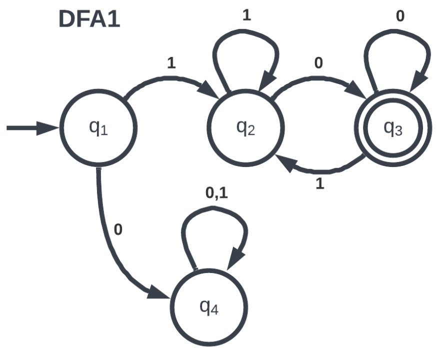
Homework2

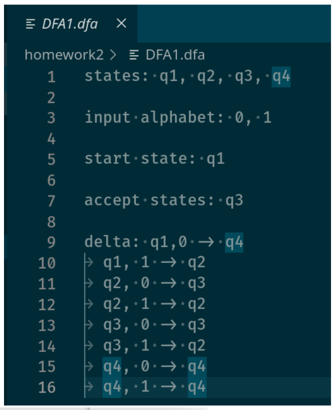
Antonio Zea Jr

September 28, 2022

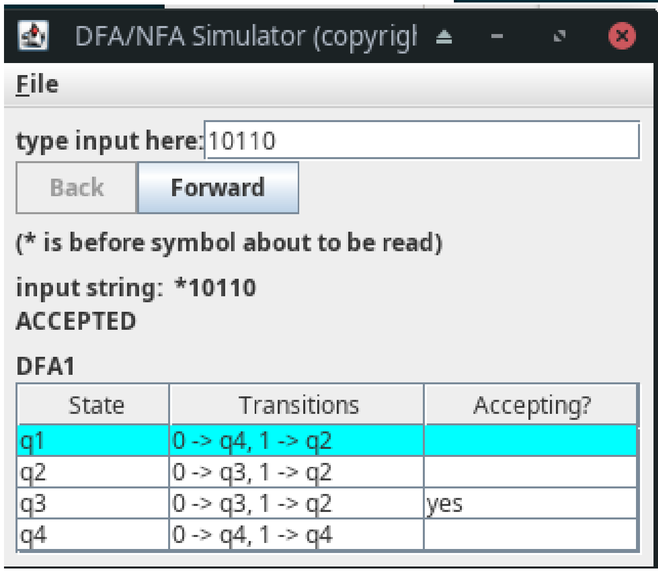
1 Design the following finite automata (DFA denotes deterministic and NFA - nondeterministic). For each one, draw the state diagram, implement it for the FASimulator and show its output (copy of the simulator window) for two inputs, one ACCEPTED and one REJECTED

1.1 DFA1 that recognizes the language $A_1 = \{w | w \in \{0,1\}^*, w$ begins with a 1 and end with a 0$\}$





```
states: q1, q2, q3, q4

input alphabet: 0, 1

start state: q1

accept states: q3

delta: q1,0 -> q4
  q1, 1 -> q2
  q2, 0 -> q3
  q2, 1 -> q2
  q3, 0 -> q3
  q3, 1 -> q2
  q4, 0 -> q4
  q4, 1 -> q4
```



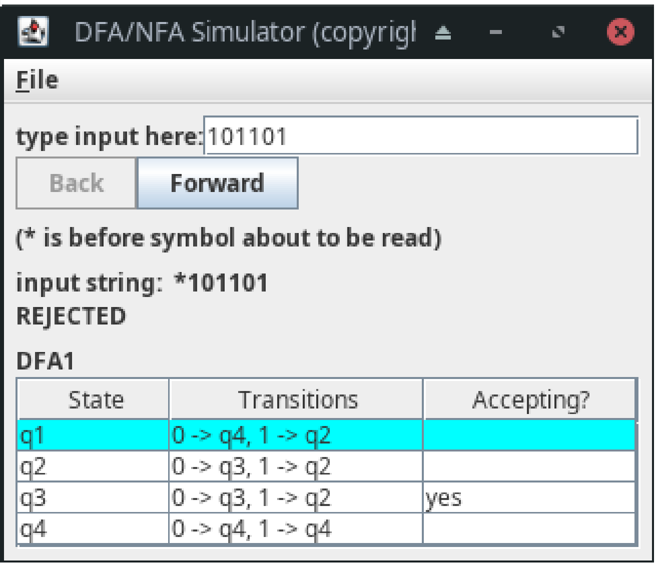
type input here: 10110

(* is before symbol about to be read)

