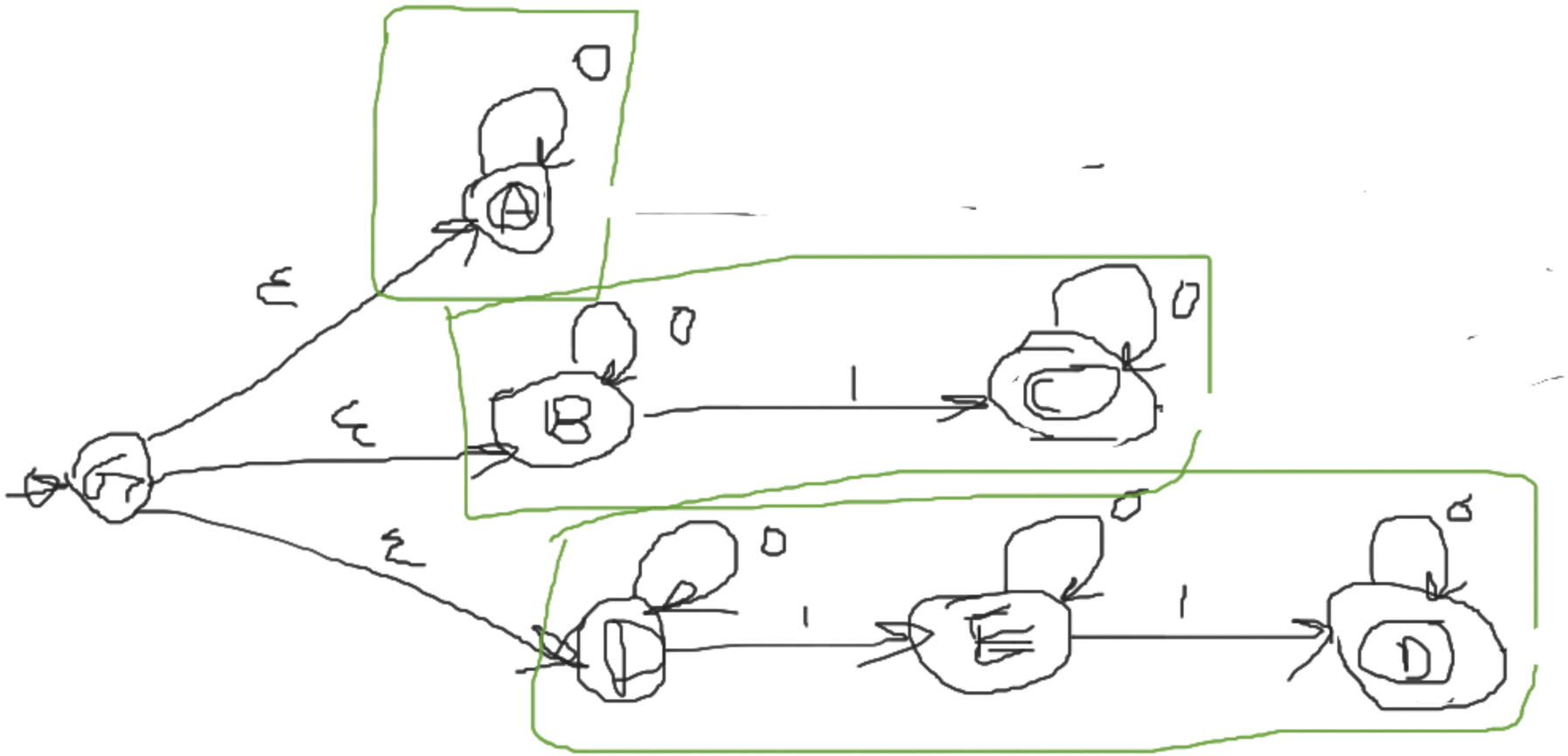
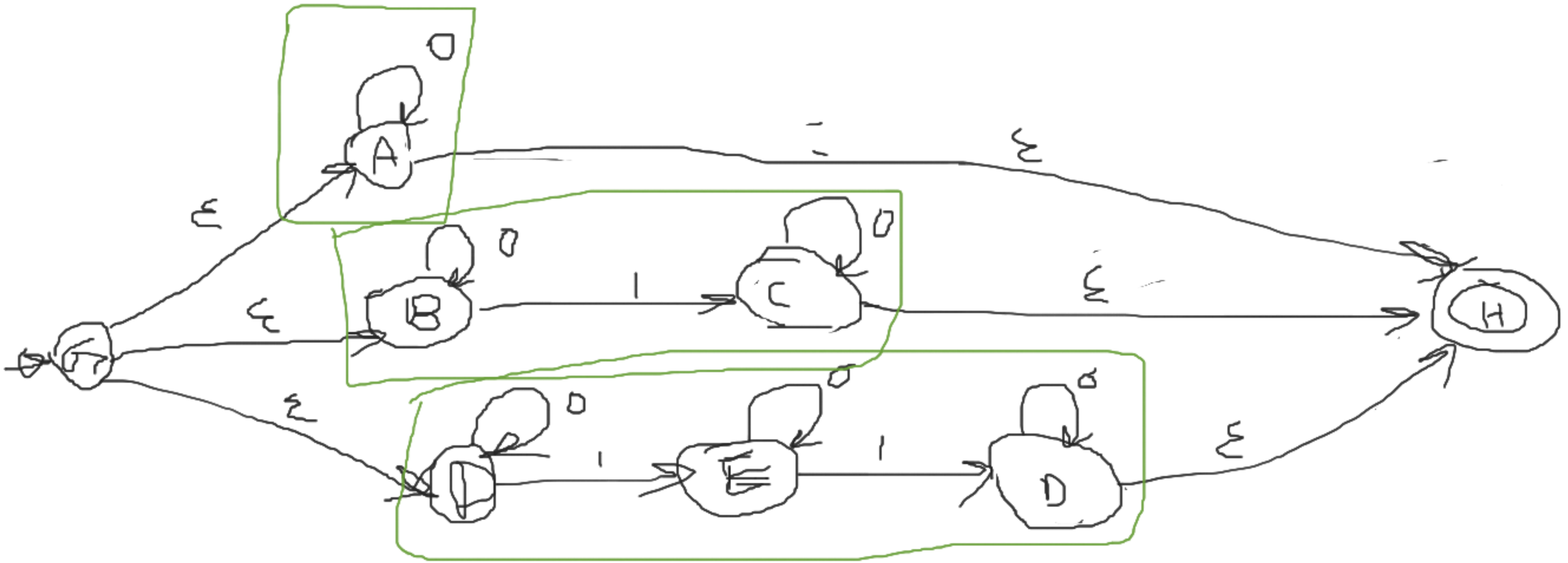
L = set of binary stirng containing at most two 1s

