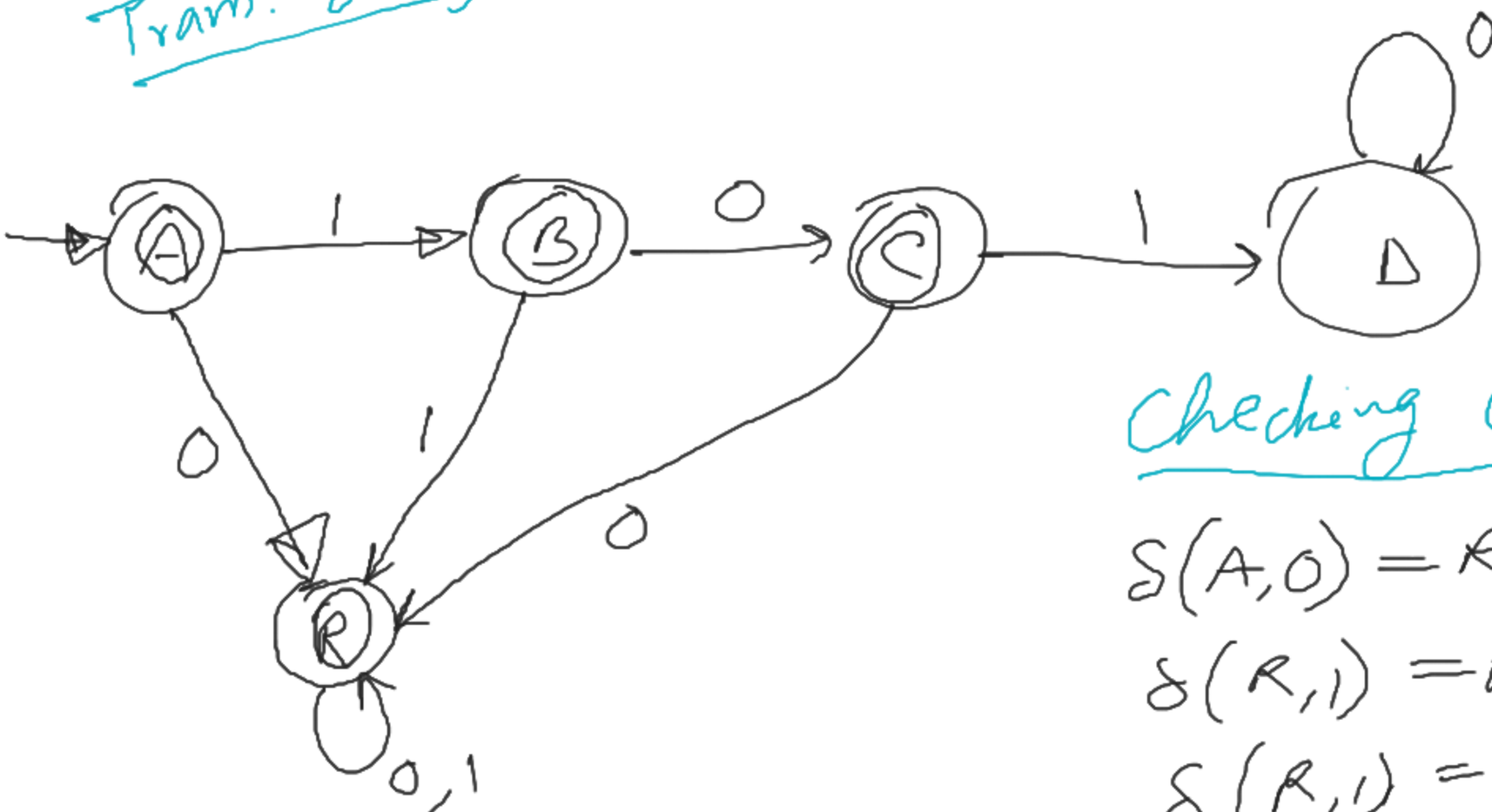


start with 101 $\Rightarrow RE = 101(0/1)^*$



B) do not start with 101

Tran. diagram



Trans. Table:

	0	1
→*A	R	B
*B	C	R
*C	R	D
D	D	D
*R	R	R

Checking 011:

