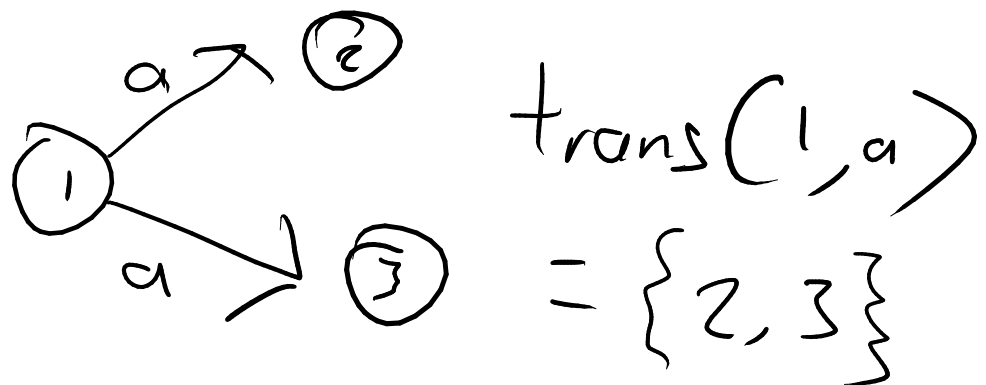


# ECE 467

- set of states  $S$
- $\Sigma$  an alphabet  $\epsilon \in \Sigma$
- start state
- $\geq 1$  final states
- transition function  
 $\text{trans}(s, c) \rightarrow \{\text{next states}\}$



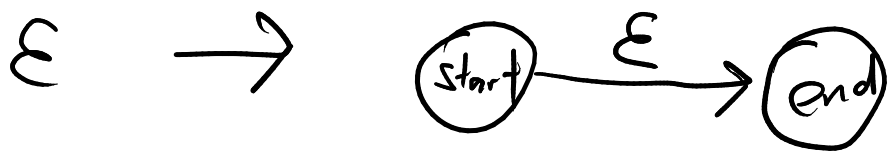
# Constructing NFA from regex

① break down  $r$  recursively

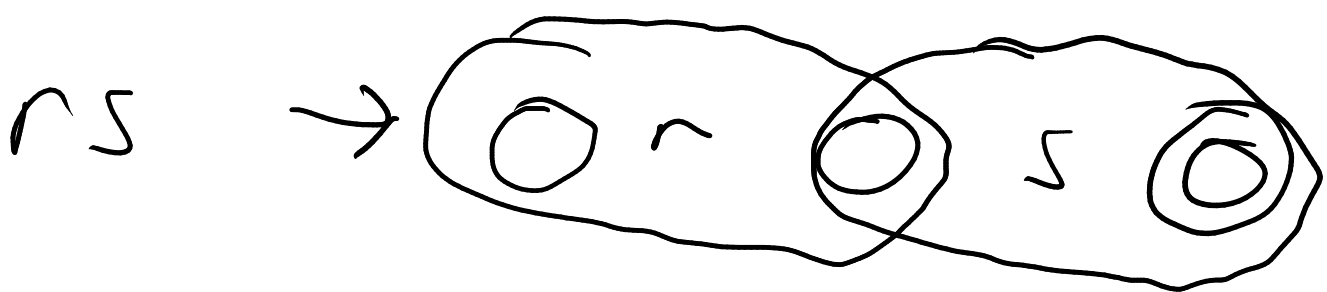
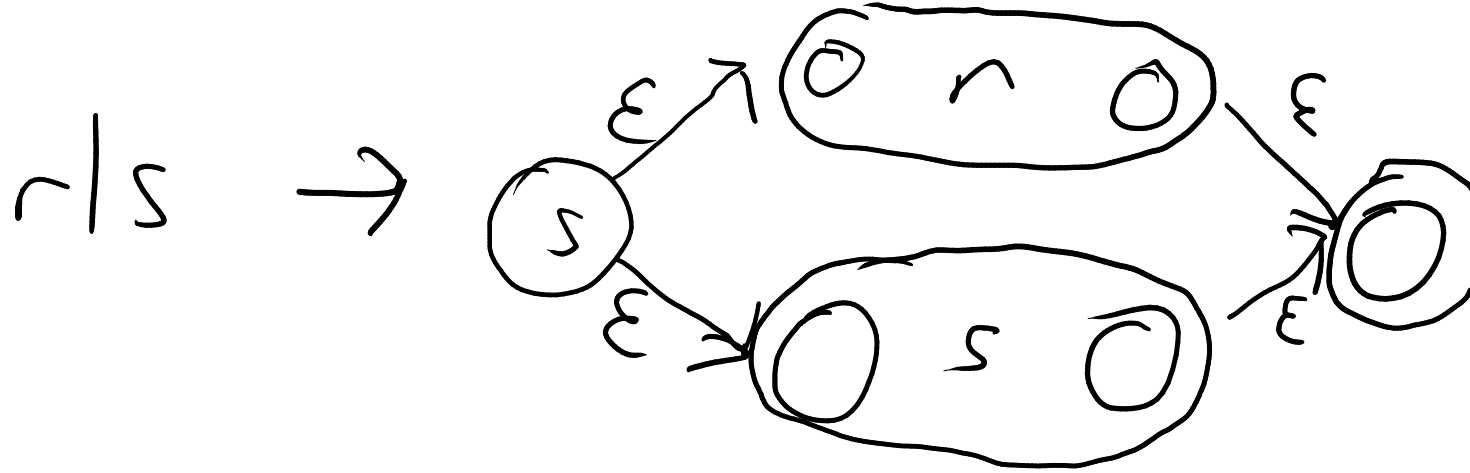
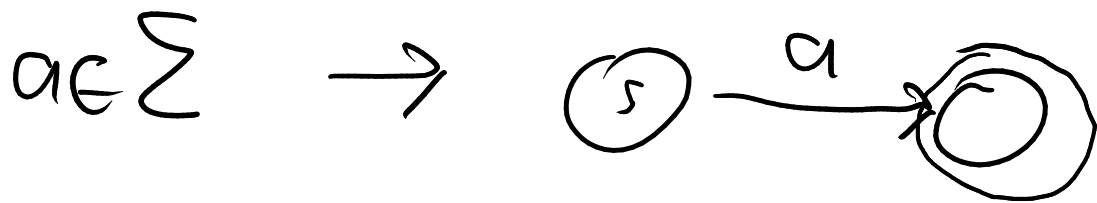
$$r, s \rightarrow rls$$

