Relating DFAs and NFAs

12 September 2023
Assignment 2

Due today

Corrections due Thursday
Where are we?
We can describe a DFA with a state transition diagram,
We can describe a DFA with a state transition diagram,

or, equivalently,

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_0$</td>
<td>$q_0$</td>
<td>$q_1$</td>
</tr>
<tr>
<td>$q_1$</td>
<td>$q_0$</td>
<td>$q_2$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$q_2$</td>
<td>$q_2$</td>
</tr>
</tbody>
</table>
More formally, a DFA is represented as a five-tuple \((Q, \Sigma, \delta, q_0, F)\) where

- \(Q\) is a finite set of states,
- \(\Sigma\) is the alphabet, a finite set of input symbols,
- \(\delta: Q \times \Sigma \rightarrow Q\) is the transition function,
- \(q_0 \in Q\) is the start state, and
- \(F \subseteq Q\) is a set of zero or more accept states.
If $D$ is a DFA, the language of $D$, denoted $L(D)$, is

$$\{w \in \Sigma^* \mid D \text{ accepts } w\}.$$  

A language is a regular language if there exists a DFA $D$ such that $L(D) = L$. 
A **nondeterministic finite automaton** (NFA) can is like a DFA, but it can have missing transitions or multiple transitions defined on the same input symbol.

An NFA accepts if *any possible series of choices* leads to an accept state.
NFAs can also 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.
Formally, an NFA is defined like a DFA:

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

Except instead of

\[ \delta: Q \times \Sigma \rightarrow Q, \]

we have

\[ \delta: Q \times (\Sigma \cup \{\epsilon\}) \rightarrow \wp(Q). \]
An NFA can be thought of as a DFA that can be in many states at once.

At each point in time, when the NFA needs to follow a transition, it tries all the options at the same time.

The NFA accepts if *any* of the states that are active at the end are accept states. Otherwise, it rejects.
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?
a, b

start

q0 → a → q1 → b → q2 → a → q3

a b a b a
a, b

start

q0

a

q1

b

q2

a

q3

a b a b a a
The given automaton starts at state $q_0$ and transitions through states $q_1$, $q_2$, and $q_3$ based on input symbols:

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

The automaton accepts the string 'a b a b a a'.
a, b

start

q_0 → a → q_1 → b → q_2 → a → q_3

a b a b a
a, b

\begin{align*}
\text{start} & \rightarrow q_0 \\
q_0 & \xrightarrow{a} q_1 \\
q_1 & \xrightarrow{b} q_2 \\
q_2 & \xrightarrow{a} q_3
\end{align*}

\rightarrow \{q_0\}
\[
q_0 \xrightarrow{a, b} q_0 \rightarrow q_1 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_3
\]

\[
\rightarrow \{q_0\}
\]
\[
\begin{align*}
\rightarrow \{q_0\}
\end{align*}
\]
\[
\begin{align*}
&\text{start} \\
&\rightarrow \{q_0\} \\
\end{align*}
\]
A finite automaton with the following states and transitions:

- States: $q_0, q_1, q_2, q_3$
- Transitions:
  - From $q_0$ to $q_1$ on input $a$
  - From $q_0$ to $q_0$ on input $a$, $b$ (loop)
  - From $q_1$ to $q_2$ on input $b$
  - From $q_2$ to $q_3$ on input $a$

Initial state: $q_0$
Final state: $q_3$

Input alphabets:
- $a$
- $b$

Transition functions:
- $a$: $\{q_0\}$
- $b$: $\{q_0, q_1\}$
\[ q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_3 \]

Transition:

- From \( q_0 \):
  - \( a \): \( \{q_0\} \)
  - \( b \): \( \{q_0, q_1\} \)

- From \( q_1 \):
  - \( a \): \( \{q_0, q_1\} \)

- From \( q_2 \):
  - \( a \): \( \{q_0\} \)

- From \( q_3 \):
  - \( \text{accepting state} \)
The given automaton has the following transitions:

- From state $q_0$, on reading 'a', move to state $q_1$.
- From state $q_1$, on reading 'b', move to state $q_2$.
- From state $q_2$, on reading 'a', move to state $q_3$.
- The initial state is $q_0$.