$\delta(A,0) = R$

$\delta(R,1) = R$

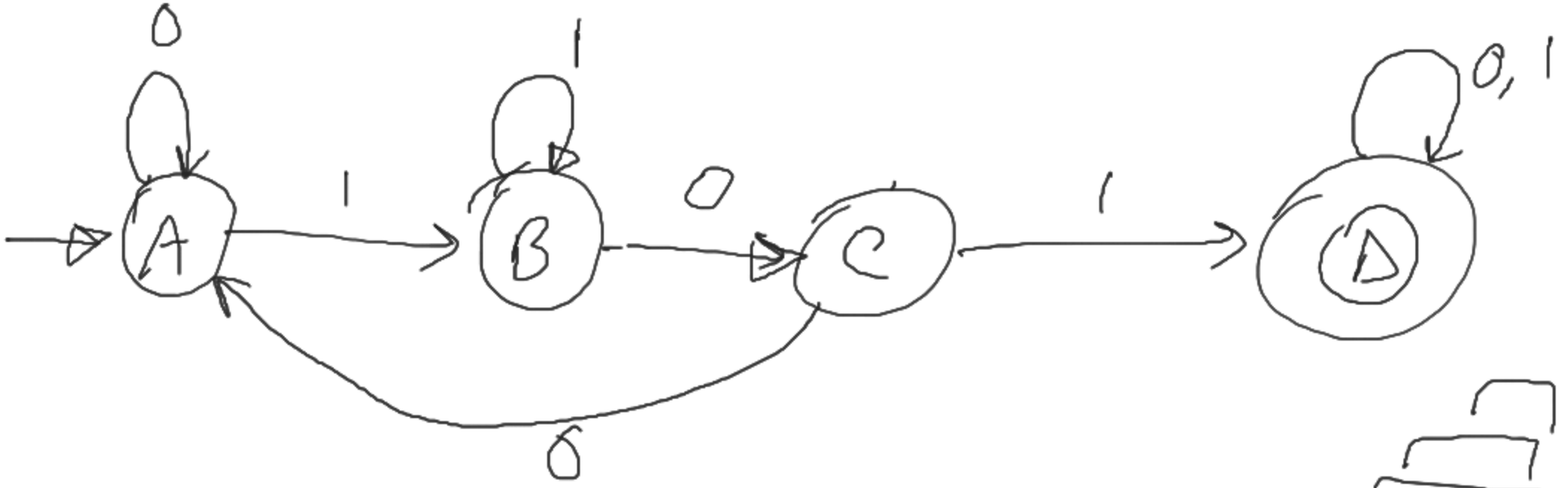
$\delta(R,1) = R \Rightarrow$ Accepted since R is a final state

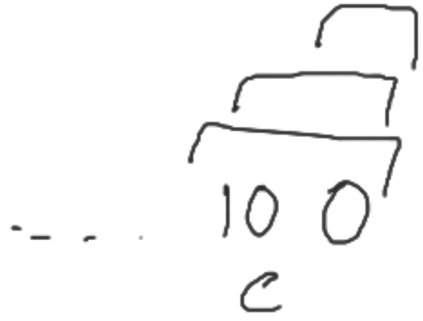
DFA for binary strings containing 101 as a substring