input string: *10110
ACCEPTED

DFA1

State	Transitions	Accepting?
q1	0 -> q4, 1 -> q2	
q2	0 -> q3, 1 -> q2	
q3	0 -> q3, 1 -> q2	yes
q4	0 -> q4, 1 -> q4	



type input here: 101101

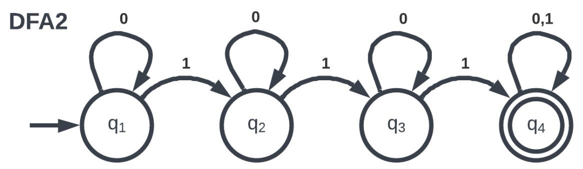
(* is before symbol about to be read)

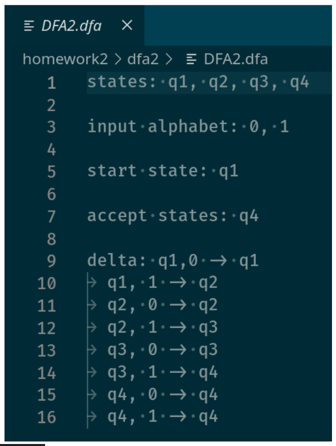
input string: *101101
REJECTED

DFA1

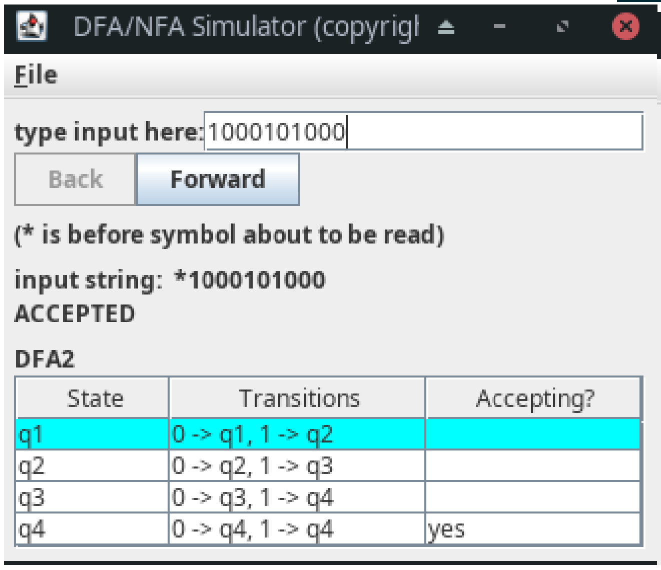
State	Transitions	Accepting?
q1	0 -> q4, 1 -> q2	
q2	0 -> q3, 1 -> q2	
q3	0 -> q3, 1 -> q2	yes
q4	0 -> q4, 1 -> q4	

1.2 DFA2 that recognizes the language $A_2 = \{w | w \in \{0,1\}^*, w$ contains at least three 1s$\}$





```
states: q1, q2, q3, q4

input alphabet: 0, 1

start state: q1

accept states: q4

delta: q1,0 → q1
  q1, 1 → q2
  q2, 0 → q2
  q2, 1 → q3
  q3, 0 → q3
  q3, 1 → q4
  q4, 0 → q4
  q4, 1 → q4
```



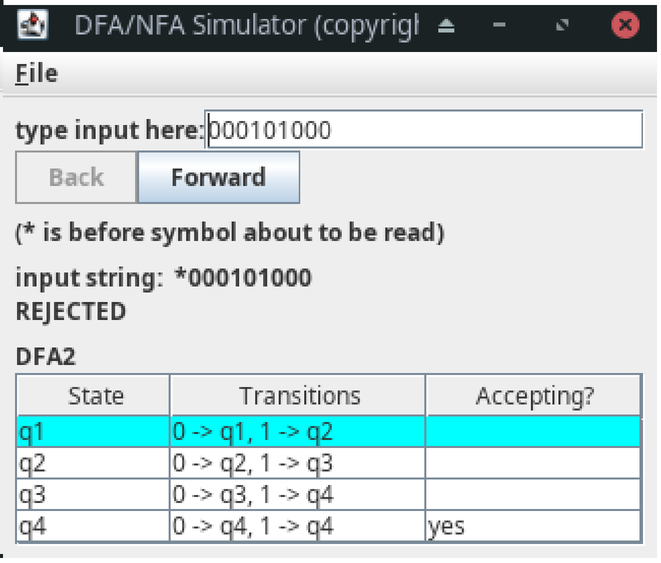
type input here: 1000101000

(* is before symbol about to be read)

