COMS 3261 – Lecture 3, Part 1:

- Closure of the regular languages under regular operations
- Regular Expressions.

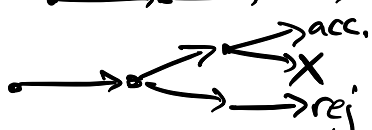
(Tuesday)
Announcements: HW 1 due 7/6/21 @ 11:59 PM EST
HW 2 (short homework) due 7/12/21 @ 11:59 EST.
(Monday)

Readings: Sipser, end 1.2 and 1.3

Today:

1. Review
2. Regular languages closed under $\cup$, $\circ$, $*$
3. Regular expressions
4. Regular expressions describe regular languages
(5. Regular languages can all be described by regular expressions.)

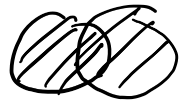
1. Review

- CS Theory $\approx$ formal science on computation
- Languages = sets of strings $\approx$ mathematical 'concepts'
- Automata — read strings and accept or reject
  "recognizing languages"
  - ↳ DFAs 
  - ↳ NFAs 

✓
- Regular languages — those recognized by DFAs (also NFAs)
- Proof structure:
  Wanted to show 'if this object(s) exist,
  [illegible] exists.'

→ that

- Regular operations.

$A \cup B$: 

$A \circ B := \{xy, x \in A, y \in B\}$

$A^* := \{x_1 \dots x_k \mid k \geq 0, x_i \in A\}$

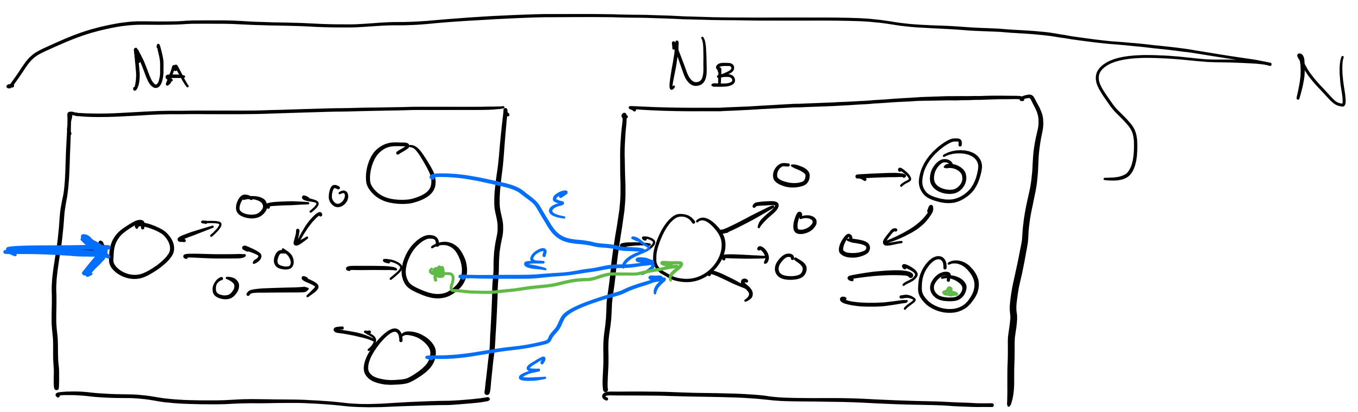
2. Closure of Reg. Languages under Reg. Operations

Theorem: The class of regular languages is closed under concatenation ($\circ$).

(Equiv: If A regular, B regular languages, then $A \circ B$ regular.)

Idea: Build an NFA that recognizes $A \circ B$, by nondeterministically guessing when to stop reading a string from A and start reading a string from B.

Proof by Picture. Suppose we have two regular languages, A and B, recognized by the NFAs N_A and N_B. We'll build a new NFA, N, that recognizes $A \circ B$.



Goal: accept xy s.t. $x \in A$, $y \in B$.

Create N by: (1) including N_A and N_B entirely

(2) let the start state of N be the start state of N_A

(3) Give each accept state in N_A an ε-arrow to the

start state of N_B, and turn accept states of N_A into ordinary states.

