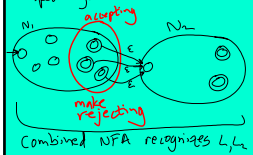
Regexes ⇄ Regular lang's

Thm: (1) Every regex denotes a regular language. (2) Every regular language is denoted by some regex.

Proofs by construction.

Lemma: If L_1 & L_2 are regular languages (over Σ), then $L_1 L_2$ $(:= \{xy : x \in L_1 \text{ & } y \in L_2\})$ is regular.

Proof: Let N_1 & N_2 be NFAs recognizing L_1 & L_2, respectively.



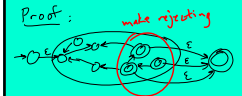
Combined NFA recognizes $L_1 L_2$

Def: Let A be an NFA. Say that A is clean if

1) A has exactly one accepting state, and it is different from the start state
2) No transitions into the start state
3) No transitions out of the accepting state.

Prop: Every NFA is equivalent to a clean NFA.

Proof:



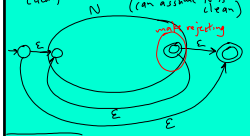
Lemma: If $L \subseteq \Sigma^*$ is regular, then L^* is regular.

Recall: $L^* := \{\varepsilon\} \cup L \cup LL \cup \cdots \cup L^n \cup \cdots$
(L^0, L^1, L^2)

Kleene *-operator
Kleene closure

Proof: Given an NFA N recognizing L, build an NFA N' recognizing L^* as follows:

(N' clean) (can assume N is clean)



Regex syntax & semantics review

Fix an alphabet Σ. A regex over Σ is an expression built inductively as follows:

Atomic regexes:

$\emptyset$ stands for the empty language

(for all $a \in \Sigma$) a stands for $\{a\}$

Nonatomic regexes:
Let s and t be regexes over Σ. Then

$s \cup t$ (union)
st (concat)
s^* (Kleene *)

Can use parens for grouping
Can drop parens abiding these precedence rules:

binary infix: $\cup$ — lowest precedence (associative); concat — next higher precedence
unary postfix: * — highest precedence except for (···)

Ex: $a^* = \{\varepsilon, a, aa, aaa, \ldots, a^n, \ldots\}$

$(a \cup b)(a \cup b) = \{aa, ab, ba, bb\}$

$a^* \cup b^* = \cdots$

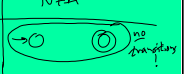
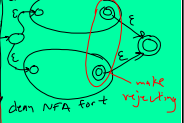
$ab^* = \{a, ab, abb, abbb, \ldots\}$

$(ab)^* = \{\varepsilon, ab, abab, ababab, \ldots\}$

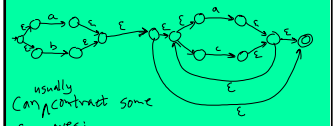
$(a \cup b)^* = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, \cdots\}$
all strings over $\{a, b\}$

Proof of (1) of the Theorem:

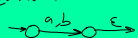
By induction on the (clean) syntax of a regex, build an NFA recognizing that language.

regex r	clean NFA
$\emptyset$	
(for any $a \in \Sigma$) a	
$s \cup t$ (s, t regexes)	
st	(by the construction for the 1st lemma)
s^*	(by the construction for the 2nd lemma)

Ex: NFA for $(a \cup b)(a \cup c)^*$:



Can usually contract some ε-moves:

 etc.

Proof of part (2) of theorem:

We start with an arbitrary reg lang L, recognized by some clean NFA.

Def: A generalized NFA (GNFA) is a tuple

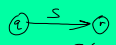
$\langle Q, \Sigma, \delta, q_0, F \rangle$

where, Q, Σ, q_0, F are as with an NFA and

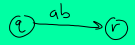
$\delta : Q \times Q \longrightarrow REX_\Sigma$

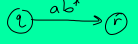
set of regexes over Σ

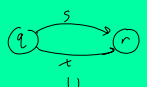
Transition diagram:

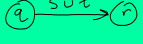
 means $\delta(q, r) = s$ (s is a regex)

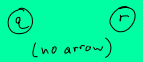
Ex:

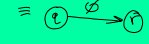
 can get from q to r by reading ab on input







 (no arrow)

$\equiv$ 

NFA $\xrightarrow{\text{trivially converts}}$ equivalent GNFA