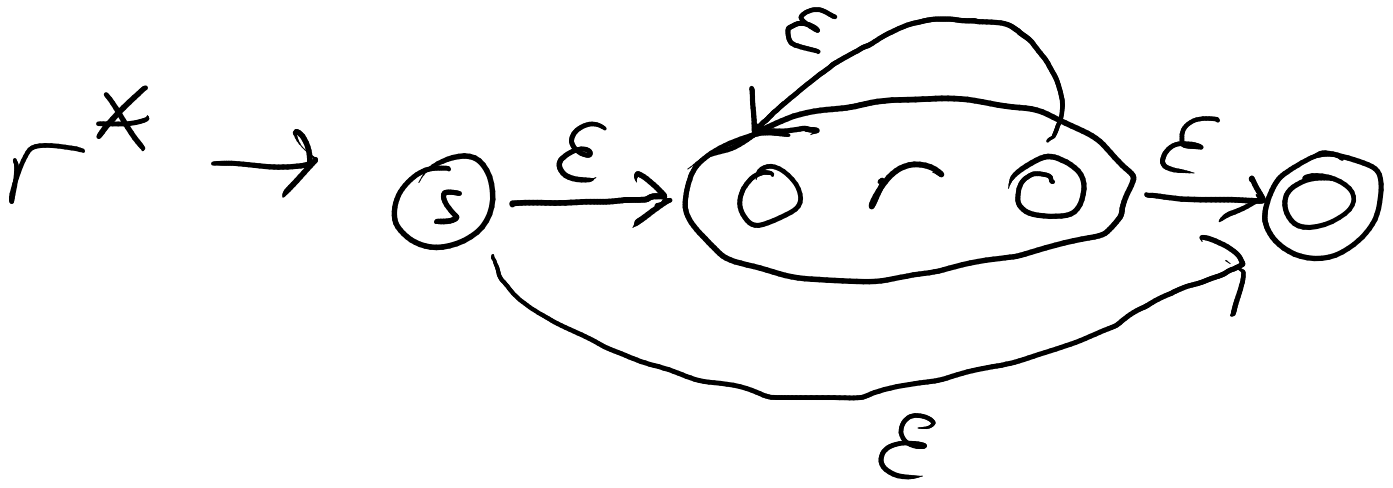
$a, b, c \in \Sigma$      $\epsilon$  (empty string)

