CSCE 355
2/5/2024

Today: regex $\longrightarrow$ ε-NFA [$\rightarrow$ NFA $\rightarrow$ DFA]

Recall: $\emptyset$, a (any $a \in \Sigma$) or
$s+t$, st, s^* (regexes s, t)

$L(\emptyset) = \emptyset$

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

$L(s+t) = L(s) \cup L(t)$

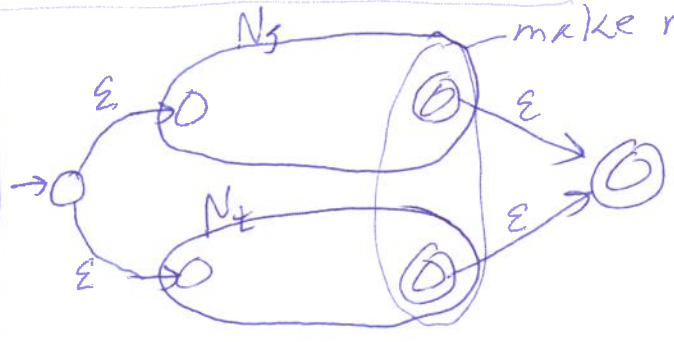
$L(st) = L(s)L(t)$

$L(s^*) = \{\varepsilon\} \cup L \cup LL \cup LLL \cup \cdots$

regex to ε-NFA Fix alphabet Σ.

Given a regex r, construct a clean ε-NFA N_r such that $L(r) = L(N_r)$ (regular).

By induction on the syntax of r:

r	N_r
$\emptyset$	 clean (see next page)
a $(a \in \Sigma)$	
$s+t$ (s, t regexes)	 make rejecting

Def. An ε-NFA is *clean* if

1) it has exactly one accept state
2) accept state is not the start state
3) no transitions into the start state
4) " " out of the accept state.

Prop: Every ε-NFA can be converted into an equivalent clean ε-NFA.

Proof:

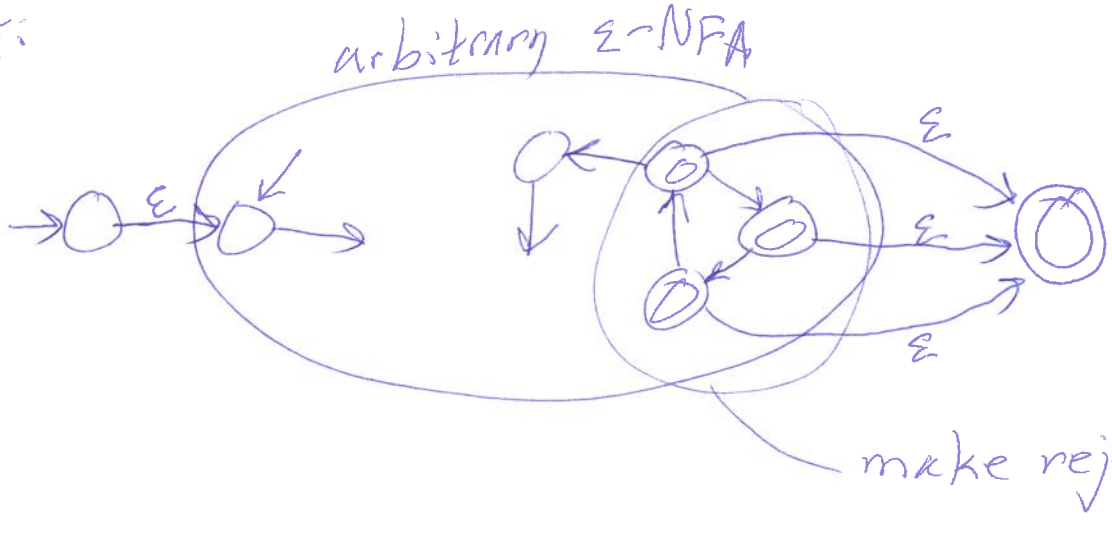
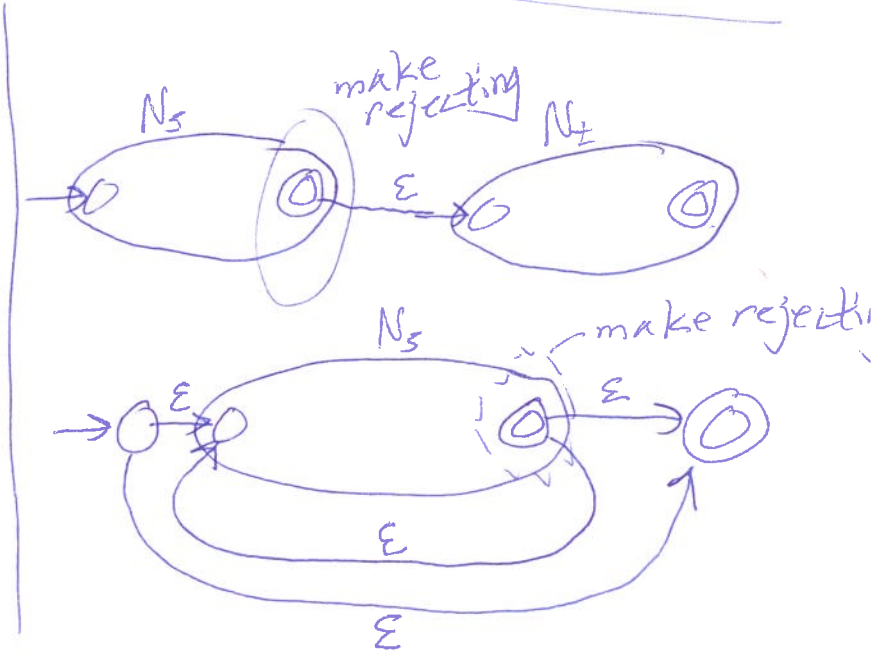


