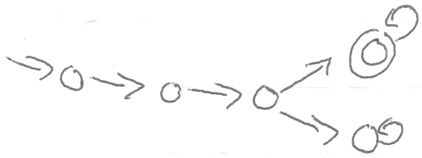
3-1

Complexity DFAs

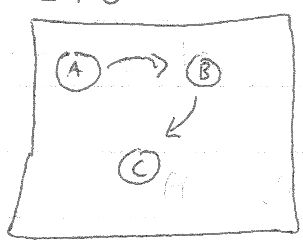
Suppose DFA has $|Q|$ states and is given a string of length $|w|$. $|Q| = M$, $|w| = N$

Time — how long to run — N steps — $O(n)$ — linear

Space — how much memory — $O(\log M)$ — logarithmic in M
— constant re: input



CPU



A // rax = 5 rbx = ... = r15 = 0

B // rax = 6 rbx = ... = r15 = 0

A → B

"incq rax"

Transducer = DFA w/ output

Γ = output alphabet

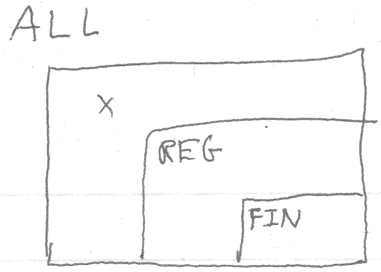
(1) $\theta : Q \to \Gamma$

output in state Q

(2) $\delta' : Q \times \Sigma \to Q \times \Gamma$

DFAs are real-time / online vs "batch"

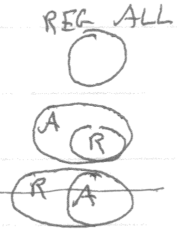
3-2

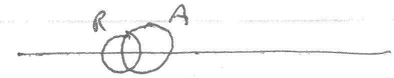


What problems do DFAs solve?

REGULAR

REG





✗ REG = ALL

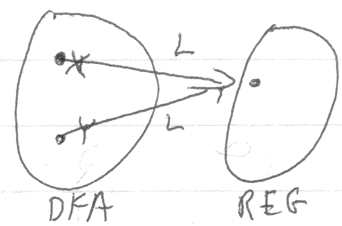
✓ REG $\subsetneq$ ALL

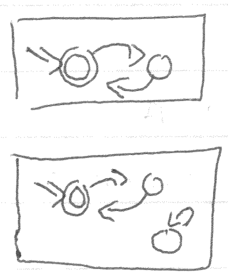
~~ALL $\subsetneq$ REG~~

REG is a set of languages (set of sets of strings)

DFA is a set of machines (set of 5-tuples)

A ∈ REG iff ∃ D ∈ DFA, L(D) = A





Is REG closed under some operations?

Take an element, do something, get some in set

N closed under add 1

N not closed under sub 1

3-3/ Union operation

Is REG closed under union?

$\forall \vec{x}\ \exists \vec{y},\ P(\vec{x},\vec{y})$

$\forall A \in REG,\ \forall B \in REG,\ A \cup B \in REG$

$\longleftrightarrow$

$\forall \hat{A} \in DFA,\ \forall \hat{B} \in DFA,\ \exists \hat{C} \in DFA,\ L(\hat{C}) = L(\hat{A}) \cup L(\hat{B})$

input: $\hat{A} = (Q_A, \Sigma, q_{0A}, \delta_A, F_A)$

$\hat{B} = (Q_B, \Sigma, q_{0B}, \delta_B, F_B)$

output: $\hat{C} = (Q_C, \Sigma, q_{0C}, \delta_C, F_C)$

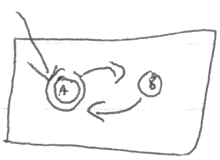
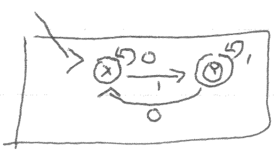
$Q_C = Q_A \times Q_B$ $\qquad F_A \cup F_B = $ X wrong

$q_{0C} = (q_{0A}, q_{0B})$ $\qquad F_A \times F_B = \cap$

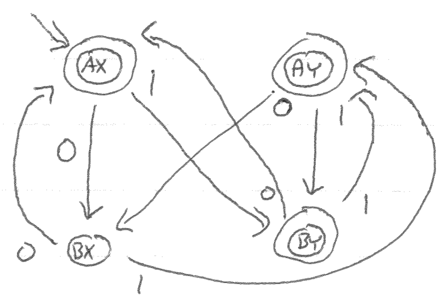
$F_C = F_A \times Q_B \cup Q_A \times F_B$

$\delta_C = Q_C * \Sigma \rightarrow Q_C$ $\qquad \delta((q_a, q_b), c)$

$(Q_A \times Q_B) \times \Sigma \Rightarrow (Q_A \times Q_B)$ $\qquad = (\delta_A(q_a, c), \delta_B(q_b, c))$

E =  ends in 1  $\Sigma = \{0, 1\}$

E ∪ ends in 1



REG = closed ∪, ∩, C = complement

$Q' = Q - F$