CSE 396 Lecture 2/16 Spr 2017 (1)

**Def$^n$:** An NFA $N = (Q, \Sigma, \delta, s, F)$ **can process** a string $x \in \Sigma^*$ **from** state $p$ **to** state $q$ if there is a {trace / **computation**}

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

such that $q_0 = p$, $w_1 \cdots w_m = x$, $q_m = q$, and for each $i$, $1 \le i \le m$, $(q_{i-1}, w_i, q_i) \in \delta$.

- Works as-is for a DFA $M$ in place of an NFA $N$, and also for "strict NFAs" (no $\epsilon$).
- If no $\epsilon$-arcs, then each $w_i$ is a char and so $m = n =_{\text{def}} |x|$.

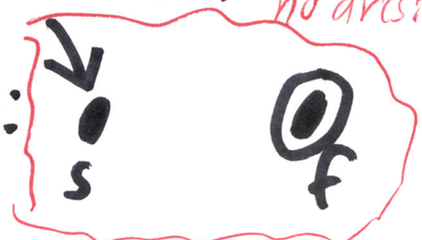
**Theorem:** For every NFA $N$ there is a DFA $M$ such that $L(M) = L(N)$. Proof → Tue. Idea previewed last lecture with the "subset trace"

# Formal Def$^n$ of Regular Expressions And "Their" NFAs. ②

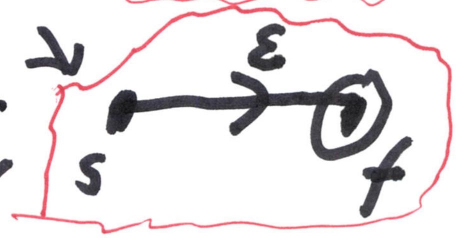
Definition By Induction.

**Basis:** | Language | NFA

$\underset{\sim}{\emptyset}$ is a regexp $\quad L(\underset{\sim}{\emptyset}) = \emptyset \quad N_{\underset{\sim}{\emptyset}}$:




$\underset{\sim}{\varepsilon}$ is a regexp $\quad L(\underset{\sim}{\varepsilon}) = \{\varepsilon\} \quad N_{\underset{\sim}{\varepsilon}}$

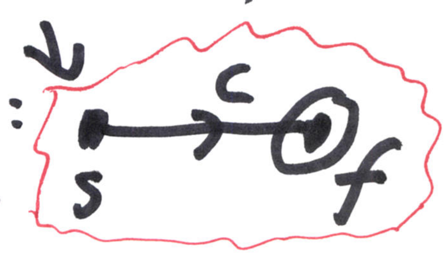



$N_{\emptyset}$ cannot process any strings from $s$ to $f$.

$N_{\underset{\sim}{\varepsilon}}$ can process $\varepsilon$ from $s$ to $f$: $(s, \varepsilon, f)$ This is $\in \delta$, $m=1$ step

Neither one can process any char(s) from $s$ to $f$. For any $c \in \Sigma$:

$\underset{\sim}{c}$ is a regexp, $L(\underset{\sim}{c}) = \{c\}$, $N_{\underset{\sim}{c}}$:



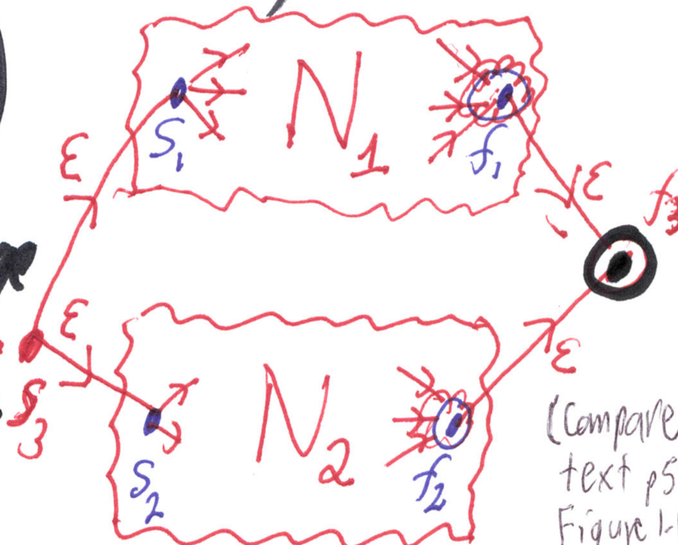

**Def$^n$:** $\underline{L(N)} =_{\text{def}} \{x \in \Sigma^* :$ for some $f \in F$, $N$ can process $x$ from $s$ to $f\}$.