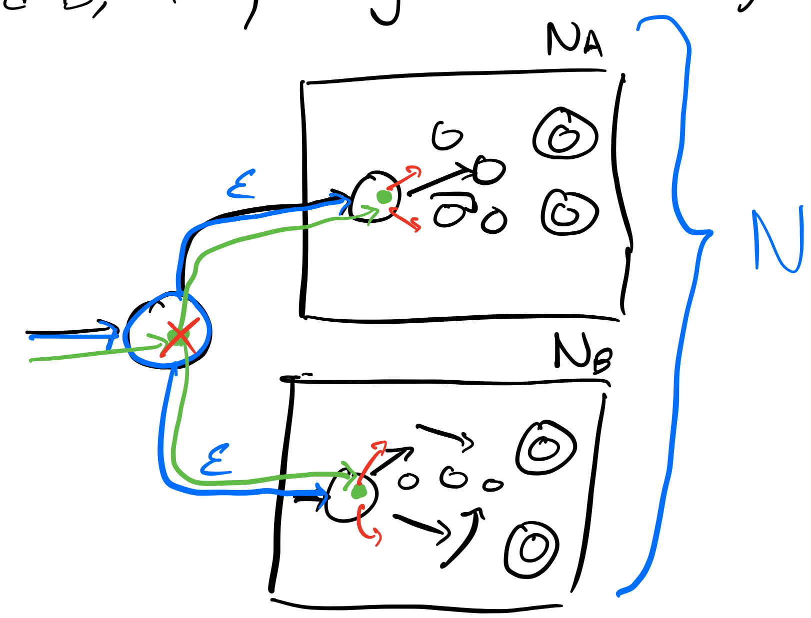
Claim: N accepts $w \iff w = xy$ for some $x \in A$, $y \in B$.

$\Leftarrow$. Suppose $x \in A$, $y \in B$. On input xy, some branch of computation reaches an accept state of N_A after reading in x. That branch then takes the ε-transition to the start of N_B, which accepts on y.

$\Rightarrow$. Suppose N accepts some string w. Then $\exists$ some branch that reaches an accept state. Let $r_1, r_2, \dots r_{(N_B)}, \dots r_m$ be the sequence of states that track our accepting branch, and let $r_{(N_B)}$ denote the start state of N_B. Then we know that $r_1 \dots r_{(N_B - 1)}$ correspond to a branch that reaches an accept state of N_A, and $r_{(N_B)} \dots r_m$ correspond to an accepting sequence in N_B. $\square$

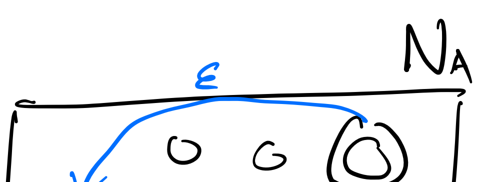
Theorem. (Already proved.) The class of Regular languages is closed under union.

Proof by picture. Given NFAs N_A and N_B that recognize languages A and B, the following NFA N recognizes $A \cup B$.

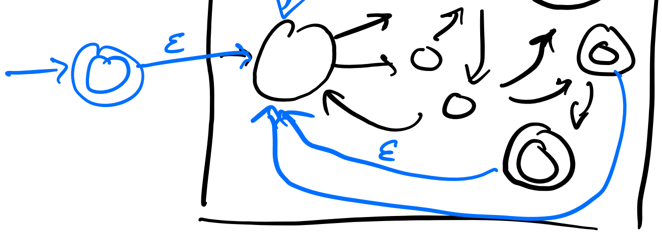


Theorem. The class of regular languages is closed under star (*).

Proof by Picture. Let N_A be an NFA that recognizes A. We'll show an NFA N that recognizes A^*.



x_1, x_2



$A^* := \{a_1 a_2 a_3 \cdots a_k \mid k \geq 0, a \in A\}$

1. ε-arrows from end states to start state. This ensures that every time a branch of computation recognizes a substring, we guess a partition of the input at that index and try to read a new substring.
2. ε case. Make sure we accept ε!

Punchline: If we know that some set of languages R contains only regular languages, then anything we build from R using regular operations ($\cup, \circ, *$) is also regular!

Next up: regular expressions!