The set of states for each transition are:

- $\{q_0\}$ for 'a' from $q_0$.
- $\{q_0, q_1\}$ for 'b' from $q_0$.
- $\{q_0\}$ for 'a' from $q_2$.
- $\{q_0, q_1\}$ for 'b' from $q_2$.
- $\{q_0\}$ for 'b' from $q_3$.
\[
\begin{align*}
\text{start} & \quad q_0 \\
q_0 & \xrightarrow{a} q_1 \\
q_1 & \xrightarrow{b} q_2 \\
q_2 & \xrightarrow{a} q_3 \\
q_0 & \xrightarrow{a} q_0 \\
q_1 & \xrightarrow{b} q_1 \\
\end{align*}
\]
A deterministic finite automaton (DFA) with the following states and transitions:

- States: \{q_0, q_1, q_2, q_3\}
- Transitions:
  - From \(q_0\) on input 'a' to \(q_1\)
  - From \(q_1\) on input 'b' to \(q_2\)
  - From \(q_2\) on input 'a' to \(q_3\)
  - On any input from \(q_3\), it loops back to itself

Initial state: \(q_0\)
Accepting states: \(\{q_0, q_1\}\)
\[
\begin{array}{c}
a, b \\
\end{array}
\]

\[
\begin{array}{c}
\text{start} \\
\end{array}
\]

\[
\begin{array}{c}
q_0 \\
\end{array}
\]

\[
\begin{array}{c}
q_1 \\
\end{array}
\]

\[
\begin{array}{c}
q_2 \\
\end{array}
\]

\[
\begin{array}{c}
q_3 \\
\end{array}
\]

\[
\begin{array}{c}
a \\
\end{array}
\]

\[
\begin{array}{c}
b \\
\end{array}
\]

\[
\begin{array}{c}
\{q_0\} \\
\{q_0, q_1\} \\
\{q_0, q_1\} \\
\{q_0\} \\
\{q_0, q_1\} \\
\{q_0, q_1\} \\
\{q_0\} \\
\{q_0, q_1\} \\
\end{array}
\]
A finite automaton with the following states and transitions:

- **States:** q0, q1, q2, q3
- **Start State:** q0
- **Accept State:** q3
- **Transitions:**
  - a: q0 → q1
  - b: q1 → q2
  - a: q2 → q3

The transitions are labeled with the input symbol and the set of states reached:

- a: \{q0\}
- b: \{q0, q1\}
- a: \{q0, q1\}
- b: \{q0\}
- a: \{q0\}
- b: \{q0\}
The diagram represents a finite automaton with the following states:

- **q₀** (start state)
- **q₁**
- **q₂**
- **q₃** (accept state)

The transitions are labeled with input symbols:

- **a**: Transition from **q₀** to **q₁**
- **b**: Transition from **q₁** to **q₂**
- **a** (dashed line): Transition from **q₂** to **q₃**

The corresponding state sets for each transition are:

- **a**: \(\{q₀\} \rightarrow \{q₀, q₁\}\)
- **b**: \(\{q₁\} \rightarrow \{q₀, q₁\}\)
- **a**: \(\{q₀\} \rightarrow \{q₀\}\)

The automaton starts in state **q₀** and can transit to **q₁** upon input **a**, to **q₂** upon input **b**, and to **q₃** upon input **a**.
A deterministic finite automaton (DFA) with states $q_0$, $q_1$, $q_2$, $q_3$.

