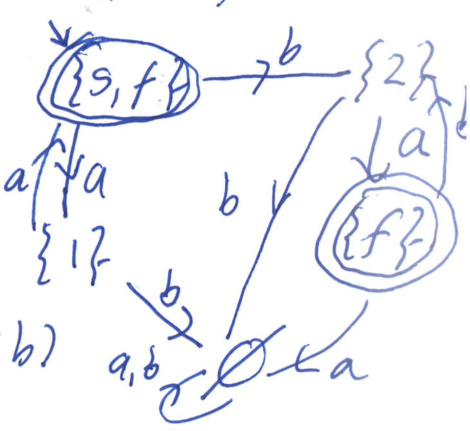
CSE 396 Lecture Thu 2/21, 2019 Spring 2019

Top Hat #2741

Table from prev. lecture:
Regexp is $(aa)^*(ba)^*$

δ	a	b
s	1	$\emptyset$
f	$\emptyset$	2
1	s, f	$\emptyset$
2	f	$\emptyset$



Need to know the δ table and: $S = \{s, f\}$

$\Delta(P, c) = \bigcup_{p \in P} \delta(p, c)$. $F = \{$anything with $f\}$.

$\Delta(S, a) = \delta(s, a) \cup \delta(f, a) = \{1\} \cup \emptyset = \{1\}$ $\quad \Delta(S, b) = \delta(s, b) \cup \delta(f, b) = \emptyset \cup \{2\} = \{2\}$

$\Delta(\{1\}, a) = \delta(1, a) = \{s, f\}$ $\quad \Delta(\{1\}, b) = \emptyset$

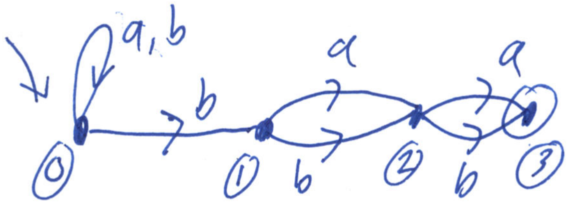
$\Delta(\{2\}, a) = \delta(2, a) = \{f\}$ $\quad \Delta(\{2\}, b) = \emptyset$

$\Delta(\{f\}, a) = \emptyset$ $\quad \Delta(\{f\}, b) = \delta(f, b) = \{2\}$ no new states, so done.

Thus we needed only 5 of 12 possible states.

Next Example:

N:



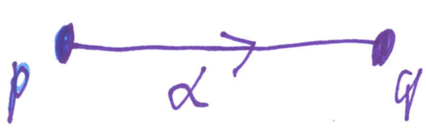
$L(N) = \{x \in \{a,b\}^* : |x| \geq 3$ and the third char from the right is a 'b' $\}$.
$= \{x \in \Sigma^* : |x| \geq 3$ and $x_{n-2} = b\}$
(numbering from 1)

$r_N = (a \cup b)^* b (a \cup b)^2$

FACT: The smallest DFA M s.t. $L(M) = L(N)$ has 8 states.

Next week: Given any $k \geq 1$, put $r_k = (a \cup b)^* b (a \cup b)^{k-1}$. Then any DFA M_k s.t. $L(M_k) = L(r_k)$
The NFA N_k still needs only $k+1$ states. "Exponential Explosion": needs 2^k states!

Defn. A "generalized NFA" (GNFA) is $N = (Q, \Sigma, \delta, s, F)$ again but with $\delta \subseteq (Q \times \text{Regexp}(\Sigma)) \times Q$. "Instructions" look like



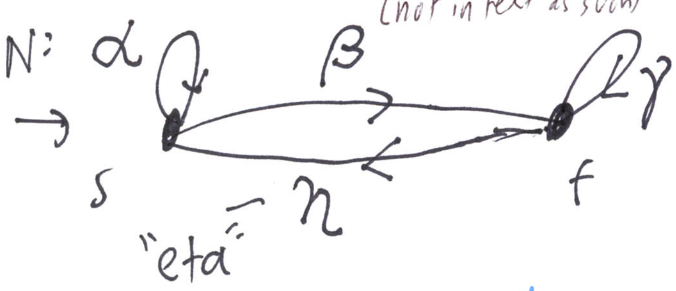
where α is a regular expression over Σ.

The instruction "processes" a substring w from p to q if w matches α.

