Finite Automata

- Deterministic Finite Automata
- Languages decided by a DFA – Regular Languages
- Closure Properties of regular languages
- Non-Deterministic Finite Automata, DFA= NFA
- Regular Expression: Computation as Description
- DFA=NFA=RegExp, Generalized NFA
- Non-Regular Languages, The Pumping Lemma
- Minimizing DFA

Imdad ullah Khan

More Closure Properties of Regular Languages

Regular Languages are closed under “Reversal”

For every NFA N, there is a DFA M such that $L(M) = L(N)$

Regular Languages = DFA-Recognizable Langauges = NFA-Recognizable Languages

Definition (Regular Language)

A language L is regular if it is accepted by an NFA

Theorem

If L is a regular language over Σ, then L^R is also regular

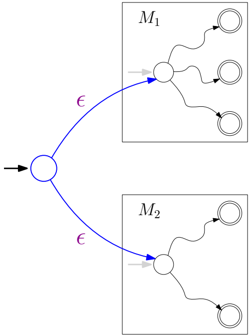
Proof: If L is a regular, then there is a DFA M recognizing it

The NFA M^R recognizes L^R, thus L^R is also regular

We can also make a DFA M' equivalent to M^R

Closure Properties via NFA

NFAs make proving closure properties simpler

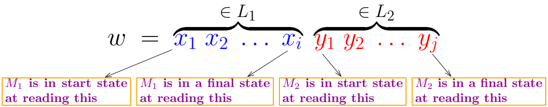


What language is accepted by the new machine?

Regular Languages are closed under "Concatenation"

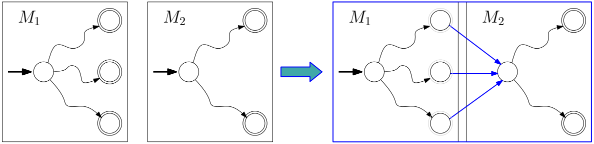
If L_1 and L_2 are regular languages over Σ, then $L_1 \circ L_2$ is also regular

M_1 and M_2: DFAs recognizing L_1 and L_2, make M to recognize $L_1 \circ L_2$



Intuitively, M should simulate M_1 until x_i (check if M_1 is accepting) and then start to simulate M_2 until y_j (and check if M_2 is accepting)

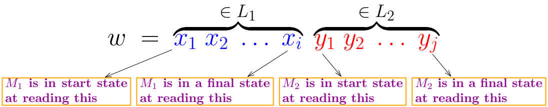
Don't know boundary of x and y, attach F_1 to q_0^2



Regular Languages are closed under "Concatenation"

If L_1 and L_2 are regular languages over Σ, then $L_1 \circ L_2$ is also regular

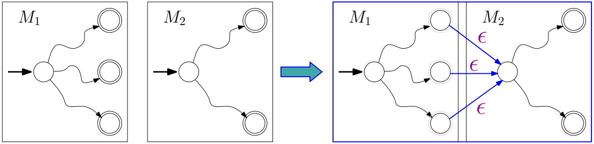
M_1 and M_2: DFAs recognizing L_1 and L_2, make M to recognize $L_1 \circ L_2$



Intuitively, M should simulate M_1 until x_i (check if M_1 is accepting) and then start to simulate M_2 until y_j (and check if M_2 is accepting)

Don't know boundary of x and y, attach F_1 to q_0^2

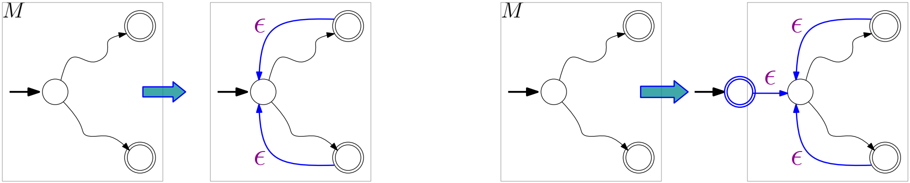
Accepting state of M_2 may be visited many times (don't indicate x_i), need to guess end of x and beginning of y



Regular Languages are closed under "Star"

L is a regular language $\implies L^* = \{w_1 \dots w_n : n \geq 0 \wedge w_i \in L\}$ is regular

M: DFAs recognizing L. M^* simulates serial cascade of M to recognize L^*



Give formal construction of NFA's to recognize the concatenation of two regular languages and the star of a regular language