- Start state: $q_0$
- Transitions:
  - $a$: $q_0 \rightarrow q_1$
  - $b$: $q_1 \rightarrow q_2$
  - $a$: $q_2 \rightarrow q_3$

**Transitions from each state for inputs $a$ and $b$:**

- $a$:
  - $q_0 \rightarrow \{q_0\}$
  - $q_0, q_1 \rightarrow \{q_0, q_1\}$
  - $q_0, q_2 \rightarrow \{q_0, q_2\}$

- $b$:
  - $q_0, q_1 \rightarrow \{q_0, q_1\}$
  - $q_0, q_2 \rightarrow \{q_0, q_2\}$
\begin{itemize}
  \item \textbf{a}: \{q_0\} \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_2\}
  \item \textbf{b}: \{q_0\} \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_2\}
\end{itemize}
A finite automaton with the following states and transitions:

- **States:** q0, q1, q2, q3
- **Start State:** q0
- **Accepting States:** q3

**Transitions:**
- From q0:
  - on input a: transition to q1
  - on input b: transition to q2
- From q1:
  - on input b: transition to q2
- From q2:
  - on input a: transition to q3
- From q3:
  - on any input: loop back to q3

**Transition Table:**

<table>
<thead>
<tr>
<th>State</th>
<th>a</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>q0</td>
<td>{q0}</td>
<td>{q0, q1}</td>
</tr>
<tr>
<td>q0, q1</td>
<td>{q0, q1}</td>
<td>{q0, q1}</td>
</tr>
<tr>
<td>q0, q2</td>
<td>{q0, q2}</td>
<td>{q0, q2}</td>
</tr>
</tbody>
</table>
The diagram represents a deterministic finite automaton (DFA) with the following transitions:

- Start state: \( q_0 \)

Transitions:

- From \( q_0 \):
  - On input 'a', go to \( q_1 \)
  - On input 'b', go to \( q_2 \)
- From \( q_1 \):
  - On input 'a', go to a loop back to \( q_0 \)
  - On input 'b', go to \( q_2 \)
- From \( q_2 \):
  - On input 'a', go to \( q_3 \)

States:

- \( q_0 \)
- \( q_1 \)
- \( q_2 \)
- \( q_3 \)
The given diagram represents a deterministic finite automaton (DFA) with the following states and transitions:

- **States**: $q_0, q_1, q_2, q_3$
- **Alphabet**: $\{a, b\}$
- **Transition Table**:
  - $(q_0, a) \rightarrow \{q_0, q_1\}$
  - $(q_0, b) \rightarrow \{q_0\}$
  - $(q_1, a) \rightarrow \{q_0\}$
  - $(q_1, b) \rightarrow \{q_0, q_1\}$
  - $(q_2, a) \rightarrow \{q_0, q_2\}$
  - $(q_3, a) \rightarrow \{q_3\}$

The start state is $q_0$, and the accepting state is $q_3$. The transitions are shown with arrows, and the accepting state is indicated by a double circle.
The diagram represents a finite automaton with the following transitions:

- **Start state**: $q_0$
- **Transitions**:
  - **a**:
    - From $q_0$ to $q_1$: $\{q_0, q_1\}$
    - From $q_0$ to $q_3$: $\{q_0\}$
    - From $q_1$ to $q_2$: $\{q_0, q_1\}$
    - From $q_1$ to $q_3$: $\{q_0, q_2\}$
  - **b**:
    - From $q_0$ to $q_1$: $\{q_0, q_1\}$
    - From $q_1$ to $q_2$: $\{q_0, q_1\}$
    - From $q_2$ to $q_1$: $\{q_0, q_1\}$
    - From $q_2$ to $q_3$: $\{q_0, q_2\}$

The accepting state is $q_3$. The diagram also shows a loop from $q_3$ back to $q_0$ labeled with $a, b$. Two configurations of the automaton are shown:

- $\{q_0\}$
- $\{q_0, q_1\}$
- $\{q_0, q_2\}$
The given automaton has the following transitions:

- **Start state:** $q_0$
- **Transitions:**
  - $a$: $q_0 \rightarrow q_1$
  - $b$: $q_1 \rightarrow q_2$
  - $a$: $q_2 \rightarrow q_3$

The corresponding state sets are:

- $a$: $\{q_0\} \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_1, q_3\}$
- $b$: $\{q_0, q_1\} \rightarrow \{q_0, q_1\} \rightarrow \{q_0, q_2\}$
a, b

start

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

\[
\begin{array}{c}
\{q_0\} \\
\{q_0, q_1\} \\
\{q_0, q_1\} \\
\{q_0, q_2\}
\end{array}
\]
\[
\begin{align*}
\text{a, b} & \quad \text{a} & \quad \text{b} \\
\text{start} & \rightarrow \{q_0\} & \{q_0, q_1\} & \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0, q_1\} & \{q_0, q_1\} & \{q_0, q_2\} \\
\{q_0, q_2\} & \rightarrow \{q_0, q_1, q_3\}
\end{align*}
\]
\[
\begin{align*}
\{q_0\} & \rightarrow \{q_0, q_1\} & \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0, q_1\} & \{q_0, q_2\} \\
\{q_0, q_2\} & \rightarrow \{q_0, q_1, q_3\}
\end{align*}
\]
\[
\begin{align*}
q_0 & \rightarrow \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0, q_1\} \quad \{q_0, q_2\} \\
\{q_0, q_2\} & \rightarrow \{q_0, q_1, q_3\}
\end{align*}
\]
\[
\begin{align*}
\text{a} & \quad \text{b} \\
\{q_0\} & \quad \{q_0, q_1\} & \quad \{q_0\} \\
\{q_0, q_1\} & \quad \{q_0, q_1\} & \quad \{q_0, q_2\} \\
\{q_0, q_2\} & \quad \{q_0, q_1, q_3\}
\end{align*}
\]
\[
\begin{array}{c|cc|c}
\text{a} & q_0 & q_1 & q_2 & a, b \\
\{q_0\} & \{q_0, q_1\} & \{q_0\} \\
\{q_0, q_1\} & \{q_0, q_1\} & \{q_0, q_2\} \\
\{q_0, q_2\} & \{q_0, q_1, q_3\} & \{q_0\} \\
\end{array}
\]
\[ q_0 \rightarrow \{q_0\} \]
\[ \{q_0, q_1\} \rightarrow \{q_0, q_1\} \]
\[ \{q_0, q_2\} \rightarrow \{q_0, q_1, q_3\} \]
\[ \{q_0, q_1, q_3\} \rightarrow \{q_0\} \]
\[ a \rightarrow \{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_0, q_1, q_3\} \]

\[ b \rightarrow \{q_0\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_0\} \]
\[
\begin{array}{c}
\text{a, b} \\
\text{start} \\
\end{array}
\]

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

Transitions:
- From $q_0$ on $a$: $\{q_0\}$
- From $q_0$ on $b$: $\emptyset$
- From $q_1$ on $a$: $\emptyset$
- From $q_1$ on $b$: $\{q_0, q_1\}$
- From $q_2$ on $a$: $\{q_0\}$
- From $q_3$ on $a$: $\{q_3\}$

Symbols:
- $\rightarrow$ is a transition on input symbol $a$ or $b$.

Equations:
- For $a$: $\{q_0\} \rightarrow \{q_0\}$
- For $b$: $\emptyset \rightarrow \{q_0, q_1\}$
- For $a$ on $q_1$: $\emptyset \rightarrow \emptyset$
- For $b$ on $q_1$: $\{q_0, q_1\} \rightarrow \{q_0, q_2\}$
- For $a$ on $q_2$: $\{q_0\} \rightarrow \{q_0\}$
- For $a$ on $q_3$: $\{q_3\} \rightarrow \{q_3\}$