$L(N_{\underset{\sim}{\emptyset}}) = \emptyset$, $L(N_{\underset{\sim}{\varepsilon}}) = \{\varepsilon\}$, and $L(N_{\underset{\sim}{c}}) = \{c\}$.

<u>Induction</u>: <u>Let any</u> regexps $r_1$ and $r_2$ along with "<u>their</u>" NFAs $N_1$ and $N_2$ <u>be given</u>.

"Their" means $L(N_1)=L(r_1)$, $L(N_2)=L(r_2)$ <u>and</u> each has a unique accepting state $f \neq s$.

<u>Then</u> $r_3 =_{def} (r_1 \cup r_2)$ is a regexp, with language $L(r_3)=L(r_1)\cup L(r_2)$ and its NFA is $N_3=$




(Compare text p59, Figure 1.46)

<u>Construction</u>: $N_3=(Q_3,\Sigma,\delta_3,s_3,F_3)$ where:

$Q_3 = Q_1 \cup Q_2 \cup \underbrace{\{s_3,f_3\}}_{\text{new states}}$ $\quad s_3$ is the new start state, $F_3=\{f_3\}$ only.

$\delta_3 = \delta_1\cup\delta_2\cup\{(s_3,\varepsilon,s_1),(s_3,\varepsilon,s_2),(f_1,\varepsilon,f_3),(f_2,\varepsilon,f_3)\}$.

<u>Verification</u>: We need $L(N_3)=L(r_3)$, using $L(r_3)=_{def} L(r_1)\cup L(r_2)$.

<u>By induction hypothesis</u>, $L(r_1)=L(N_1)$ and $L(r_2)=L(N_2)$.

So we need to show $L(N_3)=L(N_1)\cup L(N_2)$. We do by showing both

<u>$L(N_3)\subseteq L(N_1)\cup L(N_2)$, and $L(N_1)\cup L(N_2)\subseteq L(N_3)$.</u> ← more clear.

Any x processed from $s_3$ to $f_3$ must have been done by $N_1$ or by $N_2$. $\boxtimes$

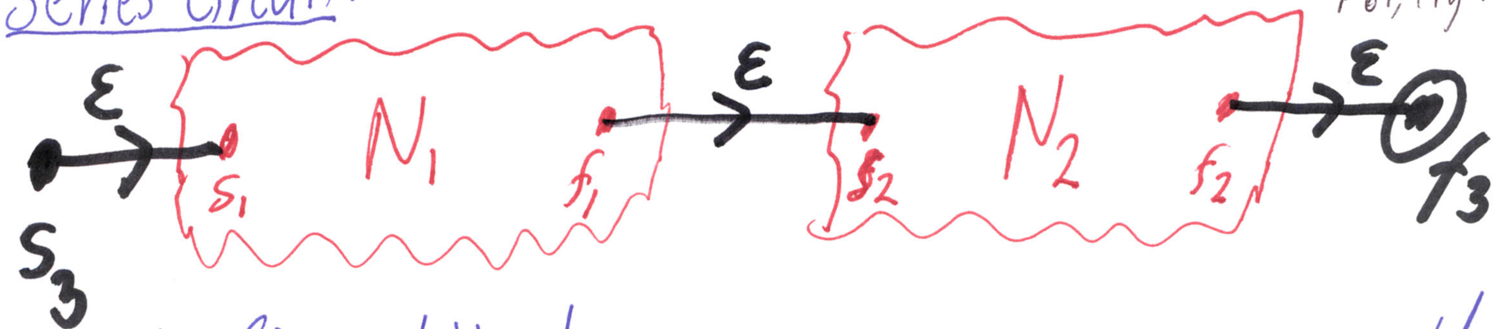
**Next Case** • [Let $r_1, r_2, N_1, N_2$ be given...] (4)

• $r_3 = (r_1 \bullet r_2)$ is a regexp.

$L(r_3) =_{def} L(r_1) \bullet L(r_2)$, and the NFA $N_3$ is:

[Compare text p61, Fig 1.48]

Series circuit.




The first and third ε arcs are unnecessary: we could define $s_3 = s_1$ and $f_3 = f_2$ and inductively abide by the rules.
The middle ε is needed: "fusing" $s_2 = f_1$ can cause ~~back~~ ...

$X \in L(r_1) \cdot L(r_2)$ $\equiv$ $X$ can be broken as $X =: Y \cdot Z$

ind hyp. $\rightarrow$ $L(N_1) \cdot L(N_2)$ such that $Y \in L(N_1)$ and $Z \in L(N_2)$.

(note: either $Y$ or $Z$ could be ε.)

We want this to be equivalent to $X \in L(N_3)$.

$L(N_1) \bullet L(N_2) \subseteq L(N_3)$ is clear.

$L(N_3) \subseteq L(N_1) \cdot L(N_2)$: the only way $N_3$ can process $X$ is for $N_1$ to process an initial part $Y$ and $N_2$ does the rest.

$\therefore L(N_3) = L(N_1) \cdot L(N_2)$.

If we allowed $s_2 = f_1$ and $f_1$ had back-arcs, then we could get "other stuff" in $L(N_3)$.

**Last Case:** [We only need $r_1$ and $N_1$:] (5)

$r_3 = (r_1{}^*)$ is a regexp.

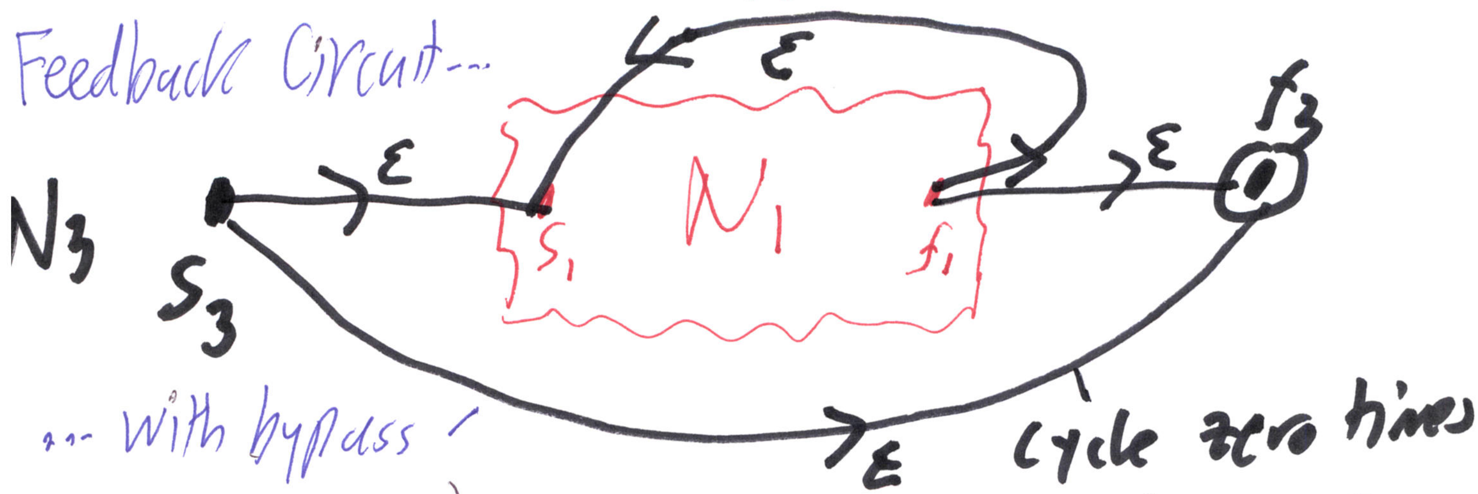
$A^* =_{def} \bigcup_{i=0}^{\infty} A^i$

$= \{\varepsilon\} \cup A \cup A \cdot A \cup A \cdot A \cdot A \cup A \cdot A \cdot A \cdot A$ etc.

$L(r_3) =_{def} = L(r_1)^* =_{def}$

$\{\varepsilon\} \cup L(r_1) \cup L(r_1) \cdot L(r_1) \cup \cdots$

$L(r_3) = \{x \in \Sigma^*:$ x can be broken into zero or more parts $y_1 \cdots y_m$ such that each $y_i$ (if any) belongs to $L(r_1)\}$




ompare text's figure 1.50 n page 62 — rather than se the bypass arc, the text nakes the start state accept too.

$L(N_3) = L(N_1)^*$. $\boxtimes$

**Added:** To verify $L(N_3) = L(N_1)^*$, enough to see how t cycles $N_1$ zero or more times. Analogy to a for-loop `for (int i=0; i<n; i++){...}` which "falls thru" if n=0.