Table (cont.)

$s t$

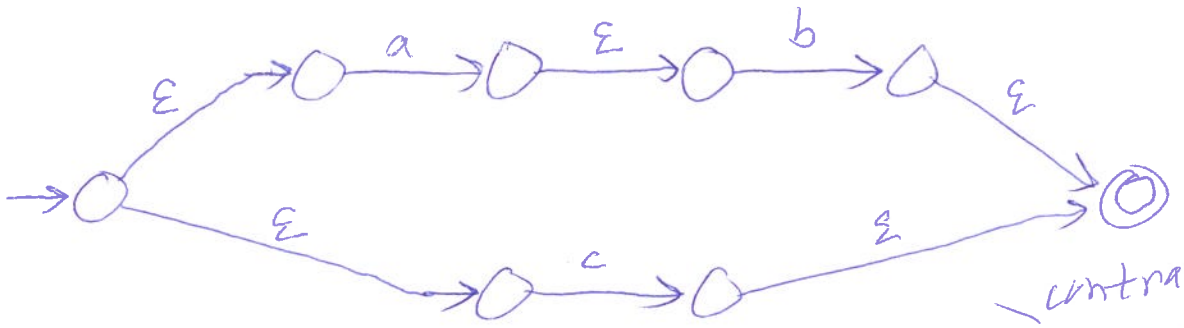
s^*



Examples: $ab + c$

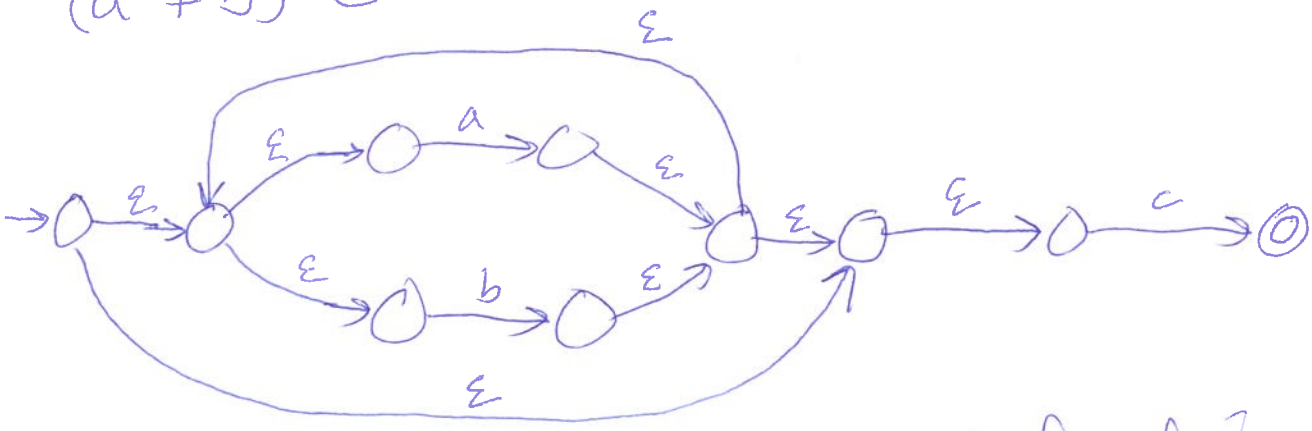
$L(ab+c) = \{ab, c\}$

work from the bottom up
atomic regexes first,
then combine.

