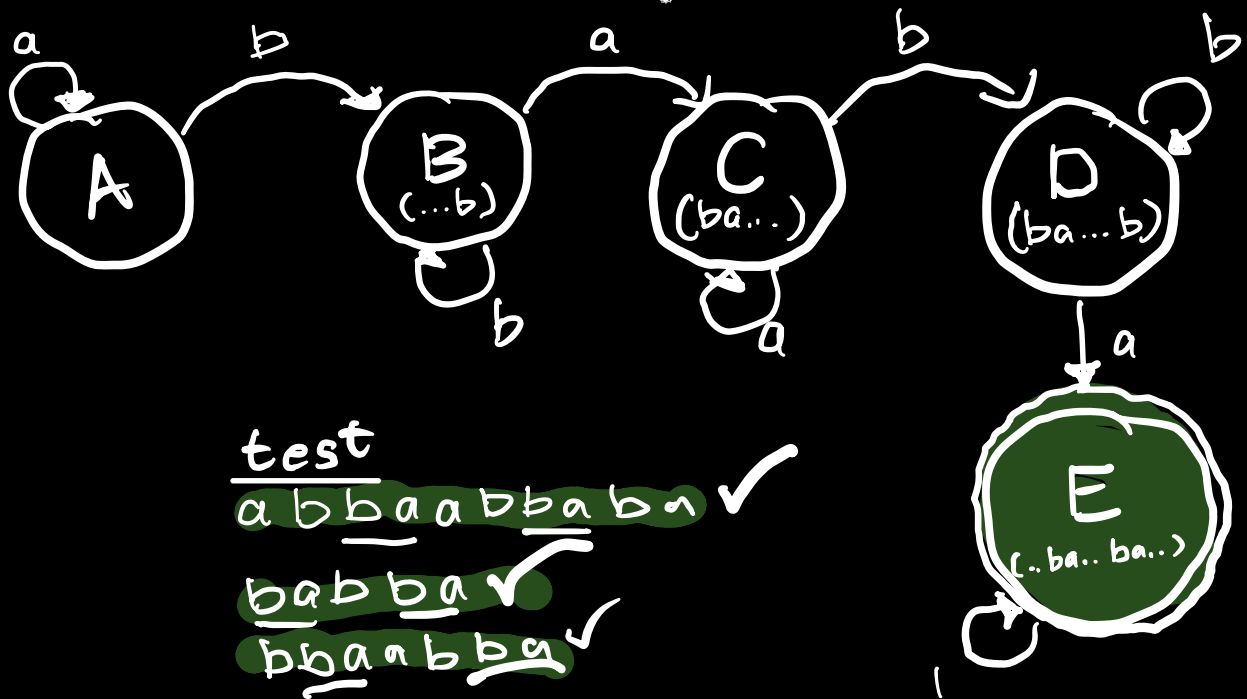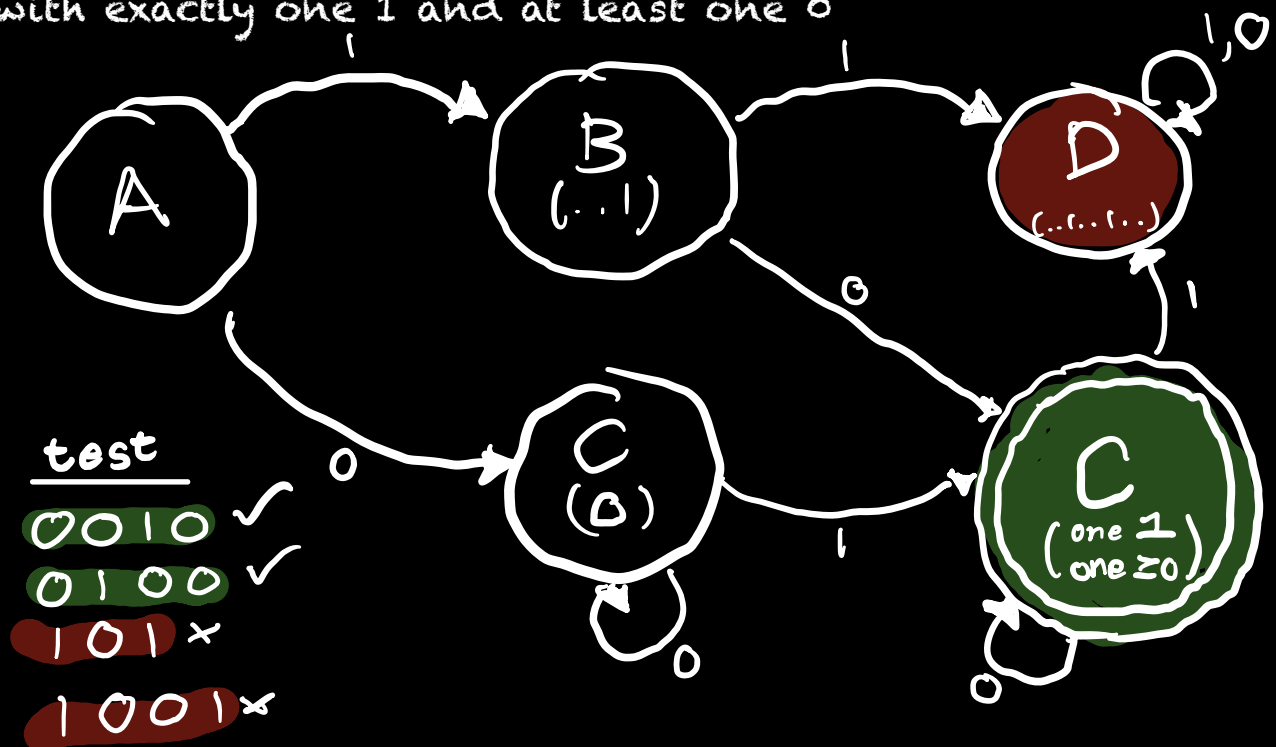1) Design a DFA for the language over the alphabet {a,b} for all the strings with at least two occurrences of ba. The two occurrences can be anywhere in the string.

