NFAs
NFAs

• An **NFA** is a
  • **N**ondeterministic
  • **F**inite
  • **A**utomaton

• Structurally similar to a DFA, but represents a fundamental shift in how we'll think about computation.
(Non)determinism

- A model of computation is **deterministic** if at every point in the computation, there is exactly one choice that can make.
- The machine accepts if that series of choices leads to an accepting state.
- A model of computation is **nondeterministic** if the computing machine may have multiple decisions that it can make at one point.
- The machine accepts if **any** series of choices leads to an accepting state.
  - (This sort of nondeterminism is technically called **existential nondeterminism**, the most philosophical-sounding term we’ll introduce all quarter.)
A Simple NFA

\[
\begin{align*}
q_0 \quad &\xrightarrow{1} q_1 \\
q_1 \quad &\xrightarrow{1} q_2 \\
q_2 \quad &\xrightarrow{0, 1} q_3 \\
q_3 \quad &\xrightarrow{0, 1} q_0
\end{align*}
\]
A Simple NFA

$q_0$ has two transitions defined on 1!
A Simple NFA

0 1 0 1 1
A Simple NFA

start

$q_0$  
1  
0, 1

$q_1$  
1

$q_2$

$q_3$

0

0, 1

0, 1

0, 1

0 1 0 1 1
A Simple NFA

start

$q_0$ 1 $q_1$

0, 1

$q_1$ 1 $q_2$

0, 1

$q_3$

$q_2$

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1

0, 1
A Simple NFA
A Simple NFA

\[ \begin{array}{c}
$q_0$ & 1 & $q_1$ & 1 & $q_2$ \\
0, 1 & 0 & 0, 1 & 0, 1
\end{array} \]
A Simple NFA

start

$q_0$ → $q_1$ on 1

$q_1$ → $q_2$ on 1

$q_1$ → $q_3$ on 0

$q_3$ → $q_2$ on 0, 1

$q_2$ → $q_3$ on 0, 1

$0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 1$
A Simple NFA
A Simple NFA

Starting at state $q_0$, the NFA transitions as follows:
- From $q_0$ to $q_1$ on input 1.
- From $q_1$ to $q_2$ on input 1.
- From $q_2$ to $q_3$ on input 0.
- From $q_3$ to $q_3$ on input 0.

The accepting states are $q_2$ and $q_3$. The input sequence 01011 is tested on the NFA.
A Simple NFA

- **Start State**: $q_0$
- **States**: $q_1$, $q_2$, $q_3$
- **Transitions**:
  - From $q_0$ on 1 to $q_1$
  - From $q_1$ on 1 to $q_2$
  - From $q_2$ on 0 to $q_3$
  - From $q_3$ on 0 or 1 to $q_3$

Input: 0 1 0 1 1
A Simple NFA

\[
\begin{array}{c}
q_0 \\
\uparrow \text{start} \\
1 \\
q_1 \\
0, 1 \\
1 \\
q_2 \\
0, 1 \\
0, 1 \\
q_3 \\
0, 1 \\
0, 1 \\
q_2 \\
\end{array}
\]
A Simple NFA

\[ q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \xrightarrow{0} q_3 \xrightarrow{0, 1} q_2 \]

Input sequence: 0 1 0 1 1 1
A Simple NFA
A Simple NFA

![A Simple NFA Diagram]

0 1 0 1 1 1
A Simple NFA
A Simple NFA

![A Simple NFA Diagram]
A Simple NFA

0 1 0 1 1
A Simple NFA

Start

$q_0$ 1 $q_1$

$q_1$ 1 $q_2$

$q_2$ 0 $q_3$

$q_3$ 0 $q_2$

$q_2$ 0, 1

$q_3$ 0, 1

0 1 0 1 1 1
A Simple NFA

0 1 0 1 1 1
A Simple NFA

\[
\begin{array}{c}
\text{start} \\
q_0 \\
\quad 1 \\
\text{0, 1} \\
q_1 \\
\quad 1 \\
q_2 \\
\quad 0, 1 \\
q_3 \\
\quad 0, 1 \\
\end{array}
\]
A Simple NFA

Start $q_0$ on input 1 to $q_1$, then $q_1$ on input 1 to $q_2$. $q_3$ accepts on 0, 1 input.
A Simple NFA

Start state: $q_0$

Transitions:
- $q_0$: 1 to $q_1$
- $q_1$: 1 to $q_2$
- $q_1$: 0, 1 to $q_3$
- $q_2$: 0, 1 to $q_3$
- $q_3$: 0, 1 to $q_2$

