CSCE 355
2/7/2022

today

Regex ⇄ ε-NFA

Recall: A regex is an expr built from
$\emptyset$, a $(a \in \Sigma)$, using $+$, $\cdot$, $*$
union, concat, Kleene closure

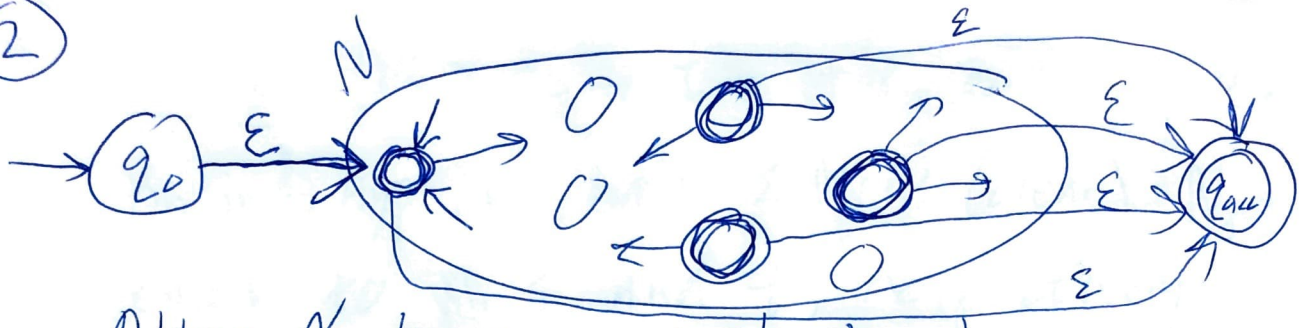
Given a regex r as input, construct an equivalent ε-NFA by induction/recursion on the syntax of r.

Def: An ε-NFA is **clean** if
1) it has exactly one accepting state
2) the accepting state is not the start state
3) no transitions into the start state
4) no transitions out of the accepting state.

Lemma: For every ε-NFA there exists an equivalent clean ε-NFA.

Proof: Let N be an ε-NFA.

(2)



Alter N to an equivalent clean ε-NFA N' by:

1) Add a new start state q_0 (rejecting) and new accept state q_{acc}
2) Add ε-move from q_0 to the old start state
3) Add ε-moves from all old accept states to q_{acc}
4) Make all the old accept states rejecting

Recursive:

Rules for converting any regex (over an alphabet Σ) into an equivalent ε-NFA:

Base cases: Atomic regexes:

$\emptyset$ ⇒ 

(clean) recognizes $\emptyset$

$a \in \Sigma$ ⇒ 

recognizes {a}

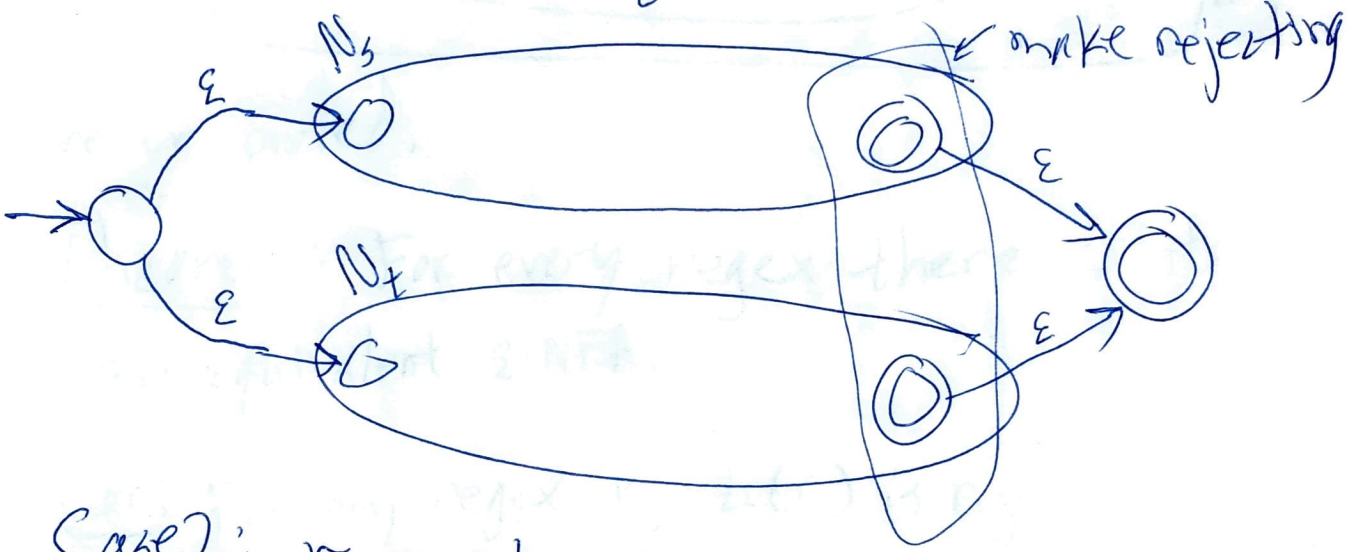
Inductive cases: nonatomic regexes.

