ε-NFA $\longrightarrow$ regex

Def: Given alphabet Σ, let REX_Σ be the set of all regexes over Σ.

A generalized (nondeterministic) finite automaton (GNFA or GFA) is a tuple $\langle Q, \Sigma, \delta, q_0, F \rangle$ where Q, Σ, q_0, F are as with an ε-NFA and

$$\delta : Q \times Q \longrightarrow REX_\Sigma.$$

Transition diagram:

means $\delta(q, r) = R$

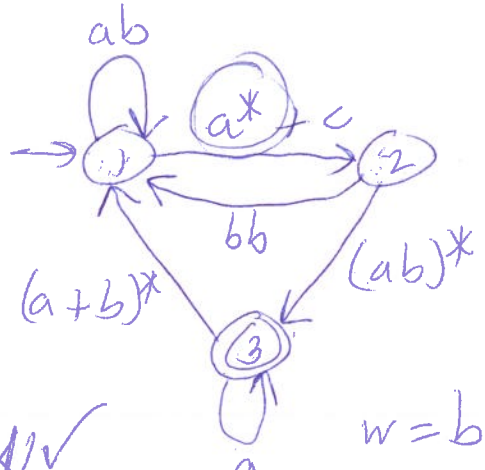
unique edge from q to r

$\delta(q, r) = \emptyset$

(same as 

)

Ex:



w = aabbc ✓

w = aab ✓

w = ε ✓

w = cc ✓

w = b ✓

no rejected strings

Tabular form

arrow tail \ arrow head	1	2	3
→1	ab	a*+c	∅
2	bb	∅	(ab)*
*3	(a+b)*	∅	a

Def (GNFA semantics): Let $G = \langle Q, \Sigma, \delta, q_0, F \rangle$ be a GNFA and let $w \in \Sigma^*$ be any string over the input alphabet. A computation of G on input w is a sequence of states $s_0, s_1, \ldots, s_n \in Q$ such that there exist strings $w_1, w_2, \ldots, w_n \in \Sigma^*$ such that

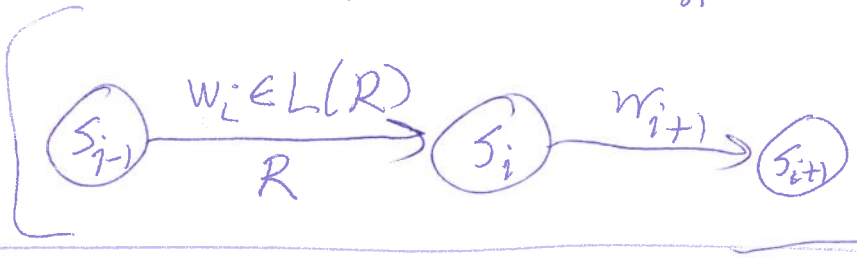
1) $s_0 = q_0$ (start state)

2) $w = w_1 \cdots w_n$

3) For each $1 \le i \le n$,

$$w_i \in L(\delta(s_{i-1}, s_i))$$

Say the computation ends in state s_n.



Plan for ε-NFA $\longrightarrow$ regex:

Start with a clean ε-NFA N.

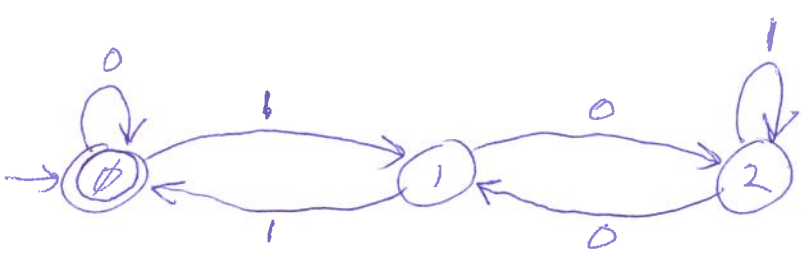
1) Convert N into an equivalent GNFA G_0

2) For $i := 1$ to (#states of N) $- 2$
   Remove some intermediate state from G_{i-1}, add bypassing edges, consolidate multiple edges, let G_i be the result.

3) // G_i has 2 states: start & accepting, with one nonempty arrow from start to accepting, with some label R.

4) Return R.