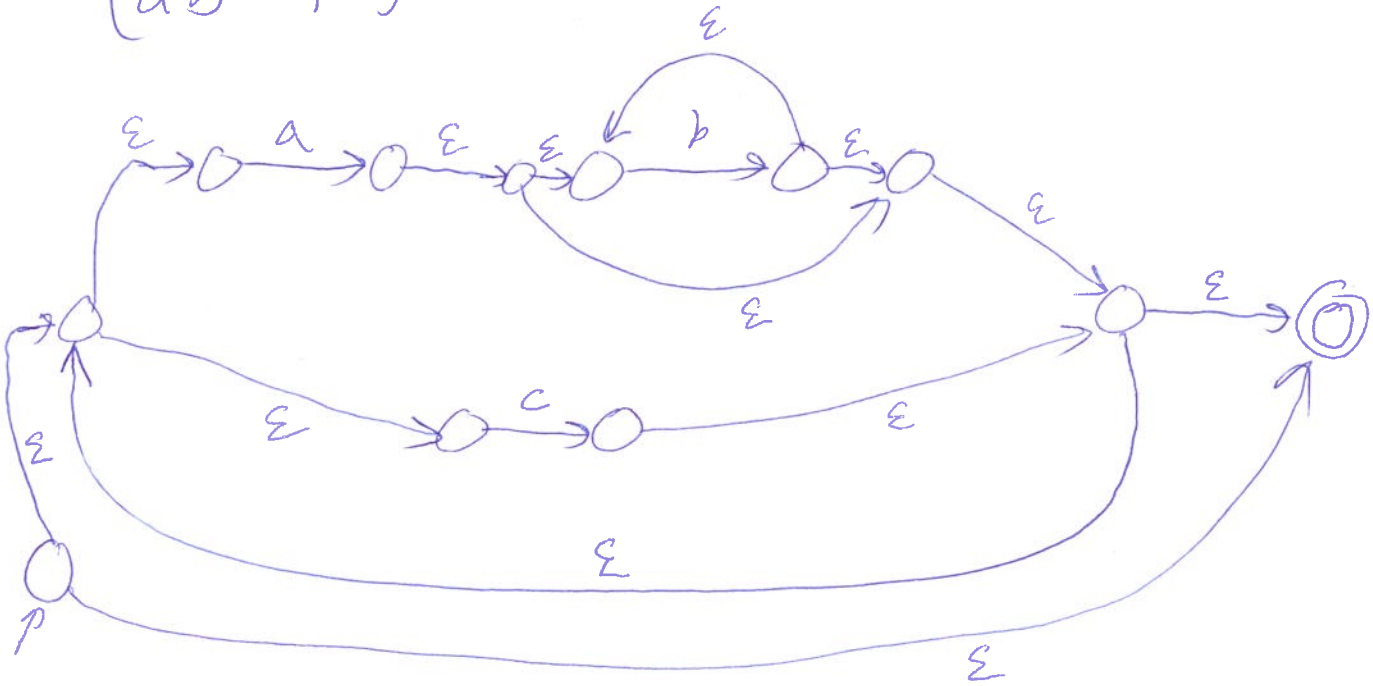


contract ε-moves

$(a + b)^* c$



$(ab^* + c)^*$



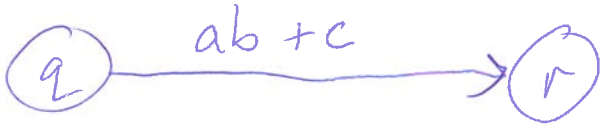
ε-NFA (or NFA or DFA) ⟶ equivalent regex

State elimination method

GNFA (GFA) — Generalized NFA

Like an NFA, but edge labels can now be arbitrary regexes.