input string: *1000101000
ACCEPTED

DFA2

State	Transitions	Accepting?
q1	0 -> q1, 1 -> q2	
q2	0 -> q2, 1 -> q3	
q3	0 -> q3, 1 -> q4	
q4	0 -> q4, 1 -> q4	yes



type input here: 000101000

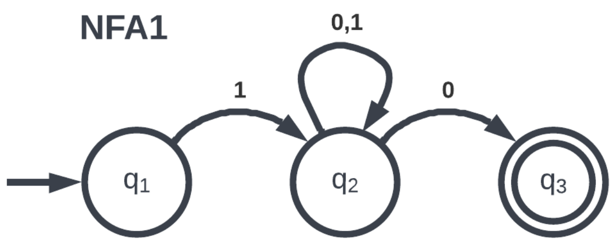
(* is before symbol about to be read)

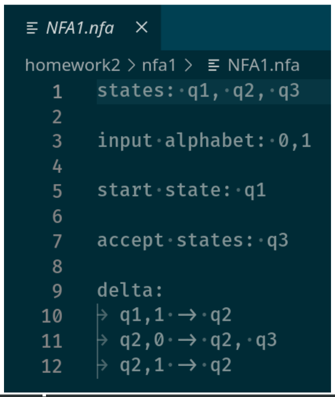
input string: *000101000
REJECTED

DFA2

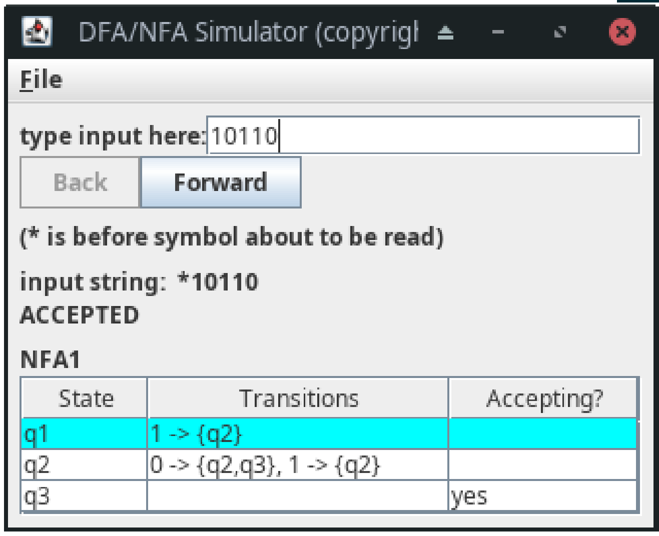
State	Transitions	Accepting?
q1	0 -> q1, 1 -> q2	
q2	0 -> q2, 1 -> q3	
q3	0 -> q3, 1 -> q4	
q4	0 -> q4, 1 -> q4	yes

1.3 NFA1 for language A_1 with three states





```
states: q1, q2, q3

input alphabet: 0,1

start state: q1

accept states: q3

delta:
    q1,1 → q2
    q2,0 → q2, q3
    q2,1 → q2
```



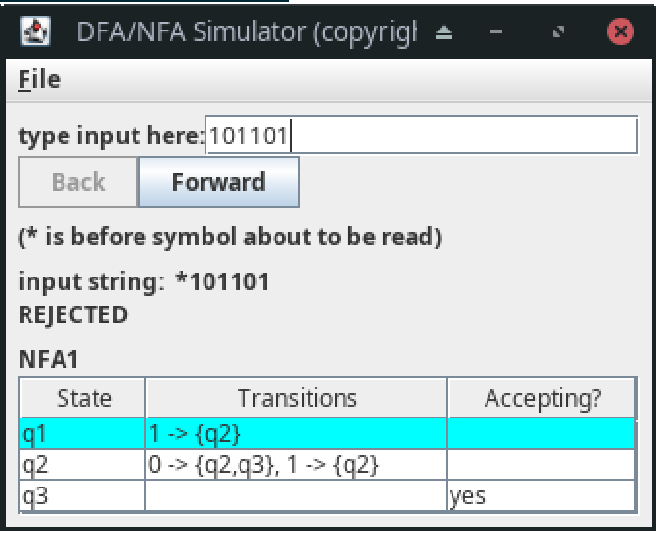
type input here: 10110

(* is before symbol about to be read)