RE = $(0/1)^* 101 (0/1)^*$

end with 10

D is an accepting/final dead state





NOT

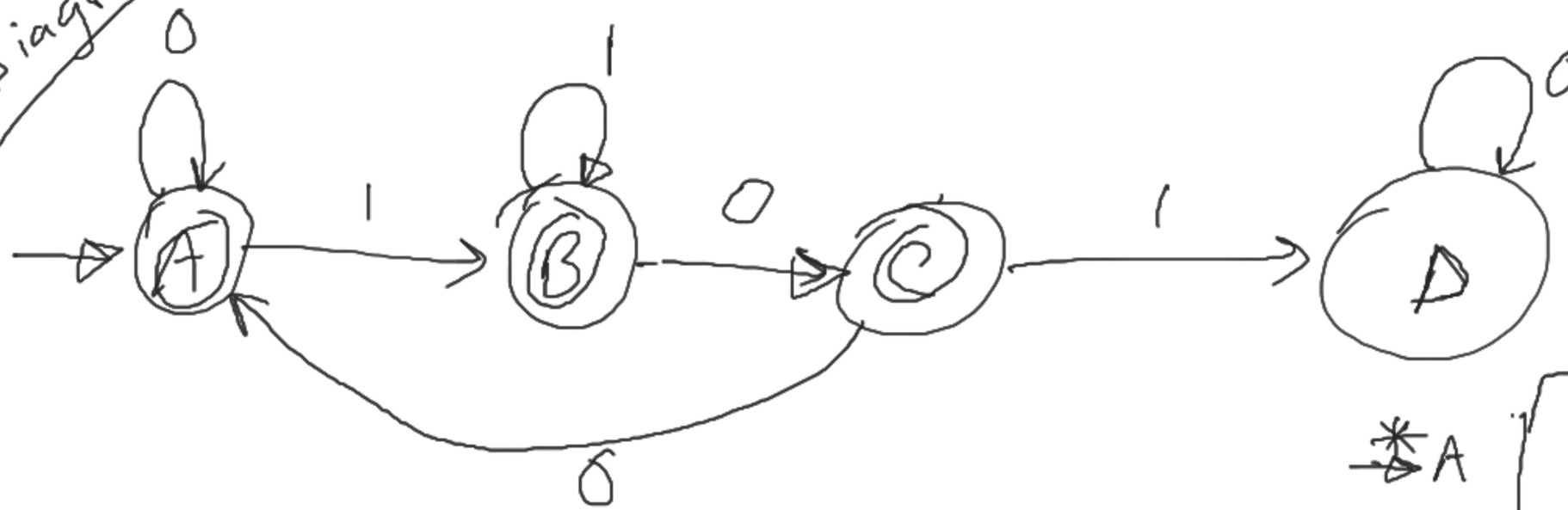
DFA for binary strings containing 101 as a substring

RE = $(0/1)^* 101 (0/1)^*$

end with 10

D is an accepting/final dead state

Trans. Diagram



Transition Table

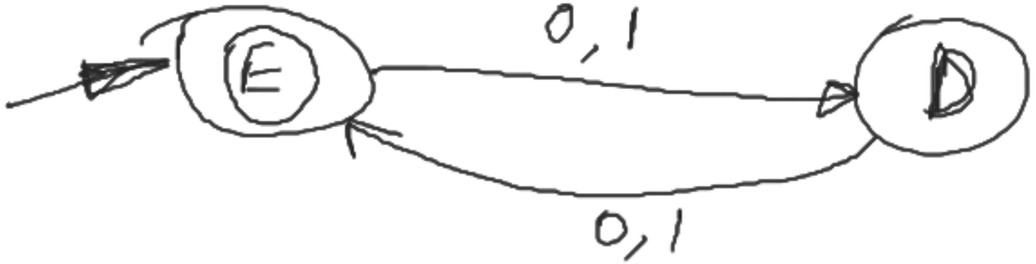
	0	1
*A	A	B
*B	C	B
*C	A	D
D	D	D

DFA for binary strings having even length

$\delta(E, 0) = D$ $\quad \delta(E, 0) = D$

$\delta(D, 1) = E$ $\quad \delta(D, 0) = E$

since E is a final state, 0100 is accepted



E : read even no. of bits so far

D : . . . . odd . . . . .

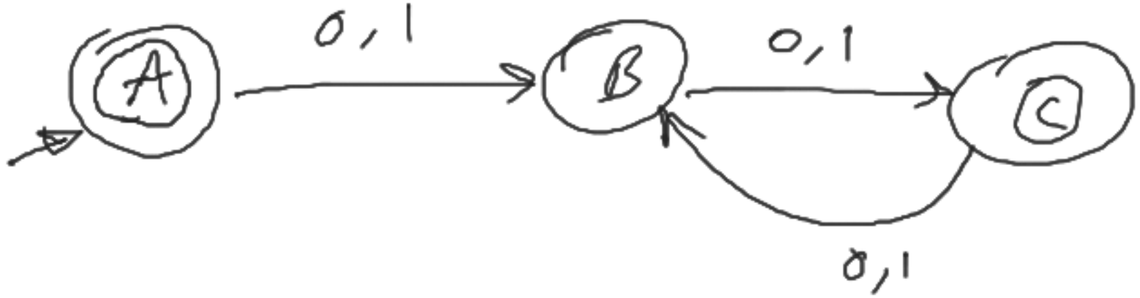
. . . .

