Lecture 11 - June 10

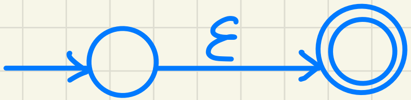
Lexical Analysis

Formulation: Ext. Subset Construction
Minimizing DFA

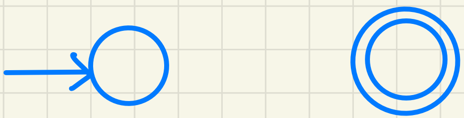
Regular Expression to epsilon-NFA

Base Cases

$\boxed{\varepsilon}$ $L(\varepsilon) = \{\varepsilon\}$

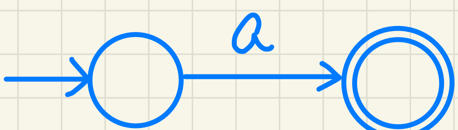


$\emptyset$ $L(\phi) = \phi$



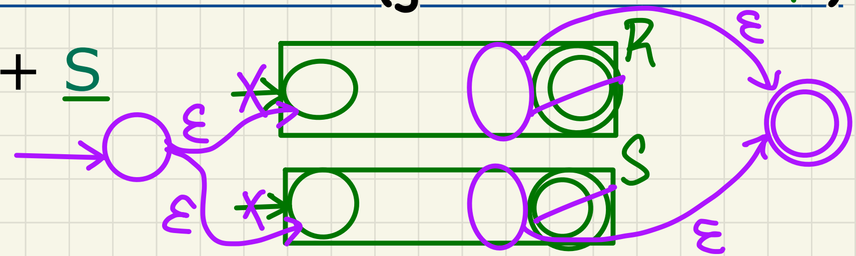
$0 \in \{0, 1\}$

a $(a \in \Sigma)$

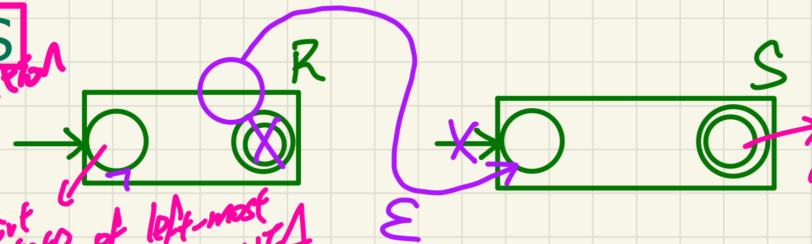


Recursive Cases (given REs ~~E~~ R and ~~F~~ S)

R + S



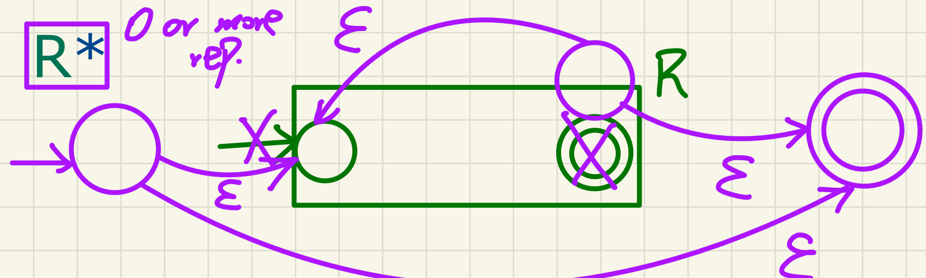
$\boxed{RS}$ correction



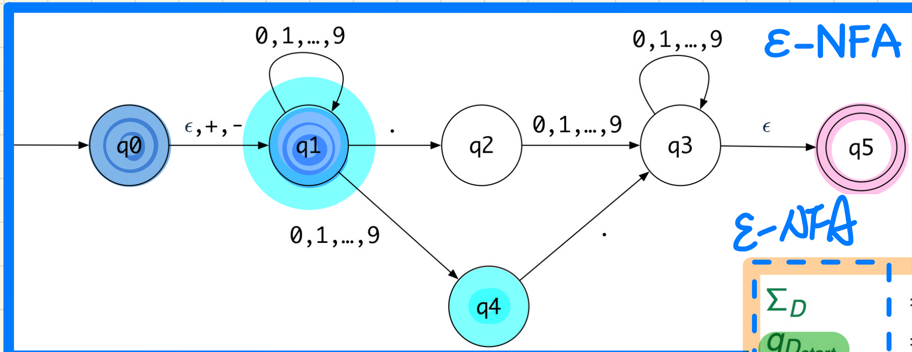
start state of left-most ε-NFA

accept state of right-most ε-NFA

$\boxed{R^*}$ 0 or more rep.



