CSE 396 Lecture Thu Feb. 18 Spr 2016 ①

Converting an NFA $N = (Q, \Sigma, \delta, s, F)$ into an equivalent DFA $M = (Q', \Sigma, \Delta, S, F')$: $Q' \subseteq \mathcal{P}(Q)$

$L(M) = L(N)$ — F' to be defined.

Economize by making $|Q'| \ll 2^{|Q|}$ if possible, by Breadth-First Search (BFS)

Main Idea: Upon reading a prefix w of the input $x \in \Sigma^*$:

The Set-State P that M is in	$=$	the set of states $p \in Q$ such that N can process w from s to p.

(IV) An "Induction Invariant".

$\therefore$ The starting Set-State S of M should equal $=$ the set of states p s.t. N can process ε from s to p. (Basis)

Example: $N =$

Whenever ①, then ②.

. Hence the set-states 3 1} and {1,3} will not occur.

By default, N can process ε from ① to ①, ie. s to s. But, N can also process ε from ① to ②.

$\therefore S = \{1,2\}$ And $S \in F'$.

Inductive Rule

For any set-state $P \subseteq Q$ and char $c \in \Sigma$:

$$\Delta(P, c) = \{r \in Q : \text{for some } p \in P,\ N \text{ can process } c \text{ from } p \text{ to } r\} \equiv R$$

$\boxed{\text{Think } w = v \cdot c}$ $= \bigcup_{p \in P} \{r \in Q : N \text{ can process } c \text{ from } p \text{ to } r\}$.

$\therefore F'$ should ~~be {r ∈ Q : N can process x from s to r}~~ $=_{def} \{R \subseteq Q : \text{some } q \in F \text{ belongs to } R\}$. ☒

Final Shortcut: Define $\underline{\delta}(p,c) = \{q : N$ can process c from p to q using $\underline{\text{trailing}}$ εs only.$\}$.

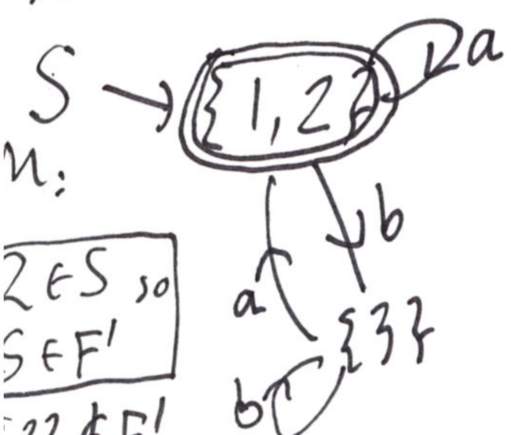
∴ If we initialize S correctly, then we will get

$$\Delta(P,c) = \bigcup_{p \in P} \underline{\delta}(p,c).$$ (Different from text)

$\underline{\delta}(1,a) = \emptyset$ $\quad\underline{\delta}(2,a) = \{1,\underline{2}\}$ by trailing ε $\quad\underline{\delta}(3,a) = \{1,2\}$

$\underline{\delta}(1,b) = \{3\}$ $\quad\underline{\delta}(2,b) = \{3\}$ $\quad\underline{\delta}(3,b) = \{3\}$.

M:



$\boxed{\varepsilon \in S \text{ so } S \in F'}$

$\{3\} \notin F'$

$F' = \{\{1,2\}\}$ only!

$\Delta(S,a) = \underline{\delta}(1,a) \cup \underline{\delta}(2,a) = \emptyset \cup \{1,2\} = \{1,2\}$.

$\Delta(S,b) = \underline{\delta}(1,b) \cup \underline{\delta}(2,b) = \{3\} \cup \{3\} = \{3\}$

$\Delta(\{3\},a) = \underline{\delta}(3,a) = \{1,2\}$ again.

$\Delta(\{3\},b) = \underline{\delta}(3,b) = \{3\}$ No more new states, so $\underline{\text{done}}$!

$\underline{\text{Intuition}}$: $L(M) = \{x \in \Sigma^* : x$ ends in an a or $x = \varepsilon\}$
$= \varepsilon \cup (a+b)^* a$.