Let r be some nonatomic regex.

(3) Case 1: $r = s + t$ for some regexes s, t.

By ind. hyp. we have Σ-NFAs (clean) N_s, N_t equiv to s and t, respectively.

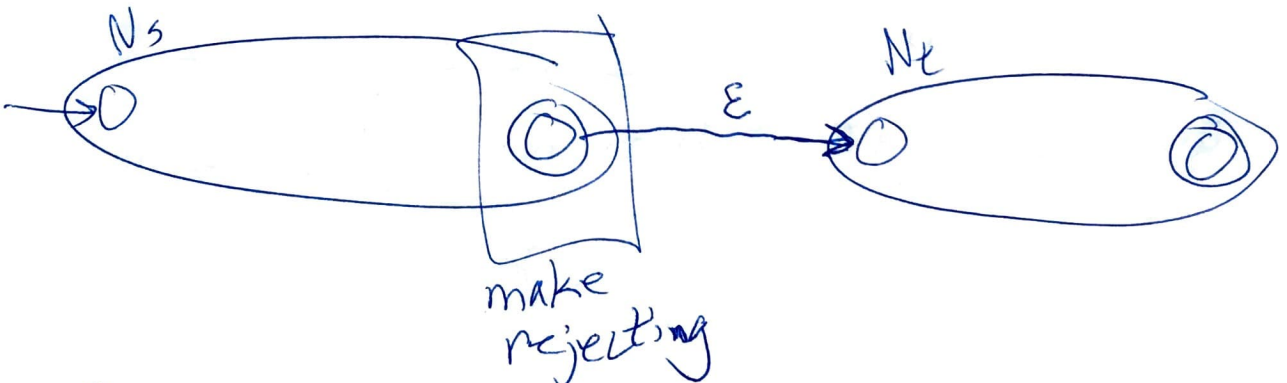
Here is an Σ-NFA equiv to $r = s + t$:



Case 2: $r = st$ for regexes s, t.

Let N_s & N_t be as in Case 1.

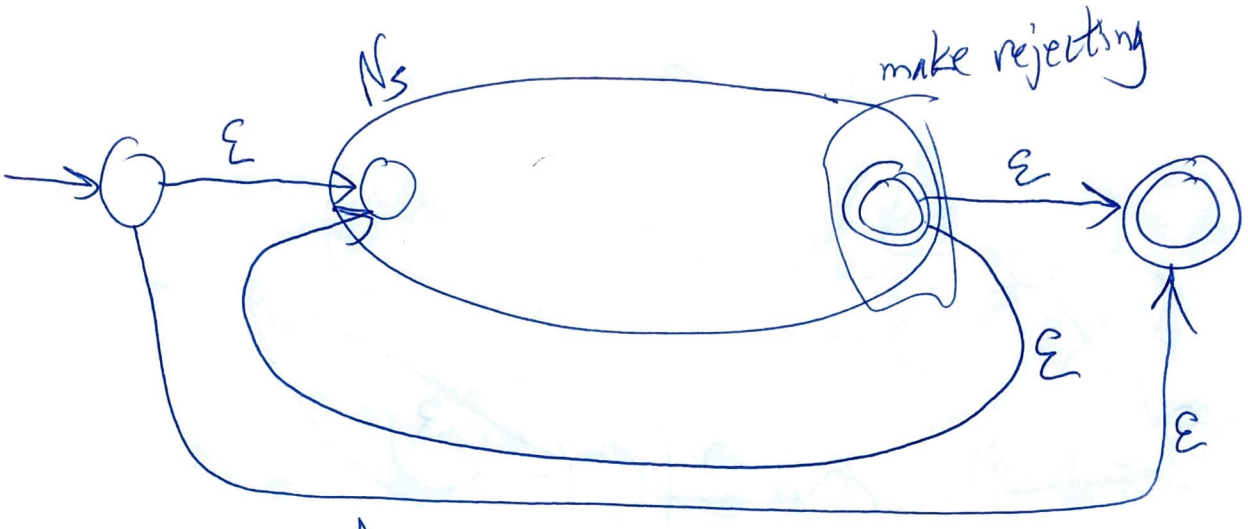
Here is an Σ-NFA equiv to $r = st$



Case 3: $r = s^*$ for some regex s

Let N_s be an clean Σ-NFA recognizing $L(s)$ (equiv to s)

(4)



We've proved:

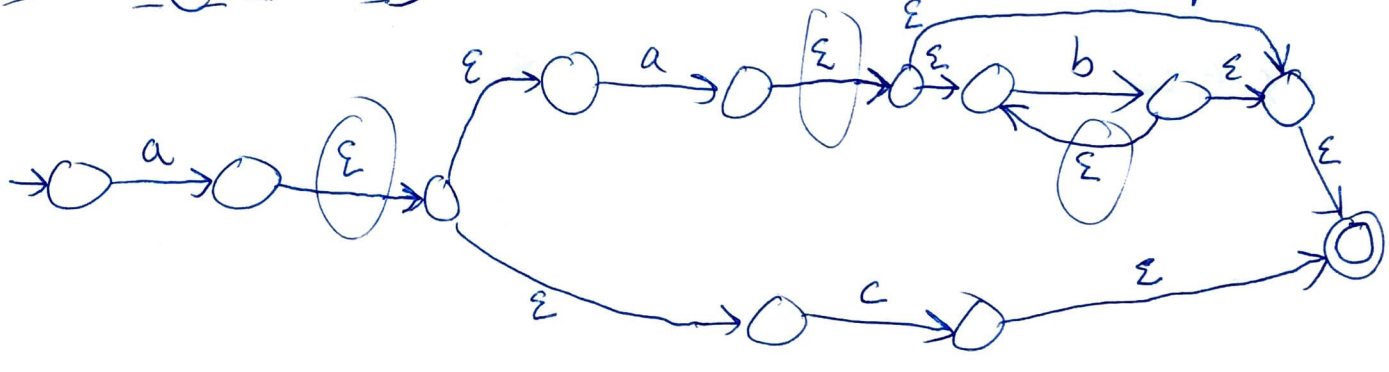
Theorem: For every regex there exists an equivalent ε-NFA.

Cor: For any regex r, L(r) is regular.

Pf: r ⟹ equiv ε-NFA ⟹ equiv NFA ⟹ equiv DFA (recog. L(r))

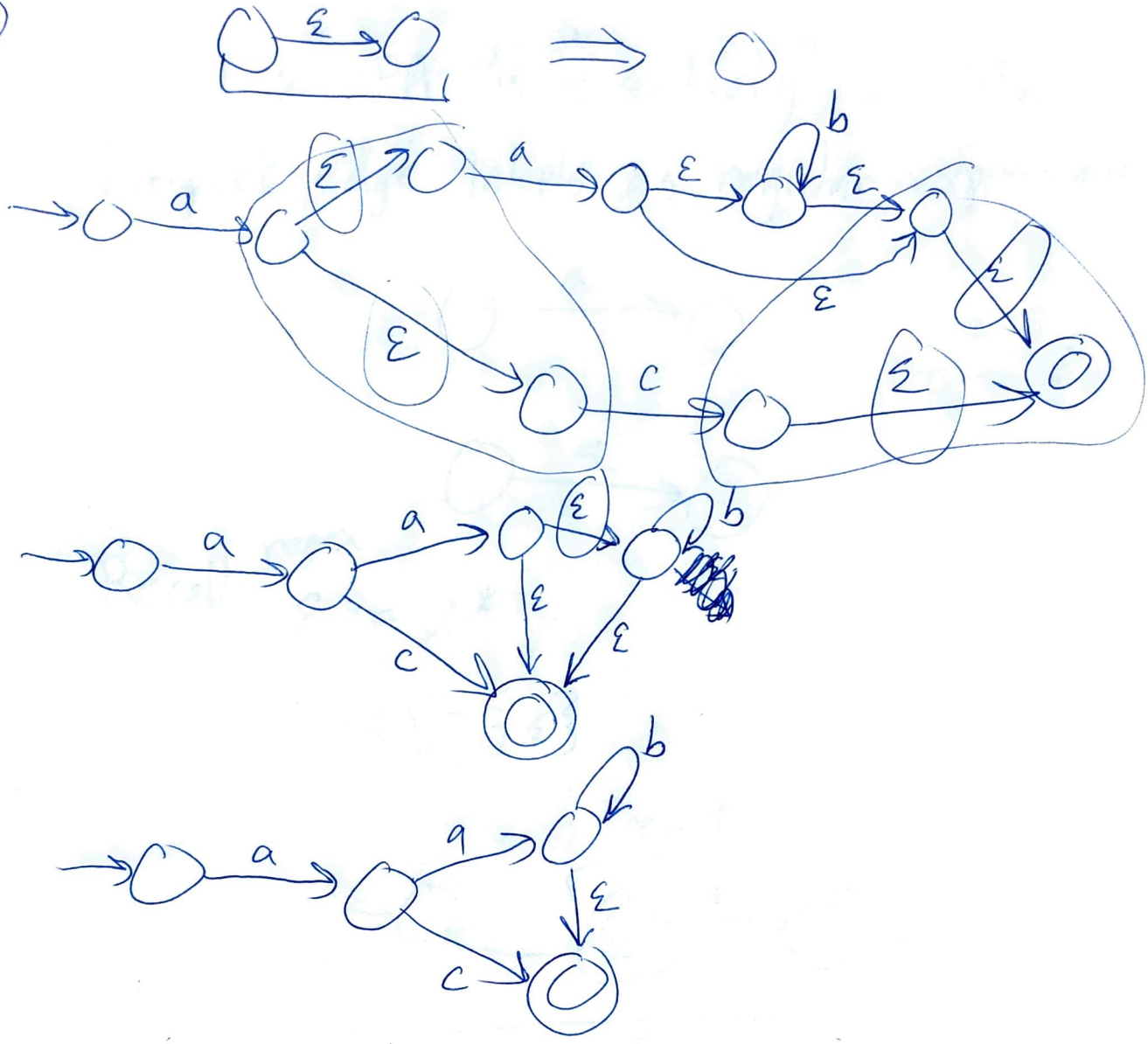
↑ today — ↑ last week — ↑ Set-of-states construction

Ex: $a(ab^* + c)$ construct ε-NFA bottom up



Not optimal. Can often contract ε-moves

(5)



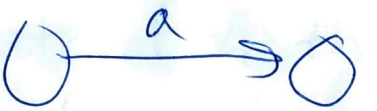
ε-NFA ⇒ regex State elimination method

ε-NFA $N \Rightarrow G_0 \Rightarrow G_1 \Rightarrow \cdots \Rightarrow G_k \Rightarrow$ regex

G_i's are called <u>generalized NFAs</u> (GNFAs)

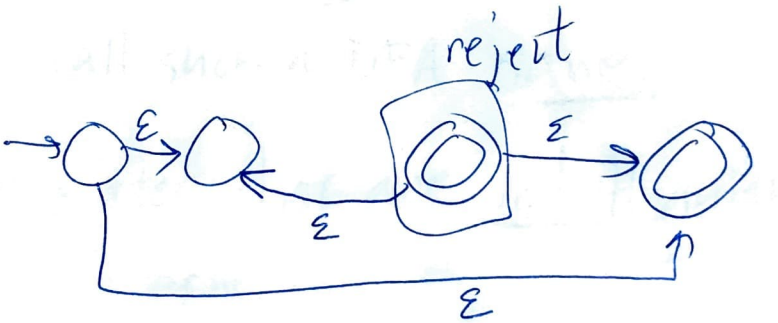
Ex: In a GNFA could have a transition

(6) Every Σ-NFA is "essentially" a GNFA:
Interpret edge labels as regular expressions



Recall regex
$\varepsilon := \emptyset^*$

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



recognizes $\{\varepsilon\}$