② regex                      NFA

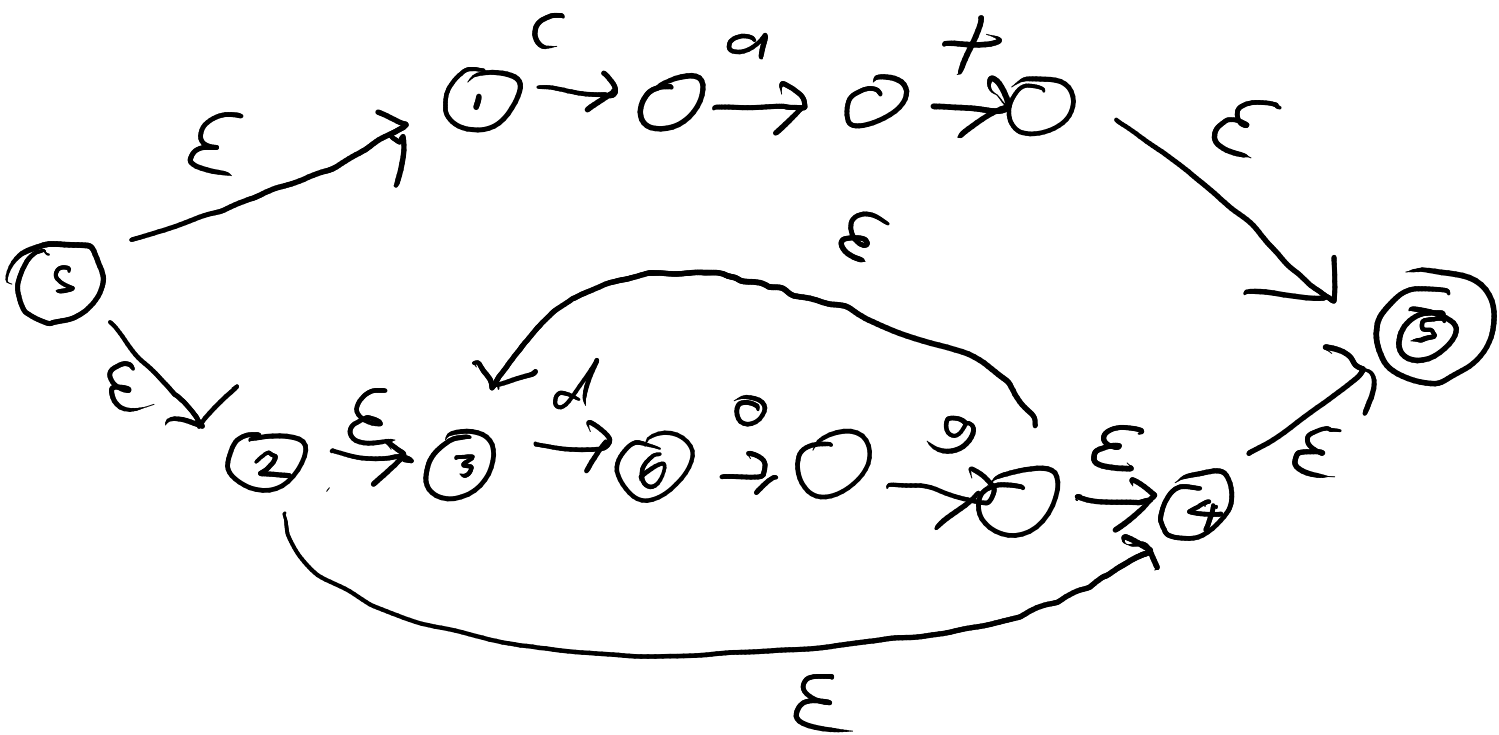


$$\text{trans}(\text{start}, a) = \{\}$$





$\underbrace{\text{cat}} \mid \underbrace{(\text{dog})^*}$



$$\epsilon\text{-closure}(\{s\}) = \{1, 2, 3, 4, 5\}$$

$$\text{move}(\{1, 3\}, d) = \{6\}$$

$\epsilon$ -closure( $S$ )  $\rightarrow$  { states }

$S$  + all states reachable via  $\epsilon$

while changed {

for each  $s \in S$  {

$S = S \cup \text{trans}(s, \epsilon)$

} // update changed

}  
}

move( $S, c$ )  $\rightarrow$  { states }

$\bigcup_{s \in S} \text{trans}(s, c)$

$O(m)$

$S = \{ \text{start state} \}$

loop {

$c = \text{nextchar}();$

if  $c == \text{EOF}$  {

break;

}

$S = \text{move}(\underbrace{\epsilon\text{-closure}(S)}_{O(m)}, c)$

}

loop executes  $\leq k = |\text{input string}|$

inside executes  $O(m)$  where  $m := |\text{transitions}|$

algorithm  $O(km)$

$m \leq 4|r| = O(|r|)$

$O(k|r|)$