Computation paths on an input x have the form $(q_0, w_1, q_1, w_2, q_2, \dots q_{m-1}, w_m, q_m)$ where $x = w_1 \cdots w_m$ and for all j, $1 \leq j \leq m$, there is an instruction (q_{j-1}, α, q_j) such that w_j matches α. Then $x \in L_{q_0 q_m}$

$m < |x|$ and $m > |x|$ are both possible

The General Two-State GNFA (not in text as such)



$$\underline{L_{ss}^{once}} = \alpha + \beta\gamma^*\eta$$

$$L_{ss} = \left(L_{ss}^{once}\right)^*$$ Always includes ε!

$$= (\alpha + \beta\gamma^*\eta)^*.$$

$$L_{sf} = L_{ss}\cdot\beta\gamma^*$$

$$= \underline{(\alpha + \beta\gamma^*\eta)^*\beta\gamma^*}.$$

If $F = \{f\}$, then $L(N) = L_{sf}$, else if $F = \{s, f\}$, $L(N) = L_{ss} \cup L_{sf}$.

Alternative Form "Centered" on f not s.

$$L_{ff} = (\gamma + \eta\alpha^*\beta)^*. \text{ Then}$$

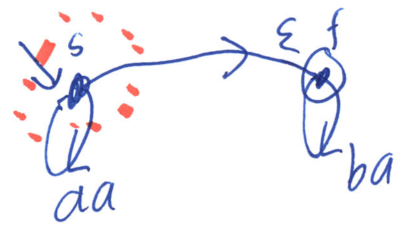
$$L_{sf} = \cancel{\alpha^*\beta^*\cdot L_{ff}}$$

$$\alpha^*\beta\cdot L_{ff} = \underline{\alpha^*\beta(\gamma + \eta\alpha^*\beta)^*}$$

Example where it simplifies further:

$$L_{sf} = (aa)^*\varepsilon\cdot(ba + \underbrace{\emptyset\cdot(aa)^*\cdot\varepsilon}_{\text{Kills it all}})^*$$

$$= (aa)^*(ba + \emptyset)^* = (aa)^*(ba)^*$$



$\alpha = aa$
$\beta = \varepsilon$
$\gamma = ba$
$\eta = \emptyset$

$$L_{sf} = (aa + \underbrace{\beta(ba)^*\emptyset}_{\text{Kills}})^*\varepsilon(ba)^*$$

$$= (aa + \emptyset)^*(ba)^*$$

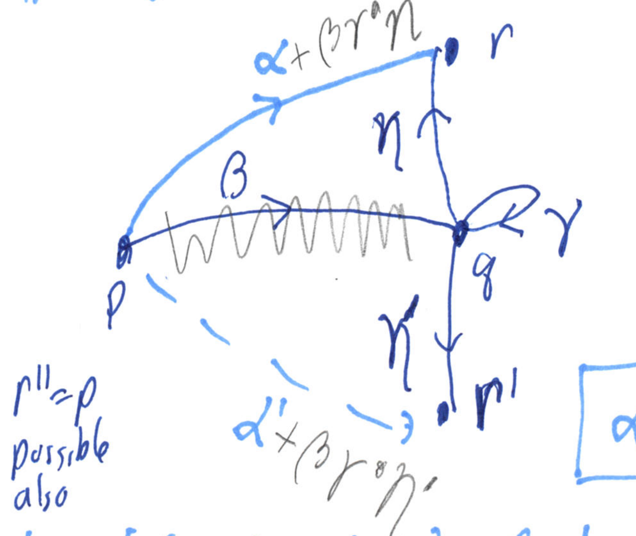
$$= (aa)^*(ba)^* \text{ again.}$$

Adding L_{ss} with $F = \{s, f\}$ does not change this particular language!

$L_{ss} = (aa)^*$ $\quad L_{ss} \cup L_{sf} = (aa)^* \cup (aa)^*(ba)^* = (aa)^*(ba)^*$ because ε matches $(ba)^*$

Idea of the Conversion from GNFA to Regular Exp:

If q is a state not in F and $q \neq s$, then we can eliminate q by bypassing all incoming arcs (p, β, q) to all outgoing (q, η, r).



Carry out the bypass by updating each (p, α, r) to

$$\boxed{\alpha + \beta\gamma^*\eta}$$

Once done for each outgoing r, r', ..., delete edge β. Once all incoming edges to q are deleted, eliminate q.

Gist of algorithm is to eliminate all nonaccepting states different from s.