Relating DFAs and NFAs

13 September 2022
Assignment 2

Due today

Corrections due Thursday
Where are we?
A language is a *regular language* iff some deterministic finite automaton recognizes it.
DFAs let us recognize languages where we only need to keep track of a fixed set of possibilities, e.g.,

All even-length strings
Any number of \texttt{a}s followed by any number of \texttt{b}s
Alternating \texttt{a}s and \texttt{b}s
Strings ending with \texttt{er}, \texttt{or}, or \texttt{ist}
**Deterministic finite automaton**: Computation proceeds according to the design of the transition function.

There are no choices to make in the computation.

**Nondeterministic finite automaton**: Zero or more options to continue the computation.

Accept if *any possible sequence* of choices would succeed.
Formally, an NFA is defined like a DFA:

\[ N = (Q, \Sigma, \delta, q_0, F) \]

Except now the output of the transition function \( \delta \)
\[ \delta(q_0, a) \]

isn’t a single state but a set of states.
NFAs can have a special type of transition called an \( \varepsilon \)-transition:

An NFA may follow any number of \( \varepsilon \)-transitions at any time without consuming any input.
Language of an NFA:

An NFA accepts an input string \( w \) if any path from the start state to an accept state is labeled \( w \).
We can think of an NFA as a DFA that can magically guess the right choices to make.

Alternatively, we can think of an NFA as a DFA that can be in many states at once – massive parallelism.

When it needs to follow a transition, it tries all of the options at the same time.
Relating DFAs and NFAs
Just how powerful are NFAs?
**NFAs must be at least as powerful as DFAs.**

Any language that can be recognized by a DFA can be recognized by an NFA.

Why? Essentially, every DFA already *is* an NFA – just one that doesn’t exploit nondeterminism.
Can every language recognized by an NFA also be recognized by a DFA?
Can every language recognized by an NFA also be recognized by a DFA?

While NFAs *seem* more powerful, surprisingly, the answer is yes!
Thought experiment: How could you simulate an NFA in software?
The given automaton starts in state $q_0$ and transitions as follows:

- From $q_0$, an input of $a$ moves to $q_1$.
- From $q_1$, an input of $b$ moves to $q_2$.
- From $q_2$, an input of $a$ moves to $q_3$.

The diagram also shows that there is a loop from $q_3$ back to $q_0$ labeled with $a, b$. The input sequence $a b a b a$ indicates the sequence of inputs that leads from the start state to $q_3$. The automaton accepts this input sequence.
a

\rightarrow \{q_0\}
\[ a \xrightarrow{} \{q_0\} \]
$\begin{align*}
q &\rightarrow \{q_0\} \\
\end{align*}$
\[
q_0 \xrightarrow{a, b} q_0 \\
q_0 \xrightarrow{a} q_1 \\
q_1 \xrightarrow{b} q_2 \\
q_2 \xrightarrow{a} q_3
\]

\[
\rightarrow \{q_0\} \quad \{q_0, q_1\}
\]
\[\begin{align*}
\text{start} & \quad \rightarrow \quad \{q_0\} \\
a, b & \quad \rightarrow \quad \{q_0, q_1\}
\end{align*}\]
\[
\begin{align*}
\text{start} & \quad \rightarrow \quad \{q_0\} \\
q_0 & \quad \xrightarrow{a, b} \quad \{q_0, q_1\} \\
q_1 & \quad \xrightarrow{a} \quad \{q_0\} \\
q_2 & \quad \xrightarrow{b} \quad \{q_0\} \\
q_3 & \quad \text{final state}
\end{align*}
\]
The diagram represents a finite automaton with states $q_0$, $q_1$, $q_2$, and $q_3$. The transitions are as follows:

- From $q_0$ on input $a$, go to $q_1$.
- From $q_1$ on input $b$, go to $q_2$.
- From $q_2$ on input $a$, go to $q_3$.
- There is a self-loop on $q_0$ on input $a, b$.

The initial state is $q_0$. The states and transitions are:

- $q_0$: $\{q_0\}$
- $q_1$: $\{q_0, q_1\}$
- $q_2$: $\{q_0\}$
- $q_3$: $\{q_3\}$
The diagram represents a finite automaton with the following states and transitions:

- **States:**
  - $q_0$
  - $q_1$
  - $q_2$
  - $q_3$

- **Transitions:**
  - From $q_0$ on $a$ to $q_1$
  - From $q_0$ on $b$ to $q_1$
  - From $q_1$ on $a$ to $q_2$
  - From $q_1$ on $b$ to $q_2$
  - From $q_2$ on $a$ to $q_3$

The arrows are labeled with input symbols $a$, $b$, and $\rightarrow$ (for start state).

The transitions are associated with sets of states:

- $\rightarrow \{q_0\}$
- $\rightarrow \{q_0, q_1\}$
- $\rightarrow \{q_0\}$
- $\rightarrow \{q_0, q_1\}$
\begin{itemize}
\item $a, b$
\item $\rightarrow \{q_0\}$
\item $\rightarrow \{q_0, q_1\}$
\item $\rightarrow \{q_0, q_1\}$
\end{itemize}
\[
\begin{align*}
q_0 & \xrightarrow{a, b} q_0 \\
v & \rightarrow \{q_0\} \\
q_0 & \xrightarrow{a} q_1 \\
v & \rightarrow \{q_0, q_1\} \\
q_1 & \xrightarrow{b} q_2 \\
v & \rightarrow \{q_0, q_1\} \\
q_2 & \xrightarrow{a} q_3 \\
v & \rightarrow \{q_0\}
\end{align*}
\]
\[ q_0 \xrightarrow{a, b} q_1 \xrightarrow{a} q_2 \xrightarrow{a} q_3 \]