RE = 0*/0*10*/0*10*10*

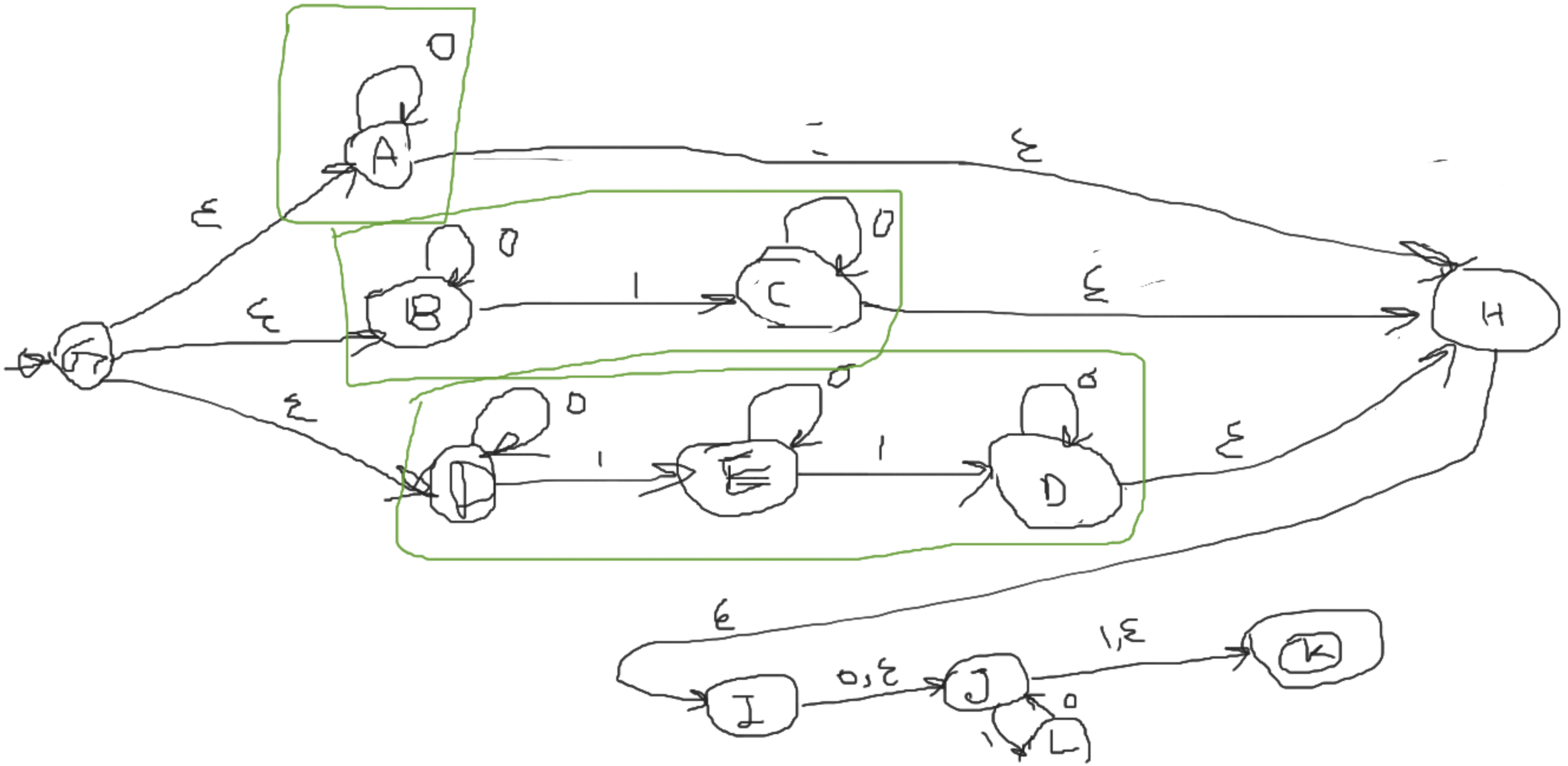


L = set of binary stirng containing at most two 1s

RE = 0*/0*10*/0*10*10*



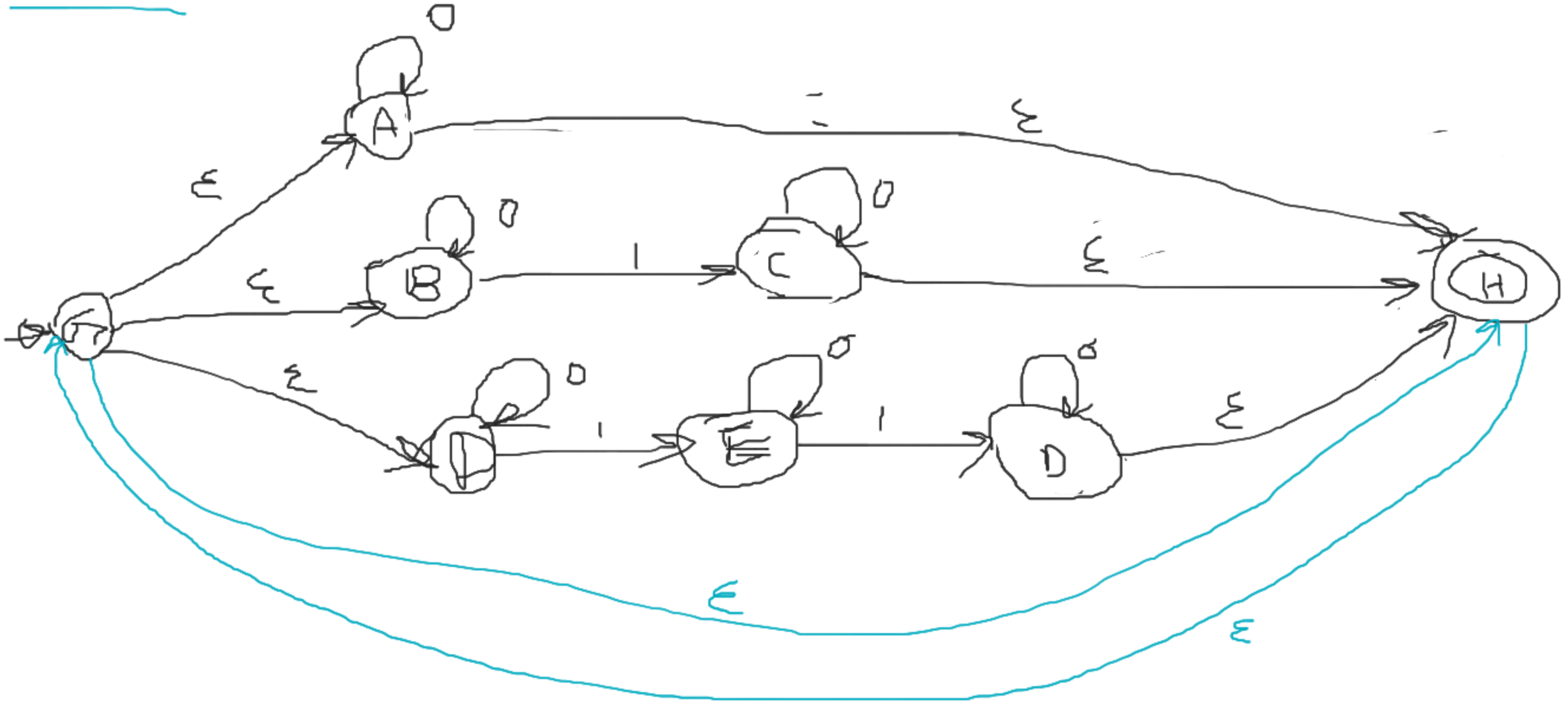
Design an eNFA for L1L2 (using closure