Input string: 0 1 0 1 1
A Simple NFA

0 1 0 1 1 1
A Simple NFA

start -> $q_0$ on 1
$q_0$ on 0, 1 -> $q_1$ on 1
$q_1$ on 0 -> $q_3$ on 0, 1
$q_3$ on 0, 1 -> $q_2$ on 0, 1

0 1 0 1 1 1
A Simple NFA

**Diagram:**
- **Start State:** $q_0$
- **States:** $q_0, q_1, q_2, q_3$
- **Transitions:**
  - $q_0$ to $q_1$: $1$
  - $q_1$ to $q_2$: $1$
  - $q_1$ to itself: $0, 1$
  - $q_1$ to $q_3$: $0$
  - $q_3$ to $q_2$: $0, 1$
  - $q_3$ to itself: $0, 1$

**Input String:** 0 1 0 1 1
A Simple NFA

\begin{itemize}
  \item \textbf{Start State}: $q_0$
  \item \textbf{Transition 1}: $q_0 \xrightarrow{1} q_1$
  \item \textbf{Transition 2}: $q_1 \xrightarrow{1} q_2$
  \item \textbf{Transition 3}: $q_2 \xrightarrow{0, 1} q_3$
  \item \textbf{Transition 4}: $q_3 \xrightarrow{0, 1} q_2$
\end{itemize}
A More Complex NFA
A More Complex NFA

If a NFA needs to make a transition when no transition exists, the automaton **dies** and that particular path does not accept.
A More Complex NFA

\[ q_0 \rightarrow 0, 1 \quad q_1 \rightarrow 1 \quad q_2 \]

\[ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \]
A More Complex NFA
A More Complex NFA
A More Complex NFA

- **Start State**: $q_0$
- **Transition 1**: $q_0 \xrightarrow{1} q_1$
- **Transition 2**: $q_1 \xrightarrow{1} q_2$

Input sequence: 0 1 0 1 1

States:
- $q_0$
- $q_1$
- $q_2$ (accepting state)
A More Complex NFA

0 1 0 1 1
A More Complex NFA

0 1 0 1 1 1
A More Complex NFA

Oh no! There's no transition defined!
A More Complex NFA

0 1 0 1 1
A More Complex NFA

\[
\begin{array}{c}
\text{start} \\
q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2
\end{array}
\]

\[
\begin{array}{c}
0, 1 \\
0, 1, 0, 1, 0, 1, 1
\end{array}
\]
A More Complex NFA

- Start state: $q_0$
- Transition:
  - From $q_0$ to $q_1$: 1
  - From $q_1$ to $q_2$: 1
  - From $q_1$: 0, 1

Input sequence: 010111
A More Complex NFA

start \rightarrow q_0 \rightarrow q_1 \rightarrow q_2

0, 1

0 1 0 1 1 1
A More Complex NFA
A More Complex NFA

![Diagram of a NFA with states q0, q1, and q2 connected by transitions 1 and 0, 1. The arrow from start to q0 is labeled 1, and the arrow from q1 to q2 is labeled 1. A string 01011 is shown as an example input.]
A More Complex NFA

\begin{itemize}
\item Start: \( q_0 \)
\item Transition: \( 0, 1 \)
\item Transition: \( 1 \)
\end{itemize}

Input: 0 1 0 1 1
A More Complex NFA

- Start state: $q_0$
- States: $q_0, q_1, q_2$
- Transitions:
  - $q_0 \xrightarrow{1} q_1$
  - $q_1 \xrightarrow{1} q_2$
  - $q_0 \xleftarrow{0, 1}$

Input: 010111
A More Complex NFA

\[ q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2 \]

Transitions:
- From \( q_0 \) to \( q_1 \) on input 1
- From \( q_1 \) to \( q_2 \) on input 1

Input sequence: 010111
A More Complex NFA

```
0 1 0 1 1
```
A More Complex NFA

Start state: $q_0$

Transitions:
- From $q_0$ to $q_1$ on input 1
- From $q_1$ to $q_2$ on input 1

States: $q_0$, $q_1$, $q_2$
As with DFAs, the language of an NFA $N$ is the set of strings that $N$ accepts:

$$\mathcal{L}(N) = \{ w \in \Sigma^* | N \text{ accepts } w \}.$$ 

What is the language of the NFA shown above?