A
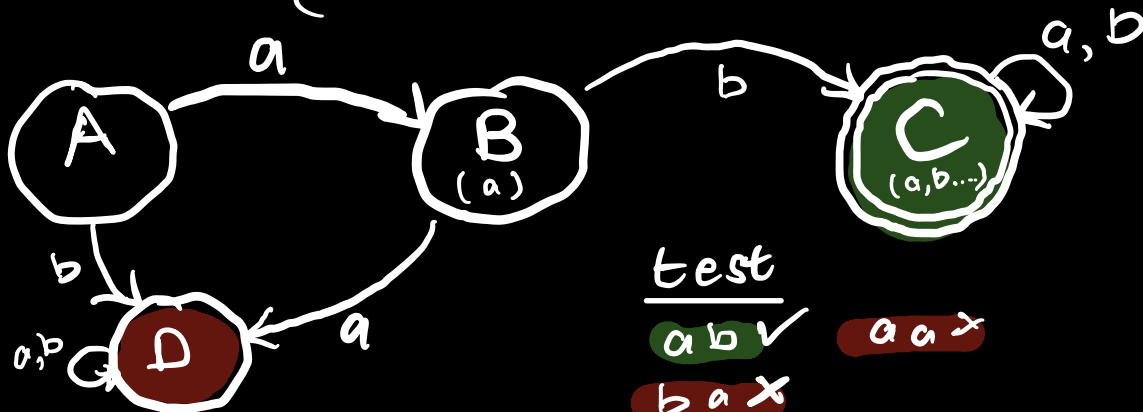
B
(...b)

C
(ba...)

D
(ba...b)

E
(..ba..ba..)

a   b   a   b   b

b

a

a

test

a̲ b̲ b̲ a̲ a̲ b̲ b̲ a̲ b̲ a̲ ✓

b̲ a̲ b̲ b̲ a̲ ✓

b̲ b̲ a̲ a̲ b̲ b̲ a̲ ✓

2) Design a DFA over the alphabet {0,1} for all the strings with exactly one 1 and at least one 0

A

B
(..1)

D
(..1..1..)