Rough :

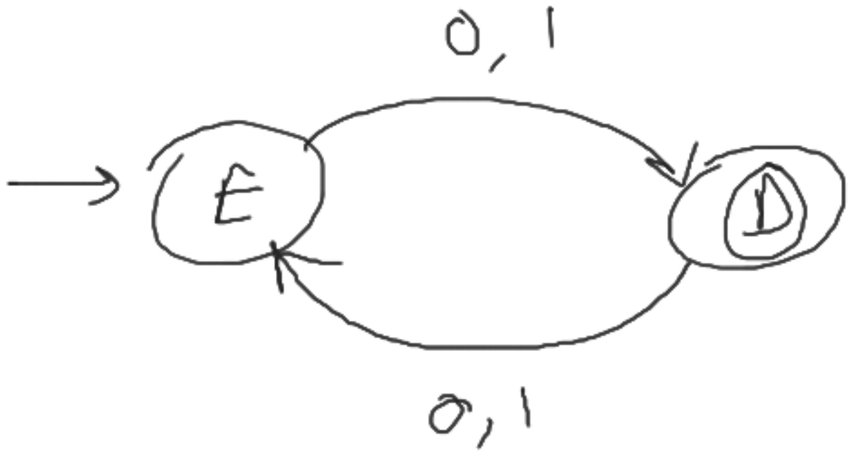
$|\Sigma| = 0$; an even no.

E D E D E ⇒ Accept (final)

ALT soln :

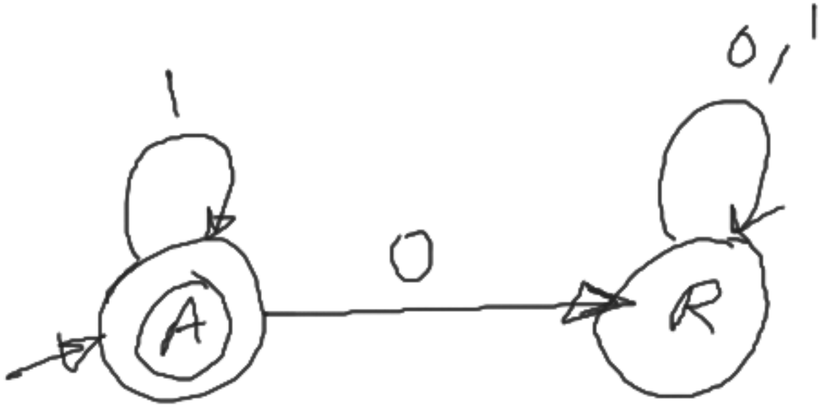


e) DFA for binary strings having odd length



DFA for binary strings having no 0s

$RE = 1^* = \varepsilon / 1^+$



Checking 1101 :