input string: *10110
ACCEPTED

NFA1

State	Transitions	Accepting?
q1	1 -> {q2}	
q2	0 -> {q2,q3}, 1 -> {q2}	
q3		yes



type input here: 101101

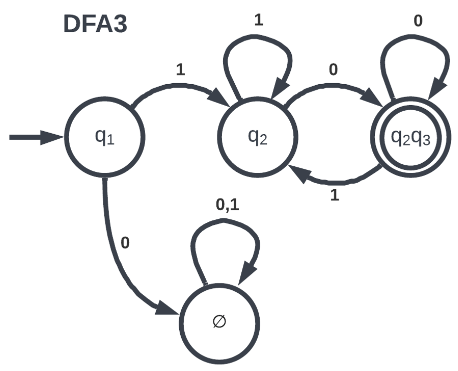
(* is before symbol about to be read)

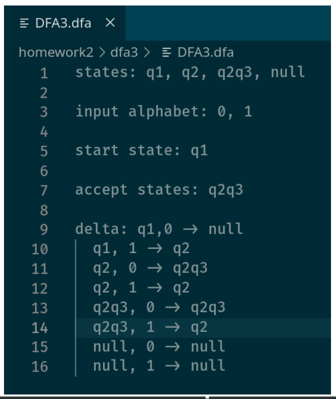
input string: *101101
REJECTED

NFA1

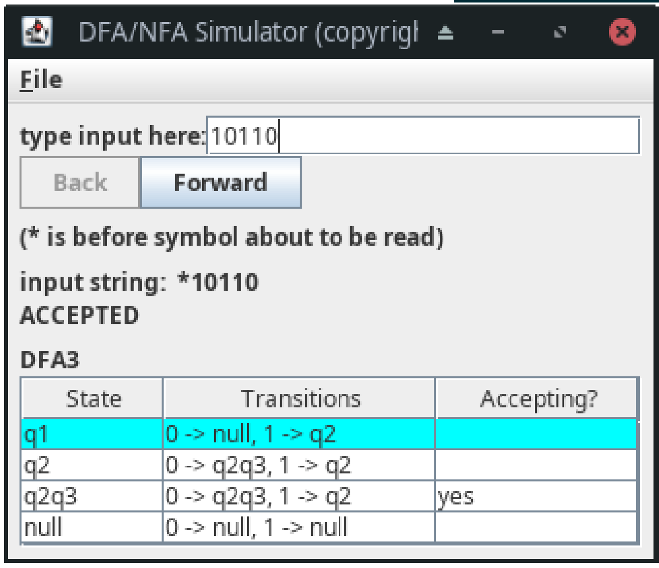
State	Transitions	Accepting?
q1	1 -> {q2}	
q2	0 -> {q2,q3}, 1 -> {q2}	
q3		yes

1.4 DFA3 converred from NFA1 usig the proof of Theorem 1.39





```
states: q1, q2, q2q3, null

input alphabet: 0, 1

start state: q1

accept states: q2q3

delta: q1,0 → null
  q1, 1 → q2
  q2, 0 → q2q3
  q2, 1 → q2
  q2q3, 0 → q2q3
  q2q3, 1 → q2
  null, 0 → null
  null, 1 → null
```



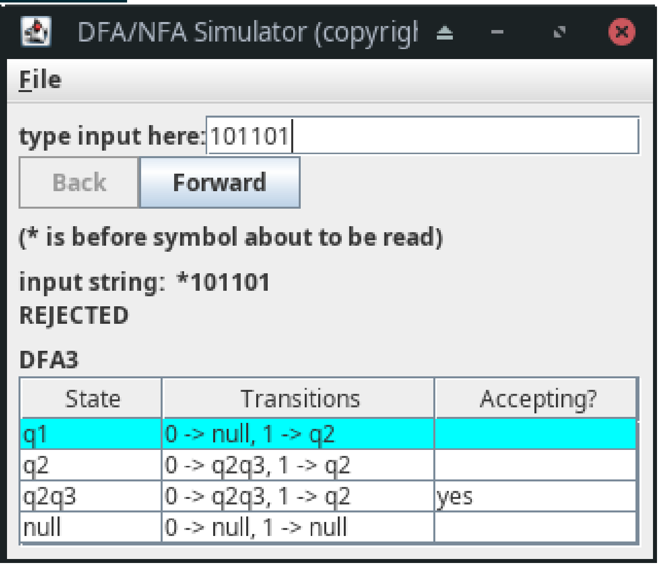
type input here: 10110

(* is before symbol about to be read)