Ex:



In general:

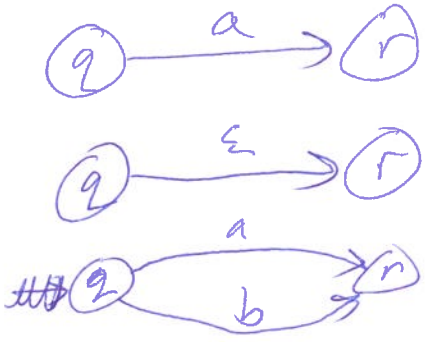
R a regex

means: can get from state q to state r by reading a string matching R.

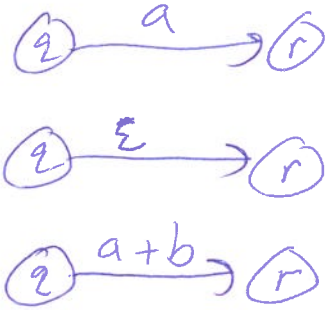
Notice: Every ε-NFA is essentially a GNFA:

ε-NFA

$a \in \Sigma$



GNFA

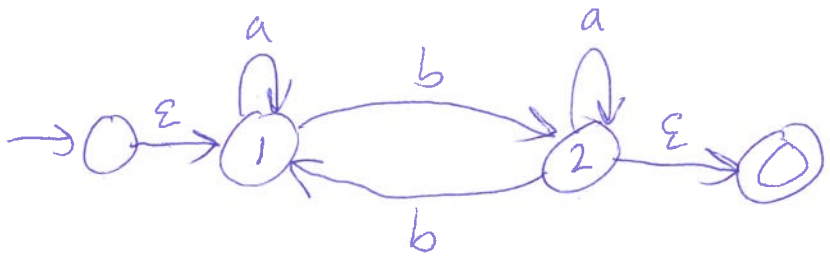


$[\varepsilon + \emptyset^*]$

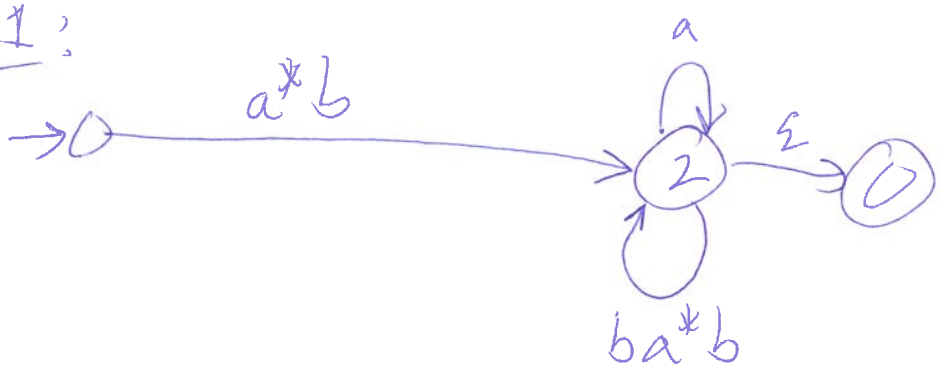
edge consolidation

Ex: # $\Sigma = \{a, b\}$

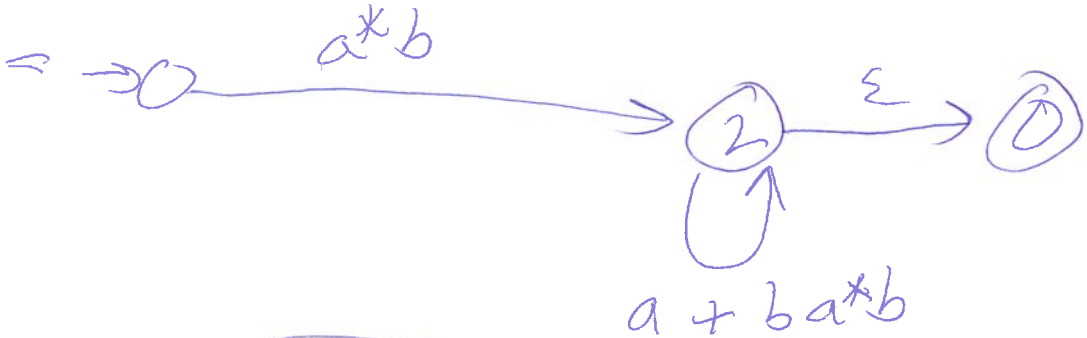
$L = \{w \in \Sigma^* : w \text{ has an odd \# of } b\text{'s}\}$



Elim 1:



=



Elim 2: