LECTURE 4

WEDNESDAY JANUARY 15

Quiz 1: Wednesday
Jan. 21

Office Hours: Friday
2:30pm ~ 3:30pm

DFA: Formulation (1)

A *deterministic finite automata (DFA)* is a 5-tuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

Language of a DFA

$$L(M) = \left\{ a_1 a_2 \ldots a_n \;\middle|\; 1 \le i \le n \wedge a_i \in \Sigma \wedge \delta(q_{i-1}, a_i) = q_i \wedge q_n \in F \right\}$$

s_1

length

DFA languages

q_i the resulting state after reading the i^{th} char

$a_1 \; a_2 \; a_3 \; a_4$

$0 \; 1 \; 0 \; 1$

s_3

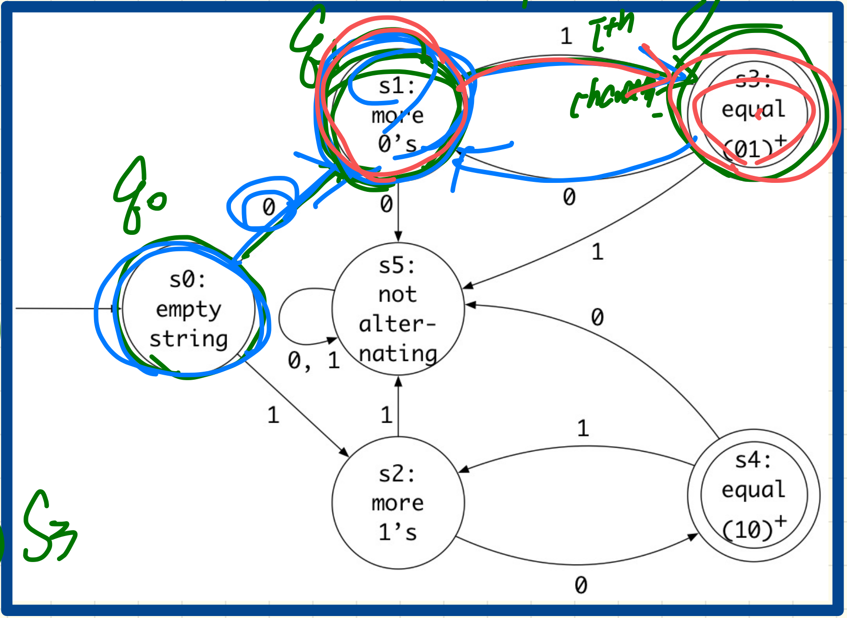
$i=1 \quad \delta(q_0, a_1) = q_1$

s_1

s_1

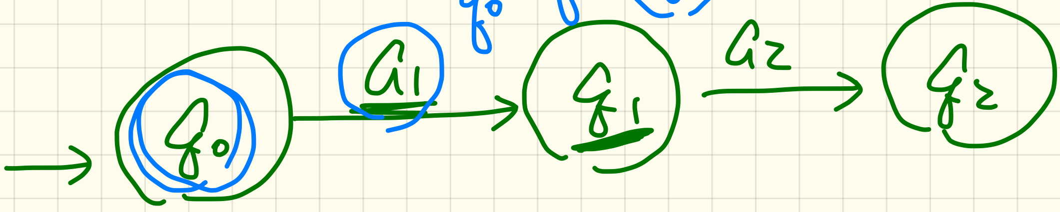
$i=2 \quad \delta(\boxed{q_1}, a_2) = q_2$

s_3



Input string: | a_1 | a_2 | . -

q_0 q_1 q_2



$$\delta(q_0, a_1) = q_1$$

DFA: Formulation (2)

A *deterministic finite automata (DFA)* is a 5-tuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

Language of a DFA

$$\hat{\delta} : (Q \times \Sigma^*) \to Q$$

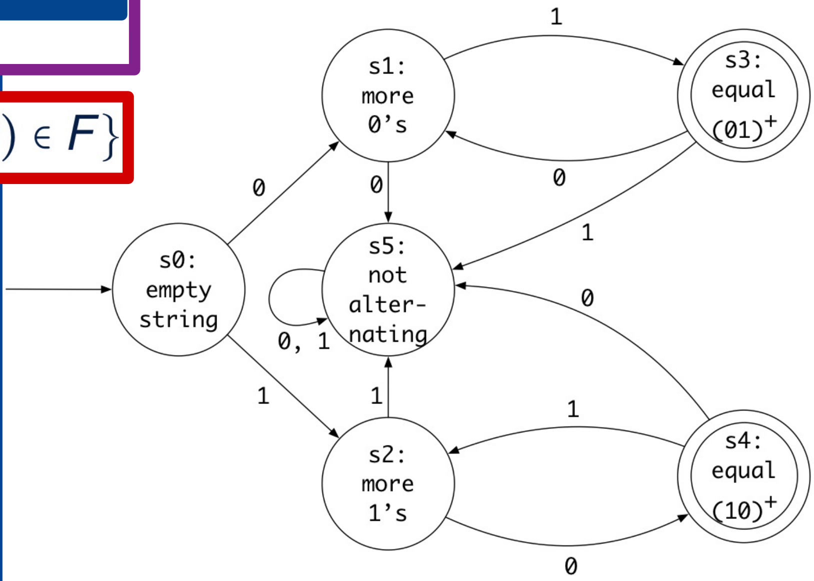
We may define $\hat{\delta}$ recursively, using δ!

$\hat{\delta}(q, \epsilon) =$ [redacted]

$\hat{\delta}(q, xa) =$ [redacted]

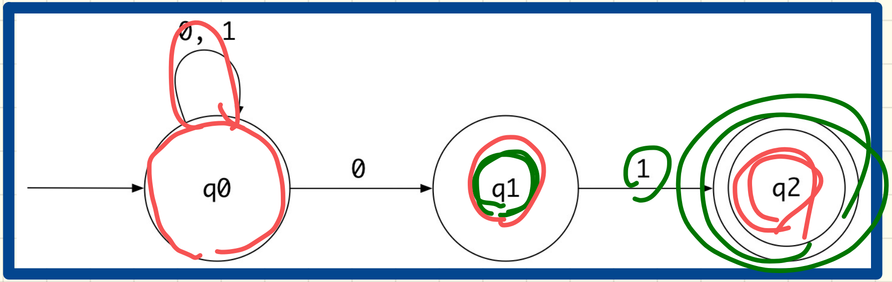
where $q \in Q$, $x \in \Sigma^*$, and $a \in \Sigma$

$$L(M) = \{w \mid w \in \Sigma^* \wedge \hat{\delta}(q_0, w) \in F\}$$



NFA: Processing Strings

How an NFA determines if an input 00101 should be processed:



- **Read 0:** $\delta(q_0, 0) = \{q_0, q_1\}$
- **Read 0:** $\overset{q_0, q_1}{\delta(q_0, 0)} \cup \overset{\emptyset}{\delta(q_1, 0)} = \{q_0, q_1\}$
- **Read 1:** $\overset{q_0}{\delta(q_0, 1)} \cup \overset{q_2}{\delta(q_1, 1)} = \{q_0, q_2\}$
- **Read 0:** $\overset{q_0, q_1}{\delta(q_0, 0)} \cup \overset{\emptyset}{\delta(q_2, 0)} = \{q_0, q_1\}$
- **Read 1:** $\overset{q_0}{\delta(q_0, 1)} \cup \overset{q_2}{\delta(q_1, 1)} = \{q_0, q_2\}$ ✓

NFA: Formulation

A *nondeterministic finite automata (NFA)* is a 5-tuple

$$M = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

Language of a NFA

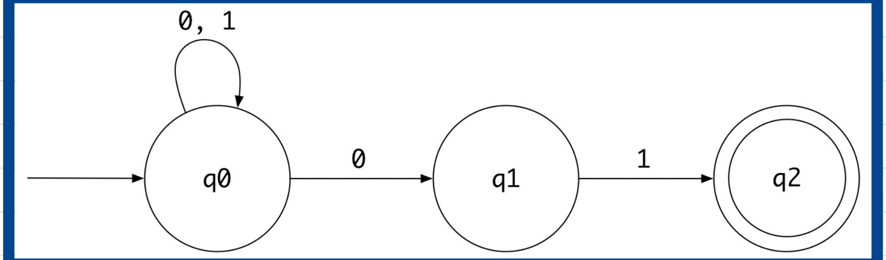
$$\hat{\delta} : (Q \times \Sigma^*) \to \mathbb{P}(Q)$$

We may define $\hat{\delta}$ recursively, using δ!

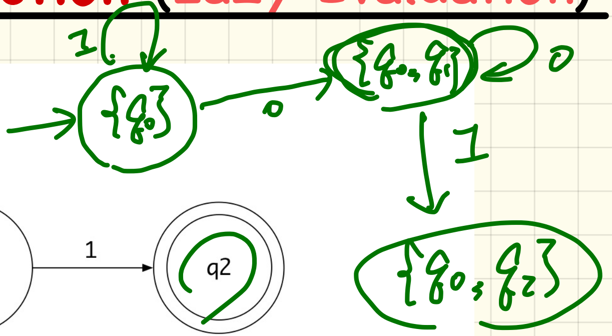
$$\begin{array}{rcl} \hat{\delta}(q, \epsilon) & = & \{q\} \\ \hat{\delta}(q, xa) & = & \bigcup\{\delta(q', a) \mid q' \in \hat{\delta}(q, x)\} \end{array}$$

where $q \in Q$, $x \in \Sigma^*$, and $a \in \Sigma$

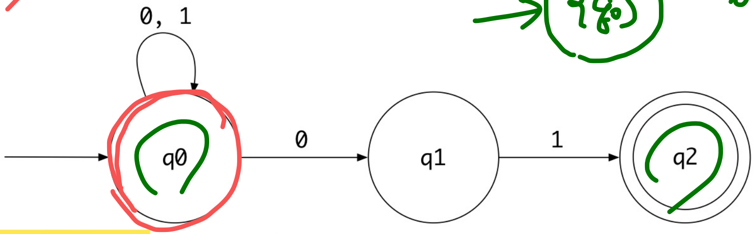
$$L(M) = \{w \mid w \in \Sigma^* \wedge \hat{\delta}(q_0, w) \cap F \neq \varnothing\}$$



NFA to DFA: Subset Construction (Lazy Evaluation)

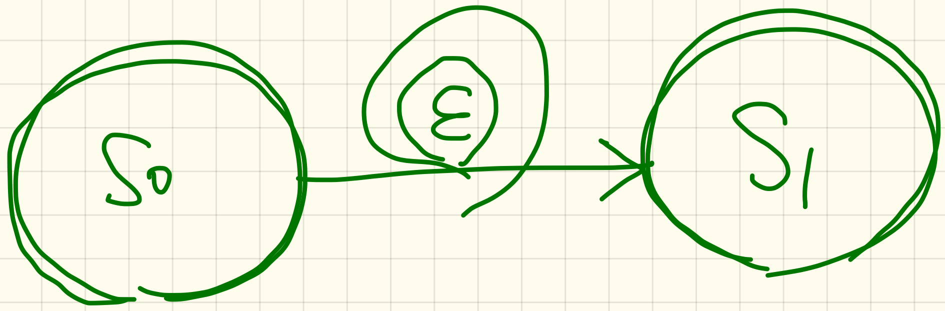


Given an *NFA*:



Subset construction (with *lazy evaluation*) produces a *DFA* transition table:

state \ input	0	1
$\{q_0\}$	$\delta(q_0, 0) = \boxed{\{q_0, q_1\}}$	$\delta(q_0, 1) = \boxed{\{q_0\}}$
$\{q_0, q_1\}$	$\delta(q_0, 0) \cup \underbrace{\delta(q_1, 0)}_{\emptyset} = \{q_0, q_1\}$	$\delta(q_0, 1) \cup \delta(q_1, 1) = \boxed{\{q_0, q_2\}}$



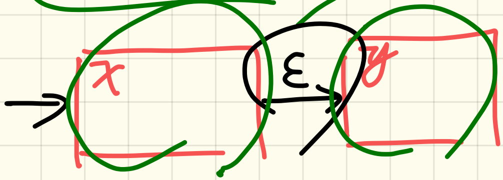
$\{S_0, S_1\}$.

epsilon-NFA: Motivation

Draw NFA

$$\left\{ xy \;\middle|\; \begin{array}{l} x \in \{0,1\}^* \\ \wedge\; y \in \{0,1\}^* \\ \wedge\; x \text{ has alternating 0's and 1's} \\ \wedge\; y \text{ has an odd \# 0's and an odd \# 1's} \end{array} \right\}$$

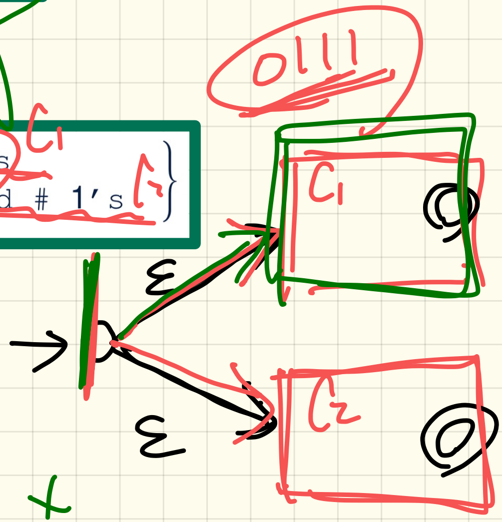
xy



$$\left\{ w : \{0,1\}^* \;\middle|\; \begin{array}{l} w \text{ has alternating 0's and 1's} \\ \vee\; w \text{ has an odd \# 0's and an odd \# 1's} \end{array} \right\}$$

C_1

C_2



$\wedge$ ("+" + "-" + ϵ)

+.23

-2.

3.24

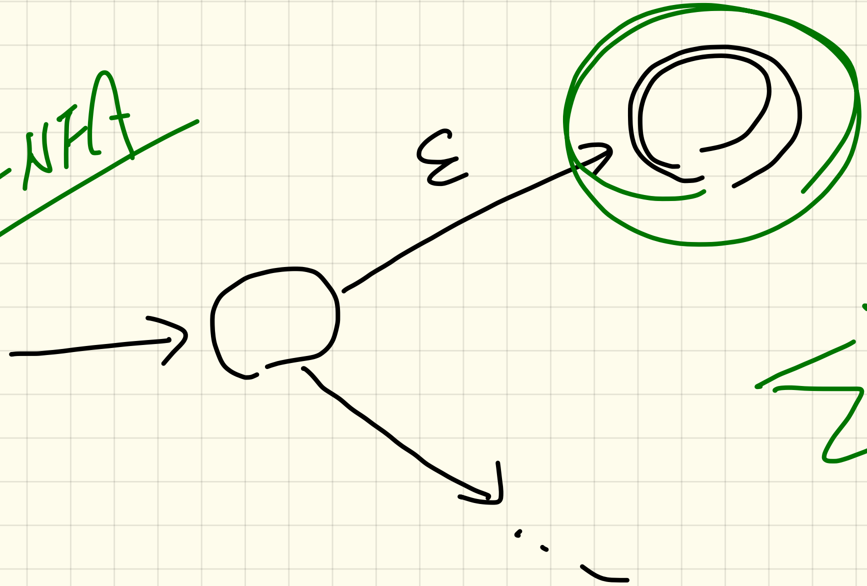
+

$$\left\{ sx.y \;\middle|\; \begin{array}{l} s \in \{+,-,\epsilon\} \\ \wedge\; x \in \Sigma_{dec}^* \\ \wedge\; y \in \Sigma_{dec}^* \\ \wedge\; \neg(x = \epsilon \wedge y = \epsilon) \end{array} \right\}$$

exercise

+ .

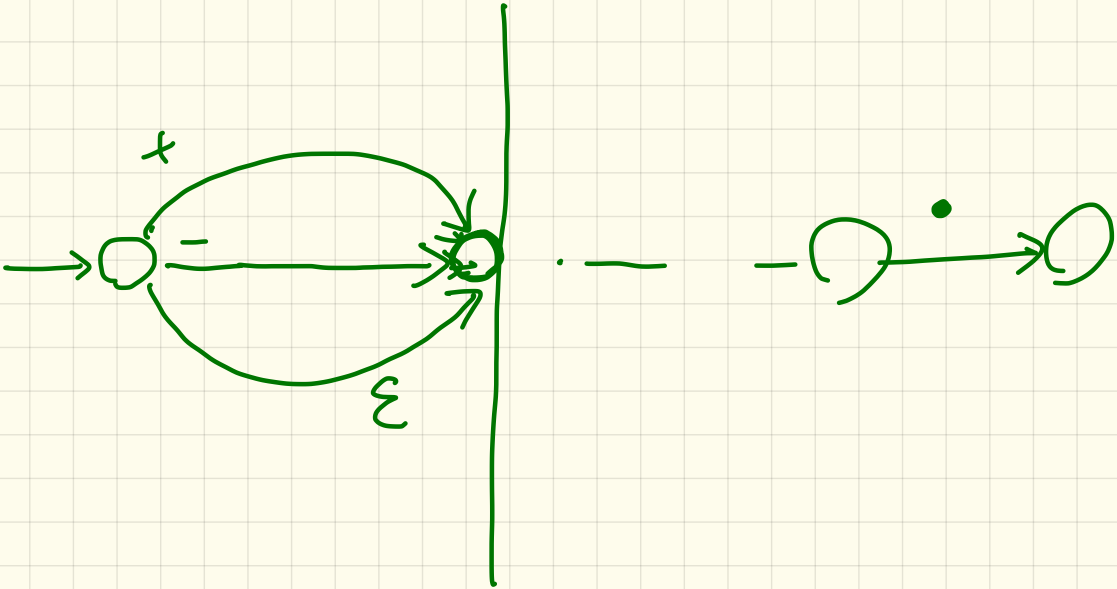
ε-NFA



Σ^*

epsilon-NFA: Example

$$\left\{ s x.y \;\middle|\; \begin{array}{ll} & s \in \{+, -, \epsilon\} \\ \wedge & x \in \Sigma_{dec}^{*} \\ \wedge & y \in \Sigma_{dec}^{*} \\ \wedge & \neg(x = \epsilon \wedge y = \epsilon) \end{array} \right\}$$

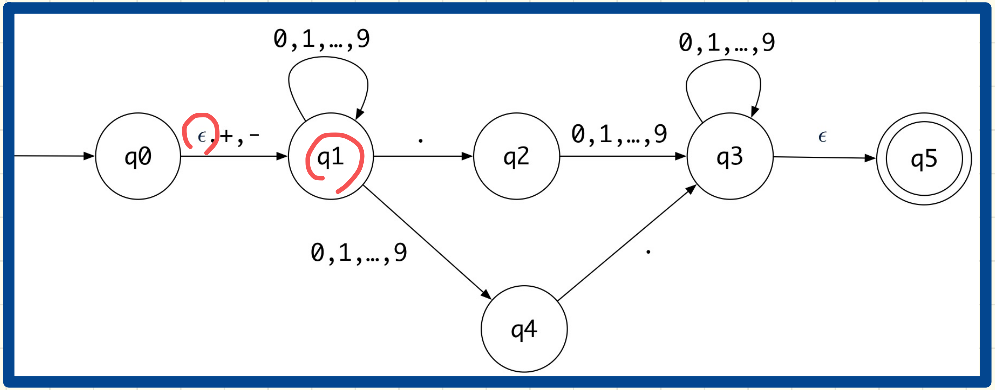


epsilon-NFA: Formulation (1)

An ϵ-NFA is a 5-tuple

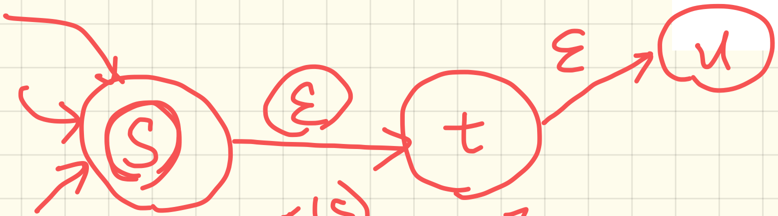
$$M = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

Example epsilon-NFA



Draw a transition table for the above NFA's δ function:

	ϵ	+, -	.	0 .. 9
q_0	$\{q_1\}$			
q_1	$\emptyset$			
q_2	$\emptyset$			
⋮				
q_5				



ECLOSE(s)

$= \{s, t, u\}$

epsilon-NFA: Formulation (2)

An *ϵ-NFA* is a 5-tuple

$$M = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

we define the *epsilon closure* (or *ϵ-closure*) as a function

$$\mathrm{ECLOSE} : Q \to \underbrace{\mathbb{P}(Q)}_{\text{set of states}}$$

For any state $q \in Q$

$$\mathrm{ECLOSE}(q) = \{q\} \cup \bigcup_{p \in \delta(q, \epsilon)} \mathrm{ECLOSE}(p)$$

union the ECLOSE of all states reachable using ϵ from q

ECLOSE(q0) ?

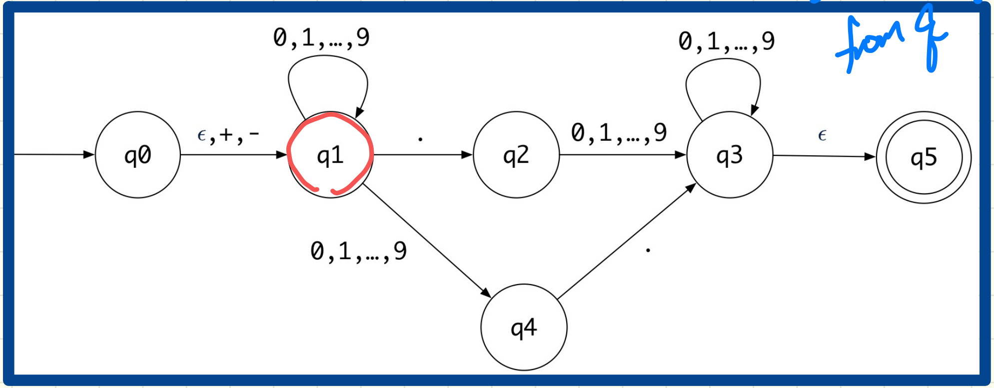
$\mathrm{ECLOSE}(q_0) =$

$\{q_0\} \cup \mathrm{ECLOSE}(q_1)$

$\{q_1\} \cup \emptyset$

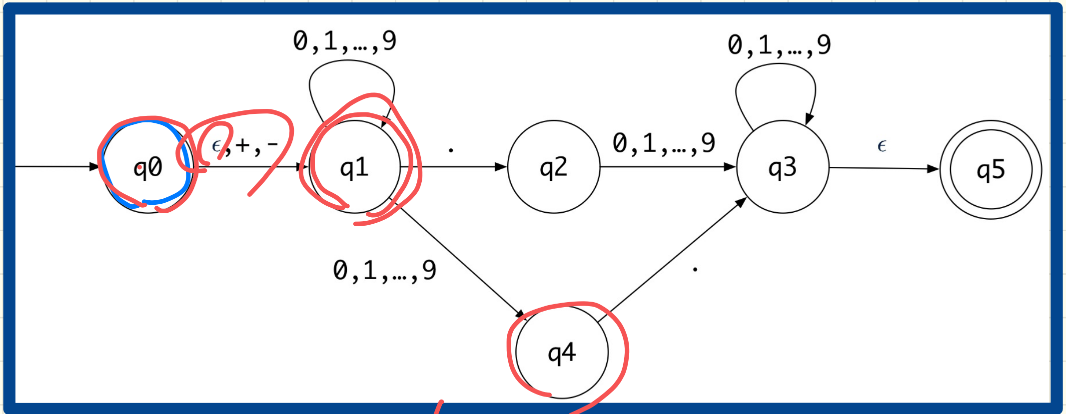
$= \{q_0, q_1\}$

Example epsilon-NFA



epsilon-NFA: Processing Strings

How an epsilon-NFA determines if input 5.6 should be processed



5.6

Starting state:

$ECLOSE(q_0) = \{q_0, q_1\}$

Read 5	$\underbrace{\delta(q_0, 5)}_{\emptyset} \cup \underbrace{\delta(q_1, 5)}_{q_1, q_4} = \{q_1, q_4\}$ $\underbrace{ECLOSE(q_1)}_{q_1} \cup \underbrace{ECLOSE(q_4)}_{q_4}$
Read .	
Read 6	

epsilon-NFA: Formulation (3)

An ϵ-*NFA* is a 5-tuple

$$M = (Q,\ \Sigma,\ \delta,\ q_0,\ F)$$

Language of a epsilon-NFA

$$\hat{\delta} : (Q \times \Sigma^*) \to \mathbb{P}(Q)$$

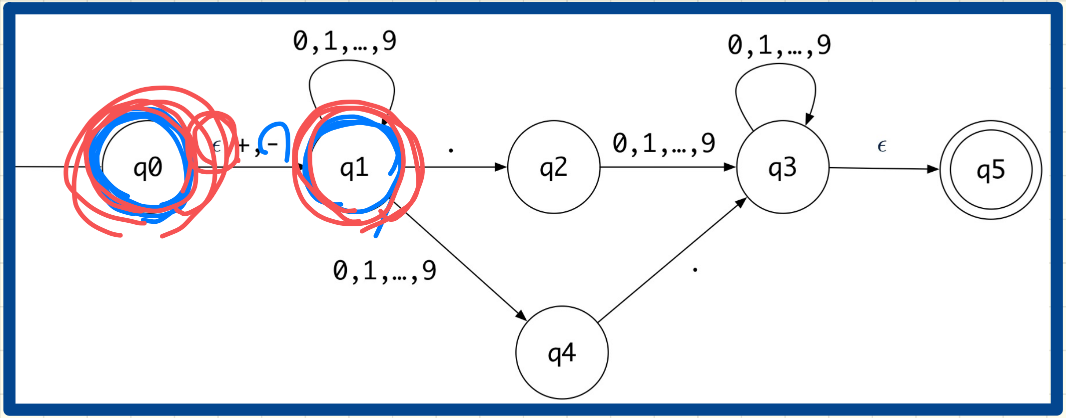
We may define $\hat{\delta}$ recursively, using δ!

$\hat{\delta}(q, \epsilon) = \text{ECLOSE}(q)$

$\hat{\delta}(q, xa) = \bigcup\{$ [redacted] $\mid q'' \in \delta(q', a) \wedge q' \in \hat{\delta}(q, x) \}$

$$L(M) = \{w \mid w \in \Sigma^* \wedge \hat{\delta}(q_0, w) \cap F \neq \varnothing\}$$

epsilon-NFA to DFA: Subset Construction



ECLOSE(q0) ?

0|

Subset construction (with *lazy evaluation* and *epsilon closures*) produces a *DFA* transition table.

	$d \in 0..9$	$s \in \{+,-\}$	.
$\{q_0, q_1\}$	$\boxed{}$		

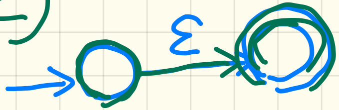
$\delta(q_0, d) \cup \delta(q_1, d)$

$= \{ \_ \_ \_ \_ \}$

Regular Expression to epsilon-NFA

Base Cases

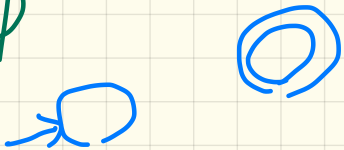
ε



↗ a

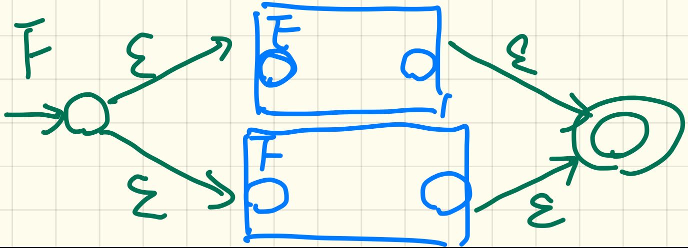
$a \in \Sigma$ a

ϕ

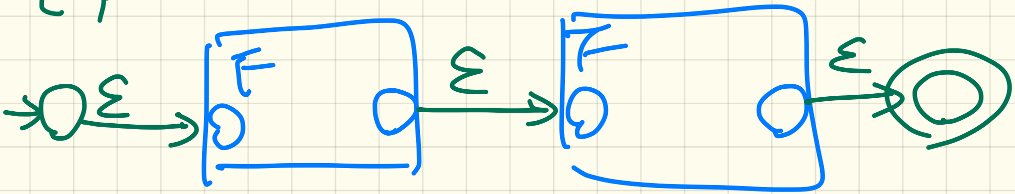


Recursive Cases (given REs E and F)

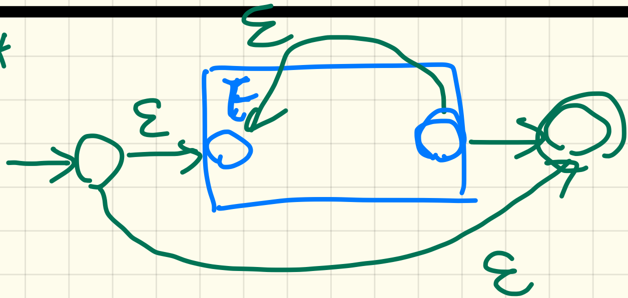
$E + F$

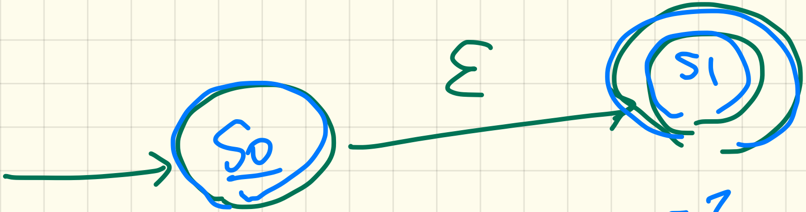


EF



E^*





a

$$ECLOSE(s_0) = \{s_0, s_1\}$$

$$\underbrace{\delta(s_0, a)}_{\phi} \cup \underbrace{\delta(s_1, a)}_{\phi} = \phi$$

Regular Expression to epsilon-NFA: Example

$(0+1)^*1(0+1)$