input string: *10110
ACCEPTED

DFA3

State	Transitions	Accepting?
q1	0 -> null, 1 -> q2	
q2	0 -> q2q3, 1 -> q2	
q2q3	0 -> q2q3, 1 -> q2	yes
null	0 -> null, 1 -> null	



type input here: 101101

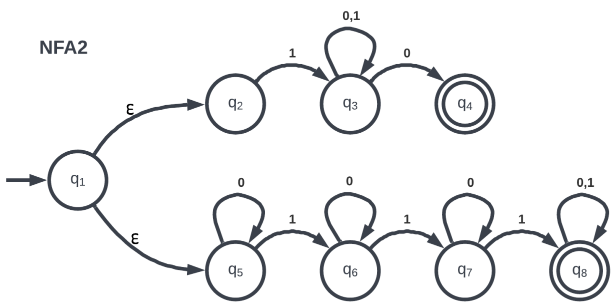
(* is before symbol about to be read)

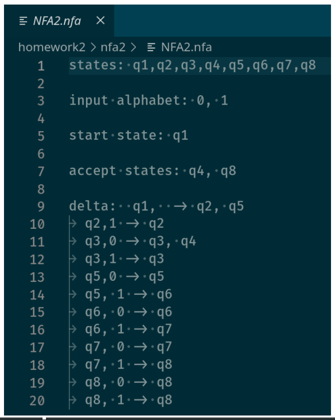
input string: *101101
REJECTED

DFA3

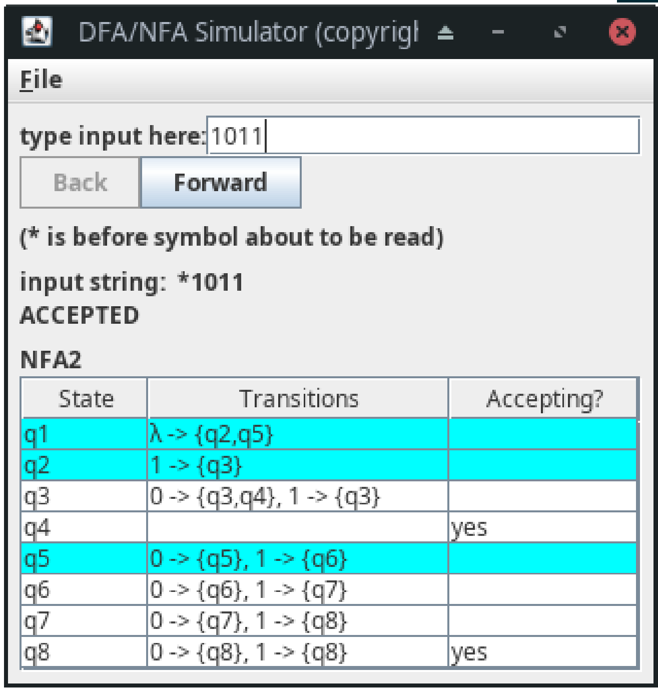
State	Transitions	Accepting?
q1	0 -> null, 1 -> q2	
q2	0 -> q2q3, 1 -> q2	
q2q3	0 -> q2q3, 1 -> q2	yes
null	0 -> null, 1 -> null	

1.5 NFA2 that recognizes the language $A_1 \cup A_2$. Use the construction in the proof of Theorem 1.45





```
states: q1,q2,q3,q4,q5,q6,q7,q8

input alphabet: 0, 1

start state: q1

accept states: q4, q8

delta:  q1,  → q2, q5
    q2,1 → q2
    q3,0 → q3, q4
    q3,1 → q3
    q5,0 → q5
    q5, 1 → q6
    q6, 0 → q6
    q6, 1 → q7
    q7, 0 → q7
    q7, 1 → q8
    q8, 0 → q8
    q8, 1 → q8
```

