15-251
Great Ideas in Theoretical Computer Science

Lecture 4:
Deterministic Finite Automaton (DFA), Part 2

January 25th, 2018

Closure properties of regular languages

Closed under complementation

Proposition:

Let Σ be some finite alphabet.
If $L \subseteq \Sigma^*$ is regular, then so is $\overline{L} = \Sigma^* \backslash L$.

Proof:

Closed under union

Theorem:
Let Σ be some finite alphabet.
If $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$ are regular, then so is $L_1 \cup L_2$.

Proof:

The mindset

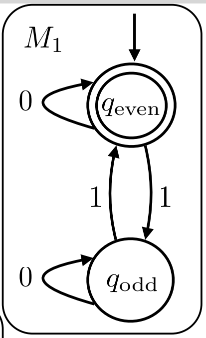
Step 1: Imagining ourselves as a DFA

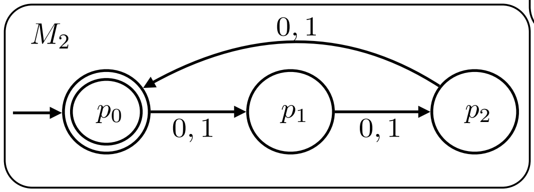
Closed under union

Example

$L_1 =$ strings with even number of 1's

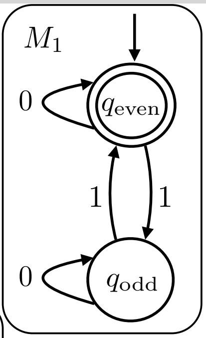
$L_2 =$ strings with length divisible by 3.

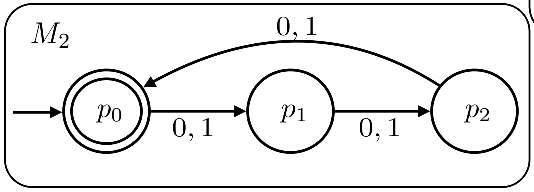




Closed under union

Input: 101001
↑

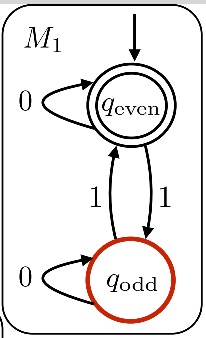


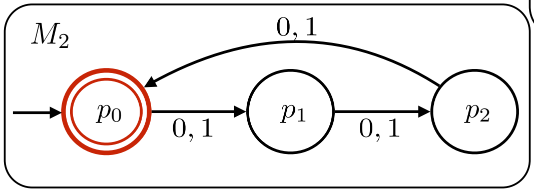


Closed under union

Input: 101001
↑

Accept

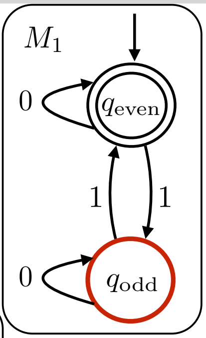


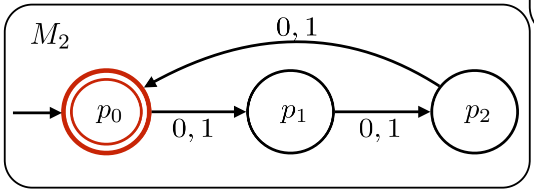


Closed under union

Main idea:

Construct a DFA that keeps track of both at once.

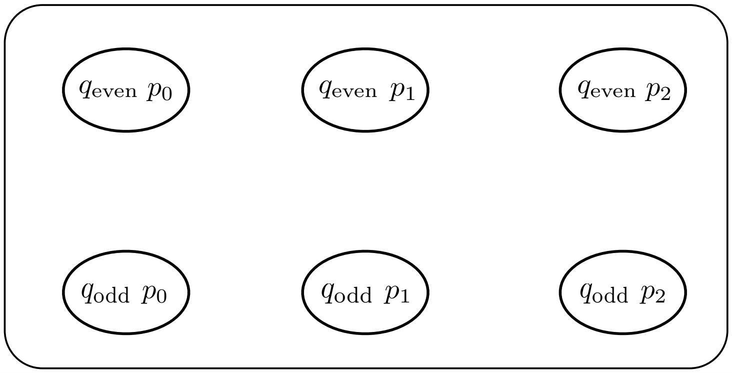




Closed under union

Main idea:

Construct a DFA that keeps track of both at once.



<u>Step 2</u>: Formally defining the DFA

Closed under union

Proof: Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA deciding L_1 and $M' = (Q', \Sigma, \delta', q_0', F')$ be a DFA deciding L_2. We construct a DFA $M'' = (Q'', \Sigma, \delta'', q_0'', F'')$ that decides $L_1 \cup L_2$, as follows:

More closure properties

Closed under union:

Closed under concatenation:

Closed under star:

---

super awesome vs regular

What is the relationship between
super awesome and regular ?

super awesome vs regular

Theorem:

Can define regular languages recursively as follows:

Closed under concatenation

Theorem:
Let Σ be some finite alphabet.
If $L_1 \subseteq \Sigma^*$ and $L_2 \subseteq \Sigma^*$ are regular, then so is $L_1 L_2$.