States:
- $q_0$
- $q_1$
- $q_2$
- $q_3$
\[
\begin{array}{c}
\{q_0\} \quad \{q_0, q_1\} \quad \{q_0\} \\
\{q_0, q_1\} \quad \{q_0, q_1\} \quad \{q_0, q_2\} \\
\{q_0, q_2\} \quad \{q_0, q_1, q_3\} \quad \{q_0\} \\
\{q_0, q_1, q_3\} \quad \{q_0, q_1\}
\end{array}
\]
- Transition:
  - \( a \rightarrow \{q_0\} \)
  - \( b \rightarrow \{q_0, q_1\} \)
  - \( a \rightarrow \{q_0, q_1\} \)
  - \( b \rightarrow \{q_0, q_2\} \)
  - \( a \rightarrow \{q_0, q_1, q_3\} \)
  - \( a \rightarrow \{q_0\} \)
  - \( a \rightarrow \{q_0, q_1, q_3\} \)
  - \( b \rightarrow \{q_0, q_1\} \)
  - \( a \rightarrow \{q_0\} \)
\[
\begin{align*}
\text{a, b} & \quad \rightarrow \quad \{q_0, q_1, q_3\} & \{q_0, q_1\} \\
\{q_0\} & \quad \rightarrow \quad \{q_0, q_1\} & \{q_0\} \\
\{q_0, q_1\} & \quad \rightarrow \quad \{q_0, q_1\} & \{q_0, q_2\} \\
\{q_0, q_2\} & \quad \rightarrow \quad \{q_0, q_1, q_3\} & \{q_0\} \\
\{q_0, q_1, q_3\} & \quad \rightarrow \quad \{q_0, q_1\} & \\
\end{align*}
\]
\[ \begin{array}{c|c|c}
\text{Input} & \text{Transition} & \text{States} \\
\hline
a & \{q_0\} & \{q_0\} \\
b & \{q_0, q_1\} & \{q_0, q_1\} \\
a & \{q_0, q_1\} & \{q_0, q_1\} \\
b & \{q_0, q_2\} & \{q_0, q_2\} \\
a & \{q_0, q_3\} & \{q_0, q_3\} \\
b & \{q_0, q_1, q_3\} & \{q_0, q_1, q_3\} \\
\end{array} \]
\[
\begin{align*}
\text{a} & \rightarrow \{q_0\} \quad \{q_0, q_1\} \quad \{q_0\} \\
\{q_0, q_1\} & \rightarrow \{q_0, q_1\} \quad \{q_0, q_2\} \\
\{q_0, q_2\} & \rightarrow \{q_0, q_1, q_3\} \quad \{q_0\} \\
\{q_0, q_1, q_3\} & \rightarrow \{q_0, q_1\} \quad \{q_0, q_2\}
\end{align*}
\]
a, b

\[ q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_3 \]

\[ a \ b \ a \ b \ a \]

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

start

\[ q_0 \]

\[ q_1 \]

\[ q_2 \]

\[ q_3 \]
The diagram shows a finite state machine (FSM) with the following states and transitions:

- Start state: $q_0$
- States: $q_0, q_1, q_2, q_3$
- Transitions:
  - $q_0 \xrightarrow{a} q_1$
  - $q_1 \xrightarrow{b} q_2$
  - $q_2 \xrightarrow{a} q_3$
- Final state: $q_3$

The input sequence is: $a b a a b a$

The corresponding automaton is shown with transitions between states and inputs $a$ and $b$. The start state is $q_0$, and the sequence of states the automaton goes through is $q_0, q_1, q_2, q_3$.
This method of transforming an NFA for a language \(L\) into a DFA for \(L\) 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 \(a\) is found as follows:

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

The state \(q\) in the DFA transitions on \(a\) to a DFA state corresponding to the set of states \(S'\).
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 *exponentially larger* than the original NFA.
A language $L$ is called a *regular language* if there exists a DFA $D$ such that $L(D) = L$. 
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 $A$:

1. $Q' = \wp(Q)$
2. $q_0' = \{q_0\}$
3. $F' = \{R \in Q' \mid R \text{ contains an accept state of } N\}$
4. For $R \in Q'$ and $a \in \Sigma$, we define $\delta'(R, a) = \bigcup_{r \in R} \delta(r, a)$

Every state $R$ of $M$ is a set of states of $N$.

