CSE 396 Lecture 2/23/17 Spr 2017 (1)

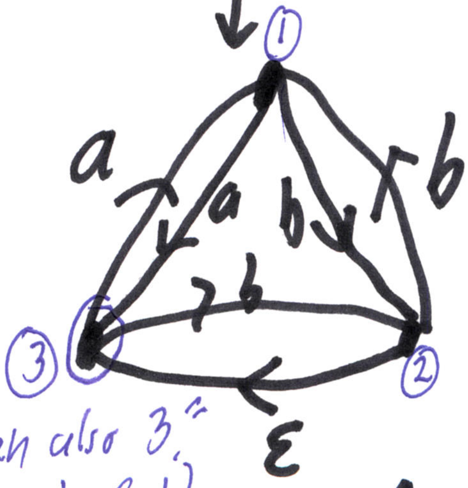
The quickest way I Know for NFA → DFA:

① Define $\underline{\delta}(p,c) = \{q :$ you can get to q by processing c <u>first</u> then any ϵ-arcs$\}$.

② $\Delta(P,c) = \bigcup_{p \in P} \underline{\delta}(p,c)$ Now the $E(\cdots)$ closure is <u>inside</u>.

S, F same as before for DFA. $\boxed{\text{Use Breadth First Search}}$

Example:

N = 

$F = \{3\}$.

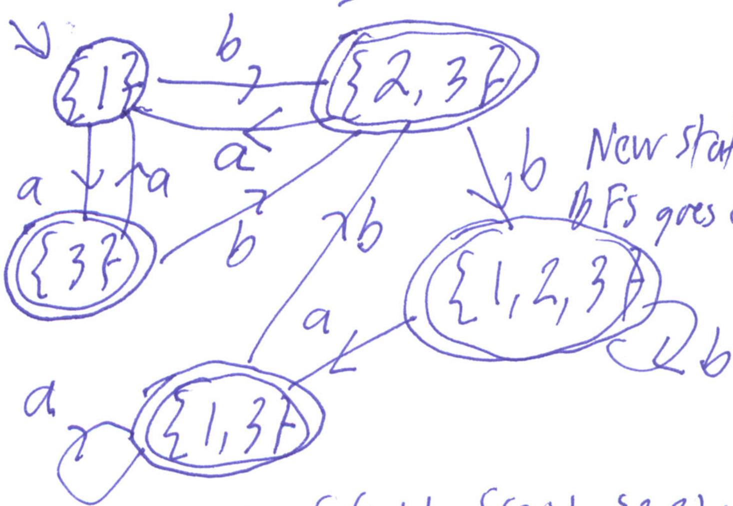
Whenever 2, then also 3.

$S = \{1\}$ (no ϵ out of 1)

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

$\underline{\delta}(2,a) = \emptyset$ (!) $\underline{\delta}(2,b) = \{1\}$

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

DFA M = 

New state BFS goes on.

$\Delta(\{2,3\},a) = \underline{\delta}(2,a) \cup \underline{\delta}(3,a)$
$= \emptyset \cup \{1\} = \{1\}$

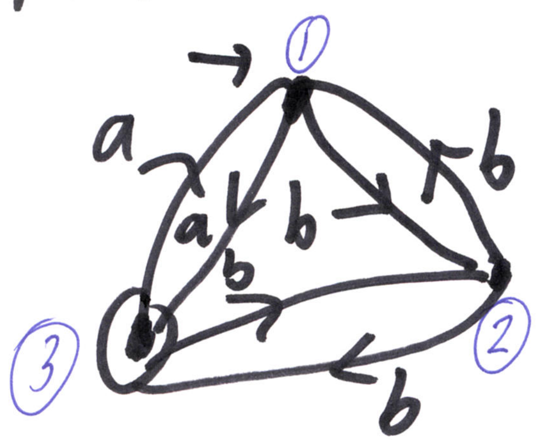
$\Delta(\{2,3\},b) = \underline{\delta}(2,b) \cup \underline{\delta}(3,b)$
$= \{1\} \cup \{2,3\} = \{1,2,3\}$

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

Let's trace the original NFA on input

$$x = \overset{S=\{1\}}{\ } \; b \; \overset{\{2,3\}}{\ } \; b \; \overset{\{1,2,3\}}{\ } \; a \; \overset{\{1,3\}}{\ } \; b \; \overset{\{2,3\}}{\ } \; a \; \overset{\{1\}}{\ } \notin L(N).$$