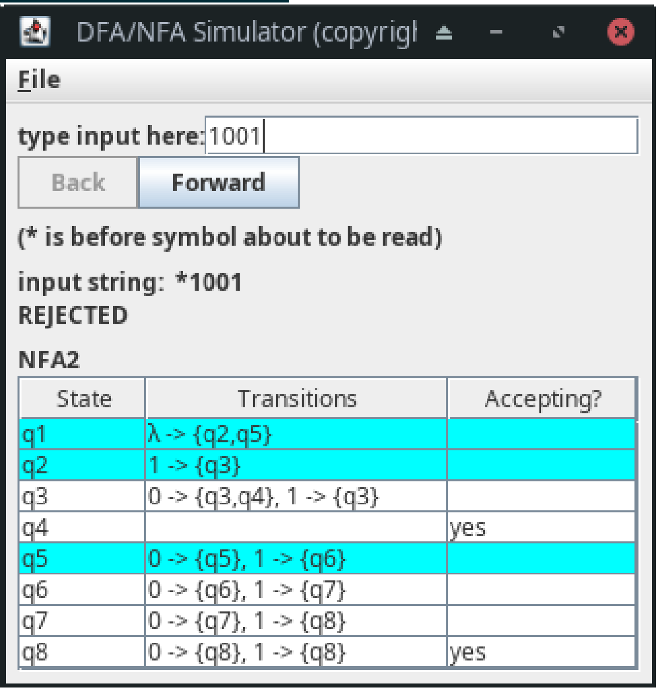


(* is before symbol about to be read)

input string: *1011
ACCEPTED

NFA2

State	Transitions	Accepting?
q1	λ -> {q2,q5}	
q2	1 -> {q3}	
q3	0 -> {q3,q4}, 1 -> {q3}	
q4		yes
q5	0 -> {q5}, 1 -> {q6}	
q6	0 -> {q6}, 1 -> {q7}	
q7	0 -> {q7}, 1 -> {q8}	
q8	0 -> {q8}, 1 -> {q8}	yes



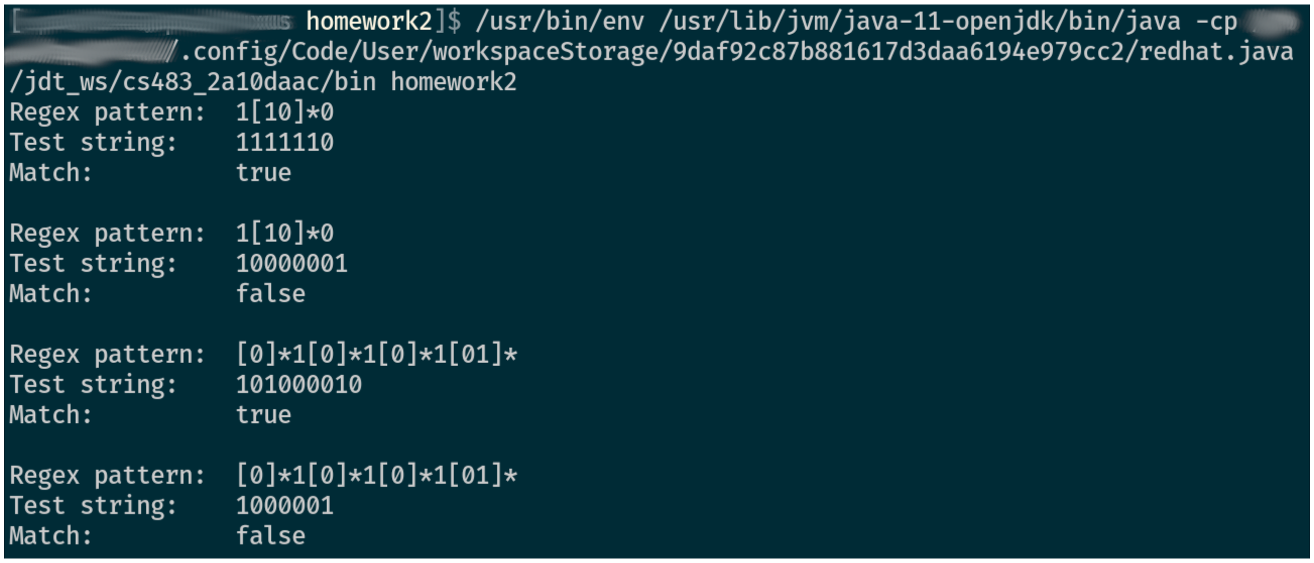
(* is before symbol about to be read)

