Lecture 2 - NFAs

Eric A. Autry

Last time: Review of Graphs, DFAs

This time: Union, Concatenation, NFAs

Next time: NFAs

Homework 1 on Sakai and due this Thursday in class.

Quiz 0 on Sakai

I am making it due this Thursday as well.

Office Hours

For this week, office hours will be:

- Wednesday from 4:30 - 8 pm
- Physics 119

Regular Operations

Let A and B be languages.

- Union: $A \cup B = \{w \mid w \in A \text{ or } w \in B\}$
- Concatenation: $A \circ B = \{w_1 w_2 \mid w_1 \in A \text{ and } w_2 \in B\}$
- Star: $A^* = \{w_1 w_2 \ldots w_k \mid k \geq 0 \text{ and each } w_i \in A\}$
  - What if k=0?
  - $\varepsilon \in A^*$

Union

Can we prove that regular languages are closed under the union operation?

In other words, can we prove that if A and B are regular languages, then $A \cup B$ is also a regular language?

Idea: Recall that a regular language is one that can be recognized by a finite automaton. So, let's build an automaton.

Union

Can we prove that if A and B are regular languages, then $A \cup B$ is also a regular language?

Since A and B are regular languages, we know that there must exist a machine M_1 that recognizes A and a machine M_2 that recognizes B.

Our goal is to create a machine M that recognizes $A \cup B$ using machines M_1 and M_2.

Can we just run M_1, see if that works, then run M_2?

Nope. We can't rewind the input!

Union

We are on the right track, but we somehow need to figure out how to simulate both machines M_1 and M_2 at the same time, and accept the input if either machine accepts.

Step 1: What information do we want to track?

- What information do we need to simulate a machine?
- We need to know what state that machine is in.
- So, we just need to keep track of two states at once.

Idea: Use ordered pairs.

Union

Idea: Use ordered pairs.

If Q_1 are the states for M_1, and Q_2 are the states for M_2, let's define the states for our new combined machine as $Q = Q_1 \times Q_2$.

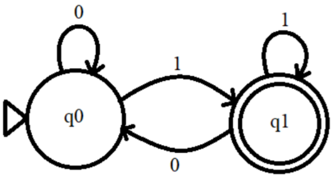
Ex: If $Q_1 = \{q_0, q_1, q_2\}$ and $Q_2 = \{q_a, q_b\}$, then

$$Q = \{\ (q_0, q_a),\ (q_0, q_b),\ (q_1, q_a),\ (q_1, q_b),\ (q_2, q_a),\ (q_2, q_b)\ \}$$

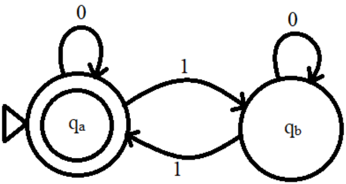
Union

Ex:

$$L(M_1) = \{w \mid w \text{ ends in a } 1\} :$$



$$L(M_2) = \{w \mid w \text{ has an even number of } 1\text{'s}\} :$$



Union

Step 2: What state is the start state?

- M_1 starts in q_0 and M_2 starts in q_a, so the combined M should start in (q_0, q_a).

Step 3: What states are the accept states?

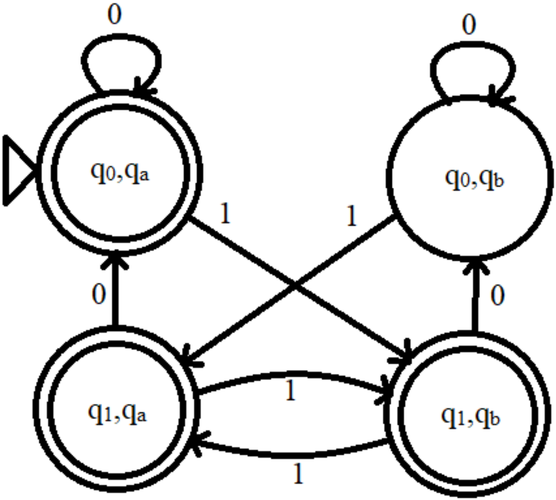
- Any state in our combined machine that corresponds to either M_1 and M_2 being in accepting states.

Union

Step 4: Fill in the transitions.

- Say we are looking at state (q_i, q_j).
- This means that M_1 is in state q_i and M_2 is in state q_j.
- What happens if next symbol in the input is a 0 that sends M_1 to state q_n and M_2 to state q_m?
- Then our combined machine will transition to state (q_n, q_m).

Union



Union

Recall:

A finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set called the states,
2. Σ is a finite set of symbols called the alphabet,
3. $\delta : Q \times \Sigma \to Q$ is the transition function,
4. $q_0 \in Q$ is the start state,
5. $F \subseteq Q$ is the set of accepting states.

So we've successfully defined a new machine M that recognizes the language $A \cup B$!

Concatenation

Can we prove that regular languages are closed under concatenation?

$A \circ B = \{w_1 w_2 \mid w_1 \in A \text{ and } w_2 \in B\}$

Idea: First run M_1 on part of the input, then run M_2 on the remainder.

Problem: Where should we break our input?

Nondeterminism

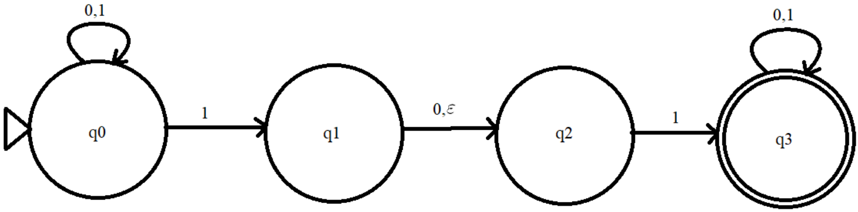
All of the finite automatons we have seen so far have been deterministic.

This means that the machine will always behave the same way for a given input.

We call these deterministic finite automata or DFAs.

Let's introduce nondeterministic finite automata or NFAs.

NFAs

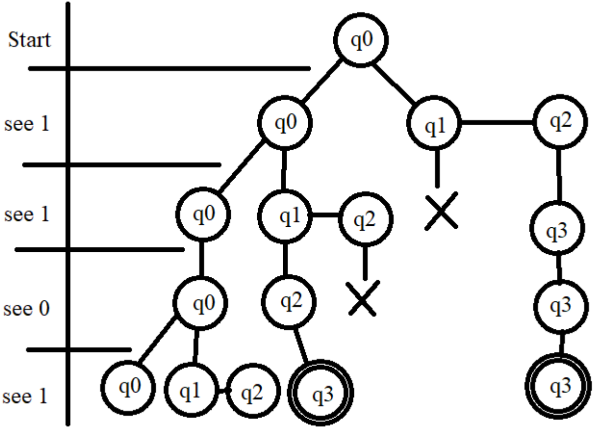


- What happens if we are in q_0 and see a 1?
- What does the ε mean between q_1 and q_2?
- What if q_2 sees a 0?
- Ex: let's try the string '1101'
- Ex: let's try the string '010'
- What does this machine do?

$$L(M) = \{w \,|\, w \text{ contains the substrings } 11 \text{ or } 101\}.$$

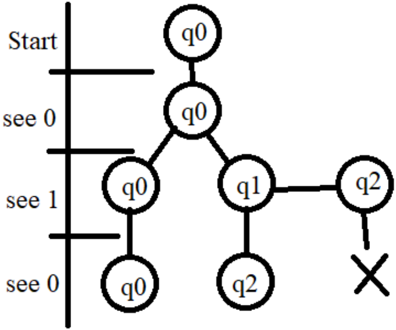
NFAs

Ex: let's try the string '1101'

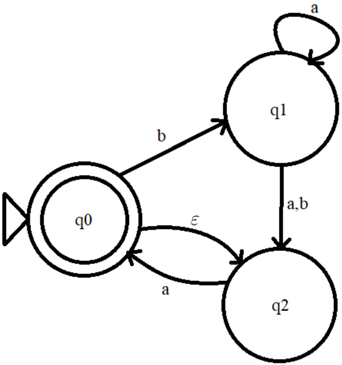


NFAs

Ex: let's try the string '010'



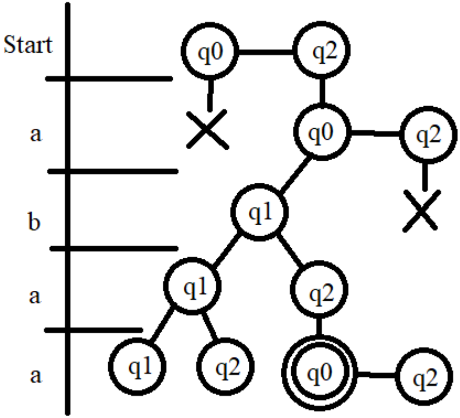
NFAs



- Ex: let's try the string ε
- Ex: let's try the string 'a'
- Ex: let's try the string 'b'
- Ex: let's try the string '$abaa$'
- Ex: let's try the string '$babba$'
- Ex: let's try the string 'baa'

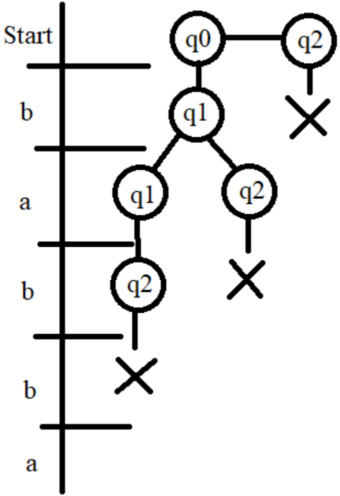
NFAs

Ex: let's try the string '*abaa*'



NFAs

Ex: let's try the string '*babba*'



NFA - Formal Definition

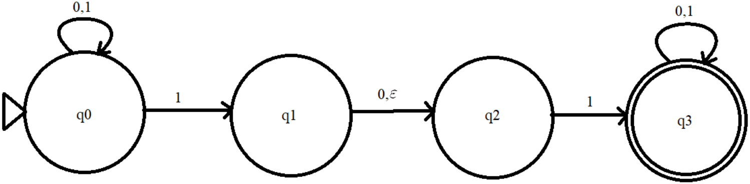
A nondeterministic finite automaton is a 5-tuple $(Q, \Sigma, \delta, q_0, F)$, where

1. Q is a finite set of states,
2. Σ is a finite alphabet,
3. $\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ is the transition function,
   - Σ_ε is the alphabet $\Sigma \cup \{\varepsilon\}$.
   - $\mathcal{P}(Q)$ is the power set (set of all subsets) of Q.
4. $q_0 \in Q$ is the start state,
5. $F \subseteq Q$ is the set of accepting states.

Note: all DFAs are also NFAs.

NFA - Formal Definition

Ex:



1. $Q = \{q_0, q_1, q_2, q_3\}$,
2. $\Sigma = \{0, 1\}$,
3. $\delta : Q \times \Sigma_\varepsilon \to \mathcal{P}(Q)$ is:

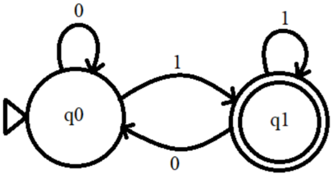
	0	1	ε
q_0	$\{q_0\}$	$\{q_0, q_1\}$	$\emptyset$
q_1	$\{q_2\}$	$\emptyset$	$\{q_2\}$
q_2	$\emptyset$	$\{q_3\}$	$\emptyset$
q_3	$\{q_3\}$	$\{q_3\}$	$\emptyset$

4. q_0 is the start state,
5. $F = \{q_3\}$.

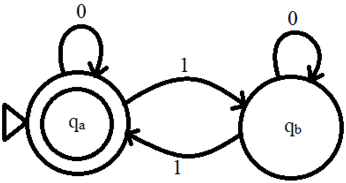
Union - NFAs

How would we create an NFA that can recognize $A \cup B$?

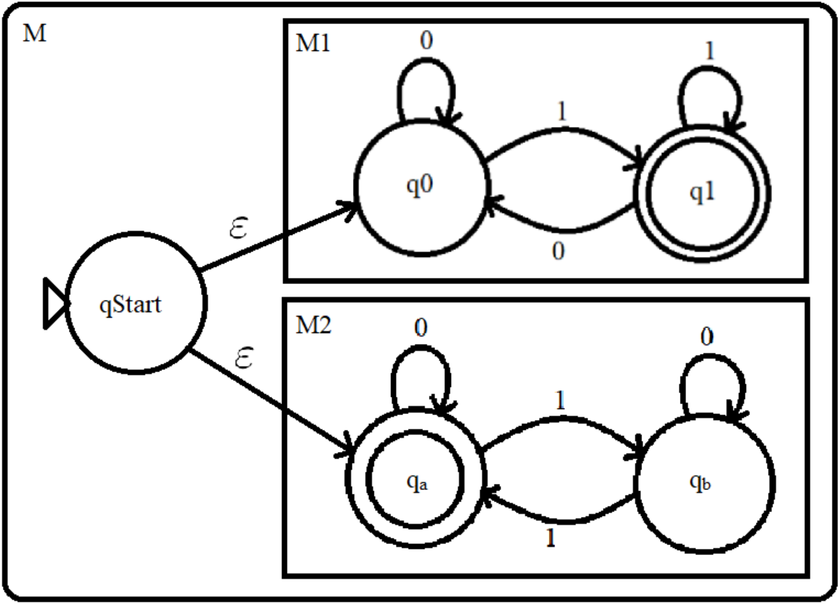
$$L(M_1) = \{w \mid w \text{ ends in a } 1\} :$$



$$L(M_2) = \{w \mid w \text{ has an even number of } 1\text{'s}\} :$$

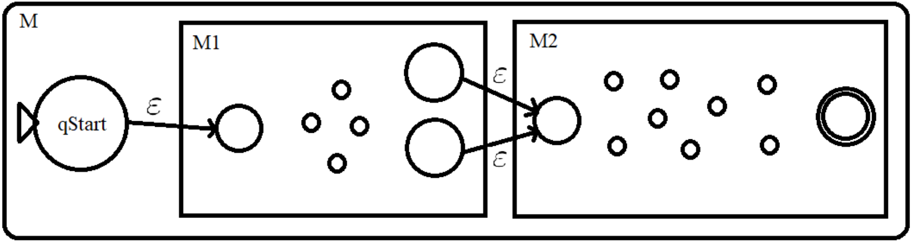


Union - NFAs



Concatenation - NFAs

How would we create an NFA that can recognize $A \circ B$?



Star - NFAs

How would we create an NFA that can recognize A^*?