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

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

- **Start state**: $q_0$
- Transitions:
  - From $q_0$: on 1, go to $q_1$
  - From $q_1$: on 1, go to $q_2$
  - From $q_2$: on 0, go to $q_3$
  - From $q_3$: on 0, go to $q_2$
  - From $q_2$: on 0, 1, loop back to $q_2$
    - From $q_3$: on 0, 1, loop back to $q_3$

States $q_0$, $q_1$, $q_2$, and $q_3$ are all part of the NFA.
A Simple NFA

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

$q_0$ -> $q_1$ with transitions 0, 1
$q_1$ -> $q_2$ with transitions 1
$q_0$ -> $q_3$ with transitions 0, 1
$q_3$ -> $q_2$ with transitions 0, 1

Input sequence: 0 1 0 1 1
A Simple NFA

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

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

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

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

\[ q_3 \xrightarrow{0, 1} q_3 \]

Input sequence: 0 1 0 1 1
A Simple NFA

- States: $q_0, q_1, q_2, q_3$
- Transitions:
  - $q_0 \xrightarrow{0, 1} q_1$
  - $q_1 \xrightarrow{1} q_2$
  - $q_2 \xrightarrow{0} q_0$
  - $q_2 \xrightarrow{0, 1} q_3$
  - $q_3 \xrightarrow{0, 1} q_2$

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

0 1 0 1 1
A Simple NFA

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

\[ q_3 \xrightarrow{0} q_2 \]

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

start

$q_0$ 1 $q_1$ 1 $q_2$

0, 1

$q_3$

0

0, 1

0, 1

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

- Start state: $q_0$
- Transitions:
  - $q_0 \overset{1}{\rightarrow} q_1$
  - $q_1 \overset{1}{\rightarrow} q_2$
  - $q_1 \overset{0, 1}{\rightarrow} q_3$
  - $q_3 \overset{0, 1}{\rightarrow} q_2$
  - $q_3 \overset{0, 1}{\rightarrow} q_3$

Input string: 010111
A Simple NFA

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

start

$q_0$ 1 $q_1$

0, 1 1 $q_2$

$q_3$

0, 1
A Simple NFA

- Start state: $q_0$
- States: $q_0$, $q_1$, $q_2$, $q_3$
- Transitions:
  - $q_0$ to $q_1$: on 1
  - $q_1$ to $q_2$: on 1
  - $q_1$ to $q_3$: on 0, 1
  - $q_3$ to $q_2$: on 0, 1

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

```
q_0 -> 1 -> q_1 -> 1 -> q_2
  0, 1     0
q_3 -> 0, 1
```

Input: 0 1 0 1 1 1
A Simple NFA

0 1 0 1 1
A Simple NFA

\[
\begin{array}{ccc}
q_0 & \xrightarrow{1} & q_1 \\
& \xrightarrow{0, 1} & \\
\text{start} & \xrightarrow{1} & q_1 \xrightarrow{1} q_2 \\
& \xrightarrow{0} & q_3 \xrightarrow{0, 1} \text{cycle} \\
& \xrightarrow{0, 1} & \\
\end{array}
\]

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

0 1 0 1 1
A Simple NFA

0 1 0 1 1
A Simple NFA

start \rightarrow q_0 \rightarrow q_1 \rightarrow q_2

0, 1 \rightarrow q_3 \rightarrow q_2

0, 1

0, 1

0, 1
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

\[\begin{array}{c}
\text{start} \\
q_0 \\
\rightarrow \\
1 \\
q_1 \\
1 \\
q_2 \\
0, 1
\end{array}\]
A More Complex NFA

\[010111\]
A More Complex NFA

0 1 0 1 1
A More Complex NFA

\[
\text{start} \rightarrow q_0 \xrightarrow{1} q_1 \xrightarrow{1} q_2
\]

0, 1

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

Start

$q_0$ $\xrightarrow{1}$ $q_1$ $\xrightarrow{1}$ $q_2$

0, 1

0 1 0 1 1 1
A More Complex NFA

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

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

Start state: \( q_0 \)

Transitions:
- \( q_0 \xrightarrow{0, 1} q_1 \)
- \( q_1 \xrightarrow{1} q_2 \)

Input sequence: 0 1 0 1 1

Final state: \( q_2 \)
A More Complex NFA

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

Start

$q_0$ → 1 → $q_1$ → 1 → $q_2$

0, 1

0 1 0 1 1 1
A More Complex NFA

![NFA Diagram]

- Start state: $q_0$
- Transitions:
  - $q_0$ to $q_1$: on 1
  - $q_1$ to $q_2$: on 1

- Accepting state: $q_2$

Input sequence: 0 1 0 1 1 1
A More Complex NFA

start

$q_0$ 1 $q_1$ 1 $q_2$

0, 1

0 1 0 1 1 1
A More Complex NFA

0 1 0 1 1
A More Complex NFA

The diagram shows a non-deterministic finite automaton (NFA) with states $q_0$, $q_1$, and $q_2$. The transitions are:

- From $q_0$ to $q_1$ on input 1.
- From $q_1$ to $q_2$ on input 1.
- From $q_0$ to $q_0$ on inputs 0 and 1.

The input string processed by the automaton is 010111.
A More Complex NFA

0 1 0 1 1 1
A More Complex NFA

\begin{center}
\begin{tikzpicture}[node distance = 2cm, thick, main node/.style = {circle,draw}]
  \node[main node] (0) {$q_0$};
  \node[main node] (1) [right of=0] {$q_1$};
  \node[main node] (2) [right of=1, xshift=1.5cm, fill=yellow] {$q_2$};

  \path[*->] (0) edge node [right] {1} (1)
                  edge [loop below] node {0, 1} (1)
  (1) edge node [right] {1} (2);
\end{tikzpicture}
\end{center}
A More Complex NFA

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

Start

\[ q_0 \]

\[ q_1 \]

\[ q_2 \]

SEAL

OF APPROVAL

0 1 1
As with DFAs, the language of an NFA $N$ is the set of strings that $N$ accepts:

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

What is the language of the NFA shown above?

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

Answer at PollEv.com/cs103 or text CS103 to 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 \textit{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 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 \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 **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.
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.
\(\varepsilon\text{-Transitions}\)

- NFAs have a special type of transition called the \textbf{\(\varepsilon\text{-transition}\)}.
- An NFA may follow any number of \(\varepsilon\)-transitions at any time without consuming any input.
- NFAs are not \textit{required} to follow \(\varepsilon\)-transitions. It's simply another option at the machine's disposal.
Suppose we run the above NFA on the string $10110$. 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 $10110$.
- This NFA rejects $10110$.

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