input string: *1001
REJECTED

NFA2

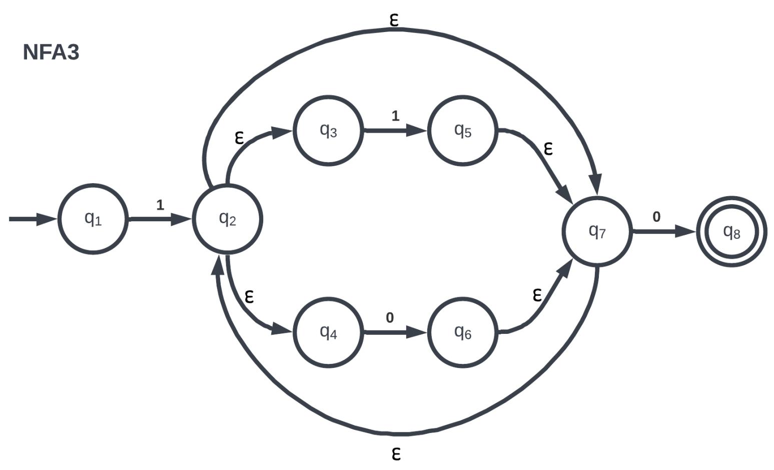
State	Transitions	Accepting?
q1	λ -> {q2,q5}	
q2	1 -> {q3}	
q3	0 -> {q3,q4}, 1 -> {q3}	
q4		yes
q5	0 -> {q5}, 1 -> {q6}	
q6	0 -> {q6}, 1 -> {q7}	
q7	0 -> {q7}, 1 -> {q8}	
q8	0 -> {q8}, 1 -> {q8}	yes

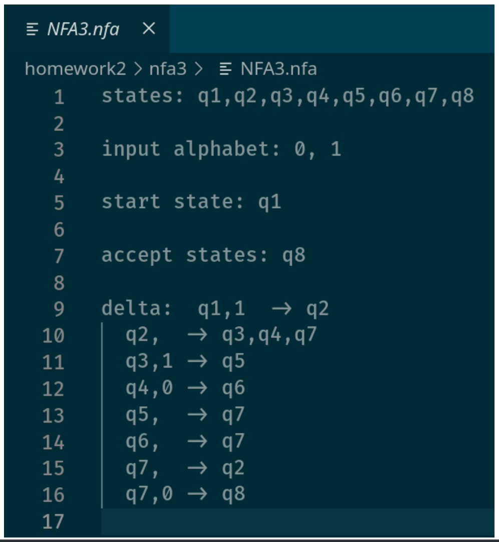
2 **Write a program in Java (using the Pattern class) for matching regular expressions and strings from the language they describe. The program should print the regex, the string and the result of matching them (true/false). Create the regular expressions describing the languages A_1 and A_2 (from Question 1) and for each one show one string that belongs to the language (matches the regex) and one that does not. Include the source code of the program in your report.**