A. \{ \texttt{01011} \}
B. \{ $w \in \{0, 1\}^* | w$ contains at least two 1s \}
D. \{ $w \in \{0, 1\}^* | w$ ends with 1 \}
C. \{ $w \in \{0, 1\}^* | w$ ends with 11 \}
E. \{ $w \in \{0, 1\}^* | w$ ends with 111 \}
F. \textbf{None} of these, or \textbf{two or more} of these.

Answer at \textbf{PollEv.com/cs103} or text \textbf{CS103} to \textbf{22333} once to join, then A, ..., or F.
NFA Acceptance

• An NFA $N$ accepts a string $w$ if there is some series of choices that lead to an accepting state.

• Consequently, an NFA $N$ rejects a string $w$ if no possible series of choices lead it into an accepting state.

• It's easier to show that an NFA does accept something than to show that it doesn't.
ε-Transitions

- NFAs have a special type of transition called the **ε-transition**.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
\textbf{ε-Transitions}

- NFAs have a special type of transition called the \textit{ε-transition}.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
\(\varepsilon\)-Transitions

- NFAs have a special type of transition called the **\(\varepsilon\)-transition**.
- An NFA may follow any number of \(\varepsilon\)-transitions at any time without consuming any input.
**ε-Transitions**

- NFAs have a special type of transition called the **ε-transition**.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the ε-transition.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
\( \varepsilon \)-Transitions

- NFAs have a special type of transition called the **\( \varepsilon \)-transition**.

- An NFA may follow any number of \( \varepsilon \)-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the ε-transition.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the **ε-transition**.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the \textbf{ε-transition}.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the ε-transition.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the **ε-transition**.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the **ε-transition**.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
**ε-Transitions**

- NFAs have a special type of transition called the **ε-transition**.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

• NFAs have a special type of transition called the \textit{ε-transition}.

• An NFA may follow any number of ε-transitions at any time without consuming any input.
\(\varepsilon\)-Transitions

- NFAs have a special type of transition called the \textbf{\(\varepsilon\)-transition}.
- An NFA may follow any number of \(\varepsilon\)-transitions at any time without consuming any input.

\begin{center}
\begin{tikzpicture}
  \node[state,initial] (q0) at (0,0) {$q_0$};
  \node[state] (q1) at (2,0) {$q_1$};
  \node[state] (q2) at (4,0) {$q_2$};
  \node[state,accepting] (q3) at (0,-2) {$q_3$};
  \node[state] (q4) at (2,-2) {$q_4$};
  \node[state] (q5) at (4,-2) {$q_5$};

  \path[->,auto]
  (q0) edge node {a} (q1)
  (q1) edge node {b, \(\varepsilon\)} (q4)
  (q2) edge node {a} (q1)
  (q3) edge node {\(\varepsilon\)} (q4)
  (q4) edge node {b} (q5)
  (q5) edge[loop below] node {b} (q5);
\end{tikzpicture}
\end{center}
\(\varepsilon\)-Transitions

- NFAs have a special type of transition called the \textit{\(\varepsilon\)-transition}.
- An NFA may follow any number of \(\varepsilon\)-transitions at any time without consuming any input.

\(\varepsilon\)-Transitions

- NFAs have a special type of transition called the \textit{\(\varepsilon\)-transition}.
- An NFA may follow any number of \(\varepsilon\)-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the *ε-transition*.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
ɛ-Transitions

• NFAs have a special type of transition called the ɛ-transition.

• An NFA may follow any number of ɛ-transitions at any time without consuming any input.
ε-Transitions

• NFAs have a special type of transition called the ε-transition.

• An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

• NFAs have a special type of transition called the **ε-transition**.

• An NFA may follow any number of ε-transitions at any time without consuming any input.
ε-Transitions

- NFAs have a special type of transition called the ε-transition.
- An NFA may follow any number of ε-transitions at any time without consuming any input.
- NFAs are not required to follow ε-transitions every time one is available. It's simply another option at the machine's disposal (stay or move).
Suppose we run the above NFA on the string \texttt{10110}. How many of the following statements are true?

- There is at least one computation that finishes in an accepting state.
- There is at least one computation that finishes in a rejecting state.
- There is at least one computation that dies.
- This NFA accepts \texttt{10110}.
- This NFA rejects \texttt{10110}.

Answer at \texttt{PollEv.com/cs103} or text \texttt{CS103} to \texttt{22333} once to join, then a number.