<table>
<thead>
<tr>
<th></th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \rightarrow )</td>
<td>{q_0}</td>
<td>{q_0, q_1}</td>
</tr>
<tr>
<td>{q_0, q_1}</td>
<td>{q_0, q_1}</td>
<td>{q_0, q_1}</td>
</tr>
</tbody>
</table>
\[
\begin{align*}
\text{start} & \rightarrow \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0, q_1\}
\end{align*}
\]
\[
\begin{align*}
\text{start} & \rightarrow \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0, q_1\}
\end{align*}
\]
\[
\begin{align*}
q_0 &\xrightarrow{a,b} q_0 \\
q_0 &\xrightarrow{a} q_1 \\
q_1 &\xrightarrow{b} q_2 \\
q_2 &\xrightarrow{a} q_3
\end{align*}
\]
The given automaton has the following transition table:

- **a**
  - From $q_0$: $\{q_0\}$
  - From $q_1$: $\{q_0, q_1\}$
  - From $q_2$: $\{q_0, q_2\}$

- **b**
  - From $q_0$: $\{q_0\}$
  - From $q_1$: $\{q_0, q_1\}$
  - From $q_2$: $\{q_0, q_2\}$

The start state is $q_0$. The transitions are labeled with $a$ and $b$. The state $q_3$ is a dead state.
\[
\begin{align*}
\text{start} & \rightarrow \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0, q_1\} \\
\{q_0, q_2\} & \rightarrow \{q_0, q_2\}
\end{align*}
\]
Start

q_0 \rightarrow \{q_0\} \rightarrow \{q_0, q_1\}
q_0 \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_1\}
q_0 \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_2\}

q_1 \rightarrow \{q_0\} \rightarrow \{q_0\}
q_1 \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_1\}
q_1 \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_2\}

q_2 \rightarrow \{q_0\} \rightarrow \{q_0\}
q_2 \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_1\}
q_2 \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_2\}

q_3
The given automaton has the following transitions:

- From state $q_0$:
  - On input $a$: $q_0$ (loop)
  - On input $b$: $q_1$ (solid arrow)

- From state $q_1$:
  - On input $a$: $q_1$ (solid arrow)
  - On input $b$: $q_2$ (solid arrow)

- From state $q_2$:
  - On input $a$: $q_3$ (solid arrow)

- From state $q_3$:
  - On input $a$: $q_0$ (solid arrow)

The input set for each state is:

- $q_0$: $\{q_0\}$
- $q_1$: $\{q_0, q_1\}$
- $q_2$: $\{q_0, q_1\}$
- $q_3$: $\{q_0, q_2\}$
\[ \begin{align*}
\text{a} & \rightarrow \{q_0\} \quad \{q_0, q_1\} \quad \{q_0\} \\
\text{b} & \rightarrow \{q_0, q_1\} \quad \{q_0, q_1\} \quad \{q_0, q_2\}
\end{align*} \]
a, b

\[
\begin{array}{ccc}
\rightarrow \{q_0\} & \{q_0, q_1\} & \{q_0\} \\
\{q_0, q_1\} & \{q_0, q_1\} & \{q_0, q_2\} \\
\{q_0, q_2\} & \{q_0, q_1, q_3\} & \\
\end{array}
\]
\[ \begin{array}{c|c|c|c|c}
\text{Input} & a & b & a, b \\
\hline
\text{States} & \{q_0\} & \{q_0, q_1\} & \{q_0\} \\
& \{q_0, q_1\} & \{q_0, q_1\} & \{q_0, q_2\} \\
& \{q_0, q_2\} & \{q_0, q_1, q_3\} & \\
\end{array} \]
q0 \rightarrow \{q_0\} 
\{q_0, q_1\} \rightarrow \{q_0, q_1\} 
\{q_0, q_2\} \rightarrow \{q_0, q_1, q_3\} 
\{q_0\} \rightarrow \{q_0\} 
\{q_0, q_1\} \rightarrow \{q_0, q_1\} 
\{q_0, q_2\} \rightarrow \{q_0, q_2\}
\begin{align*}
\rightarrow & \quad \{q_0\} \quad \{q_0, q_1\} \quad \{q_0\} \\
& \quad \{q_0, q_1\} \quad \{q_0, q_1\} \quad \{q_0, q_2\} \\
& \quad \{q_0, q_2\} \quad \{q_0, q_1, q_3\}
\end{align*}
\begin{align*}
\text{a} & \quad \{q_0\} \quad \{q_0, q_1\} \\
\text{b} & \quad \{q_0, q_1\} \quad \{q_0, q_2\} \\
\end{align*}
\begin{itemize}
  \item \textbf{a}:
    \begin{align*}
    \{q_0\} & \rightarrow \{q_0\} \\
    \{q_0, q_1\} & \rightarrow \{q_0, q_1\} \\
    \{q_0, q_2\} & \rightarrow \{q_0, q_1, q_3\}
    \end{align*}
  \\
  \item \textbf{b}:
    \begin{align*}
    \{q_0, q_1\} & \rightarrow \{q_0, q_1\} \\
    \{q_0, q_2\} & \rightarrow \{q_0, q_2\}
    \end{align*}
\end{itemize}
\[
\begin{align*}
\text{a} & : \\
\rightarrow & \{q_0\} \quad \{q_0, q_1\} \quad \{q_0\} \\
\{q_0, q_1\} & \quad \{q_0, q_1\} \quad \{q_0, q_2\} \\
\{q_0, q_2\} & \quad \{q_0, q_1, q_3\} \quad \{q_0\} \\
\{q_0, q_1, q_3\} & \\
\end{align*}
\]
\[ a, b \]

\[
\begin{align*}
q_0 &\rightarrow \{q_0\} \\
\{q_0, q_1\} &\rightarrow \{q_0, q_1\} \\
q_0 &\rightarrow \{q_0, q_1\} \\
q_0 &\rightarrow \{q_0, q_1, q_2\} \\
q_0 &\rightarrow \{q_0, q_1, q_3\} \\
\{q_0, q_1, q_2\} &\rightarrow \{q_0, q_1, q_3\} \\
\{q_0, q_1, q_3\} &\rightarrow \{q_0\}
\end{align*}
\]
$\{q_0, q_1\}$  $\{q_0, q_1\}$  $\{q_0\}$