properties)where L1 = set of binary strings containing at most two 1s, L2 = set of binary strings which are alternating sequences of 0s and 1s.



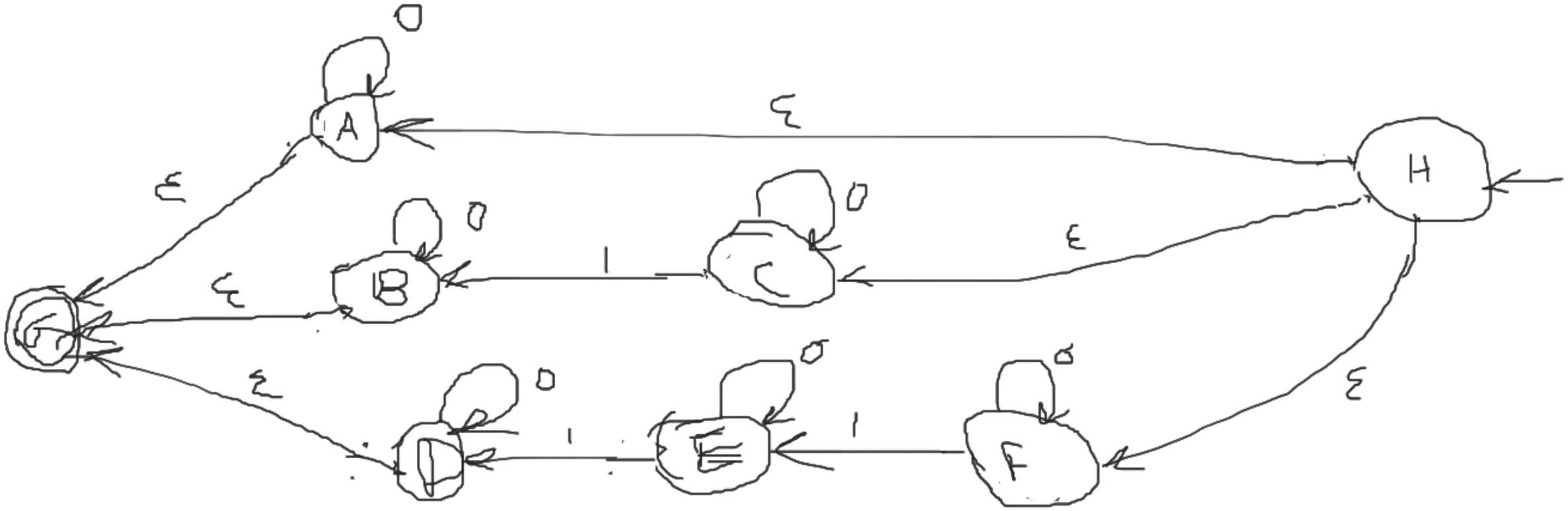
L = set of binary stirng containing at most two 1s