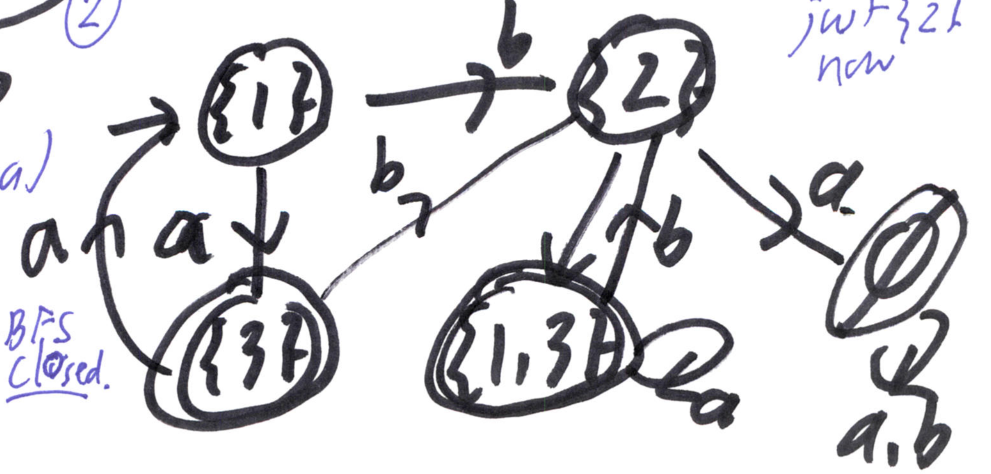
M and L(N) have no "nirvana" condition.
Do they have a "dead" condition? <u>No</u>: No $\emptyset$ state in M.

N' = 

$\delta(1,a)=\{3\}$ $\delta(1,b)=\{2\}$
$\delta(2,a)=\cancel{\emptyset}$ $\delta(2,b)=\{1,3\}$
$\delta(3,a)=\{1\}$ $\delta(3,b)=\{2\}$ (no 3 anymore)

just {2} now

$\Delta(\{1,3\},a) = \delta(1,a) \cup \delta(3,a)$
$= \{3\} \cup \{1\} = \{1,3\}$
$\Delta(\{1,3\},b) = \{2\} \cup \{2\} = \{2\}$. BFS closed.



A <u>GNFA</u> can't convert to DFA or even NFA directly because it can "jump" 2 or more chars in a step.

Helpful? intuition: An NFA has arcs labeled
- c where $c \in \Sigma$
- ϵ
- or "$\emptyset$" for arcs not there.

} It has <u>basic</u> regexps as entries in a table $T[p,q]$. GNFA liberalizes to all regexps.

Def^n: <u>Regexp(Σ)</u> denotes the set of all legal regular expressions r over Σ. ③

"generalized"

A <u>GNFA</u> is a 5-tuple $G = (Q, \Sigma, \delta, s, F)$ where Q, s, F are as with NFAs, but

$$\delta \subseteq (Q \times \underline{\text{Regexp}(\Sigma)} \times Q).$$

Def^n: A GNFA can "process" a string x from state p to state q if there is a sequence

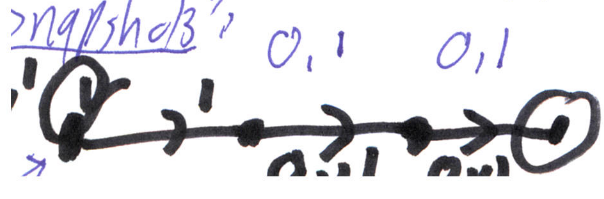
$$(q_0, w_1, q_1, w_2, q_2, \cdots, q_{m-1}, w_m, q_m)$$

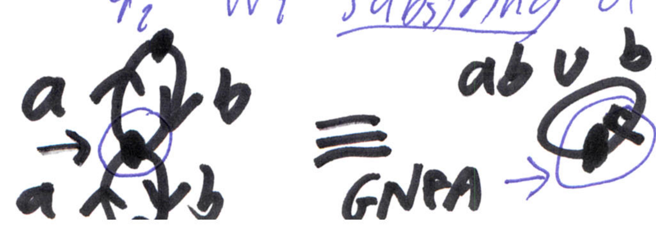
s.t. $q_0 = p$, $q_m = q$, $w_1 \cdots w_m = x$, and for $1 \le i \le m$

$$\boxed{\begin{array}{l}\text{there is an instruction } (q_{i-1}, r, q_i) \\ \text{where } r \in \text{Regexp}(\Sigma) \text{ and } w_i \in L(r).\end{array}}$$

There is an arc $q_{i-1} \xrightarrow{r} q_i$ such that r <u>matches</u> the w_i <u>substring</u> of x.

