DFA AND NFA

NAME: ATHRESH KUMAR LABDE
(RA1911033010146)
M1 CSE(S.E)

NFA to DFA

Steps for converting NFA to DFA:

Step 1: Initially Q' = ϕ

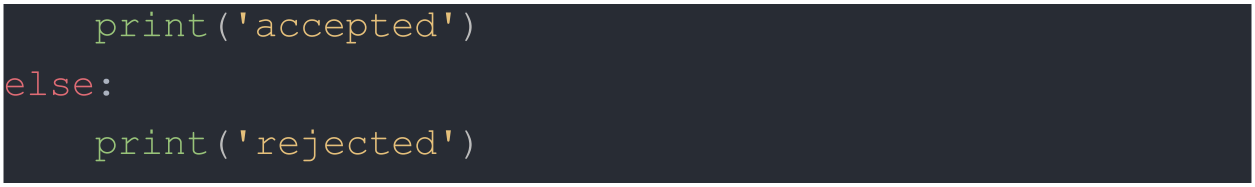
Step 2: Add q0 of NFA to Q'. Then find the transitions from this start state.

Step 3: In Q', find the possible set of states for each input symbol. If this set of states is not in Q', then add it to Q'.

Step 4: In DFA, the final state will be all the states which contain F(final states of NFA)

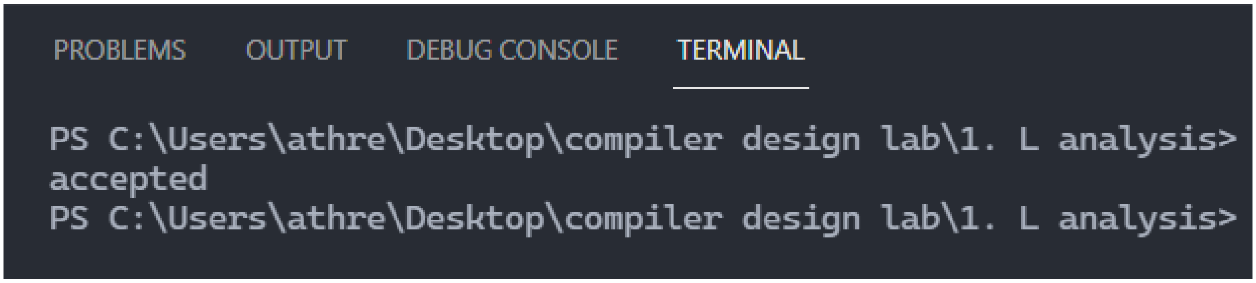
Code:

```
from automata.fa.dfa import DFA
dfa = DFA(
    states={'q0', 'q1', 'q2'},
    input_symbols={'0', '1'},
    transitions={
        'q0': {'0': 'q0', '1': 'q1'},
        'q1': {'0': 'q2', '1': 'q1'},
        'q2': {'0': 'q2', '1': 'q1'}
    },
    initial_state='q0',
    final_states={'q2'}
)
my_input_str = '010'
dfa.read_input_stepwise(my_input_str)
if dfa.accepts_input(my_input_str):
```