C
(0)

C
(one 1
one ≥0)

1   1   1,0

0   1

0   0

1

test

0̲0̲1̲0̲ ✓
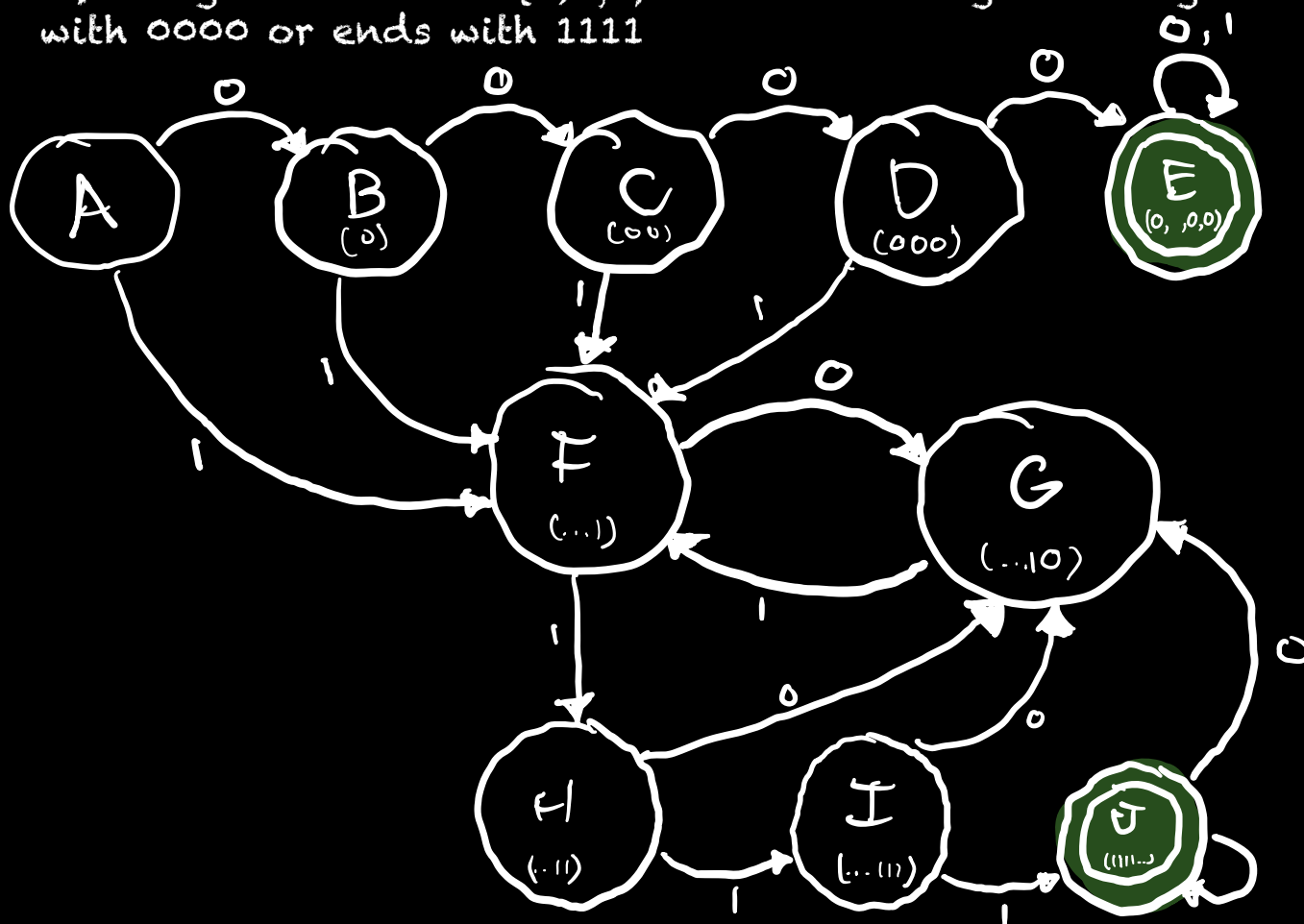
0̲1̲0̲0̲ ✓

1̲0̲1̲ ✗

1̲0̲0̲1̲ ✗

3) Design a language L over {a,b} for all the strings starting with ab