epsilon-NFA to DFA: Extended Subset Construction



$\varepsilon \notin \Sigma_N$

DFA ·

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

Any subset state (a state in DFA) containing an accept state (ε-NFA) is considered accepting

** $S \xrightarrow{a} S' \rightsquigarrow \mathrm{ECLOSE}(S')$ reachable states

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

DFA

* S is a subset state, which can be a state in the DFA

** S is reachable from q_0 via w (w.r.t. $\hat{\delta}$)

exit when reaching a fix point: $P = T$

Minimizing DFA: Algorithm

① What if $M' = M$ ⇒ no optimization was necessary

② What if $|Q_{M'}| > |Q_M|$?

should not be possible!

```
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):
      find the maximal S ⊂ p s.t. splittable(p, S)
      if S ≠ ∅ then
        T := T ∪ {S, p − S}
      else
        T := T ∪ {p}
      end
```

accept states

non-accept states

as soon as $P = T$ ⇒ no further optimization can be done.

examine each part. from last it.

exit: $P = T$

result from last it.

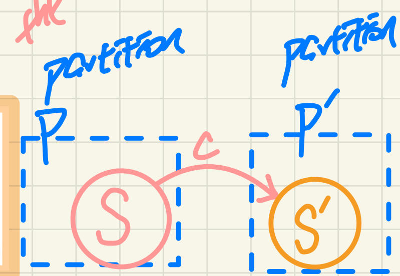
result from current it.

current part. being examined

given a set P of states as a partition, can we find a proper subset of P that can be split from the rest of P.

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

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



Partitions of States

subsets (non-empty)

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

P'
- Smallest number of partitions

P''
- Largest number of partitions

P'''
- Partitions somewhere in-between
- Analogy from Software Testing: Equivalent Classes

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

↳ 3 partitions.

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

↳ a single partition.

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

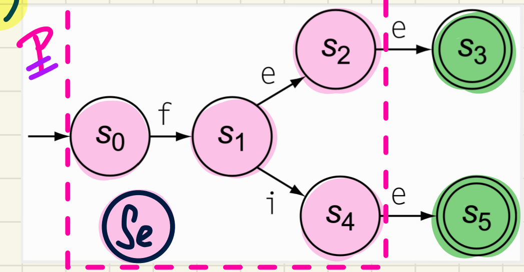
↳ 4 partitions.

Minimizing DFA: Example (1)

input symbols to focus on: f, i, e, other

(1)

P



f $\{s_0\} \rightarrow \{s_1\}$

$\{s_e\} \rightarrow \{s_e\}$

i, other

↳ no split

CP

e $\{s_2, s_4\} \rightarrow \{s_3, s_5\}$

$\{s_0, s_1, s_e\} \rightarrow$

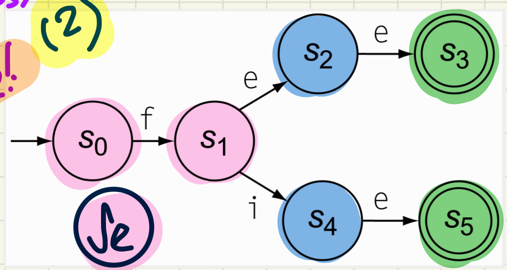
$P = \{s_3, s_5\}$

$\{s_3, s_5\} \xrightarrow{e} \{s_e\}$ splittable

$\not\subset \{s_3, s_5\}$

↳ not splittable!

(2)



f: no split

i: splittable

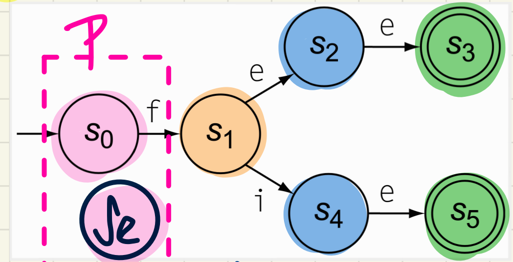
$\{s_1\} \xrightarrow{i} \{s_4\}$

$\{s_0, s_1\} \xrightarrow{i} \{s_e\}$

↳ not splittable

(3)