$\{q_0, q_2\}$  $\{q_0, q_1, q_3\}$  $\{q_0\}$

$\{q_0, q_2\}$  $\{q_0, q_1\}$  $\{q_0, q_2\}$

$\{q_0\}$  $\{q_0\}$  $\{q_0\}$
\[
\begin{align*}
\text{Transition Labels:} & \quad \{q_0\} & \quad \{q_0, q_1\} & \quad \{q_0\} \\
& \quad \{q_0, q_1\} & \quad \{q_0, q_1\} & \quad \{q_0, q_2\} \\
& \quad \{q_0, q_2\} & \quad \{q_0, q_1, q_3\} & \quad \{q_0, q_2\} \\
& \quad \{q_0, q_1, q_3\} & \quad \{q_0\} & \quad \{q_0\}
\end{align*}
\]
\[
\begin{align*}
\rightarrow & \{q_0\} & \{q_0, q_1\} & \{q_0\} \\
& \{q_0, q_1\} & \{q_0, q_1\} & \{q_0, q_2\} \\
& \{q_0, q_2\} & \{q_0, q_1, q_3\} & \{q_0\} \\
& \{q_0, q_1, q_3\} & & \{q_0\}
\end{align*}
\]
\[
\begin{align*}
\text{a} & \quad \text{b} \\
\{q_0\} & \quad \{q_0, q_1\} \quad \{q_0\} \\
\{q_0, q_1\} & \quad \{q_0, q_1\} \quad \{q_0, q_2\} \\
\{q_0, q_2\} & \quad \{q_0, q_1, q_3\} \quad \{q_0\} \\
\{q_0, q_1, q_3\} & \quad \{q_0, q_1\} \\
\end{align*}
\]
\[
\begin{array}{ccc}
\text{a} & \text{b} \\
\{q_0\} & \{q_0, q_1\} & \{q_0\} \\
\{q_0, q_1\} & \{q_0, q_1\} & \{q_0, q_2\} \\
\{q_0, q_2\} & \{q_0, q_1, q_3\} & \{q_0\} \\
\{q_0, q_1, q_3\} & \{q_0, q_1\} & \\
\end{array}
\]
\[
\begin{array}{ccc}
a & b \\
\{q_0\} & \{q_0, q_1\} & \{q_0\} \\
\{q_0, q_1\} & \{q_0, q_1\} & \{q_0, q_2\} \\
\{q_0, q_2\} & \{q_0, q_1, q_3\} & \{q_0\} \\
\{q_0, q_1, q_3\} & \{q_0, q_1\} \\
\end{array}
\]
\[
\begin{array}{c}
\text{a} \\
\text{b}
\end{array}
\]

\[
\begin{array}{c}
\{q_0\} \\
\{q_0, q_1\} \\
\{q_0, q_2\} \\
\{q_0, q_1, q_3\} \\
\{q_0, q_1, q_3\}
\end{array}
\]

\[
\begin{array}{c}
\{q_0\} \\
\{q_0, q_1\} \\
\{q_0, q_1\} \\
\{q_0, q_2\} \\
\{q_0, q_1, q_3\}
\end{array}
\]

\[
\begin{array}{c}
\{q_0\} \\
\{q_0, q_1\} \\
\{q_0, q_1, q_3\} \\
\{q_0\}
\end{array}
\]
\[
\begin{array}{ccc}
\text{a} & \rightarrow & \{q_0\} \\
\{q_0, q_1\} & \rightarrow & \{q_0, q_1\} \\
\{q_0, q_2\} & \rightarrow & \{q_0, q_1, q_2\} \\
\{q_0, q_1, q_3\} & \rightarrow & \{q_0, q_1, q_3\} \\
\end{array}
\]
\[
\begin{align*}
\rightarrow & \quad \{q_0\} & \quad \{q_0, q_1\} & \quad \{q_0\} \\
& \quad \{q_0, q_1\} & \quad \{q_0, q_1\} & \quad \{q_0, q_2\} \\
& \quad \{q_0, q_2\} & \quad \{q_0, q_1, q_3\} & \quad \{q_0\} \\
& \quad \{q_0, q_1, q_3\} & \quad \{q_0, q_1\}
\end{align*}
\]
The following is a regular expression and a finite automaton for it.

**Regular Expression:**
- \( q_0 \)
- \( \{ q_0 \} \)
- \( q_0 \) \( , \) \( q_1 \)
- \( \{ q_0 , q_1 \} \)
- \( q_0 \) \( , \) \( q_1 \) \( , \) \( q_3 \)
- \( \{ q_0 , q_1 , q_3 \} \)
- \( q_0 \) \( , \) \( q_1 \)
- \( \{ q_0 \} \)
- \( q_0 \) \( , \) \( q_1 \) \( , \) \( q_3 \)
- \( \{ q_0 , q_1 , q_3 \} \)
- \( q_0 \) \( , \) \( q_1 \)
- \( \{ q_0 \} \)
- \( q_0 \) \( , \) \( q_1 \) \( , \) \( q_3 \)
- \( \{ q_0 , q_1 , q_3 \} \)
- \( q_0 \) \( , \) \( q_1 \)
- \( \{ q_0 \} \)
- \( q_0 \) \( , \) \( q_1 \)
- \( \{ q_0 \} \)
- \( a, b \)

**Finite Automaton:**
- Start state: \( q_0 \)
- Transitions:
  - \( a \) from \( q_0 \) to \( q_1 \)
  - \( b \) from \( q_1 \) to \( q_2 \)
  - \( a \) from \( q_2 \) to \( q_3 \)
  - Looping transition from \( q_0 \) to \( q_0 \) on any input

The automaton and its transitions are consistent with the given regular expression.
The diagram shows a finite automaton with states labeled $q_0$, $q_1$, $q_2$, and $q_3$. The transitions are as follows:

- From $q_0$ on input $a$, move to $q_1$.
- From $q_1$ on input $a$, move to $q_0$.
- From $q_0$ on input $b$, move to $q_1$.
- From $q_1$ on input $b$, move to $q_2$.
- From $q_2$ on input $a$, move to $q_3$.