When $M$ is in state $R$ and reads a symbol $a$, it tracks where $N$ would go on $a$ each state in $R$.

As in Sipser, we’re using $R$ both as a state of $M$ and as a set of states of $N$. 
Wait... What about NFAs with $\epsilon$-transitions?
It’ll be easier for us to represent the equivalent DFA as a table rather than a (big, messy) diagram.
Let’s start off by thinking what the start state of our DFA is going to be.
Let’s start off by thinking what the start state of our DFA is going to be.
Let’s start off by thinking what the start state of our DFA is going to 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.
\[
\begin{align*}
\text{start} & \rightarrow q_0 \\
q_0 & \xrightarrow{\varepsilon} q_3 \\
q_0 & \xrightarrow{a} q_1 \\
q_1 & \xrightarrow{\text{dotted line}} q_1 \\
q_2 & \xrightarrow{\Sigma} q_2 \\
q_3 & \xrightarrow{\Sigma} q_4 \\
q_3 & \xrightarrow{b} q_3 \\
q_4 & \rightarrow \{q_0, q_3\}
\end{align*}
\]
A finite automaton with transitions:

- Start state: $q_0$
- Transitions:
  - $q_0 \xrightarrow{a} q_1$
  - $q_0 \xrightarrow{\varepsilon} q_2$
  - $q_0 \xrightarrow{b} q_3$
  - $q_1 \xrightarrow{a} q_2$
  - $q_2 \xrightarrow{b} q_3$
  - $q_3 \xrightarrow{b} q_4$

States:
- $q_0$
- $q_1$
- $q_2$
- $q_3$
- $q_4$

Transitions on symbols $a$, $b$, and $\varepsilon$.

States $q_0$ and $q_3$ are accepting states.

Language accepted: $\{q_0, q_3\}$.
\[ \{q_0, q_3\} \]
\[ \rightarrow \{q_0, q_3\} \]
\[
\begin{align*}
\text{states:} & \quad \{q_0, q_3\} \\
\text{accept states:} & \quad \{q_1, q_4\}
\end{align*}
\]
\[ \begin{align*}
\{q_0, q_3\} & \rightarrow a \\
\{q_1, q_4\} & \rightarrow b
\end{align*} \]
\( \begin{align*}
\{q_0, q_3\} & \rightarrow \{q_0, q_3\} \\
\{q_1, q_4\} & \rightarrow \{q_1, q_4\}
\end{align*} \)
\( q_0 \xrightarrow{\varepsilon} q_3 \xrightarrow{\Sigma} \{q_0, q_3\} \)

\( q_0 \xrightarrow{a} q_1 \rightarrow \{q_0, q_3\} \)

\( q_0 \xrightarrow{b} q_2 \rightarrow \{q_1, q_4\} \)

\( \Sigma \rightarrow \{q_1, q_4\} \)
\[
\begin{align*}
\text{start} & \rightarrow q_0 \\
q_0 & \xrightarrow{a} q_1 \\
q_0 & \xrightarrow{\varepsilon} q_2 \\
q_0 & \xrightarrow{b} q_3 \\
q_2 & \xrightarrow{\Sigma} q_0 \cup \{q_3\} \\
q_2 & \xrightarrow{b} q_4 \\
q_1 & \xrightarrow{a} q_0 \cup \{q_3\} \\
q_3 & \xrightarrow{b} q_4 \\
q_4 & \xrightarrow{\Sigma} q_4 \\
\end{align*}
\]

\[
\begin{align*}
a \quad & \rightarrow \{q_0, q_3\} \\
b \quad & \rightarrow \{q_1, q_4\}
\end{align*}
\]
\( q_0 \rightarrow \{q_0, q_3\} \)

\( q_1 \rightarrow \{q_1, q_4\} \)

\( \Sigma \)

\( \epsilon \)

\( \Sigma \)

\( a \)