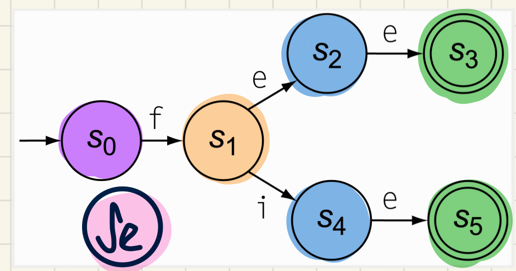
P



f $\{s_0\} \xrightarrow{f} \{s_1\}$ → splittable

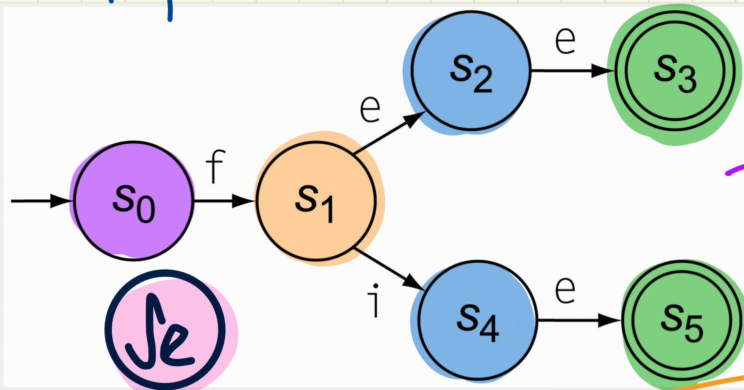
$\{s_e\} \xrightarrow{f} \{s_e\}$ → not splittable

(4)



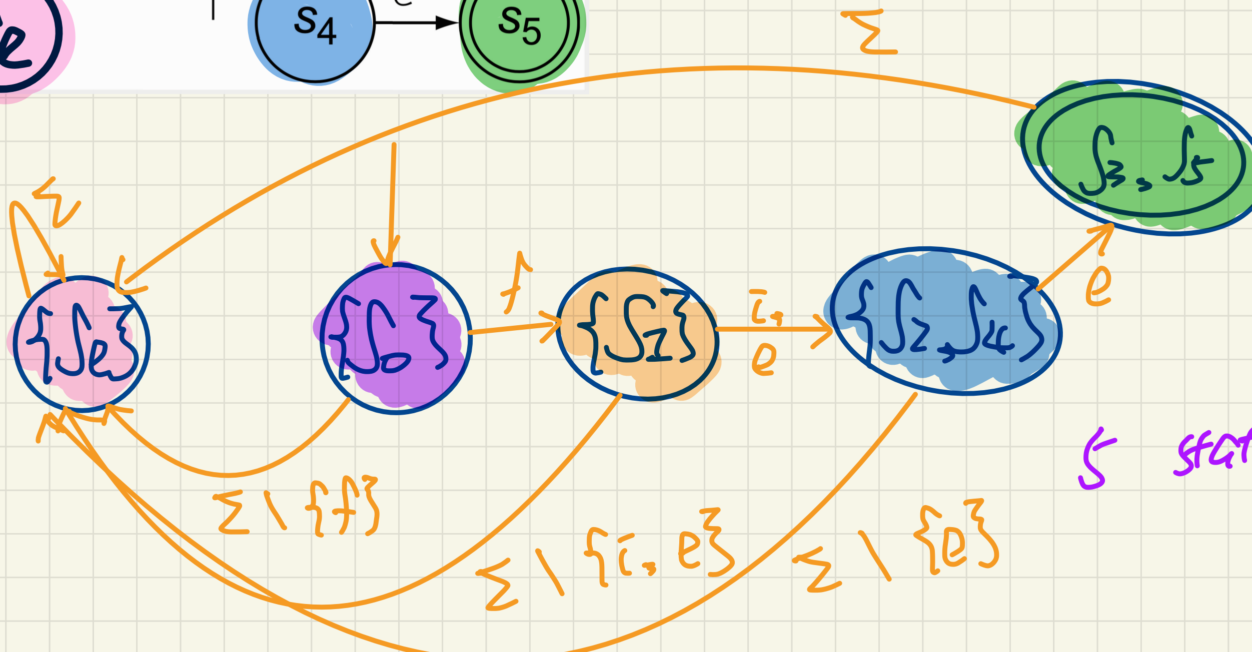
↳ not splittable for any partition anymore.

Maximal # partitions → min # of DFA states



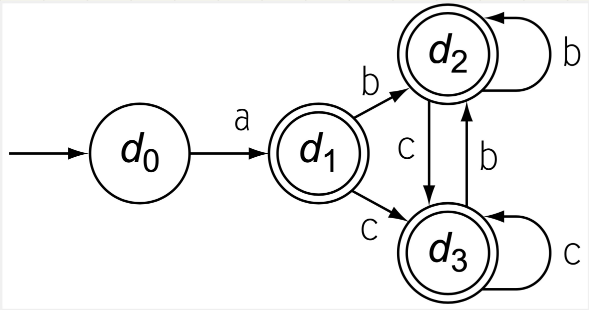
7 states

2/7 ≈ 30% reduction



5 states

Minimizing DFA: Example (2)



Minimizing DFA: Example (3)