RE = 0*/0*10*/0*10*10*

eNFA for L*:



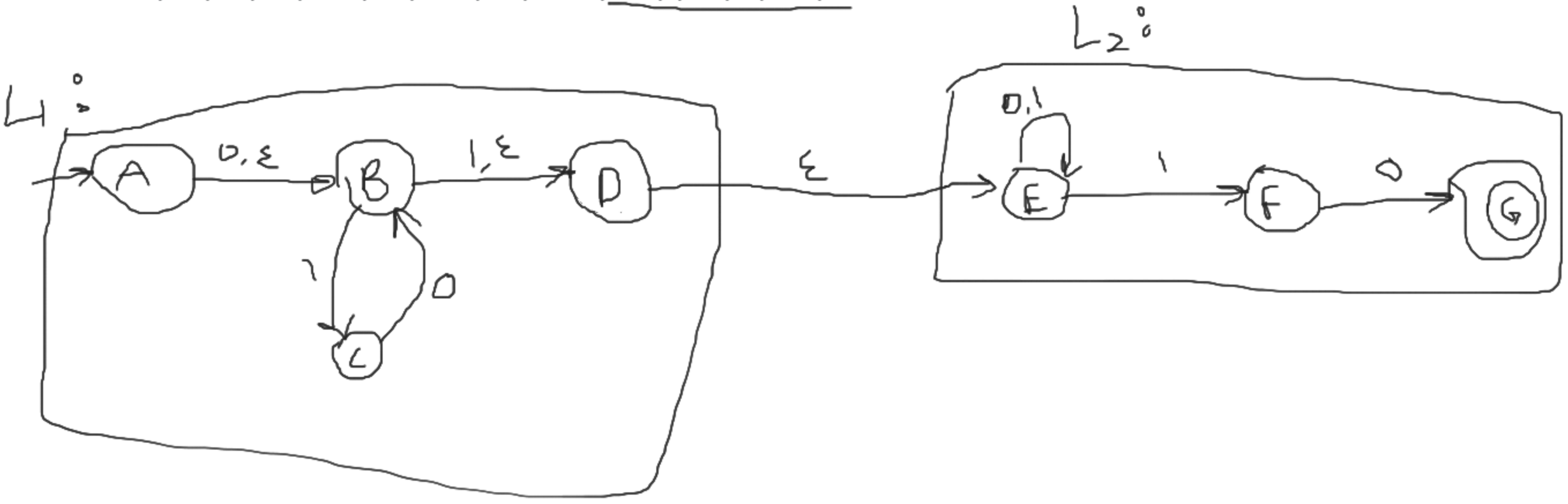
Design an eNFA for L^R where L = set of binary strings which contain at most two 1s.



Design an eNFA for L1 = set of binary strings which are alternating sequences of 0s and 1s.
Design another eNFA for L2 = set of binary strings which end with 10.
Then design an eNFA for L1L2.

RE for L2 = (0/1)*10

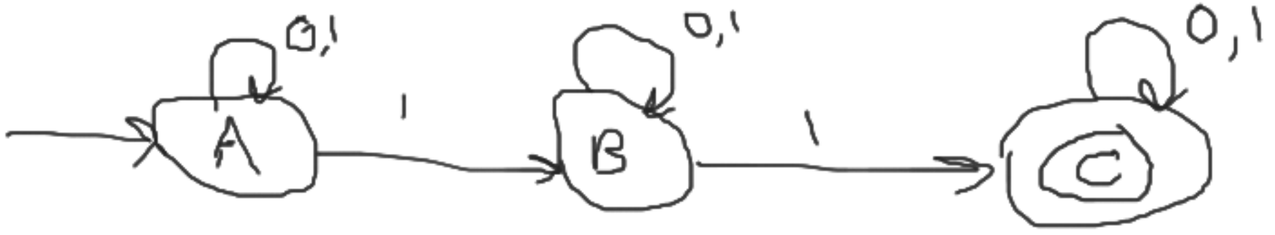
L1 = {e, 0, 1, 10, 01, 101, 010, 1010, 0101, ...}.
RE = (10)*/(01)*/0(10)*/1(01)* = (0/e)(10)*(1/e)



Design an eNFA for L^R where L is the set of binary strings which contain at least two 1s.