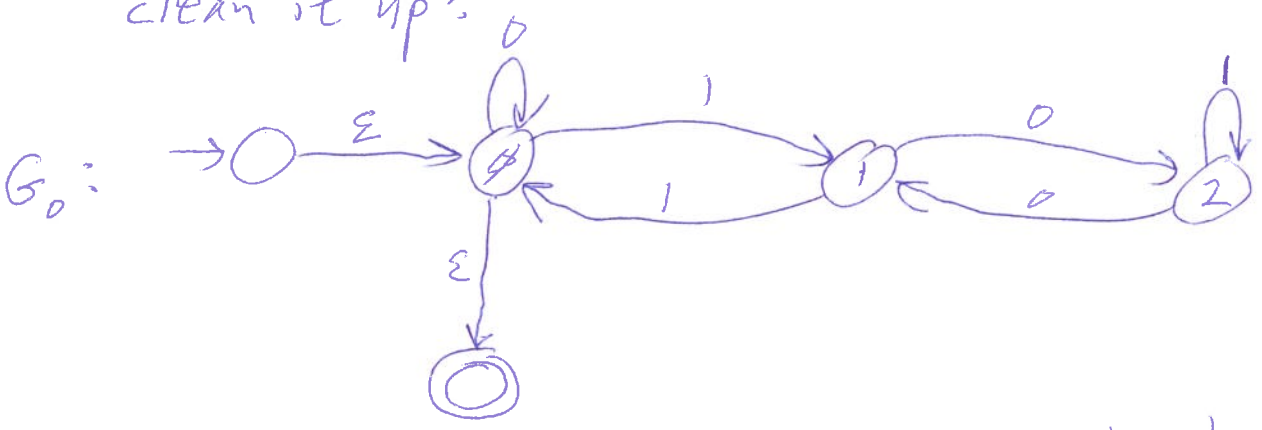
Examples:



recognizes multiples of 3 in binary (ε represents 0)

clean it up:

G_0:

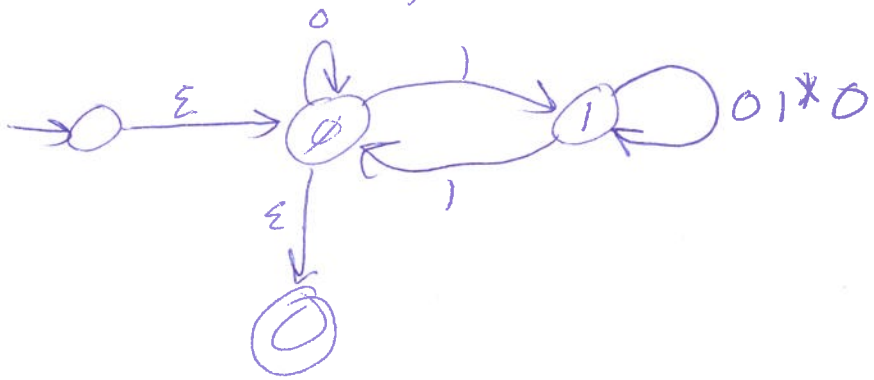


Lazy strategy: remove a state ~~with~~ that causes the fewest bypasses;

#bypasses = (#incoming edges) · (#outgoing edges)
ignoring self-loops

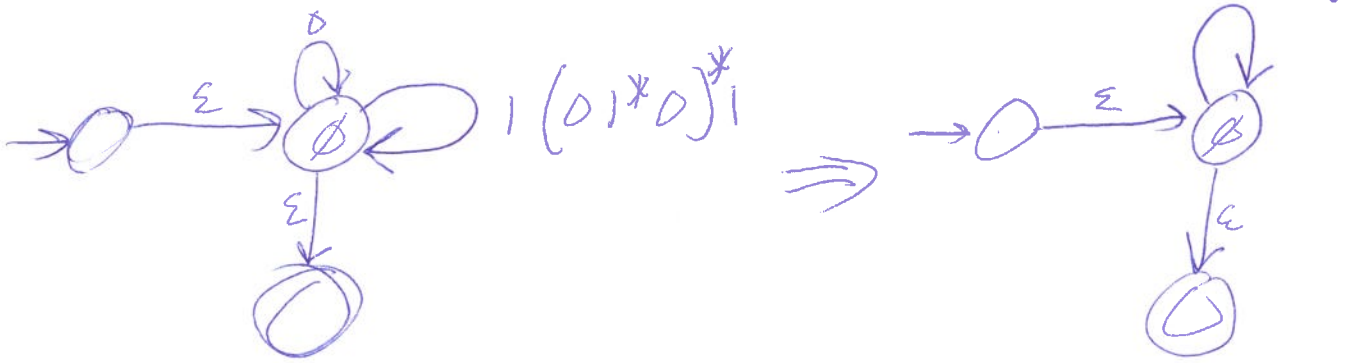
Remove state 2 (1 bypass)