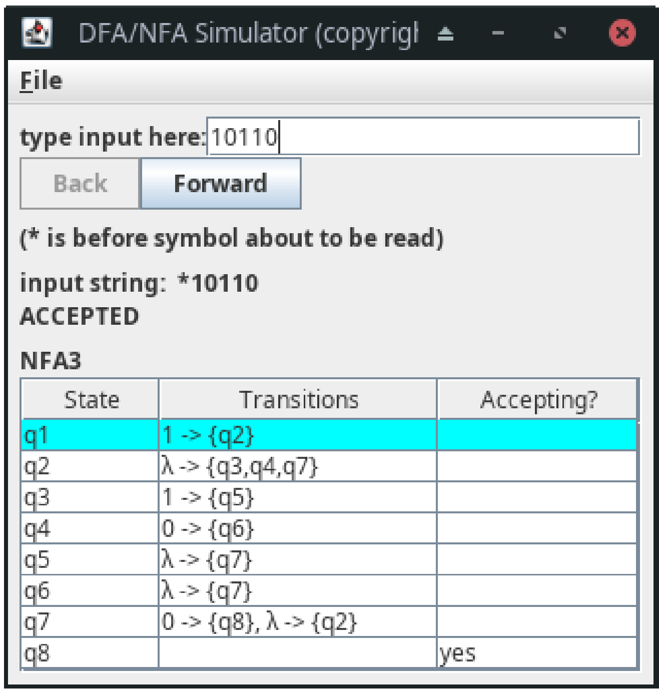
```
[[redacted] homework2]$ /usr/bin/env /usr/lib/jvm/java-11-openjdk/bin/java -cp [redacted]
[redacted]/.config/Code/User/workspaceStorage/9daf92c87b881617d3daa6194e979cc2/redhat.java
/jdt_ws/cs483_2a10daac/bin homework2
Regex pattern:  1[10]*0
Test string:    1111110
Match:          true

Regex pattern:  1[10]*0
Test string:    10000001
Match:          false

Regex pattern:  [0]*1[0]*1[0]*1[01]*
Test string:    101000010
Match:          true

Regex pattern:  [0]*1[0]*1[0]*1[01]*
Test string:    1000001
Match:          false
```

3 Convert the regular expression $1(1\cup 0)^*0$ to an NFA using the proof of Lemma 1.55. Show the state diagram, implement it for the FASimulator and show its output (copy of the simulator window) for two inputs - one ACCEPTED and one REJECTED. Note that this NFA recognizes the same language as DFA1, NFA1, and DFA3, but it is designed differently.





```
states: q1,q2,q3,q4,q5,q6,q7,q8

input alphabet: 0, 1

start state: q1

accept states: q8

delta:  q1,1  → q2
  q2,   → q3,q4,q7
  q3,1  → q5
  q4,0  → q6
  q5,   → q7
  q6,   → q7
  q7,   → q2
  q7,0  → q8
```



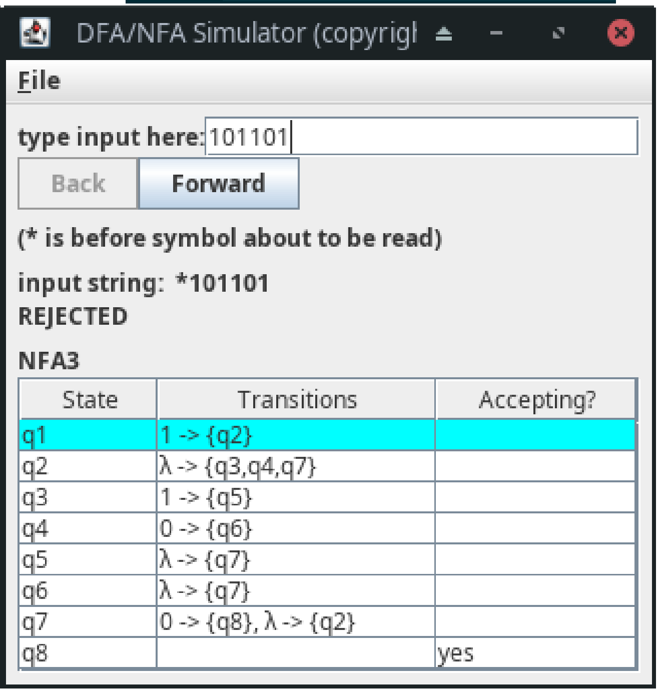
type input here: 10110

(* is before symbol about to be read)

input string: *10110
ACCEPTED

NFA3

State	Transitions	Accepting?
q1	1 -> {q2}	
q2	λ -> {q3,q4,q7}	
q3	1 -> {q5}	
q4	0 -> {q6}	
q5	λ -> {q7}	
q6	λ -> {q7}	
q7	0 -> {q8}, λ -> {q2}	
q8		yes



type input here: 101101

(* is before symbol about to be read)

input string: *101101
REJECTED

NFA3

State	Transitions	Accepting?
q1	1 -> {q2}	
q2	λ -> {q3,q4,q7}	
q3	1 -> {q5}	
q4	0 -> {q6}	
q5	λ -> {q7}	
q6	λ -> {q7}	
q7	0 -> {q8}, λ -> {q2}	
q8		yes