$$L = \{ab, aba, abb, abba \ldots\}$$
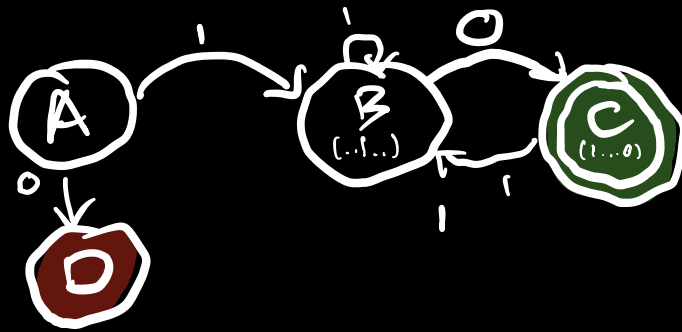

test

ab ✓   aa ✗

ba ✗

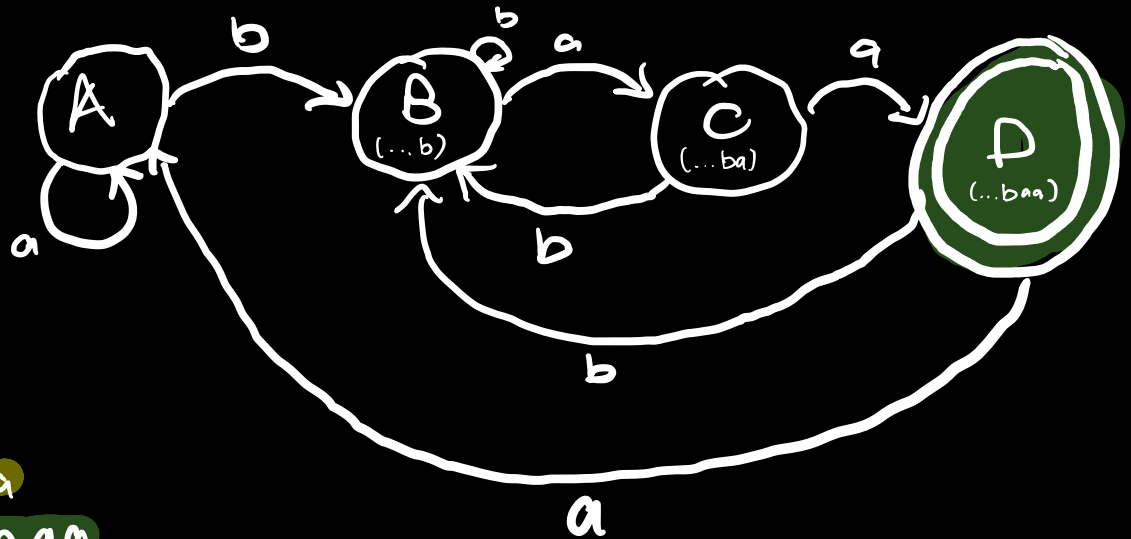4) Design a DFA over {0,1} for all the strings that begin with 0000 or ends with 1111

