CSE322 Theory of Computation(L3,4)

Recap of last lecture

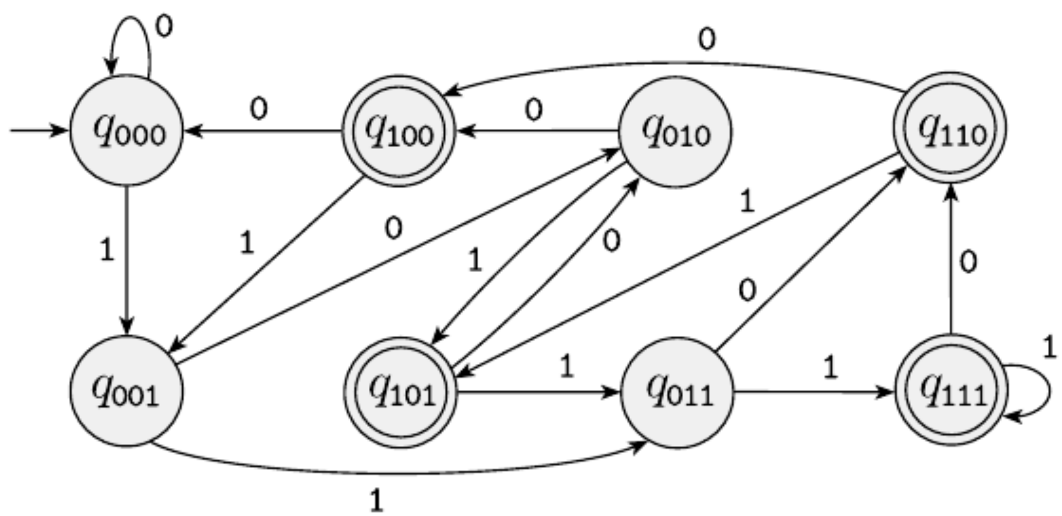
https://automatonsimulator.com/

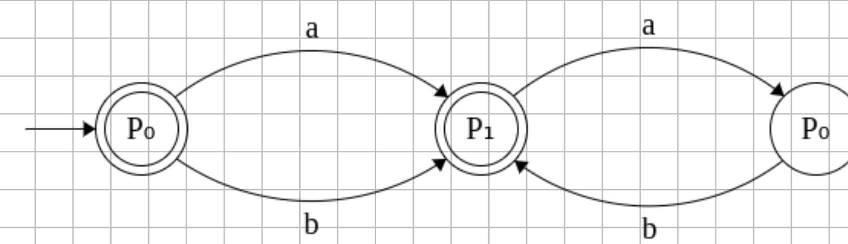
Today

~~Correctness of~~ DFA

NFA

Formalization of DFA





Does this DFA accept e?

Does this DFA accept e?

$DFA = \langle Q, \Sigma, \delta, q0, F \rangle$

DFA "accepts" $\overset{e}{s}$ if its ends up in an accept state after reading s. Formal definition

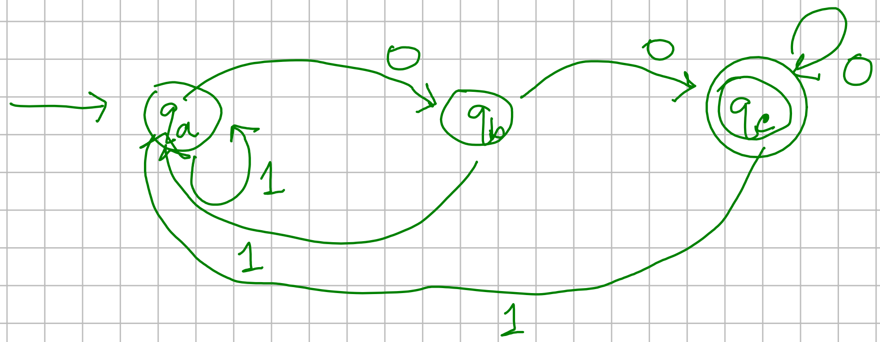
Suppose $\overset{\varepsilon}{s} = \overset{n=0}{w_1} \cdots w_n$ where n denotes $|s|$. Then there must exist states $\overset{r_0}{r_0} \cdots r_n$ st.:

1. $r_0 = q_0$
2. $\overset{r_0}{r_n} \in F$ $\Big\}$ $q_0 \in F$
3. $\forall i = 0$ to $n-1$, $\delta(r_i, w_{i+1}) = r_{i+1}$

DFA accepts the empty string e if ...

https://automatonsimulator.com/

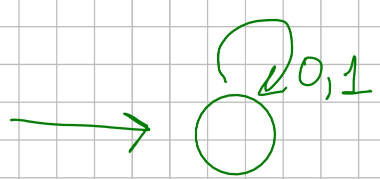
Q. Construct a DFA whose language is { binary string ending with 00 }



1001

1001000

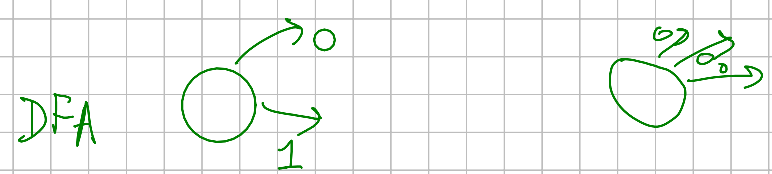
Q. Construct a DFA whose language is { }.



Non-deterministic FA

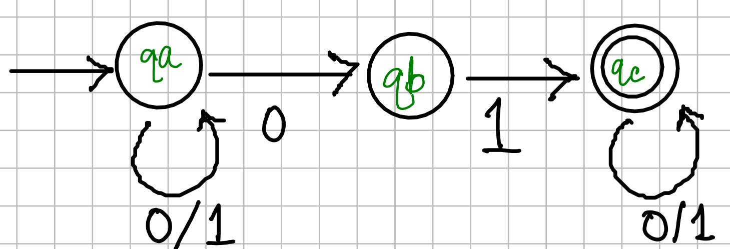
" If at each stage the motion of a machine ... is completely determined by the configuration, we shall call the machine an "automatic machine" (or a-machine). For some purposes we might use machines (choice machines or c-machines) whose motion is only partially determined by the configuration .... When such a machine reaches one of these ambiguous configurations, it cannot go on until some arbitrary choice has been made by an external operator." – Turing (On Computable Numbers)

Rabin, M. O.; Scott, D. (1959).
"Finite automata and their decision problems.".



NFA accepts s if there is "any" sequence of transitions for which NFA ends up in an "accept" state after reading s.

NFA



Trace the behaviour for 0010

Trace the behaviour for 1110

qa → (0) qa → (0) qb → (1) qc → (0) qc
accepted

qa → (0) qa → (0) qa → (1) qa → (0) qb
not a final state

qa → (0) qb → (0)
not a valid choice

0010 is accepted

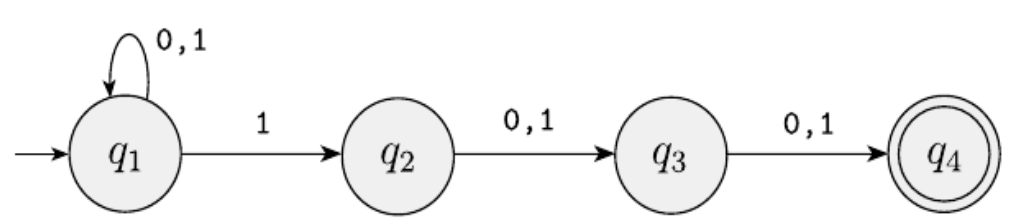
N accepts s if there is any path that ends in an accepting state.

N does not accept s otherwise. (no explicit notion of "reject")

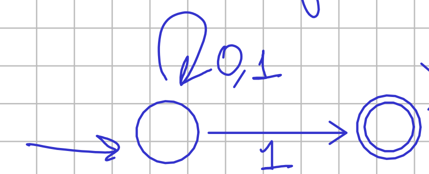
At state X, NFA "non-deterministically" decides on one of the allowed transitions. If there are no allowed transitions, then the relevant non-deterministic choices are thought to be invalid ones.

NFA accepts x if there are SOME VALID non-deterministic choices THAT lead the NFA to A FINAL STATE.

DFA is also an NFA

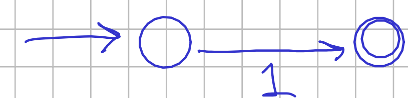


$L(N) = ?$ $\{w : |w| \geq 3$, 3rd last symbol of w is $1\}$



× also accepts 01

01 ×



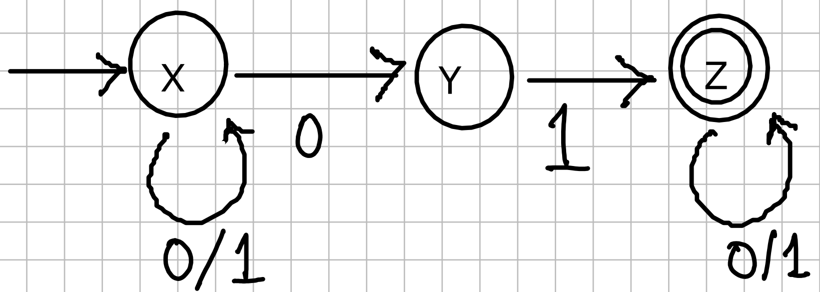
✓

Q: Construct an NFA recognizing $\{1\}$

alphabet $\{0,1\}$

Formalization of NFA

δ	0	1
$\rightarrow X$	$\{X, Y\}$	$\{X\}$
Y	$\emptyset$	$\{Z\}$
$* Z$	$\{Z\}$	$\{Z\}$



NFA does not care about invalid choices AND valid choices that do not lead to a final state.

$\delta(\text{state}, \text{symbol})$ = single state DFA
= set of states NFA

$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$

NFA = $\langle Q, \Sigma, \delta, q0, F \rangle$

NFA accepts x if ... x can be written as $x = w_1 \cdots w_n$ where $w_i \in \Sigma$, and

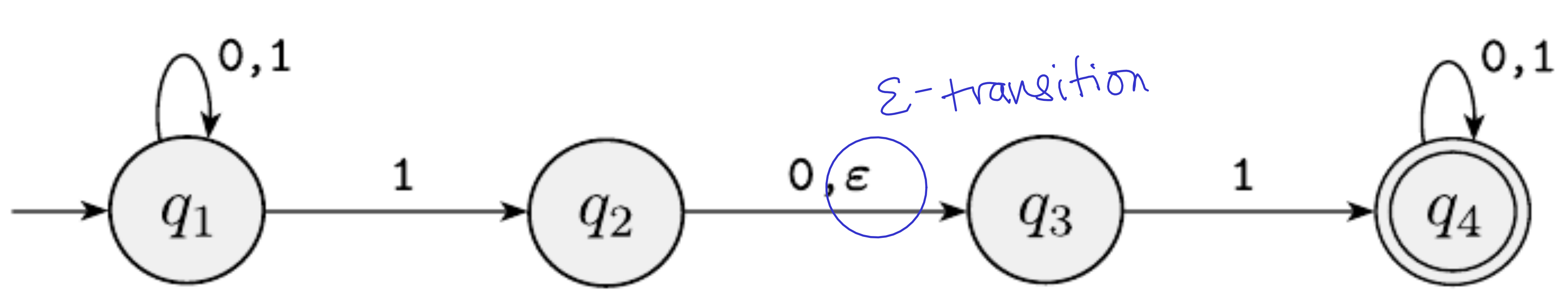
$\exists$ states $r_0 \cdots r_n$ s.t. → sequence of states on a valid sequence of transitions

(1) $r_0 = q0$ (2) $r_n \in F$ (3) $\forall i = 1 \cdots n$, $\cancel{\delta(r_{i-1}, w_i) = r_i}$ X (DFA)

$r_i \in \delta(r_{i-1}, w_i)$
(r_i is a valid choice)

ε : empty string , $|\varepsilon| = 0$, $\varepsilon \cdot a = a = a \cdot \varepsilon$

ε-NFA



Is 0 accepted? ✗

Is 11 accepted? $q_1 \xrightarrow{1} q_2 \xrightarrow{\varepsilon} q_3 \xrightarrow{1} q_4$

Is 00 accepted? ✗

Is 0101 accepted? $q_1 \xrightarrow{0} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{1} q_4$

Is 010101 accepted? ✓

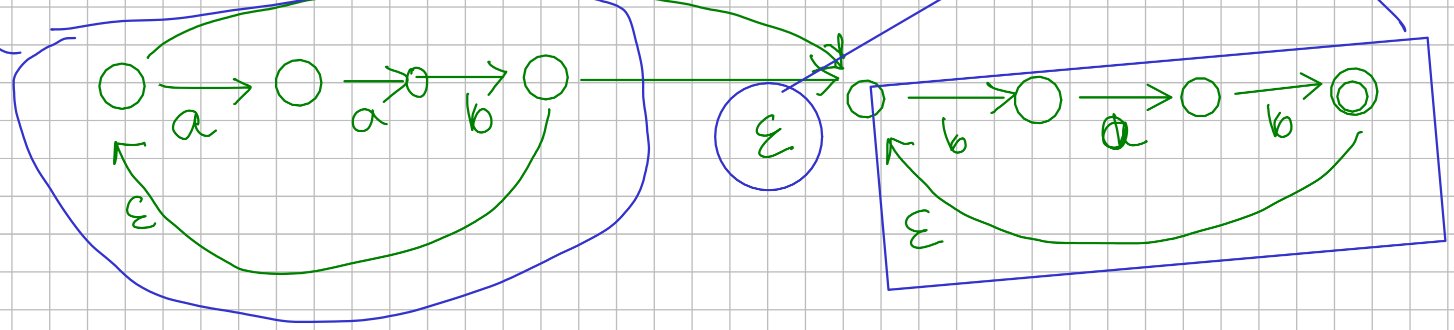
Is 01010101010101 accepted? ✓

ε-NFA for strings of the form ...

ε, aab, bab, aab bab, aabaab bab, babbab, ...

zero or more occurrence of (aab) followed by

zero of more occurrence of (bab)



Can you construct a DFA for the above language?

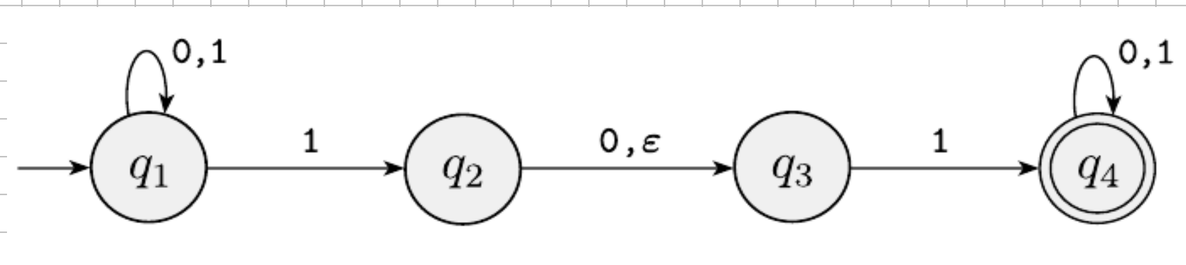
Can you prove that your NFA is correct?

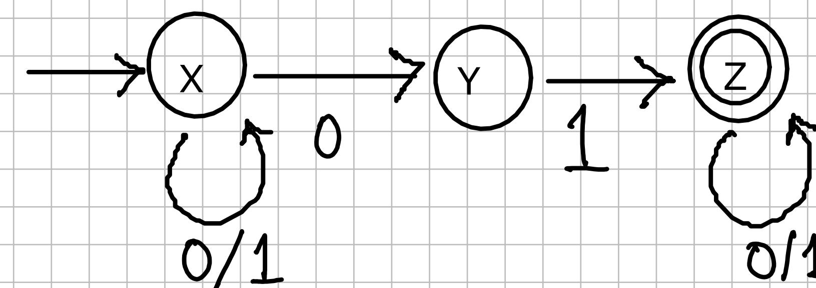
1 1 is accepted: $q_1 \xrightarrow{1} q_2 \xrightarrow{\varepsilon} q_3 \rightarrow q_4$

Formalization of e-NFA

NFA $\delta: Q \times \Sigma \rightarrow P(Q)$

$\delta(q, a) \subseteq Q$





ε-NFA $= \langle Q, \Sigma, \delta, q_0, F \rangle$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

$\delta : Q \times \Sigma_\varepsilon \rightarrow P(Q)$

ε-NFA accepts x if ...

$\exists m \geq n$, X can be written as $x = w_1 \cdots w_m$ where $w_i \in \Sigma_\varepsilon$,

$\exists r_0 \cdots r_m$, $r_i \in Q$, $\forall i = 0 \cdots m$, st.

① $r_0 = q_0$ ② $r_m \in F$ ③ $\forall i = 1 \cdots m$,

$r_i \in \delta(r_{i-1}, w_i)$

NFA does not care about invalid choices AND valid choices that do not lead to a final state.

δ	0	1	ε
q_1	$\{q_1\}$		$\emptyset$
q_2			
q_3		Ex: fill ...	
q_4			