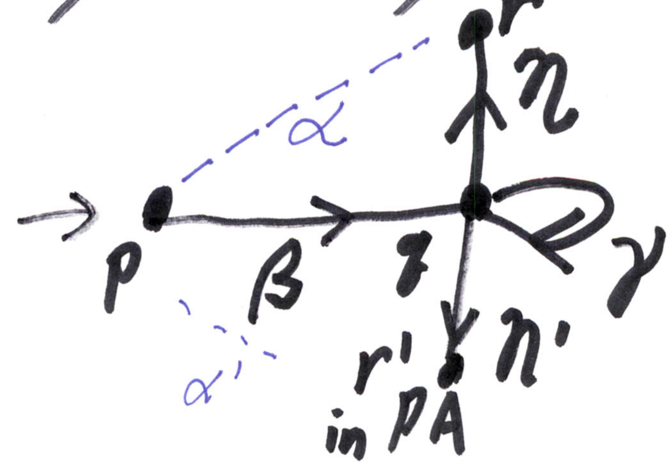
"<u>Snapshots</u>"



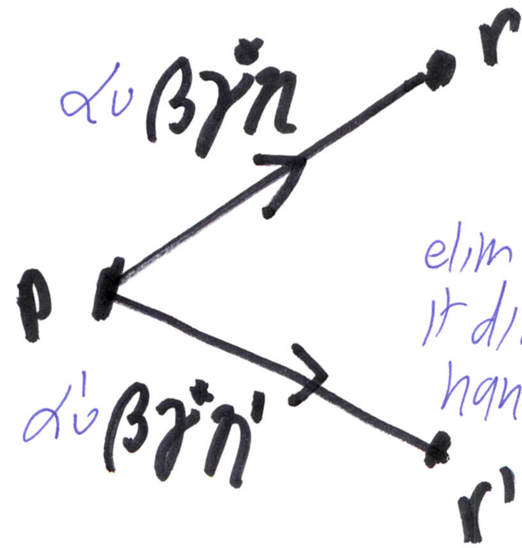


"Incoming + Outgoing Tracking Idea" to

Eliminate a Nonaccepting State q of a GNFA.



Proof builds a bypass highway



added

Strategy: DON'T do text step with new s or f - yet

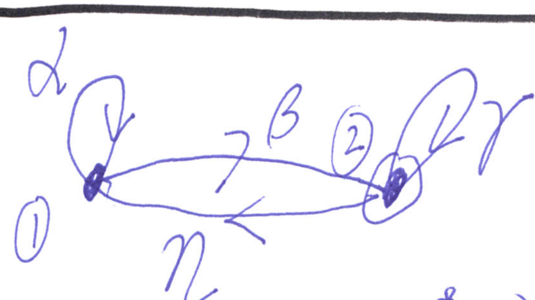
1) Eliminate all $q \notin F$, $q \neq s$ by bypassing all incoming edges (p, β, q) as diagrammed, then simply deleting q.

elim q. Any processing it did has been handled by "new road".

Added → Tue 2/28

2) If you have at most 2 states left, read off answer(s) at right:

Else, do trick of adding new f with ε arcs to it from all acc states. Then eliminate all other $q \neq s$. You will get case at right with $\gamma, \eta = \emptyset$.



$$L_{12} = (\alpha + \beta\gamma^*\eta)^*\beta\gamma^*$$

$$\text{OR} = \alpha^*\beta(\gamma + \eta\alpha^*\beta)^*$$

If state 1 is also accepting then add $L_{11} = (\alpha + \beta\gamma^*\eta)^*$.

Theorem: For every GNFA G, we can build a regexp r s.t. $L(r) = L(G)$.