The transitions for inputs $a$ and $b$ are summarized in the table:

<table>
<thead>
<tr>
<th>Input</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a$</td>
<td>${q_0}$</td>
</tr>
<tr>
<td>$a$</td>
<td>${q_0, q_1}$</td>
</tr>
<tr>
<td>$a$</td>
<td>${q_0, q_1}$</td>
</tr>
<tr>
<td>$a$</td>
<td>${q_0, q_2}$</td>
</tr>
<tr>
<td>$a$</td>
<td>${q_0, q_1, q_3}$</td>
</tr>
<tr>
<td>$a$</td>
<td>${q_0, q_1}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${q_0}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${q_0, q_1}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${q_0, q_2}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${q_0, q_1, q_3}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${q_0}$</td>
</tr>
<tr>
<td>$b$</td>
<td>${q_0, q_2}$</td>
</tr>
</tbody>
</table>
The image shows two diagrams of deterministic finite automata (DFA). The top diagram has states labeled as follows: start (q₀), q₁, q₂, and q₃. Transitions are labeled 'a' and 'b'. The bottom diagram has states labeled as {q₀}, {q₀, q₁}, {q₀, q₂}, and {q₀, q₁, q₃}. Transitions are also labeled 'a' and 'b'. The green text 'a b a a b a' represents the input string that the automata process.
A non-deterministic finite automaton (NFA) with the following transitions:

- Start state: $q_0$
- Transitions:
  - From $q_0$: On 'a' go to $q_1$, On 'b' go to $q_2$
  - From $q_1$: On 'a' go to $q_3$
  - From $q_2$: On 'a' go to $q_3$

Input string: $abaabaaba$
The diagram shows a deterministic finite automaton (DFA) with states labeled \( q_0, q_1, q_2, \) and \( q_3 \). The transitions are labeled with symbols \( a \) and \( b \). The start state is \( q_0 \), and the input string is \( a \ b \ a \ a \ b \ a \ a \).
\[
\begin{align*}
\text{start} & \rightarrow q_0 \ (a, b) \rightarrow q_1 \ (a) \rightarrow q_2 \ (b) \rightarrow q_3 \\
0, 1, 3 & \in \{q_0, q_1, q_2\} \\
\{q_0\} & \rightarrow \{q_0, q_1\} \ (a) \\
\{q_0, q_1\} & \rightarrow \{q_0, q_2\} \ (b) \rightarrow \{q_0, q_1, q_3\} \\
\end{align*}
\]
This method of transforming an NFA into a DFA is called the *subset construction*.

Each state in the DFA corresponds to a set of states in the NFA. The start state in the DFA corresponds to the start state of the NFA. The accept states in the DFA correspond to the sets of states that would be considered to accept in the NFA when using the massive parallel intuition.

If a state $q$ in the DFA corresponds to a set of states $S$ in the NFA, then the transition from state $q$ on a character $\alpha$ is found as follows:

Let $S'$ be the set of states in the NFA that can be reached by following a transition labeled $\alpha$ from any of the states in $S$.

The state $q$ in the DFA transitions on $\alpha$ to a DFA state corresponding to the set of states $S'$. 

*Introduced by Rabin & Scott, 1959*
In converting an NFA to a DFA, the DFA’s states correspond to sets of NFA states.

Useful fact: $|\mathcal{P}(S)| = 2^{|S|}$ for any finite set $S$.

In the worst case, the construction can result in a DFA that is \textit{exponentially larger} than the original NFA.
THEOREM  Every nondeterministic finite automaton has an equivalent deterministic finite automaton.

PROOF  Let \( N = (Q, \Sigma, \delta, q_0, F) \) be an NFA recognizing some language \( A \).

We can construct a DFA \( M = (Q', \Sigma, \delta', q_0', F') \) that recognizes \( A \):

\[
Q' = \wp(Q)
\]
\[
q_0' = \{q_0\}
\]
\[
F' = \{R \in Q' \mid R \text{ contains an accept state of } N\}
\]

For \( R \in Q' \) and \( \alpha \in \Sigma \), we define \( \delta'(R, \alpha) = \bigcup_{r \in R} \delta(r, \alpha) \)

As in Sipser, we use \( R \) both as a state of \( M \) and as a set of states of \( N \).

Every state \( R \) of \( M \) is a set of states of \( N \).