RE for L = (0/1)*1(0/1)*1(0/1)*

=> NFA:

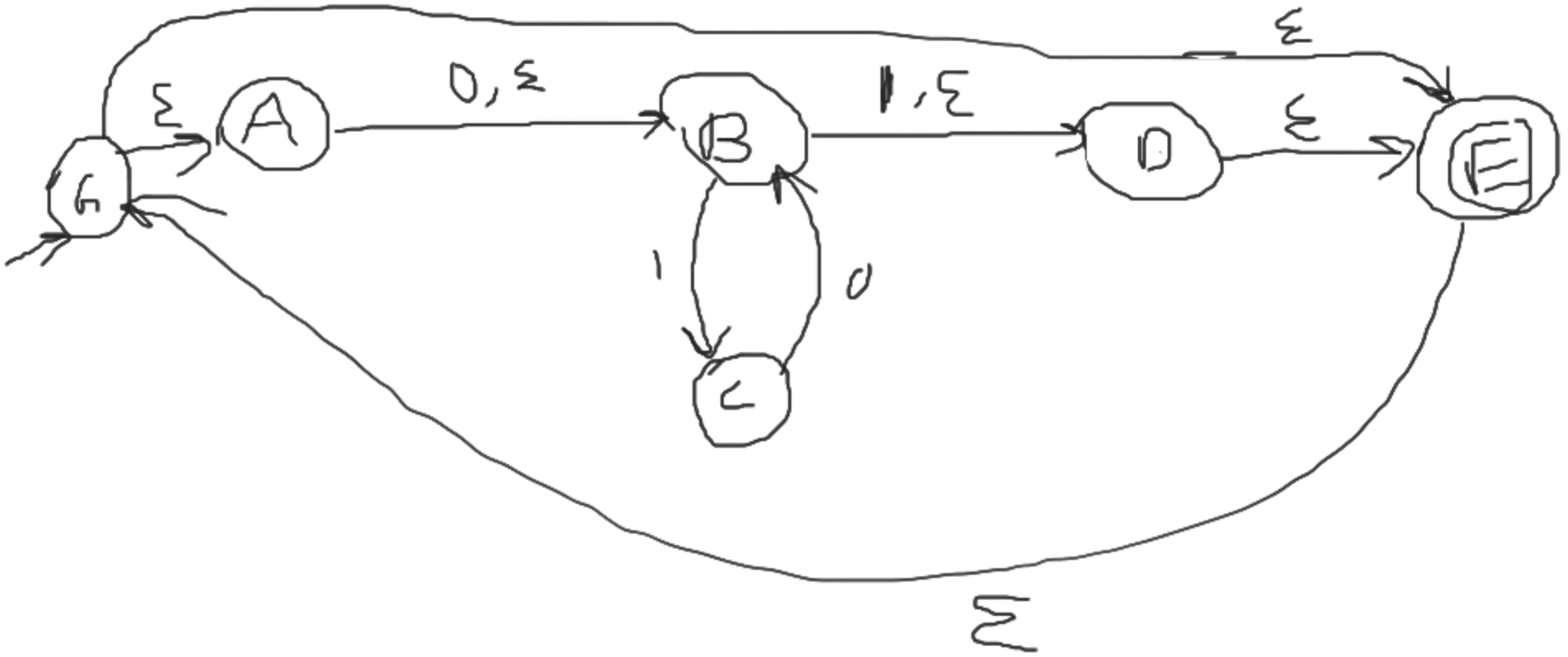


NFA for L^R:

Design an eNFA for L = set of alternating sequence of 0s and 1s. Then use that to design an eNFA for L*

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

eNFA for L*:



Design an eNFA for L = set of alternating sequence of 0s and 1s. Then use that to design an eNFA for L^R

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

eNFA for L^R: