Lecture 8 - Oct. 4

Lexical Analysis

ε-NFA: ε-Closure & Conversion to DFA
From Regular Expressions to ε-NFA
Minimizing DFA

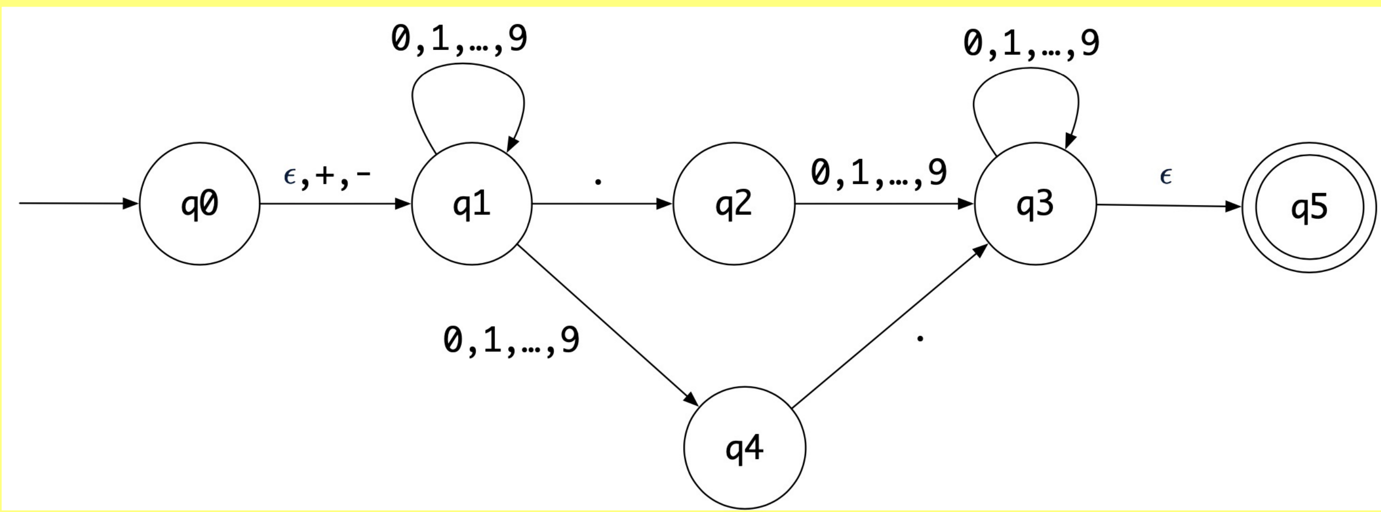
epsilon-NFA: Example

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

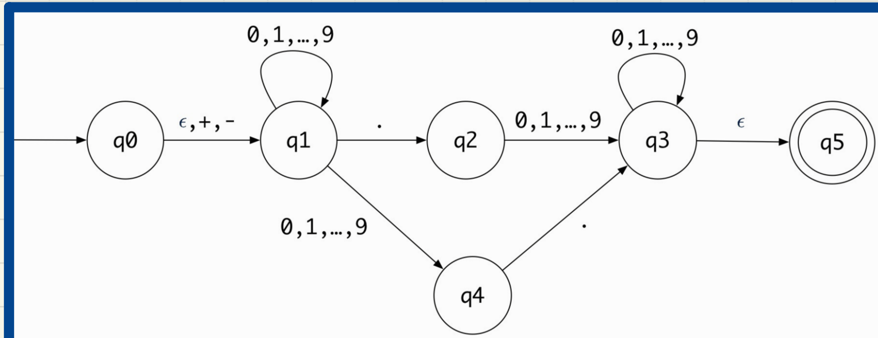
Is this a DFA? N.

Is this an NFA? N.

Is this an ε-NFA? Y.



epsilon-NFA: Formulation (1)



An ϵ-*NFA* is a 5-tuple

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

Draw the transition table.

	ϵ	+, -	.	0 .. 9
q_0	$\{q_1\}$	$\{q_1\}$	$\varnothing$	$\varnothing$
q_1	$\varnothing$	$\varnothing$	$\{q_2\}$	$\{q_1, q_4\}$
q_2	$\varnothing$	$\varnothing$	$\varnothing$	$\{q_3\}$
q_3	$\{q_5\}$	$\varnothing$	$\varnothing$	$\{q_3\}$
q_4	$\varnothing$	$\varnothing$	$\{q_3\}$	$\varnothing$
q_5	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$

Σ

epsilon-NFA: Formulation (2)

An ϵ-NFA is a 5-tuple

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

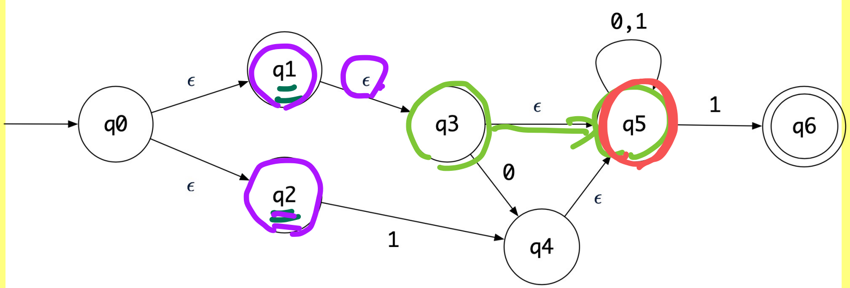
we define the *epsilon closure* (or *ϵ-closure*) as a function

$$\mathrm{ECLOSE} : Q \rightarrow \mathbb{P}(Q)$$

For any state $q \in Q$

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

ECLOSE of all states reachable from p via ε.



Derive ECLOSE(q0).

$\mathrm{ECLOSE}(q_0)$

$= \{q_0\} \cup \mathrm{ECLOSE}(q_1) \cup \mathrm{ECLOSE}(q_2)$

$\mathrm{ECLOSE}(q_1) = \{q_1\} \cup \boxed{\mathrm{ECLOSE}(q_3)}$

$\mathrm{ECLOSE}(q_2) = \{q_2\}$

$\mathrm{ECLOSE}(q_3) = \{q_3\} \cup \mathrm{ECLOSE}(q_5)$

$\mathrm{ECLOSE}(q_5) = \{q_5\}$

Answer:

$\{q_0, q_1, q_3, q_2, q_5\}$

epsilon-NFA: Formulation (3)

An ϵ-*NFA* is a 5-tuple

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

DFA: q

NFA: $\{q\}$

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

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

compare with $\hat{\delta}$ of NFA

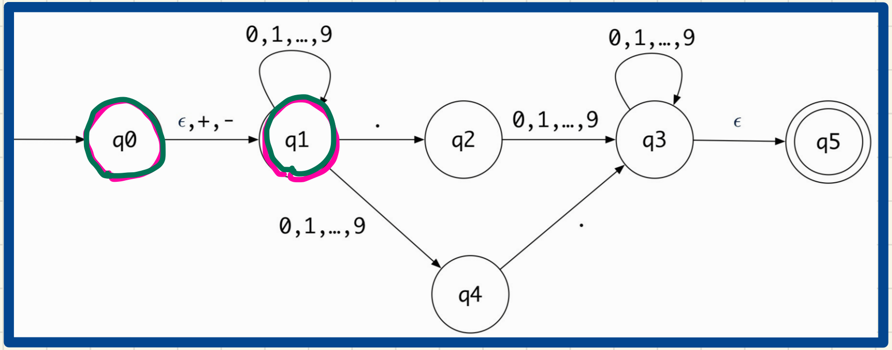
$$\begin{aligned} \boxed{\hat{\delta}(q, \epsilon)} &= \mathrm{ECLOSE}(q) \\ \hat{\delta}(q, xa) &= \bigcup\{\, \mathrm{ECLOSE}(q'') \mid q'' \in \delta(q', a) \wedge q' \in \hat{\delta}(q, x) \,\} \end{aligned}$$

$$q \xrightarrow{x} q' \xrightarrow{a} q''$$

Language of a epsilon-NFA

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

epsilon-NFA: Processing Strings



Exercises

① .6

② +23

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

$\hat{\delta}(q_0, \epsilon) =$ $\{q_0, q_1\}$

- **Read 5:** $\delta(q_0, 5) \cup \delta(q_1, 5) = \emptyset \cup \{q_1, q_4\} = \{q_1, q_4\}$

$\hat{\delta}(q_0, 5) =$ $ECLOSE(q_1) \cup ECLOSE(q_4) = \{q_1\} \cup \{q_4\} = \{q_1, q_4\}$

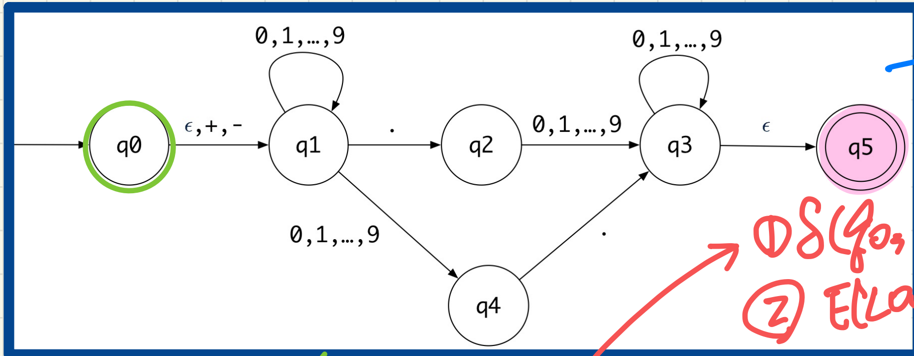
- **Read .:**

$\hat{\delta}(q_0, 5.) =$ Exercise

- **Read 6:**

$\hat{\delta}(q_0, 5.6) =$

epsilon-NFA to DFA: Extended Subset Construction



→ ε-NFA

① $\delta(q_0, d) \cup \delta(q_1, d) = \boxed{\because}$

② ECLOSE(⋯)

δ of DFA. (no ε transition)

subset state

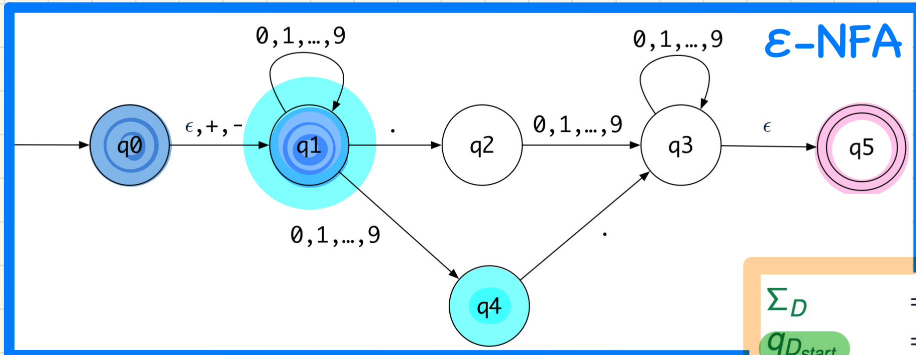
ECLOSE(q_0)

initial state of DFA.

	$d \in 0..9$	$s \in \{+,-\}$	.
$\{q_0, q_1\}$		$\{q_1\}$	$\{q_2\}$
$\{q_1, q_4\}$	$\{q_1, q_4\}$	$\varnothing$	$\{q_2, q_3, q_5\}$
$\{q_1\}$	$\{q_1, q_4\}$	$\varnothing$	$\{q_2\}$
$\{q_2\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_2, q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$

accepting (subset) states of DFA.

epsilon-NFA to DFA: Extended Subset Construction



$$
\begin{array}{lcl}
\Sigma_D & = & \Sigma_N \\
q_{D_{start}} & = & \text{ECLOSE}(q_0) \\
F_D & = & \{\, S \mid S \subseteq Q_N \land S \cap F_N \neq \varnothing \,\} \\
Q_D & = & \{\, S \mid \boxed{S \subseteq Q_N} \land (\exists w \bullet w \in \Sigma^* \land S = \hat{\delta}_N(q_0, w)) \,\} \\
\delta_D(S, a) & = & \bigcup \{\, \text{ECLOSE}(s') \mid s \in S \land s' \in \delta_N(s, a) \,\}
\end{array}
$$

each DFA state is a subset of states in ε-NFA

w is a string

all subset states reachable from q_0

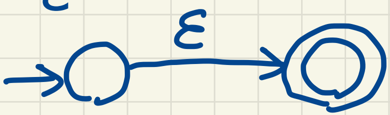
	$d \in 0..9$	$s \in \{+, -\}$	.
$\{q_0, q_1\}$	$\{q_1, q_4\}$	$\{q_1\}$	$\{q_2\}$
$\{q_1, q_4\}$	$\{q_1, q_4\}$	$\varnothing$	$\{q_2, q_3, q_5\}$
$\{q_1\}$	$\{q_1, q_4\}$	$\varnothing$	$\{q_2\}$
$\{q_2\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_2, q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$
$\{q_3, q_5\}$	$\{q_3, q_5\}$	$\varnothing$	$\varnothing$

DFA

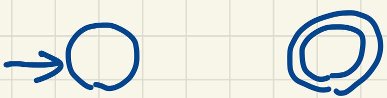
Regular Expression to epsilon-NFA

Base Cases

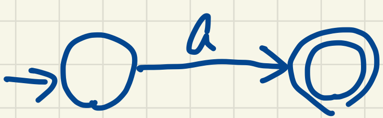
ε



$\emptyset$

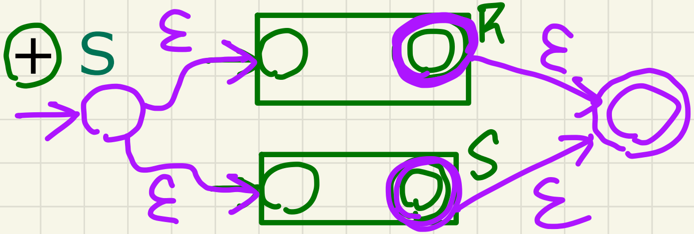


a $\quad (a \in \Sigma)$

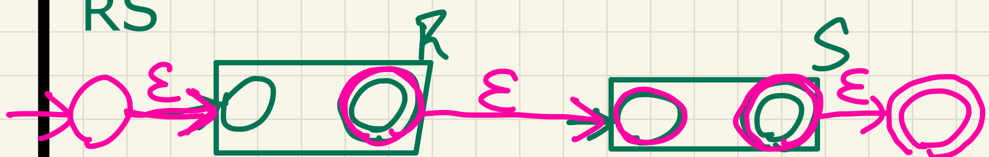


Recursive Cases (given REs E and F)

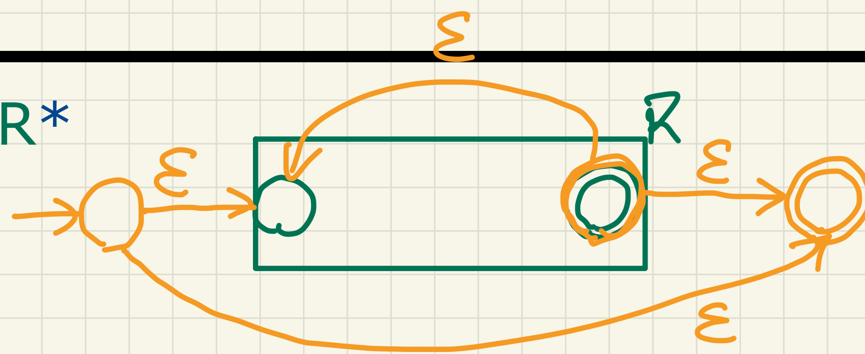
$R \oplus S$



RS

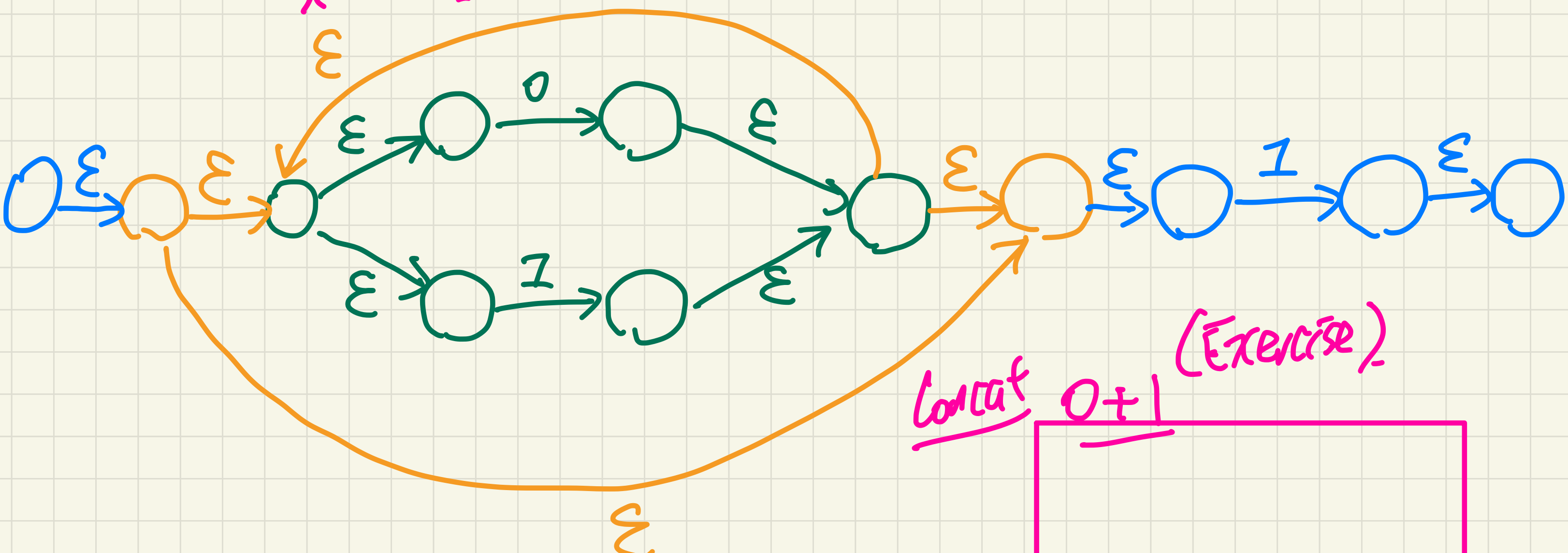


R*



Regular Expression to epsilon-NFA: Example

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



(Exercise)

Concat $0+1$

Minimizing DFA: Algorithm

① What if $M' = M \Rightarrow$ no optimization can be done
② Is $|Q(M')| > |Q(M)|$ possible? $\Rightarrow$ algo. not achieving what it's supposed to.

```
ALGORITHM: MinimizeDFAStates
  INPUT: DFA M = (Q, Σ, δ, q0, F)
  OUTPUT: M' s.t. minimum |Q| and equivalent behaviour as M
PROCEDURE:
  P := ∅ /* refined partition so far */
  T := { F, Q − F } /* last refined partition */
  while (P ≠ T):
    P := T
    T := ∅
    for(p ∈ P s.t. |p| > 1):
      find the maximal S ⊂ p s.t. splittable(p, S)
      if S ≠ ∅ then
        T := T ∪ {S, p − S}
      else
        T := T ∪ {p}
      end
```

splittable(p, S) holds iff there is $c \in \Sigma$ s.t.

1. $S \subset p$ (or equivalently: $p - S \neq \varnothing$)
2. Transitions via c lead all $s \in S$ to states in ***same partition*** $p1$ ($p1 \neq p$).

Partitions of States

input

e.g., $Q = \{s0, s1, s2, s3\}$

- Smallest number of partitions.
- Largest number of partitions.
- Partitions somewhere in-between
- Analogy from Software Testing: Equivalent Classes

single partition

$Q' = \{ \boxed{\{s_0, s_1, s_2, s_3\}} \}$

$Q' = \{ \{s_0\}, \{s_1\}, \{s_2\}, \{s_3\} \}$

no optimization