When \( M \) is in state \( R \) and reads a symbol \( \alpha \), it tracks where \( N \) would go on \( \alpha \) each state in \( R \).
What about NFAs with $\varepsilon$-transitions?
It’ll be easier for us to represent the equivalent DFA as a table rather than a (big, messy) diagram.
Start by thinking what the start state of the DFA will be.
Start by thinking what the start state of the DFA will be.
Start by thinking what the start state of the DFA will be.
Start by thinking what the start state of the DFA will be.
The start state of the NFA includes the start state $q_0$ of the DFA and $q_3$ since you can get to $q_3$ from $q_0$ by an $\varepsilon$-transition.
\begin{itemize}
\item \texttt{start} \rightarrow q_0
\item q_0, q_3 \rightarrow \{q_0, q_3\}
\item a, b \rightarrow \{q_0, q_3\}
\end{itemize}
\( \{q_0, q_3\} \)
\[
\begin{align*}
\text{start} & \quad q_0 & \quad a, b & \quad q_1 & \quad a \quad b\
\varepsilon & \quad q_2 & \quad a, b & \quad q_3 & \quad b\end{align*}
\]

\[\rightarrow \{q_0, q_3\}\]
\begin{align*}
\{q_0, q_3\} \rightarrow \{q_0, q_3\}
\end{align*}
\[
\begin{align*}
q_0 & \xrightarrow{a, b} q_2 & \xrightarrow{b} q_1 \\
q_0 & \xrightarrow{\epsilon} q_3 & \xrightarrow{a, b} q_4 \\
q_1 & \rightarrow \{q_0, q_3\} \\
q_4 & \rightarrow \{q_1, q_4\}
\end{align*}
\]
\[
\begin{align*}
\{q_0, q_3\} & \quad \rightarrow \quad \{q_1, q_4\} \\
\end{align*}
\]
\[
\begin{align*}
\text{start} & \rightarrow q_0 \\
q_0 & \rightarrow a \rightarrow q_1 \\
q_0 & \rightarrow \varepsilon \rightarrow q_3 \\
q_3 & \rightarrow a, b \rightarrow q_4 \\
q_2 & \rightarrow a, b \rightarrow q_0, q_3 \\
q_3 & \rightarrow b \rightarrow q_4 \\
q_1 & \rightarrow b \rightarrow q_2 \\
q_2 & \rightarrow a \rightarrow q_1 \\
q_1 & \rightarrow \varepsilon \rightarrow q_3 \\
q_4 & \rightarrow b \rightarrow q_3 \\
\end{align*}
\]
\[
\begin{aligned}
q_0 &\rightarrow \{q_0, q_3\} \\
q_1 &\rightarrow \{q_1, q_4\}
\end{aligned}
\]
a, b
\{q_0, q_3\} 

\{q_1, q_4\}
\[ \begin{align*}
& \text{start} \quad \rightarrow \quad \{q_0, q_3\} \\
\end{align*} \]
\[
\left\{ q_0, q_3 \right\} \quad \text{and} \quad \left\{ q_1, q_4 \right\}
\]
\[
\begin{align*}
    q_2 & \xrightarrow{\text{a, b}} q_0 \quad \xrightarrow{\text{b}} q_1 \\
    q_0 & \xrightarrow{\epsilon} q_3 \\
    q_3 & \xrightarrow{\text{a, b}} q_4 \\
    q_4 & \xrightarrow{\text{b}} q_3 \quad \xrightarrow{\epsilon} q_4
\end{align*}
\]

\[\begin{align*}
    \rightarrow & \{q_0, q_3\} \\
    a & \rightarrow \{q_1, q_4\} \\
    b & \rightarrow \{q_4\}\]
\[ \begin{align*}
q_0 & \quad \xrightarrow{a,b,a,b} \quad q_2 \\
q_0 & \quad \xrightarrow{\varepsilon} \quad q_3 \\
q_3 & \quad \xrightarrow{a,b} \quad q_4 \\
q_1 & \quad \xrightarrow{a,b} \quad q_0 \quad \xrightarrow{a,b} \quad q_1
\end{align*} \]

\[
\rightarrow \{q_0, q_3\} \quad \rightarrow \{q_1, q_4\} \quad \rightarrow \{q_4\}
\]
A non-deterministic finite automaton (NFA) with states $q_0, q_1, q_2, q_3, q_4$.

- $q_0$ is the start state.
- Transitions:
  - $q_0 \xrightarrow{a} q_1$ (accepting states: $\{q_0, q_3\}$)
  - $q_0 \xrightarrow{\varepsilon} q_4$ (accepting states: $\{q_1, q_4\}$)
  - $q_1 \xrightarrow{a} q_0$ (accepting states: $\{q_1, q_4\}$)
  - $q_1 \xrightarrow{b} q_4$ (accepting state: $\{q_4\}$)
  - $q_2 \xrightarrow{a, b} q_1$ (accepting states: $\{q_1, q_4\}$)
  - $q_2 \xrightarrow{b} q_4$ (accepting state: $\{q_4\}$)
  - $q_3 \xrightarrow{a, b} q_4$ (accepting state: $\{q_4\}$)

The NFA accepts strings over the alphabet $\{a, b\}$.
\[
\begin{align*}
q_0 &\rightarrow \{q_0, q_3\} \\
q_1 &\rightarrow \{q_1, q_4\} \\
q_2 &\rightarrow \{q_2\} \\
q_3 &\rightarrow \{q_3\} \\
q_4 &\rightarrow \{q_4\}
\end{align*}
\]
\begin{align*}
\{q_0, q_3\} & \rightarrow \{q_1, q_4\} \\
\{q_1, q_4\} & \rightarrow \{q_4\}
\end{align*}
\[
\begin{align*}
\begin{array}{ccc}
\{q_0, q_3\} & \rightarrow & \{q_1, q_4\} \\
\{q_1, q_4\} & \rightarrow & \{q_4\}
\end{array}
\end{align*}
\]
\[
\begin{align*}
\varepsilon & \rightarrow \{q_0, q_3\} \\
\{q_1, q_4\} & \rightarrow \varnothing
\end{align*}
\]

\[
\begin{array}{c}
a \rightarrow \{q_1, q_4\} \\
\{q_4\} \rightarrow b
\end{array}
\]
Several steps later…
\[
\begin{align*}
q_0 &\rightarrow \{q_0, q_3\} \\
q_1 &\rightarrow \{q_1, q_4\} \\
q_2 &\rightarrow \{q_4\} \\
q_3 &\rightarrow \{q_2, q_3\} \\
q_4 &\rightarrow \{q_3\}
\end{align*}
\]
Transition table:

<table>
<thead>
<tr>
<th>Input</th>
<th>States</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>{q0, q3}, {q1, q4}, {q4}</td>
</tr>
<tr>
<td>b</td>
<td>{q2, q3}</td>
</tr>
<tr>
<td>ε</td>
<td>{q2, q3}</td>
</tr>
</tbody>
</table>

Diagram:

- Start state: q0
- Final states: q2, q4

Transitions:
- q0 -> q1: a, b
- q1 -> q0: a
- q1 -> q4: b
- q2 -> q3: a, b
- q3 -> q4: b

\[
\begin{align*}
q_0 &\rightarrow \{q_0, q_3\} \\
q_1 &\rightarrow \{q_1, q_4\} \\
&\quad \rightarrow \{q_4\} \\
&\quad \rightarrow \emptyset \\
&\quad \rightarrow \emptyset \\
q_3 &\rightarrow \{q_2, q_3\} \\
q_4 &\rightarrow \{q_3\}
\end{align*}
\]
\[
\begin{align*}
\{q_0, q_3\} &
\rightarrow \{q_1, q_4\} \\
\{q_1, q_4\} &
\rightarrow \{q_2, q_3\}
\end{align*}
\]

**Diagram:**
- Start state: \(q_0\)
- States: \(q_0, q_1, q_2, q_3, q_4\)
- Transitions:
  - \(q_0 \xrightarrow{a} q_1\)
  - \(q_0 \xrightarrow{b} q_2\)
  - \(q_1 \xrightarrow{a} q_2\)
  - \(q_1 \xrightarrow{b} q_3\)
  - \(q_2 \xrightarrow{a} q_4\)
  - \(q_2 \xrightarrow{b} q_3\)
  - \(q_3 \xrightarrow{a} q_4\)
  - \(q_3 \xrightarrow{b} q_3\)
  - \(q_4 \xrightarrow{a} q_4\)
  - \(q_4 \xrightarrow{b} q_4\)

**Transitions:**
- \(q_0 \rightarrow \{q_1, q_4\}\)
- \(q_1 \rightarrow \{q_2, q_3\}\)
- \(q_2 \rightarrow \{q_0, q_3\}\)
- \(q_3 \rightarrow \{q_2, q_3\}\)
- \(q_4 \rightarrow \{q_3\}\)
Several steps later…
What row is missing?
\[ q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a, b} q_0 \]

\[ q_3 \xrightarrow{a, b} q_4 \]

\[ q_0, q_3 \]
\[ q_1, q_4 \]
\[ q_3 \]
\[ q_0, q_3, q_4 \]
\[ \emptyset \]

\[ q_1 \xrightarrow{b} q_4 \]
\[ q_3 \xrightarrow{b} q_4 \]
\[ q_0, q_3, q_4 \]
\[ q_0, q_3, q_4 \]
\[ q_1, q_4 \]
\[ q_1, q_4 \]
\[ q_4 \]
\[ q_4 \]

\[ a \]
\[ \{q_1, q_4\} \]
\[ \emptyset \]
\[ \emptyset \]
\[ \{q_0, q_3, q_4\} \]
\[ \{q_0, q_3, q_4\} \]

\[ b \]
\[ \{q_4\} \]
\[ \{q_2, q_3\} \]
\[ \{q_3\} \]
\[ \{q_0, q_3, q_4\} \]
\[ \{q_0, q_3, q_4\} \]
\[ \{q_1, q_4\} \]
\[ \{q_3, q_4\} \]
\[ q_0 \xrightarrow{a} q_1 \quad q_0 \xrightarrow{\varepsilon} q_3 \quad q_0 \xrightarrow{a, b} q_3 \quad q_0 \xrightarrow{a, b} q_4 \quad q_1 \xrightarrow{b} q_2 \quad q_2 \xrightarrow{b} q_1 \quad q_3 \xrightarrow{a, b} q_3 \quad q_3 \xrightarrow{a, b} q_4 \quad q_3 \xrightarrow{\varepsilon} q_4 \]

\[
\begin{align*}
\{q_0, q_3\} & \quad \{q_0, q_3, q_4\} \\
\{q_1, q_4\} & \quad \{q_1, q_4\} \\
\{q_1, q_4\} & \quad \{q_1, q_4\} \\
\{q_2, q_3\} & \quad \{q_2, q_3\} \\
\{q_3\} & \quad \{q_3\} \\
\{q_3\} & \quad \{q_3, q_4\} \\
\{q_4\} & \quad \{q_4\} \\
\{q_4\} & \quad \{q_3, q_4\}
\end{align*}
\]
\[
\begin{align*}
q_0 &\xrightarrow{a} q_1 & \quad q_0 &\xrightarrow{b} q_2 \\
q_0 &\xrightarrow{\varepsilon} q_3 & \quad q_3 &\xrightarrow{a, b} q_4 \\
\end{align*}
\]

**Transitions:**
- \(q_0\) on \(a\) to \(q_1\)
- \(q_0\) on \(b\) to \(q_2\)
- \(q_0\) on \(\varepsilon\) to \(q_3\)
- \(q_3\) on \(a, b\) to \(q_4\)

**Final States:**
- \(q_0, q_3\)
- \(q_1, q_4\)
- \(q_4\)

**Acceptance Conditions:**
- \(a\): \([q_1, q_4]\) \(\emptyset\) \(\emptyset\)
- \(b\): \([q_4]\) \([q_2, q_3]\) \([q_3]\) \([q_0, q_3, q_4]\) \([q_0, q_3, q_4]\) \([q_4]\) \([q_4]\) \([q_3, q_4]\) \([q_3, q_4]\) \(\emptyset\) \(\emptyset\)
Creating a DFA from an NFA with ε-transitions

1. Compute the **ε-closure** for each state, i.e., the set of states reachable from that state following only ε-transitions.