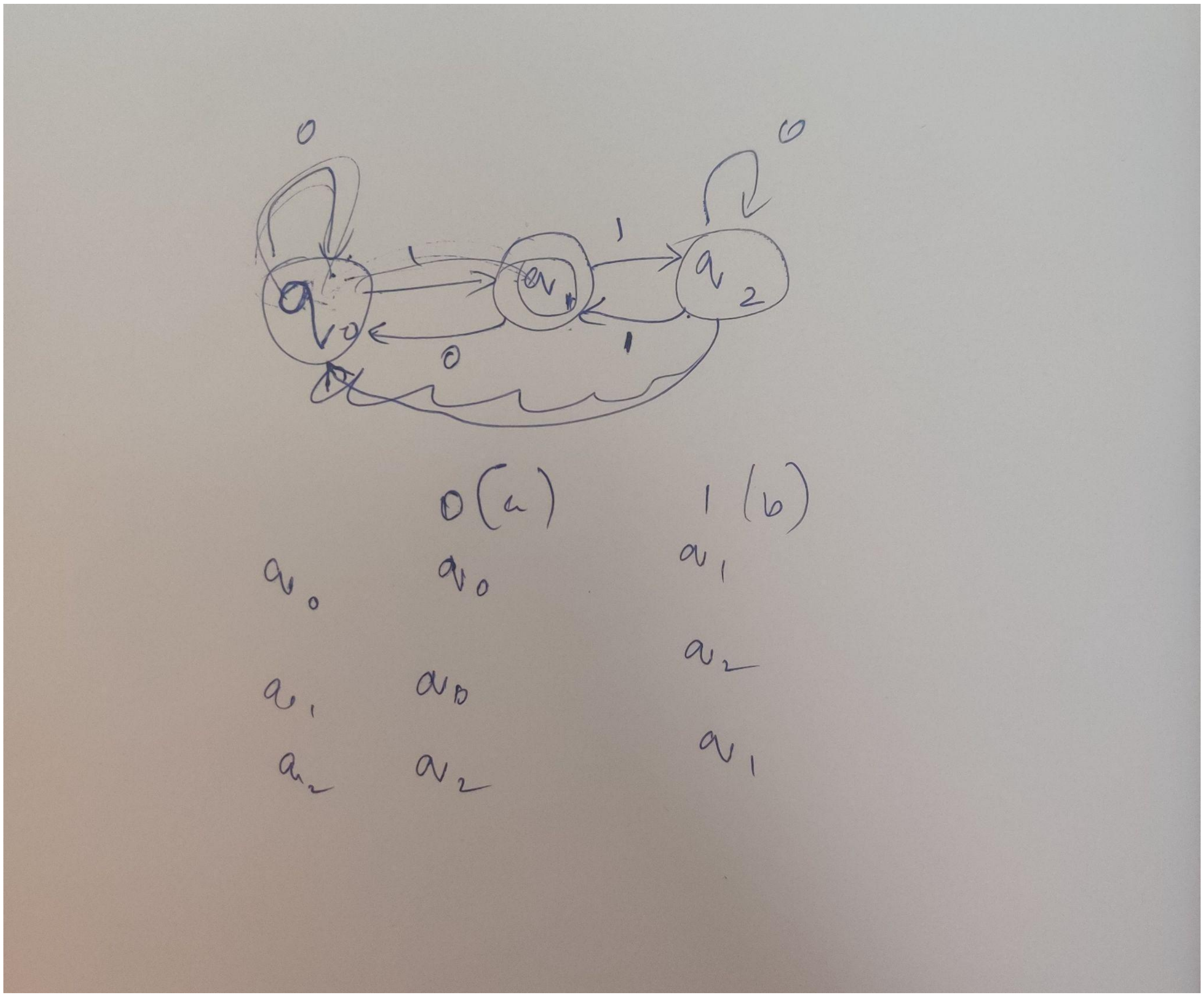
```
    print('accepted')
else:
    print('rejected')
```

INPUT - 010
OUTPUT -



```
PS C:\Users\athre\Desktop\compiler design lab\1. L analysis>
accepted
PS C:\Users\athre\Desktop\compiler design lab\1. L analysis>
```

Rough Diagram:



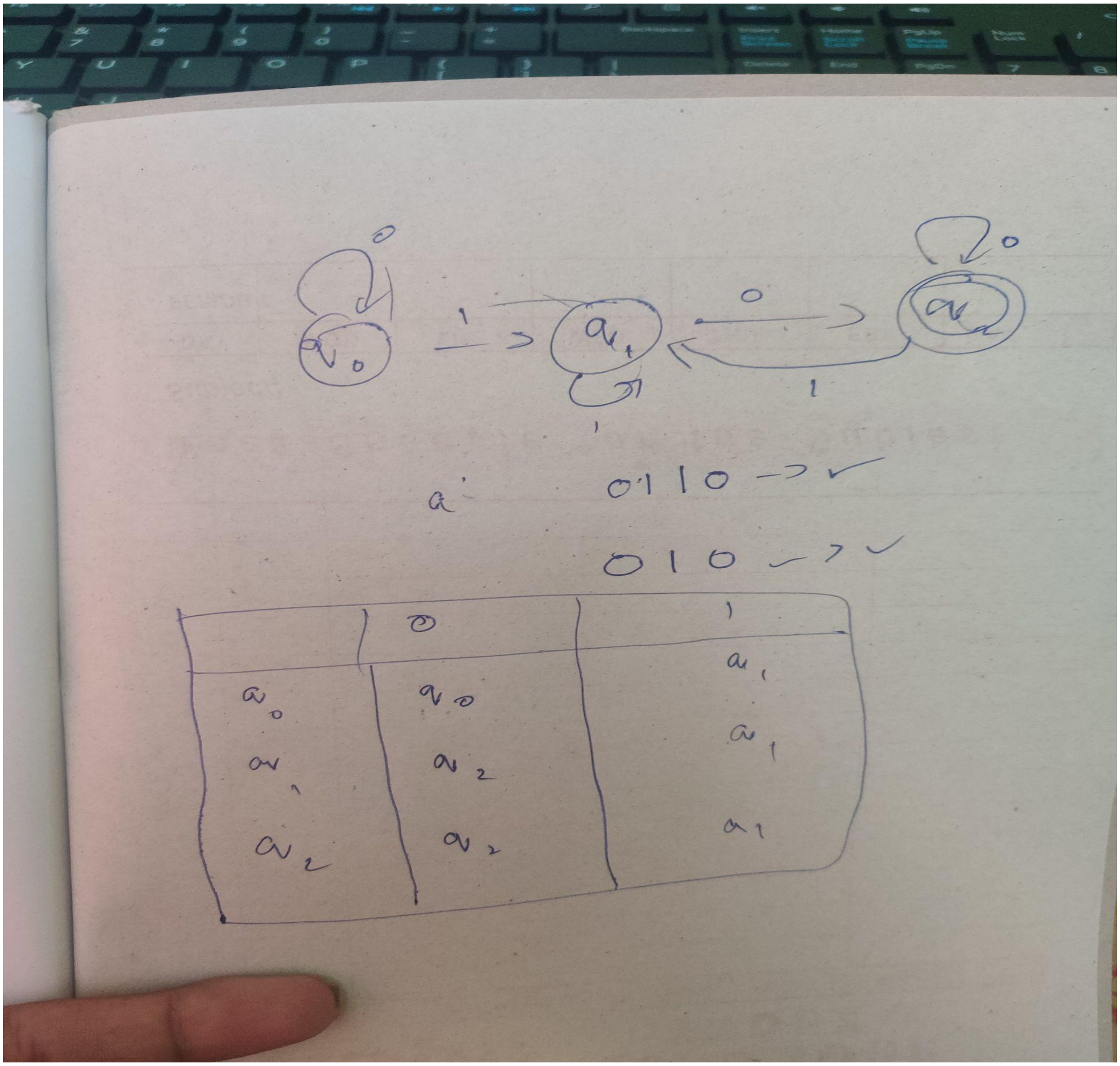
	0(a)	1(b)
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_2	q_1

DFA TO NFA
CODE:

```
from automata.fa.nfa import NFA
# NFA which matches strings beginning with 'a', ending with 'a', and containing
# no consecutive 'b's
nfa = NFA(
    states={'q0', 'q1'},
    input_symbols={'a', 'b'},
    transitions={
        'q0': {'a': {'q0'}, 'b': {'q1'}},

        'q1': {'a': {'q1'}, 'b': {'q0'}},
    },
    initial_state='q0',
    final_states={'q1'}
)
my_input_str = 'ab'
nfa.read_input_stepwise(my_input_str)
if nfa.accepts_input(my_input_str):
    print('accepted')
else:
    print('rejected')
```

ROUGH DIAGRAM



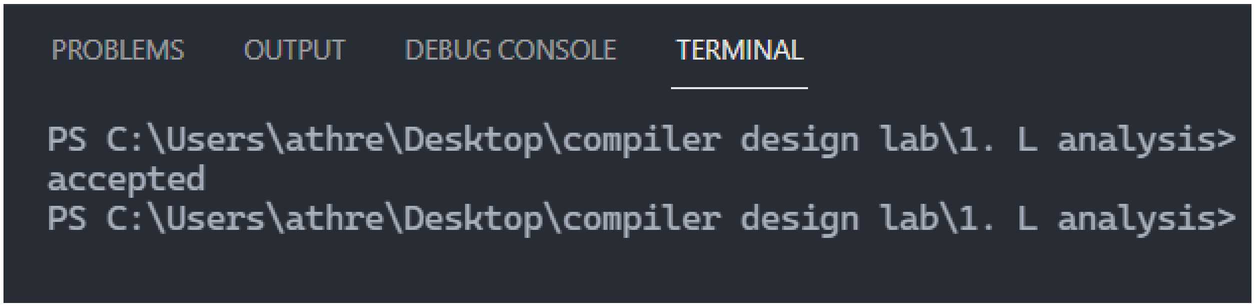
a'

0110 -> ✓

010 -> ✓

	0	1
q_0	q_0	q_1
q_1	q_2	q_1
q_2	q_2	q_1

INPUT - ‘ab’
OUTPUT - accepted



```
PS C:\Users\athre\Desktop\compiler design lab\1. L analysis>
accepted
PS C:\Users\athre\Desktop\compiler design lab\1. L analysis>
```