5) Design a DFA over {0,1} for all the strings that start with 1 and ends with 0
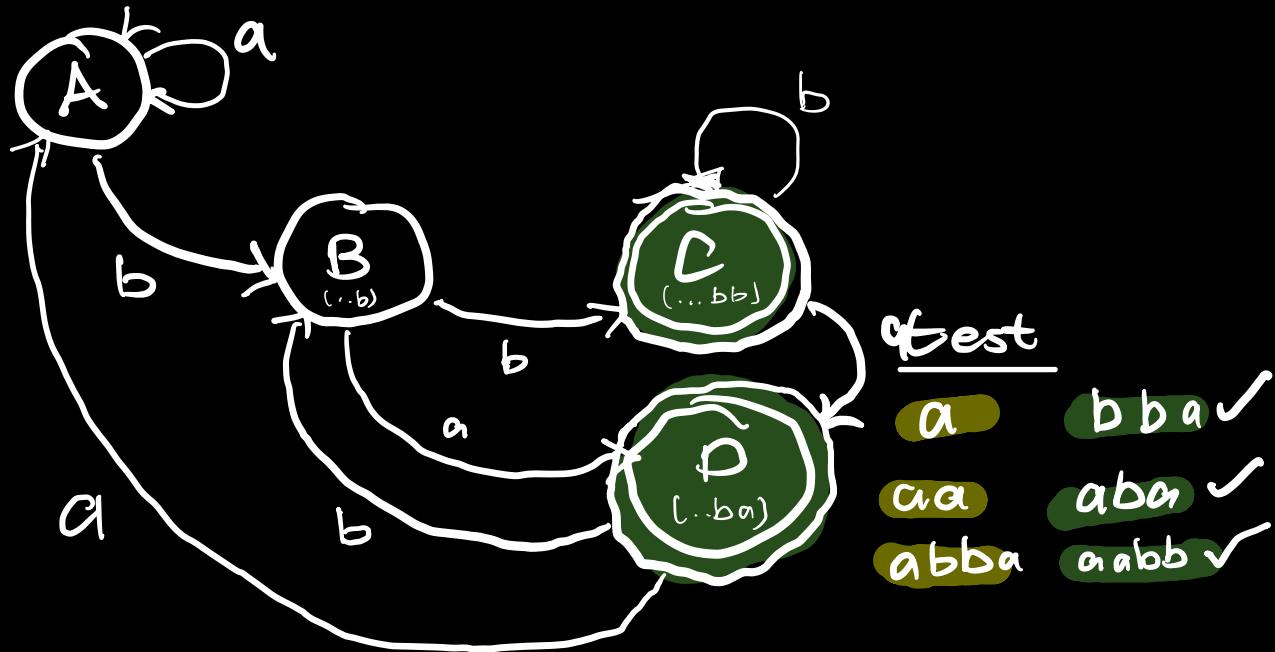

test

0 ✗
10 ✓
1010 ✓
1110 ✓

10110

6) Design a DFA over {a,b} for all the strings ending with baa
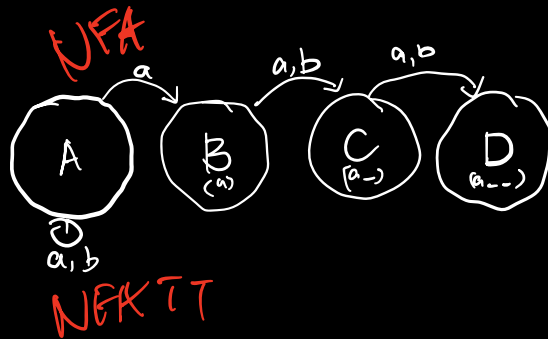

test

a ba
aa bb
aab aba
abaa abbaa

7) Design a DFA for all the strings where the second to last symbol of the word is b. Therefore, all the strings are ending with bb or ba

a

A

b

B
(..b)

b

C
(...bb)

a

a

b

b

D
(..ba)

Test

a   bba ✓
aa   aba ✓
abba   aabb ✓

8) Design a DFA set over all the strings over {a,b} for all the strings that the third symbol to the right is an a, meaning all the strings must end in one of the following: abb, aaa, aab, aba

NFA

NFA TT

	a	b
A	A,B	A
B	C	C
C	D	D
D	∅	∅

DFA TT

	a	b
A	A,B	A
AB	ABC	AC
AC	ABD	AD
AD	AB	A
ABC	ABCD	ACD
ABD	ABC	AC
ACD	ABD	AD
ABCD	ABCD	ACD

DFA