5-1/ NFA - non deterministic finite automata

$\circ$ - catenation

$DFA = (Q, \Sigma, q_0 \in Q, \delta : Q \times \Sigma \to Q, F \subseteq Q)$

$NFA = \qquad \delta' : Q \times \text{maybe } \Sigma \to P(Q)$

rule 1: one character may go to multiple next states

DFAs

$\bigcirc \xrightarrow{c} \bigcirc$

NFAs

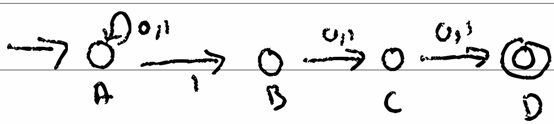
$$\begin{tikzcd} & \bigcirc \\ \bigcirc \arrow[ur, "c"] \arrow[dr, "c"'] & \\ & \bigcirc \end{tikzcd}$$

rule 2: we may move w/o seeing a character

$\bigcirc \xrightarrow{\varepsilon} \bigcirc$

S-2/ the third character from end is 1

11100 00100 0100 100 111



[A]0100 → [A]100 → [B]00 → [C]0 → [D]ε → ✓

→ [A]00 → [A]0 → [A]ε → ✗

→ ~~[B]00~~

~~convert DFA into an NFA~~

5-3/ NFA semantics

$x \in nfa\ (Q, \Sigma, q_0 \in Q, \delta : Q \times M(\Sigma) \to P(Q),$
$F \subseteq Q)$

iff $[q_0]\, x \to\to\to\to [q_f]\, \varepsilon$

$[q_i]\, w \to [q_j]\, w$ iff $\delta(q_i, \varepsilon) \ni q_j$

$[q_i]\, cw \to [q_j]\, w$ iff $\delta(q_i, c) \ni q_j$

convert a dfa into an nfa

convert : $(Q, \Sigma, q_0 \in Q, \delta : Q \times \Sigma \to Q, F \subseteq Q)$

$\to (Q', \Sigma, q_0' \in Q, \delta' : Q' \times M(\Sigma) \to P(Q'), F' \subseteq Q)$

$Q' = Q \quad q_0' = q_0 \quad F' = F$

$\delta'(q_i, c) = \{ \delta(q_i, c) \} \quad \delta'(q_i, \varepsilon) = \emptyset$

5-4/ oracle : NFA × string × trace (list(config))
$\rightarrow$ bool

is a valid trace

oracle n w ts = h n q_0 w ts

h n q_i w [] = w is empty?

h n q_i w (q_j, w') :: ts' =

if w = w' then $q_j \in \delta(q_i, \varepsilon)$ and
h n q_j w' ts'

o.w. w = cw' then $q_j \in \delta(q_i, c)$ and
h n q_j w' ts'

o.w. false

5-5/ trace tree $\overset{TT}{=}$ Y | N | Branch State List $(\mathcal{M}(\varepsilon), TT)$

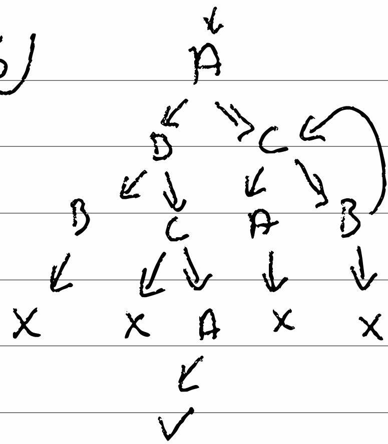
explore : NFA × String → TT

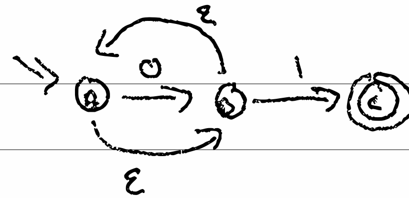
explore n w = h n q_0 w

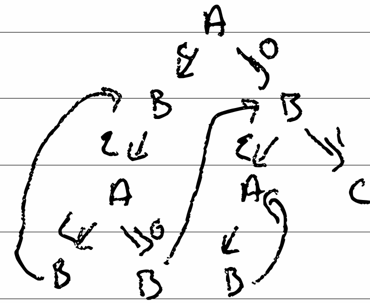
h n q_i ε = if $q_i \in F$, then Y
o.w. N

h n q_i (c w) = Branch q_i opts
where opts = (ε, h n q_j cw)
$\forall q_j \in \delta(q_i, \varepsilon)$
++ (c, h n q_j , w)
$\forall q_j \in \delta(q_i, c)$

5-6)







5-7/ accepts : $NFA \times String \to Bool$

accepts $(Q, \Sigma, q_0, \delta, F)$ w =

$V = \{\}$ $\quad P = \{(q_0, w)\}$

while $(P \neq \text{empty})$ {

let $(q_i, w) \leftarrow$ remove from P

if $w = \varepsilon$ and $q_i \in F$, return true

for $q_j \in \delta(q_i, \varepsilon)$

if $(q_j, w) \notin V$, $P = P \cup \{(q_j, w)\}$

$V = V \cup \{(q_j, w)\}$

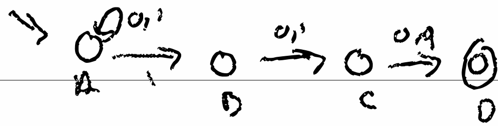
if $w = cw'$, then for $q_j \in \delta(q_i, c)$

if $(q_j, w') \notin V$, $P = P \cup \{(q_j, w')\}$

$V = V \cup \{(q_j, w')\}$

return false

5-8



0100

$V = \{\}$ $P = \{(A, 0100)\}$

$V = \{(A, 100)\}$ $P = \{(A, 100)\}$

$V = \{(A, 100)\ (A, 00)\ (B, 00)\}$ $P = \{(A, 00)\ (B, 00)\}$

$V = \{\ldots, (A, 0)\}$ $P = \{(B, 00)\ (A, 0)\}$

$V = \{\ldots, (C, 0)\}$ $P = \{(A, 0)\ (C, 0)\}$

$V = \{\ldots, (A, \varepsilon)\}$ $P = \{(C, 0)\ (A, \varepsilon)\}$

$V = \{\ldots, (D, \varepsilon)\}$ $P = \{(A, \varepsilon)\ (D, \varepsilon)\}$

$V =$ $P = \{D, \varepsilon)$

✓

$[A]0100 \to [A]100 \to [A]00 \to [A]0 \to [A]\varepsilon$ ~~ㅋ~~

$\hookrightarrow [B]00 \to [C]0 \to [D]\varepsilon$ ㅋ✓