2. The start state is the ε-closure of $q_0$, i.e., $E(\{q_0\})$.

3. Define $\delta$ for each $\alpha \in \Sigma$ and each ε-closed set $S$:

   - If a state $p \in S$ can reach state $q$ on input $\alpha$ (not ε!), then add a transition on input $\alpha$ from $S$ to $E(q)$.

4. The set of accept states for the DFA now includes those sets that contain at least one accept state of the NFA.
Exercise

Convert this NFA to a DFA.
Step 1

\[ E(q_0) = \{q_0, q_1, q_6\} \]

\[ E(q_1) = \{q_1\} \]

\[ E(q_2) = \{q_2, q_3\} \]

\[ E(q_3) = \{q_3\} \]

\[ E(q_4) = \{q_4, q_5\} \]

\[ E(q_5) = \{q_5\} \]

\[ E(q_6) = \{q_6\} \]

\[ E(q_7) = \{q_7, q_5\} \]
Step 1

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]
\[ E(\{q_1\}) = \{q_1\} \]
\[ E(\{q_2\}) = \{q_2, q_3\} \]
\[ E(\{q_3\}) = \{q_3\} \]
\[ E(\{q_4\}) = \{q_4, q_5\} \]
\[ E(\{q_5\}) = \{q_5\} \]
\[ E(\{q_6\}) = \{q_6\} \]
\[ E(\{q_7\}) = \{q_7, q_5\} \]

Step 2

Start state

\[ \longrightarrow \{q_0, q_1, q_6\} \]

\[ \longrightarrow \{q_0, q_1, q_6\} \]
Step 1

<table>
<thead>
<tr>
<th>State</th>
<th>Transition</th>
<th>Next States</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td></td>
<td>$q_1, q_6$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$a$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$\varepsilon$</td>
<td>$q_3$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$a$</td>
<td>$q_4$</td>
</tr>
<tr>
<td>$q_4$</td>
<td>$\varepsilon$</td>
<td>$q_5$</td>
</tr>
<tr>
<td>$q_5$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_6$</td>
<td>$\varepsilon$</td>
<td>$q_7$</td>
</tr>
<tr>
<td>$q_7$</td>
<td>$a$</td>
<td>$q_0$</td>
</tr>
</tbody>
</table>

Step 2

- **Start state**: $q_0, q_1, q_6$
- **Compute transitions, using $\varepsilon$-closure $E$.**

Step 3

- **Compute transitions, using $\varepsilon$-closure $E$.**
- **Input**: $a$
- **Next states**: $q_0, q_1, q_6$
Step 1

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]
\[ E(\{q_1\}) = \{q_1\} \]
\[ E(\{q_2\}) = \{q_2, q_3\} \]
\[ E(\{q_3\}) = \{q_3\} \]
\[ E(\{q_4\}) = \{q_4, q_5\} \]
\[ E(\{q_5\}) = \{q_5\} \]
\[ E(\{q_6\}) = \{q_6\} \]
\[ E(\{q_7\}) = \{q_7, q_5\} \]

Step 2

Start state

Step 3

Compute transitions, using \( \varepsilon \)-closure \( E \).

\[ \rightarrow \{q_0, q_1, q_6\} \]
\[ \{q_2, q_3, q_7, q_5\} \]
Step 1

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]
\[ E(\{q_1\}) = \{q_1\} \]
\[ E(\{q_2\}) = \{q_2, q_3\} \]
\[ E(\{q_3\}) = \{q_3\} \]
\[ E(\{q_4\}) = \{q_4, q_5\} \]
\[ E(\{q_5\}) = \{q_5\} \]
\[ E(\{q_6\}) = \{q_6\} \]
\[ E(\{q_7\}) = \{q_7, q_5\} \]

Step 2

\[ \text{Start state} \]

Step 3

\[ \text{Compute transitions, using } \varepsilon\text{-closure } E. \]
Step 1

\[
E(\{q_0\}) = \{q_0, q_1, q_6\} \\
E(\{q_1\}) = \{q_1\} \\
E(\{q_2\}) = \{q_2, q_3\} \\
E(\{q_3\}) = \{q_3\} \\
E(\{q_4\}) = \{q_4, q_5\} \\
E(\{q_5\}) = \{q_5\} \\
E(\{q_6\}) = \{q_6\} \\
E(\{q_7\}) = \{q_7, q_5\}
\]

Step 2

Start state

Step 3

Compute transitions, using \( \varepsilon \)-closure \( E \).
Step 1

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]
\[ E(\{q_1\}) = \{q_1\} \]
\[ E(\{q_2\}) = \{q_2, q_3\} \]
\[ E(\{q_3\}) = \{q_3\} \]
\[ E(\{q_4\}) = \{q_4, q_5\} \]
\[ E(\{q_5\}) = \{q_5\} \]
\[ E(\{q_6\}) = \{q_6\} \]
\[ E(\{q_7\}) = \{q_7, q_5\} \]

Step 2

Start state

Step 3

Compute transitions, using \(\epsilon\)-closure \(E\).
Step 1

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]
\[ E(\{q_1\}) = \{q_1\} \]
\[ E(\{q_2\}) = \{q_2, q_3\} \]
\[ E(\{q_3\}) = \{q_3\} \]
\[ E(\{q_4\}) = \{q_4, q_5\} \]
\[ E(\{q_5\}) = \{q_5\} \]
\[ E(\{q_6\}) = \{q_6\} \]
\[ E(\{q_7\}) = \{q_7, q_5\} \]

Step 2

Start state

Step 3

Compute transitions, using \( \varepsilon \)-closure \( E \).