$\delta(A, 1) = A$

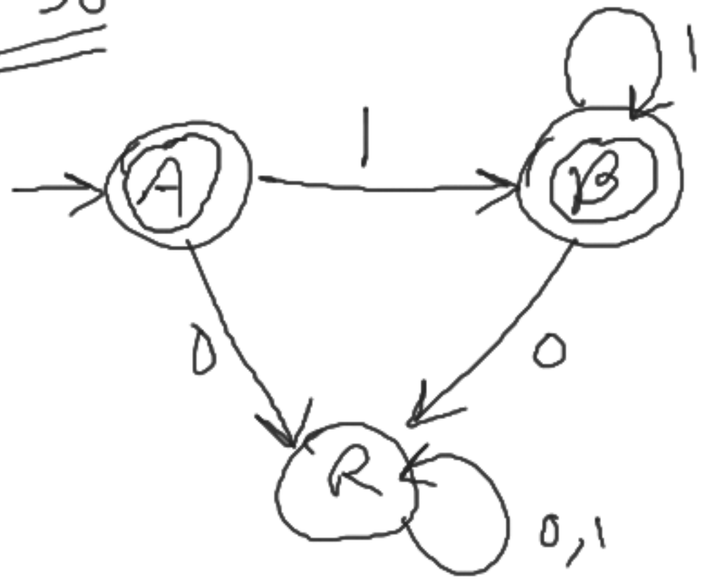
$\delta(A, 1) = A$

$\delta(A, 0) = R$

$\delta(R, 1) = R$

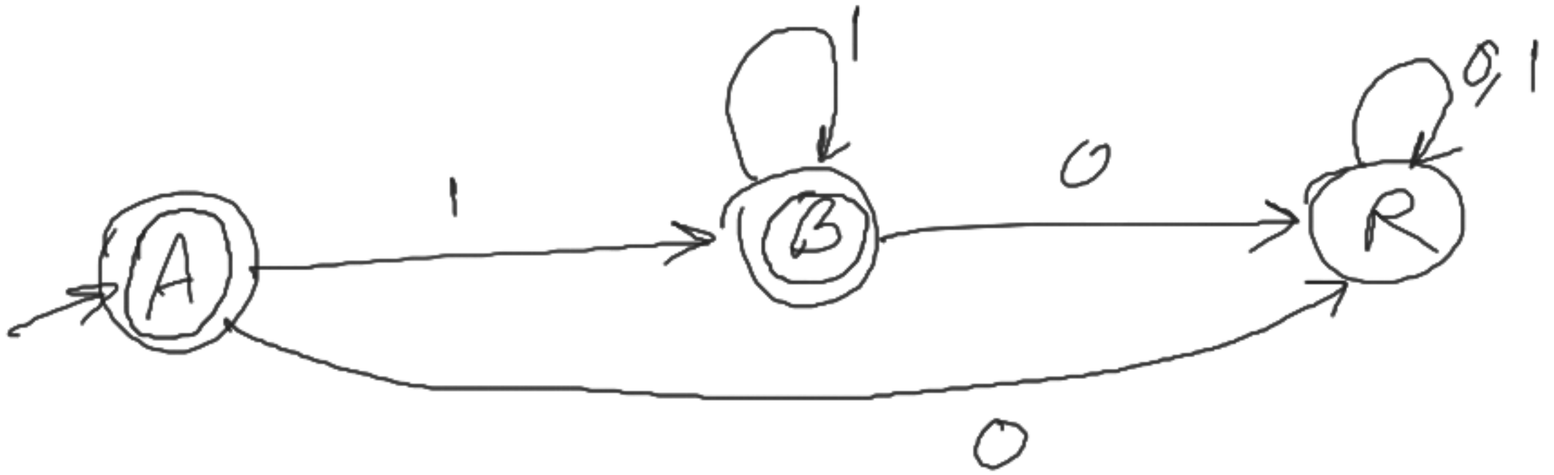
$\Rightarrow$ 1101 is rejected since R is not a final state

ALT. Soln:



DFA for binary strings having no 0s

ALT soln:



DFA for binary strings having exactly one 0

$RE = 1^*01^*$

R is a non-final dead state

Checking if DFA accepts the string 11010110:

$\delta(A, 1) = A$

$\delta(A, 1) = A$

$\delta(A, 0) = B$

$\delta(B, 1) = B$

$\delta(B, 0) = R$

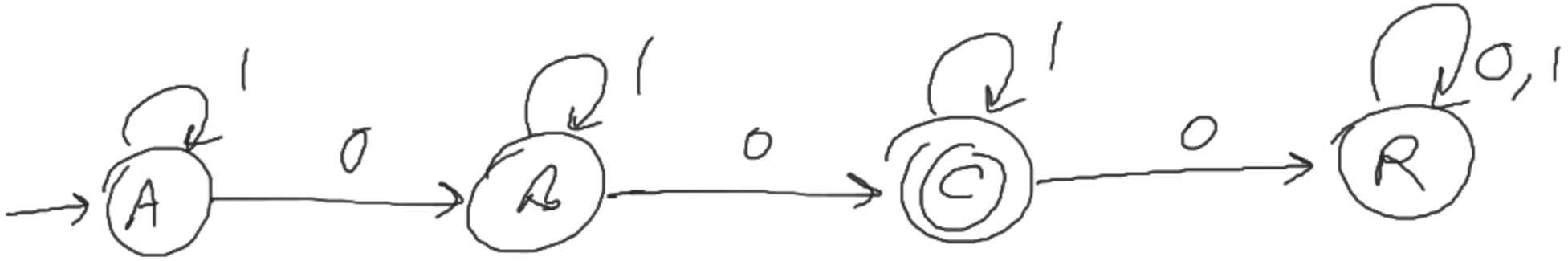
$\delta(R, 1) = R$

$\delta(R, 1) = R$

$\delta(R, 0) = R$

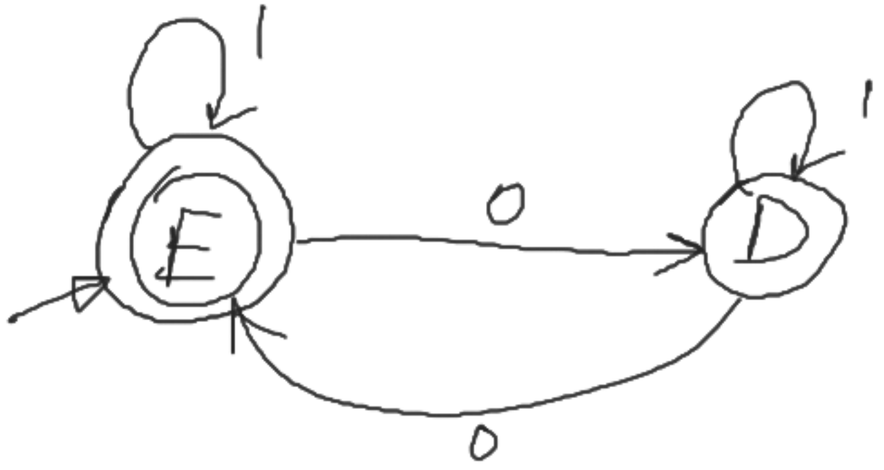
since R is non-final this string is rejected.

h) DFA for binary strings having exactly two 0s $RE = 1^*01^*01^*$



i) DFA for binary strings having even no. of 0s

Σ contains even(0) no. of 0s.



E : read even no. of 0s so far

D : - - - odd - - - - -

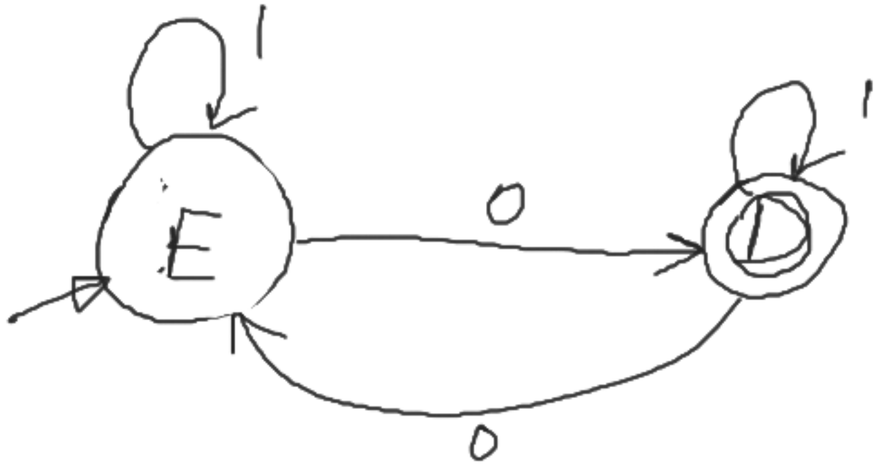
Rough:

110100

EEEDDED ⇒ rejected

DFA for binary strings having odd no. of 0s

Σ contains even(0) no. of 0s.