The mindset

Imagine yourself as a DFA.

Rules:

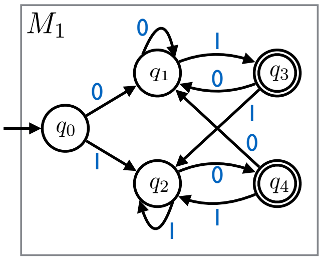
1) Can only scan the input once, from left to right.

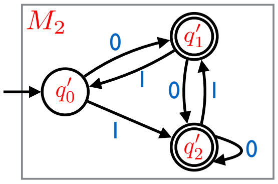
2) Can only remember "constant" amount of information.

↓

should not change
based on input length

<u>Step 1</u>: Imagining ourselves as a DFA



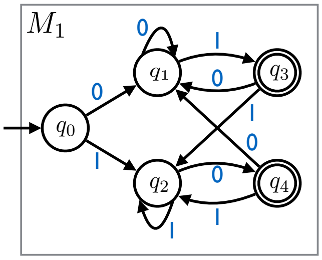


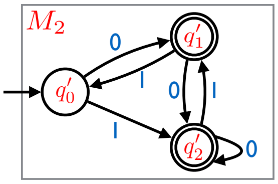
Given $w \in \Sigma^*$, we need to decide if

$$w = uv \quad \text{for} \quad u \in L_1,\ v \in L_2.$$

Problem: Don't know where u ends, v begins.

When do you stop simulating M_1 and start simulating M_2 ?

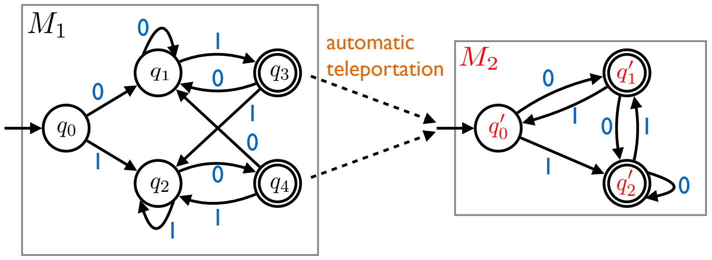




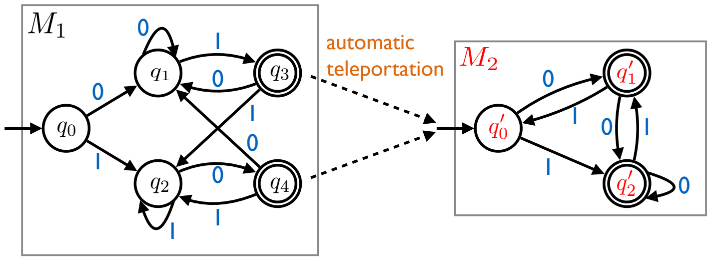
Suppose God tells you u ends at w_3 .

	0	0	1	1	0	0	1	0	0	1
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
q_0	q_1	q_1	$\textcircled{q_3}$							
			q_0'	q_2'	q_2'	q_2'	q_1'	q_2'	q_2'	q_1'

<u>thread</u>:



		0	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**
		w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}
	q_0	q_1	q_1	q_3	q_2	q_4	q_1	q_3	q_1	q_1	q_3	q_2
thread1				q'_0	q'_2	q'_2	q'_2	q'_1	q'_2	q'_2	q'_1	q'_0
thread2						q'_0	q'_1	q'_0	q'_1	q'_2	q'_1	q'_0
thread3								q'_0	q'_1	q'_2	q'_1	q'_0
thread4											q'_0	q'_2



		0	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**
		w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}
	q_0	q_1	q_1	q_3	q_2	q_4	q_1	q_3	q_1	q_1	q_3	q_2
thread1				q'_0	q'_2	q'_2	q'_2	q'_1	q'_2	q'_2	q'_1	q'_0
thread2						q'_0	q'_1	q'_0	q'_1	q'_2	q'_1	q'_0
thread3								q'_0	q'_1	q'_2	q'_1	q'_0
thread4											q'_0	q'_2

	0	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**0**	**1**	**1**
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}
q_0	q_1	q_1	q_3	q_2	q_4	q_1	q_3	q_1	q_1	q_3	q_2
			q'_0	q'_2	q'_2 q'_0	q'_2 q'_1	q'_1 q'_0 q'_0	q'_2 q'_1 q'_1	q'_2 q'_2 q'_2	q'_1 q'_1 q'_1 q'_0	q'_0 q'_0 q'_0 q'_2
	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$	$\subseteq$
	Q'	Q'	Q'	Q'	Q'	Q'	Q'	Q'	Q'	Q'	Q'

Step 2: Formally defining the DFA

$M_1 = (Q, \Sigma, \delta, q_0, F)$ $M_2 = (Q', \Sigma, \delta', q_0', F')$

$Q'' =$

$\delta'' :$

$q_0'' =$

$F'' =$