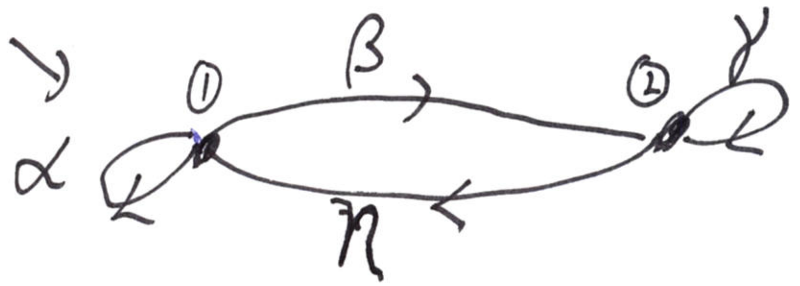
$\underline{\text{Def}^n}$: A $\underline{\text{generalized}}$ $\underline{\text{NFA}}$ (GNFA) differs from an NFA by allowing instructions (p, R, q) where R is any regular expression. A $\underline{\text{computation}}$ by ~~an~~ GNFA $G = (Q, \Sigma, \delta, s, F)$, $\delta \subseteq Q \times \underline{\underline{\text{Regexps}(\Sigma)}} \times Q$

is a sequence of instructions $(q_0, R_1, u_1, q_1)(q_1, R_2, u_2, q_2), \ldots, (q_{m-1}, R_m, u_m, q_m)$

Where for each i, $1 \le i \le m$, u_i $\underline{\text{matches}}$ R_i $\quad(u_i \in L(R_i))$
and $w = u_1 \cdot u_2 \cdots u_m$ is the string processed, $\underline{\text{and}}$ $\underline{(q_{i-1}, R_i, q_i) \in \delta}$.

Then we say G can process w from state q_0 to state q_m.
If $p = q_0$ and $r = q_m$, let $\underline{\underline{L_{pr}}}$ stand for the set of all such $w \in \Sigma^*$.
$L(G) = \bigcup_{f \in F} L_{sf} = \{x : G$ can process x from s to an accepting state.

The General Two-state GNFA. ③



α
β
γ
η <u>eta</u>

Now use for regular expressions

Claim: $L_{11} = (\underbrace{\alpha \cup \beta\gamma^*\eta}_{\text{one go-round from ① back to ①}})^*$

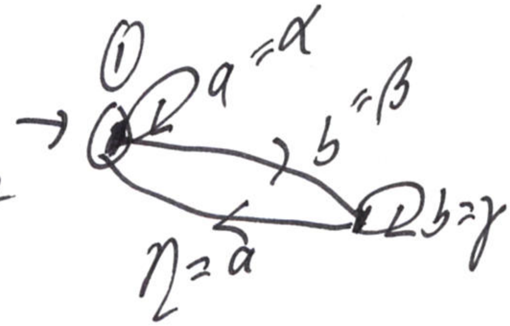
$L'_{11} = \alpha \cup \beta\gamma^*\eta$

$L_{11} = (L'_{11})^*$

$L_{12} = (\alpha \cup \beta\gamma^*\eta)^* \underbrace{\beta\gamma^*}_{\text{last time down the track w/o coming back to start.}}$ $= L_{11} \cdot \beta\gamma^*$

also $= \underbrace{\alpha^*\beta}_{\text{first time to ②}}(\gamma + \eta\alpha^*\beta)^*$

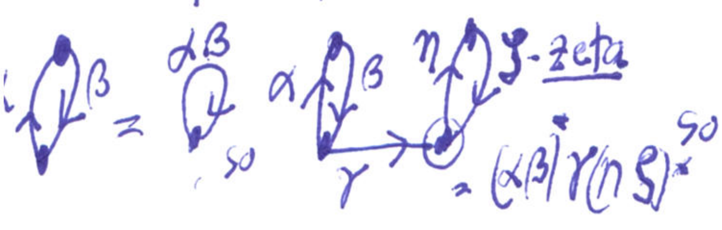
all ways to come back to ②.

$M =$ 

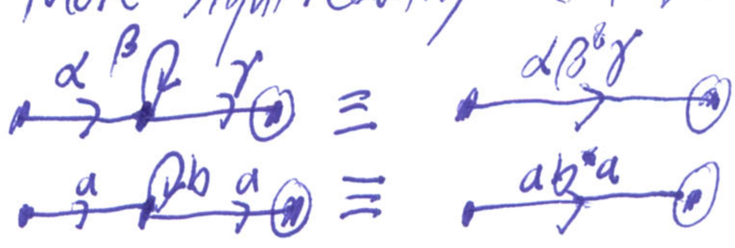
$L(M) = L_{11} = (a \cup bb^*a)^*$

$\stackrel{?}{=} (a+b)^*a \cup \varepsilon$ <u>Yes</u>, but... how?