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

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

\[
\begin{align*}
a & \rightarrow \{q_1, q_4\} \\
b & \rightarrow \{q_4\}
\end{align*}
\]
A few steps later…
\[
\begin{align*}
\Sigma & \rightarrow \{q_0, q_3\} \\
\{q_1, q_4\} & \rightarrow \{q_4\} \\
\emptyset & \rightarrow \{q_2, q_3\} \\
\} & \rightarrow \{q_3\}
\end{align*}
\]
\[
\begin{align*}
\text{start} & \rightarrow q_0 \\
\varepsilon & \rightarrow q_3 \\
\Sigma & \rightarrow \{q_0, q_3\} \\
b & \rightarrow \{q_1, q_4\} \\
a & \rightarrow \{q_1, q_4\} \\
\emptyset & \rightarrow \{q_2, q_3\}
\end{align*}
\]
\[ \begin{align*}
q_0 & \rightarrow \{q_0, q_3\} \\
q_1 & \rightarrow \{q_1, q_4\} \cup \{q_4\} \cup \emptyset \\
q_2 & \rightarrow \{q_4\} \cup \emptyset \\
q_3 & \rightarrow \{q_2, q_3\} \\
q_4 & \rightarrow \{q_4\} \cup \{q_2, q_3\} \\
\end{align*} \]
A deterministic finite automaton (DFA) with the following transitions:

- From state $q_0$: On input $a$, go to state $q_1$, and on input $\varepsilon$, go to state $q_3$.

- From state $q_1$: On input $b$, go to state $q_4$.

The accepting states are $\{q_0, q_3\}$ for input $a$ and $\{q_1, q_4\}$ for input $b$.
\[
\begin{align*}
\delta(q_0, a) &= \{q_0, q_3\} \\
\delta(q_0, b) &= \{q_1, q_4\} \\
\delta(q_0, \varepsilon) &= \emptyset \\
\delta(q_1, a) &= \{q_1, q_4\} \\
\delta(q_1, b) &= \{q_2, q_3\} \\
\delta(q_2, a) &= \{q_1, q_4\} \\
\delta(q_2, b) &= \emptyset \\
\delta(q_3, a) &= \{q_4\} \\
\delta(q_4, a) &= \{q_2, q_3\} \\
\delta(q_4, b) &= \{q_3\}
\end{align*}
\]
\[
\begin{align*}
\{q_0, q_3\} & \quad \{q_1, q_4\} \\
\{q_1, q_4\} & \quad \{q_4\} \\
\emptyset & \quad \{q_2, q_3\} \\
\emptyset & \quad \{q_3\} \\
\{q_0, q_3, q_4\} & \quad \{q_0, q_3, q_4\}
\end{align*}
\]
A few steps later…
What row is missing?
\[
q_0 \xrightarrow{a} q_1 \xrightarrow{\epsilon} q_0 \xrightarrow{\epsilon} q_3 \xrightarrow{\Sigma} q_4 \xrightarrow{b} q_3 \xrightarrow{\Sigma} q_4 \\
\{q_0, q_3\} \xrightarrow{\Sigma} \{q_0, q_3, q_4\} \xrightarrow{\Sigma} \{q_3, q_4\} \\
\{q_1, q_4\} \xrightarrow{} \{q_1, q_4\} \\
\{q_2, q_3\} \xrightarrow{} \{q_2, q_3\} \\
\{q_4\} \xrightarrow{} \{q_4\} \\
\{q_0, q_3, q_4\} \xrightarrow{} \{q_0, q_3, q_4\} \\
\{q_1, q_4\} \xrightarrow{} \{q_3, q_4\} \\
\{q_4\} \xrightarrow{} \{q_3, q_4\} \\
\emptyset \xrightarrow{} \{q_2, q_3\} \\
\{q_3\} \xrightarrow{} \{q_3\} \\
\{q_4\} \xrightarrow{} \{q_4\} \\
\{q_0, q_3, q_4\} \xrightarrow{} \{q_0, q_3, q_4\} \\
\{q_1, q_4\} \xrightarrow{} \{q_3, q_4\} \\
\{q_4\} \xrightarrow{} \{q_3, q_4\} \\
\emptyset \xrightarrow{} \emptyset
\]
\begin{align*}
\begin{array}{c}
q_2 \\
\sum \\
q_0 \\
\varepsilon \\
q_3 \\
\sum \\
q_4
\end{array}
\end{align*}

\begin{array}{l}
\text{start} \\
a \\
\text{a} \\
\{q_0, q_3\} \\
\{q_1, q_4\} \\
\{q_2, q_3\} \\
\{q_3\} \\
\{q_0, q_3, q_4\} \\
\emptyset \\
\{q_1, q_4\} \\
\{q_3, q_4\} \\
\{q_4\} \\
\{q_0, q_3, q_4\} \\
\emptyset \\
\{q_3, q_4\} \\
\emptyset
\end{array}
\begin{array}{l}
b \\
\text{b} \\
\{q_1, q_4\} \\
\{q_2, q_3\} \\
\emptyset \\
\{q_3\} \\
\{q_0, q_3, q_4\} \\
\{q_0, q_3, q_4\} \\
\{q_4\} \\
\{q_3, q_4\} \\
\emptyset \\
\{q_3, q_4\} \\
\emptyset
\end{array}
Creating a DFA from an NFA with \(\varepsilon\)-transitions

1. Compute the \(\varepsilon\)-closure for each state, i.e., the set of states reachable from that state following only \(\varepsilon\)-transitions.

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

3. Define \(\delta\) for each \(a \in \Sigma\) and each \(\varepsilon\)-closed set \(S\):
   
   If a state \(p \in S\) can reach state \(q\) on input \(a\) (not \(\varepsilon\)), then add a transition on input \(a\) 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

$q_6$ moves towards $q_0, q_1, q_6$
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, \( E \).

\[ \rightarrow \{q_0, q_1, q_6\} \]
\[ \varepsilon \]
\[ q_0 \]
\[ q_1 \]
\[ q_2 \]
\[ q_3 \]
\[ q_4 \]
\[ q_5 \]
\[ q_6 \]
\[ q_7 \]
\[ \varepsilon \]
\[ a \]
\[ \varepsilon \]
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 \( E \).

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

Start state

Step 3

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

\[ a \]

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

Start state

Step 3

Compute transitions, using ε-closure E.

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

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

\[ \begin{align*}
\epsilon & \rightarrow \{q_0, q_1, q_6\} \\
\epsilon & \rightarrow \{q_2, q_3, q_7, q_5\} \\
a & \rightarrow \{q_0, q_1, q_6\} \\
a & \rightarrow \{q_2, q_3, q_7, q_5\} \\
a & \rightarrow \{q_4, q_5\}
\end{align*} \]
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 $E$.

$a$

$\{q_0, q_1, q_6\}$
$\{q_2, q_3, q_7, q_5\}$
$\{q_4, q_5\}$
$\emptyset$
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 ε-closure E.

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

Step 3

Compute transitions, using $\epsilon$-closure $E$.

Accept states

Step 4

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

$E(\{q_1\}) = \{q_1\}$

$E(\{q_2\}) = \{q_2, q_3\}$

$E(\{q_3\}) = \{q_3\}$

$E(\{q_4\}) = \{q_4, q_5\}$

$E(\{q_5\}) = \{q_5\}$

$E(\{q_6\}) = \{q_6\}$

$E(\{q_7\}) = \{q_7, q_5\}$

$\epsilon$-closure $E$ results in:

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

Step 3

Compute transitions, using \(\varepsilon\)-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: $E(\{q_0\}) = \{q_0, q_1, q_6\}$

### 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

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

This is the same process shown without using a table.
This method of transforming an NFA into a DFA is called the **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, *plus all states reachable via ε-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 \( S \).

*Let \( S'' \) be the set of states in the NFA reachable from some state in \( S' \) by following zero or more \( \varepsilon \)-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 if and only if 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 recognized by some NFA, we can use the subset construction to convert it into a DFA that recognizes 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