\[ \{q_0, q_1, q_6\} \]
\[ \{q_2, q_3, q_7, q_5\} \]
\[ \{q_4, q_5\} \]
Step 1

\[
E(\{q_0\}) = \{q_0, q_1, q_6\} \\
E(\{q_1\}) = \{q_1\} \\
E(\{q_2\}) = \{q_2, q_3\} \\
E(\{q_3\}) = \{q_3\} \\
E(\{q_4\}) = \{q_4, q_5\} \\
E(\{q_5\}) = \{q_5\} \\
E(\{q_6\}) = \{q_6\} \\
E(\{q_7\}) = \{q_7, q_5\}
\]

Step 2

Start state

Step 3

Compute transitions, using \(\varepsilon\)-closure \(E\).

\[
\{q_0, q_1, q_6\} \rightarrow \{q_0, q_1, q_6\} \\
\{q_2, q_3, q_7, q_5\} \rightarrow \{q_2, q_3, q_7, q_5\} \\
\{q_4, q_5\} \rightarrow \{q_4, q_5\} \\
\emptyset
\]
**Step 1**

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]

\[ E(\{q_1\}) = \{q_1\} \]

\[ E(\{q_2\}) = \{q_2, q_3\} \]

\[ E(\{q_3\}) = \{q_3\} \]

\[ E(\{q_4\}) = \{q_4, q_5\} \]

\[ E(\{q_5\}) = \{q_5\} \]

\[ E(\{q_6\}) = \{q_6\} \]

\[ E(\{q_7\}) = \{q_7, q_5\} \]

**Step 2**

Start state

**Step 3**

Compute transitions, using \(\varepsilon\)-closure \(E\).
Step 1

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]
\[ E(\{q_1\}) = \{q_1\} \]
\[ E(\{q_2\}) = \{q_2, q_3\} \]
\[ E(\{q_3\}) = \{q_3\} \]
\[ E(\{q_4\}) = \{q_4, q_5\} \]
\[ E(\{q_5\}) = \{q_5\} \]
\[ E(\{q_6\}) = \{q_6\} \]
\[ E(\{q_7\}) = \{q_7, q_5\} \]

Step 2

Start state

Step 3

Compute transitions, using \( \epsilon \)-closure \( E \).

Step 4

Accept states
Step 1

\[ E(\{q_0\}) = \{q_0, q_1, q_6\} \]
\[ E(\{q_1\}) = \{q_1\} \]
\[ E(\{q_2\}) = \{q_2, q_3\} \]
\[ E(\{q_3\}) = \{q_3\} \]
\[ E(\{q_4\}) = \{q_4, q_5\} \]
\[ E(\{q_5\}) = \{q_5\} \]
\[ E(\{q_6\}) = \{q_6\} \]
\[ E(\{q_7\}) = \{q_7, q_5\} \]

Step 2

Start state

Step 3

Compute transitions, using \(\varepsilon\)-closure \(E\).

Step 4

Final states

\[ a \]

\[ \{q_2, q_3, q_7, q_5\} \]

\[ \{q_4, q_5\} \]

\[ \emptyset \]

\[ \emptyset \]
Step 1

$E(\{q_0\}) = \{q_0, q_1, q_6\}$
$E(\{q_1\}) = \{q_1\}$
$E(\{q_2\}) = \{q_2, q_3\}$
$E(\{q_3\}) = \{q_3\}$
$E(\{q_4\}) = \{q_4, q_5\}$
$E(\{q_5\}) = \{q_5\}$
$E(\{q_6\}) = \{q_6\}$
$E(\{q_7\}) = \{q_7, q_5\}$

Step 2

Start $= E(\{q_0\}) = \{q_0, q_1, q_6\}$

Step 3

$\delta(\{q_0, q_1, q_6\}, a) = E(\{q_2, q_7\}) = \{q_2, q_3, q_7, q_5\}$
$\delta(\{q_2, q_3, q_7, q_5\}, a) = E(\{q_4\}) = \{q_4, q_5\}$
$\delta(\{q_4, q_5\}, a) = \emptyset$
$\delta(\emptyset) = \emptyset$

Step 4

Final $= \{\{q_2, q_3, q_7, q_5\}, \{q_4, q_5\}\}$

This is the same process shown without using a table.
This method of transforming an NFA into a DFA is called the \textit{subset construction}.

Each state in the DFA corresponds to a set of states in the NFA.

The start state in the DFA corresponds to the start state of the NFA, \textit{plus all states reachable via }\varepsilon\textit{-transitions}.

The accept states in the DFA correspond to the sets of states that would be considered to accept in the NFA when using the massive parallel intuition.

If a state $q$ in the DFA corresponds to a set of states $S$ in the NFA, then the transition from state $q$ on a character $\alpha$ is found as follows:

Let $S'$ be the set of states in the NFA that can be reached by following a transition labeled $\alpha$ from any of the states in $S$.

\textit{Let }$S''$\textit{ be the set of states in the NFA reachable from some state in }$S'$\textit{ by following zero or more }$\varepsilon$\textit{-transitions}.

The state $q$ in the DFA transitions on $\alpha$ to a DFA state corresponding to the set of states $S''$. 
Wrap-up
A language is called a *regular language* if there exists a DFA $D$ such that $L(D) = L$. 
THEOREM  A language $L$ is regular iff there is some NFA $N$ such that $L(N) = L$.

PROOF SKETCH  If $L$ is regular, there exists some DFA for it, which we can easily convert into an NFA.

If $L$ is accepted by some NFA, we can use the subset construction to convert it into a DFA that accepts the same language, so $L$ is regular.
We now have two perspectives on regular languages:

- Regular languages are languages recognized by DFAs.
- Regular languages are languages recognized by NFAs.

We can now reason about regular languages in two different ways, and we can use whichever model is more convenient.
Acknowledgments

This lecture incorporates material from:

Nancy Ide, Vassar College
Keith Schwarz, Stanford University
Michael Sipser, *Introduction to the Theory of Computation*
Jeffrey Ullman, Stanford University