1 1 0 1 0 0

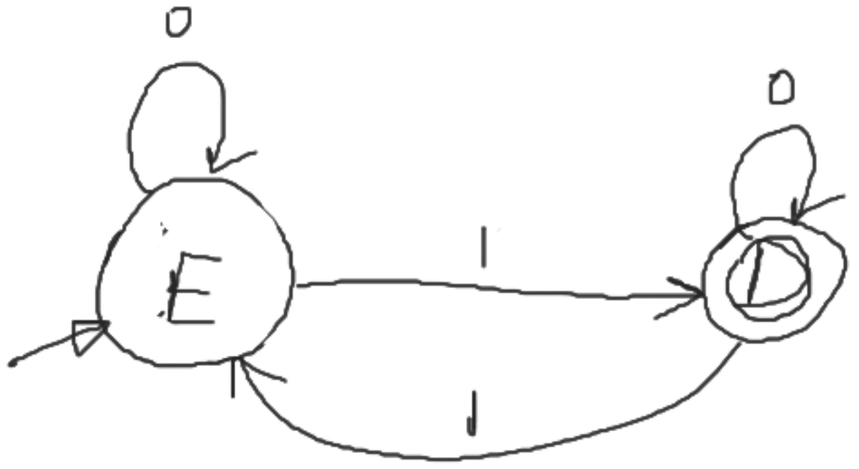
E E E D D E D ⇒ reject

E : read even no. of 0s so far

D : - - - odd - - — — -

j) DFA for binary strings having odd no. of 1s

Σ contains even(0) no. of 1s.



1 1 0 1 0 0

E E E D D E D ⇒ reject

E : read even no. of 1s so far

D : - - - odd - - — — -

m) DFA for binary strings having exactly three 1s:

$RE = 0^* 1 0^* 1 0^* 1 0^*$

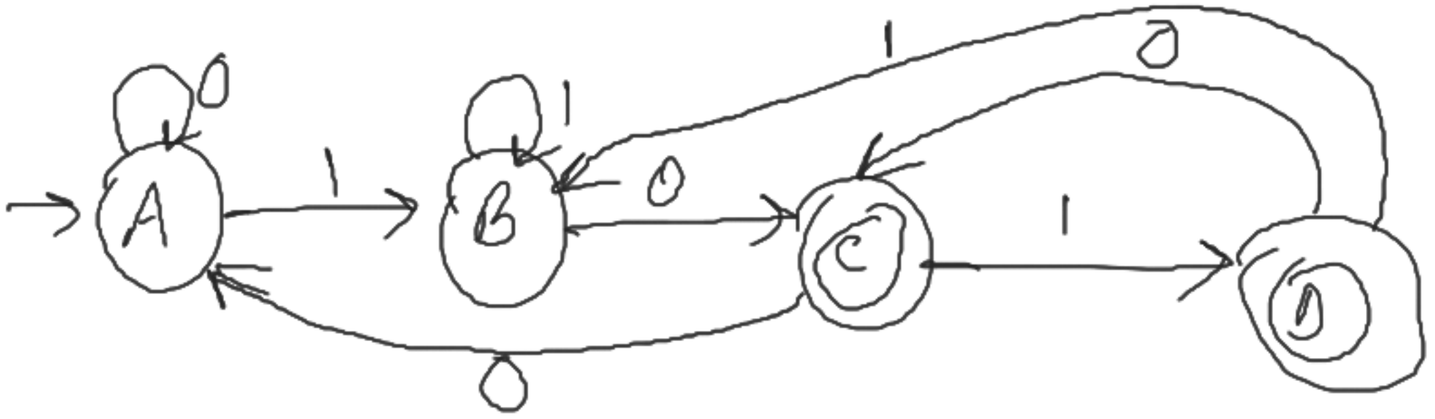
k) DFA for binary strings having at most three 1s:

$$RE = \underbrace{0^*10^*10^*10^*}_{C} / \underbrace{0^*10^*10^*}_{B} / \underbrace{0^*10^*}_{A} / 0^*$$

I) DFA for binary strings having at least three 1s:

$RE = 0^* 1 0^* 1 0^* 1 (0/1)^*$

DFA for binary strings ending with 10 or 101



$RE = (0/1)^* (10/101)$

$= (0/1)^* 10 (\varepsilon/1)$

B C D

* *

$\delta(A, 1010) = ?$