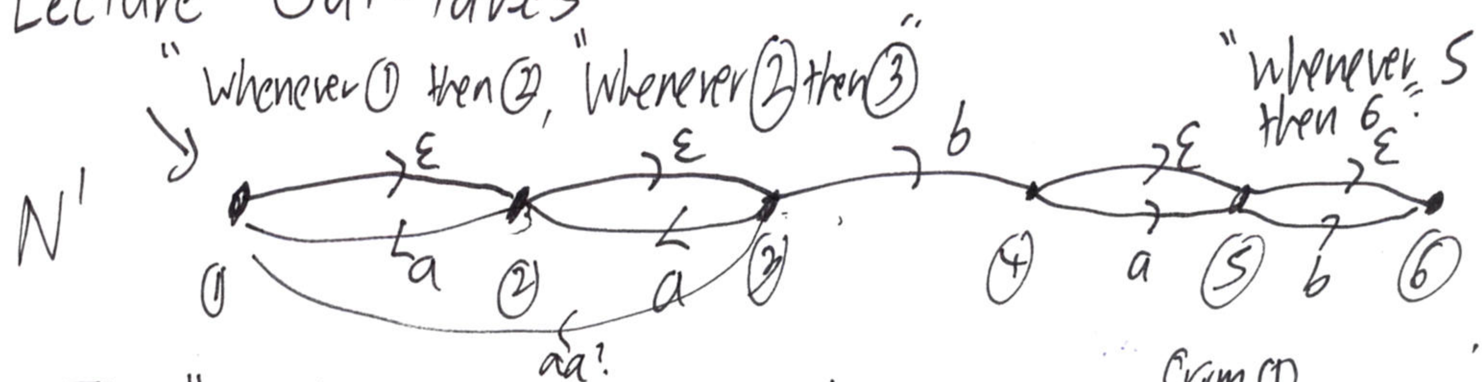
Extra Note:
Note that in this NFA → DFA conversion we did <u>not</u> get a dead state. Not every DFA has or needs one.



More "sightreading" examples:



Lecture Out-Takes



The "Epsilon Closure" $E(\{1\})$ equals $\{1,2,3\}$.

because N' can process ε from 1 to any of 1, 2, 3.

N' can process "b" from 1 to any of 4, 5, or 6.

$(3, b, 4) \in \delta$ but $\Delta(\{3\}, b) =$ the epsilon closure $E(\{4\}) = \{4,5,6\}$.

$\Delta(S, b) = \Delta(\{1,2,3\}, b) = \{4,5,6\}$.

$\Delta(S, a) \overset{?}{=} \Delta(\{1,2,3\}, a) \overset{?}{=} \{1,2,3\}$. $\Delta(\{4\}, a) = \{5,6\}$ but not 4.

$\underline{\delta}(2, b) = \emptyset$ $\underline{\delta}(3, b) = \{4,5,6\}$ using trailing εs.

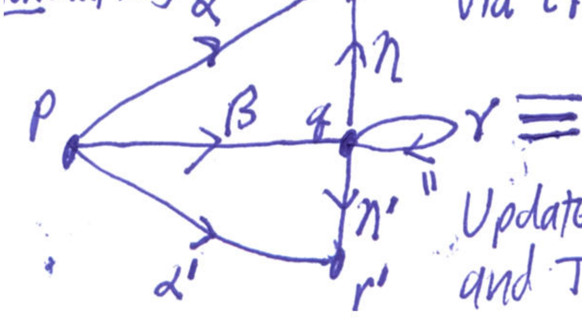
Also $\underline{\delta}(1, b) = \emptyset$ discipline.

OK since we start with $S = \{1,2,3\}$ and

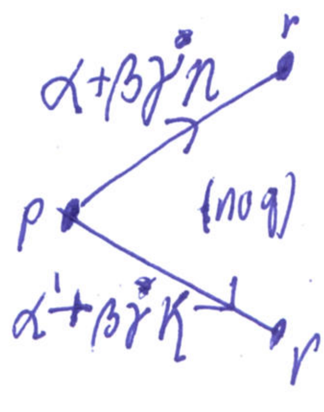
$$\underline{\delta}(S, b) = \underline{\delta}(1,b) \cup \underline{\delta}(2,b) \cup \underline{\delta}(3,b) = \bigcup_{p \in S} \underline{\delta}(p, b) = \emptyset \cup \emptyset \cup \{4,5,6\} = \{4,5,6\}$$ like we want.

Preview of Tue 2/23 Lecture:

Goal: Eliminate every nonaccepting state q: "Bypass (p,q) via (q,r), (q,r')."



"Update $T(p,r)$ and $T(p,r')$"



When all arcs into q are bypassed, then we can delete state q. Since $q \notin F$, no ability to process is altered.