G_1:



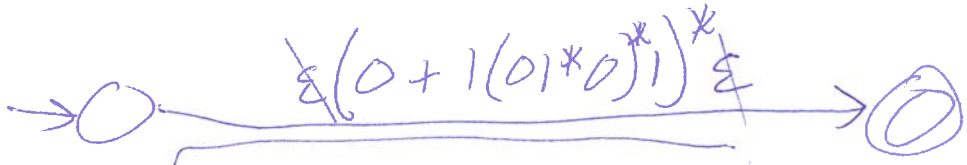
Remove state 1 (1 bypass)

G_2:



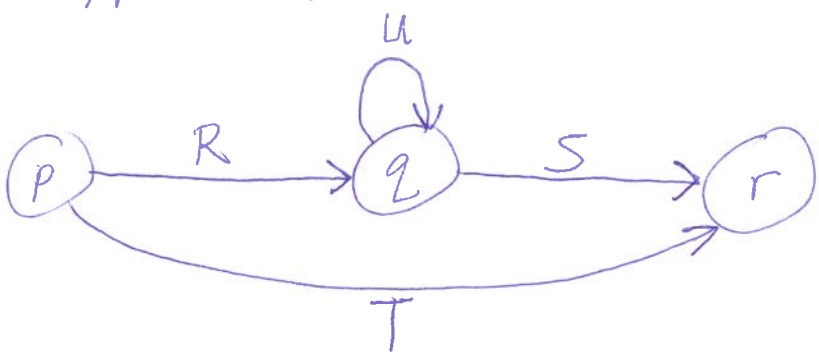
~~Remove~~ Remove state 0 (1 bypass)

G_3:



Answer: $\boxed{(0+1(01*0)*1)*}$

What is a bypass: Bypass q, let p, r be arbitrary states $\neq q$

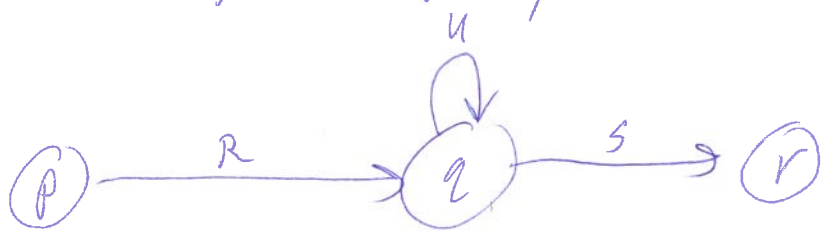


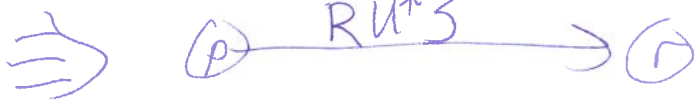
⇛

do this for every ordered pair (p, r) of states distinct from q

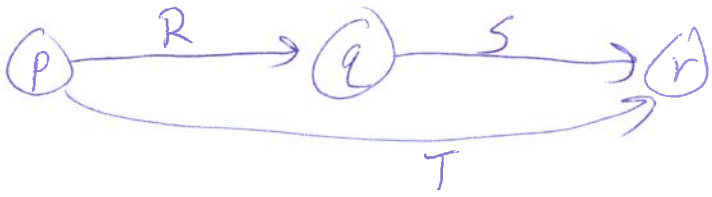
If any of these regexes are $\emptyset$, then can ~~[illegible]~~ simplify:

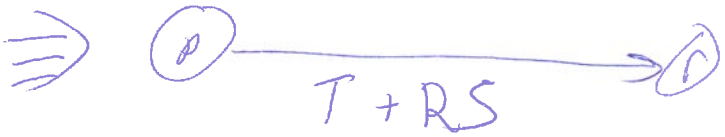
Ex:





Ex

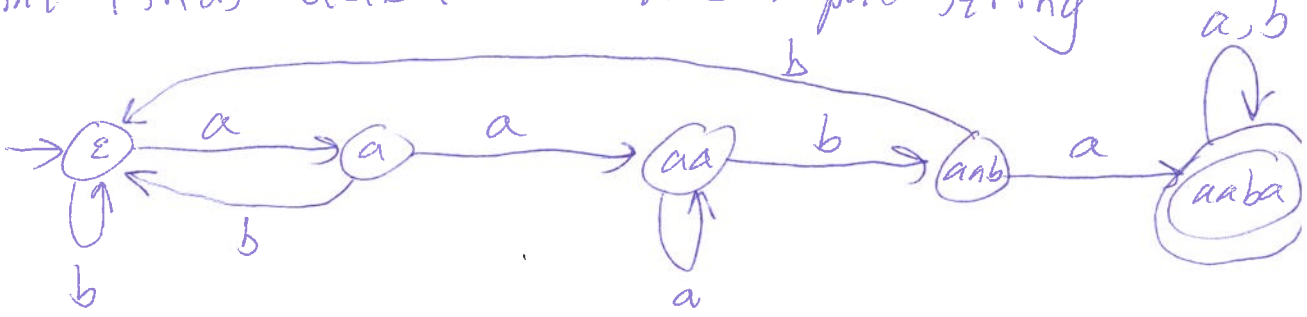




Ex: If $R = \emptyset$ or $S \neq \emptyset$ then no change (no bypass)

Ex DFA that finds aaba in the input string

$\Sigma = \{a, b\}$

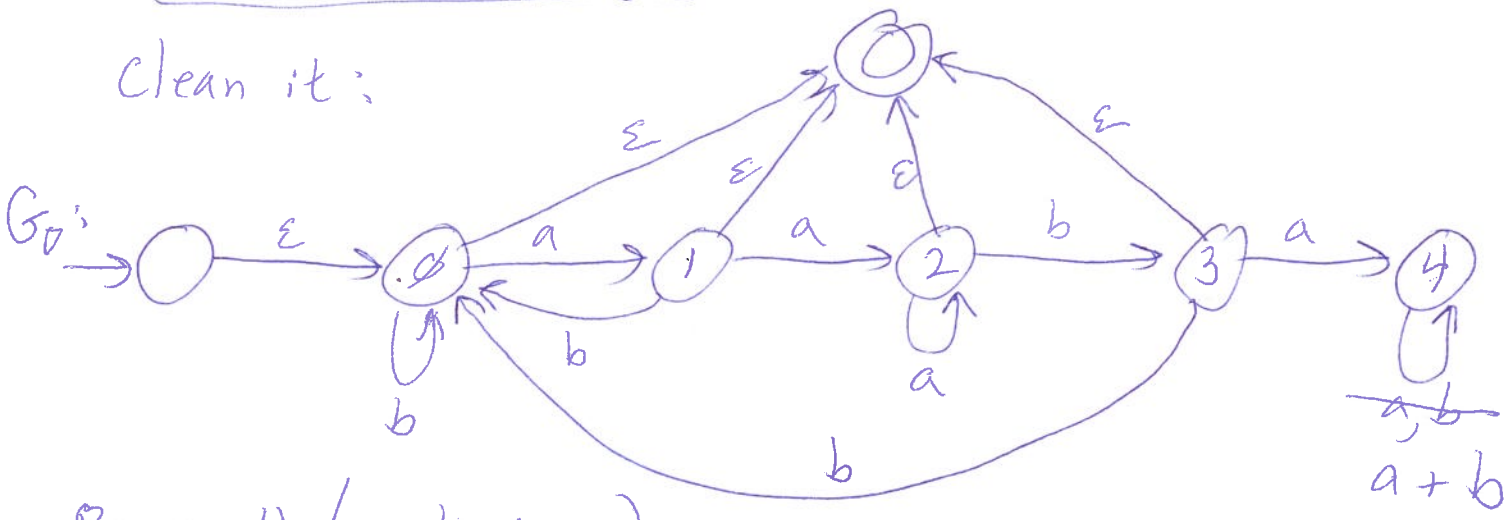


Equiv regex: $(a+b)^*aaba(a+b)^*$

DFA for the complement (accepts string iff no aaba as substring)

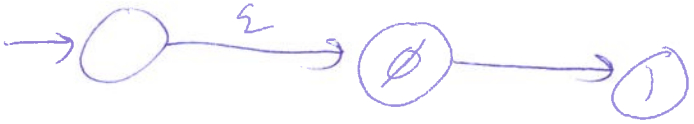
Want equivalent regex

Clean it:



Remove 4 (no bypasses)

Remove 2



Will finish next time