Nondeterministic Finite Automata

Nondeterminism

Subset Construction
Nondeterminism

◆ A **nondeterministic finite automaton** has the ability to be in several states at once.

◆ Transitions from a state on an input symbol can be to any set of states.
Nondeterminism – (2)

- Start in one start state.
- Accept if any sequence of choices leads to a final state.
- *Intuitively*: the NFA always “guesses right.”
Example: Moves on a Chessboard

◆ States = squares.

◆ Inputs = r (move to an adjacent red square) and b (move to an adjacent black square).

◆ Start state, final state are in opposite corners.
Example: Chessboard – (2)

Accept, since final state reached
Formal NFA

- A finite set of states, typically $Q$.
- An input alphabet, typically $\Sigma$.
- A transition function, typically $\delta$.
- A start state in $Q$, typically $q_0$.
- A set of final states $F \subseteq Q$. 
Transition Function of an NFA

- $\delta(q, a)$ is a set of states.
- Extend to strings as follows:
  - **Basis**: $\delta(q, \varepsilon) = \{q\}$
  - **Induction**: $\delta(q, wa) = \text{the union over all states } p \text{ in } \delta(q, w) \text{ of } \delta(p, a)$
Language of an NFA

- A string $w$ is accepted by an NFA if $\delta(q_0, w)$ contains at least one final state.
- The language of the NFA is the set of strings it accepts.
Example: Language of an NFA

- For our chessboard NFA we saw that rbb is accepted.
- If the input consists of only b’s, the set of accessible states alternates between \{5\} and \{1,3,7,9\}, so only even-length, nonempty strings of b’s are accepted.
- What about strings with at least one r?
Equivalence of DFA’s, NFA’s

- A DFA can be turned into an NFA that accepts the same language.
- If $\delta_D(q, a) = p$, let the NFA have $\delta_N(q, a) = \{p\}$.
- Then the NFA is always in a set containing exactly one state – the state the DFA is in after reading the same input.
Equivalence – (2)

◆ Surprisingly, for any NFA there is a DFA that accepts the same language.
◆ Proof is the *subset construction*.
◆ The number of states of the DFA can be exponential in the number of states of the NFA.
◆ Thus, NFA’s accept exactly the regular languages.
Subset Construction

Given an NFA with states $Q$, inputs $\Sigma$, transition function $\delta_N$, state state $q_0$, and final states $F$, construct equivalent DFA with:

- States $2^Q$ (Set of subsets of $Q$).
- Inputs $\Sigma$.
- Start state $\{q_0\}$.
- Final states = all those with a member of $F$. 
Critical Point

◆ The DFA states have *names* that are sets of NFA states.

◆ But as a DFA state, an expression like \{p,q\} must be read as a single symbol, not as a set.

◆ **Analogy:** a class of objects whose values are sets of objects of another class.
The transition function $\delta_D$ is defined by:

$$\delta_D(\{q_1, \ldots, q_k\}, a)$$

is the union over all $i = 1, \ldots, k$ of $\delta_N(q_i, a)$.

Example: We’ll construct the DFA equivalent of our “chessboard” NFA.
**Example: Subset Construction**

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>1,3,5</td>
</tr>
<tr>
<td>3</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2,8</td>
<td>1,5,7</td>
</tr>
<tr>
<td>5</td>
<td>2,4,6,8</td>
<td>1,3,7,9</td>
</tr>
<tr>
<td>6</td>
<td>2,8</td>
<td>3,5,9</td>
</tr>
<tr>
<td>7</td>
<td>4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>4,6</td>
<td>5,7,9</td>
</tr>
<tr>
<td>9</td>
<td>6,8</td>
<td>5</td>
</tr>
</tbody>
</table>

Alert: What we’re doing here is the *lazy* form of DFA construction, where we only construct a state if we are forced to.
Example: Subset Construction

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>1,3,5</td>
</tr>
<tr>
<td>3</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2,8</td>
<td>1,5,7</td>
</tr>
<tr>
<td>5</td>
<td>2,4,6,8</td>
<td>1,3,7,9</td>
</tr>
<tr>
<td>6</td>
<td>2,8</td>
<td>3,5,9</td>
</tr>
<tr>
<td>7</td>
<td>4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>4,6</td>
<td>5,7,9</td>
</tr>
<tr>
<td>9</td>
<td>6,8</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
\rightarrow \quad \{1\} \\
\{2,4\} \\
\{5\} \\
\{2,4,6,8\} \\
\{1,3,5,7\} 
\]

\[
\rightarrow \quad \{2,4\} \\
\{2,4,6,8\} \\
\{5\} \\
\{1,3,5,7\} 
\]
**Example: Subset Construction**

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>1,3,5</td>
</tr>
<tr>
<td>3</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2,8</td>
<td>1,5,7</td>
</tr>
<tr>
<td>5</td>
<td>2,4,6,8</td>
<td>1,3,7,9</td>
</tr>
<tr>
<td>6</td>
<td>2,8</td>
<td>3,5,9</td>
</tr>
<tr>
<td>7</td>
<td>4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>4,6</td>
<td>5,7,9</td>
</tr>
<tr>
<td>9</td>
<td>6,8</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
17 & \rightarrow \quad \{1\} \quad \{2,4\} \quad \{5\} \\
2 & \rightarrow \quad \{2,4\} \quad \{2,4,6,8\} \quad \{1,3,5,7\} \\
3 & \rightarrow \quad \{2,4,6,8\} \quad \{1,3,5,7\} \quad \{1,3,7,9\}
\end{align*}
\]
Example: Subset Construction

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>1,3,5</td>
</tr>
<tr>
<td>3</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2,8</td>
<td>1,5,7</td>
</tr>
<tr>
<td>5</td>
<td>2,4,6,8</td>
<td>1,3,7,9</td>
</tr>
<tr>
<td>6</td>
<td>2,8</td>
<td>3,5,9</td>
</tr>
<tr>
<td>7</td>
<td>4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>4,6</td>
<td>5,7,9</td>
</tr>
<tr>
<td>9</td>
<td>6,8</td>
<td>5</td>
</tr>
</tbody>
</table>

* {1,3,5,7}
* {1,3,7,9}

\[\rightarrow \{1\}\]
\[\rightarrow \{2,4\}\]
\[\rightarrow \{5\}\]
\[\rightarrow \{2,4,6,8\}\]
\[\rightarrow \{2,4,6,8\}\]
\[\rightarrow \{2,4,6,8\}\]
\[\rightarrow \{1,3,5,7\}\]
\[\rightarrow \{1,3,7,9\}\]
\[\rightarrow \{1,3,5,7,9\}\]
### Example: Subset Construction

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>1,3,5</td>
</tr>
<tr>
<td>3</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2,8</td>
<td>1,5,7</td>
</tr>
<tr>
<td>5</td>
<td>2,4,6,8</td>
<td>1,3,7,9</td>
</tr>
<tr>
<td>6</td>
<td>2,8</td>
<td>3,5,9</td>
</tr>
<tr>
<td>7</td>
<td>4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>4,6</td>
<td>5,7,9</td>
</tr>
<tr>
<td>9</td>
<td>6,8</td>
<td>5</td>
</tr>
</tbody>
</table>

- $\rightarrow \{1\}$
- $\rightarrow \{2,4\}$
- $\rightarrow \{5\}$
- $\rightarrow \{2,4,6,8\}$

- $\rightarrow \{1,3,5,7\}$
- $\rightarrow \{1,3,7,9\}$
- $\rightarrow \{1,3,5,7,9\}$

- $\star \{1,3,7,9\}$
- $\star \{1,3,5,7,9\}$
### Example: Subset Construction

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>1,3,5</td>
</tr>
<tr>
<td>3</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2,8</td>
<td>1,5,7</td>
</tr>
<tr>
<td>5</td>
<td>2,4,6,8</td>
<td>1,3,7,9</td>
</tr>
<tr>
<td>6</td>
<td>2,8</td>
<td>3,5,9</td>
</tr>
<tr>
<td>7</td>
<td>4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>4,6</td>
<td>5,7,9</td>
</tr>
<tr>
<td>9</td>
<td>6,8</td>
<td>5</td>
</tr>
</tbody>
</table>

- \( r_1 = \{1\} \rightarrow \{2,4\}\)
- \( r_2 = \{2,4\}\)
- \( b_2 = \{1,3,5,7,9\}\)
- \( r_3 = \{2,4,6,8\}\)
- \( b_3 = \{1,3,5,7,9\}\)
- \( r_4 = \{2,4,6,8\}\)
- \( b_4 = \{1,3,5,7,9\}\)
- \( r_5 = \{2,4,6,8\}\)
- \( b_5 = \{1,3,5,7,9\}\)
- \( r_6 = \{2,4,6,8\}\)
- \( b_6 = \{1,3,5,7,9\}\)
- \( r_7 = \{1,3,7,9\}\)
- \( b_7 = \{5\}\)
- \( r_8 = \{1,3,5,7,9\}\)
- \( b_8 = \{5\}\)
- \( r_9 = \{1,3,5,7,9\}\)
- \( b_9 = \{5\}\)
**Example: Subset Construction**

<table>
<thead>
<tr>
<th></th>
<th>r</th>
<th>b</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2,4</td>
<td>5</td>
</tr>
<tr>
<td>2</td>
<td>4,6</td>
<td>1,3,5</td>
</tr>
<tr>
<td>3</td>
<td>2,6</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>2,8</td>
<td>1,5,7</td>
</tr>
<tr>
<td>5</td>
<td>2,4,6,8</td>
<td>1,3,7,9</td>
</tr>
<tr>
<td>6</td>
<td>2,8</td>
<td>3,5,9</td>
</tr>
<tr>
<td>7</td>
<td>4,8</td>
<td>5</td>
</tr>
<tr>
<td>8</td>
<td>4,6</td>
<td>5,7,9</td>
</tr>
<tr>
<td>9</td>
<td>6,8</td>
<td>5</td>
</tr>
</tbody>
</table>

\[
\rightarrow \{1\} \rightarrow \{2,4\} \rightarrow \{5\}
\]

\[
\{2,4\} \rightarrow \{2,4,6,8\} \rightarrow \{1,3,5,7\}
\]

\[
\{5\} \rightarrow \{2,4,6,8\} \rightarrow \{1,3,7,9\}
\]

\[
\{2,4,6,8\} \rightarrow \{2,4,6,8\} \rightarrow \{1,3,5,7,9\}
\]

\[
\{1,3,5,7\} \rightarrow \{2,4,6,8\} \rightarrow \{1,3,5,7,9\}
\]

\[
\{1,3,7,9\} \rightarrow \{2,4,6,8\} \rightarrow \{5\}
\]

\[
\{1,3,5,7,9\} \rightarrow \{2,4,6,8\} \rightarrow \{1,3,5,7,9\}
\]
Proof of Equivalence: Subset Construction

◆ The proof is almost a pun.
◆ Show by induction on \(|w|\) that

\[ \delta_N(q_0, w) = \delta_D\left(\{q_0\}, w\right) \]

◆ **Basis**: \(w = \epsilon\): \(\delta_N(q_0, \epsilon) = \delta_D(\{q_0\}, \epsilon) = \{q_0\}\).
Induction

- Assume IH for strings shorter than \( w \).
- Let \( w = xa \); IH holds for \( x \).
- Let \( \delta_N(q_0, x) = \delta_D(\{q_0\}, x) = S \).
- Let \( T = \) the union over all states \( p \) in \( S \) of \( \delta_N(p, a) \).
- Then \( \delta_N(q_0, w) = \delta_D(\{q_0\}, w) = T \).
  - For NFA: the extension of \( \delta_N \).
  - For DFA: definition of \( \delta_D \) plus extension of \( \delta_D \).
    - That is, \( \delta_D(S, a) = T \); then extend \( \delta_D \) to \( w = xa \).
NFA’s With \( \epsilon \)-Transitions

- We can allow state-to-state transitions on \( \epsilon \) input.
- These transitions are done spontaneously, without looking at the input string.
- A convenience at times, but still only regular languages are accepted.
Example: $\epsilon$-NFA
Closure of States

- $CL(q) =$ set of states you can reach from state $q$ following only arcs labeled $\varepsilon$.
- **Example**: $CL(A) =$ \{A\}; $CL(E) =$ \{B, C, D, E\}.
- Closure of a set of states = union of the closure of each state.
Extended Delta

◆ **Basis:** \( \delta(q, \epsilon) = CL(q) \).

◆ **Induction:** \( \delta(q, xa) \) is computed as follows:
  1. Start with \( \delta(q, x) = S \).
  2. Take the union of \( CL(\delta(p, a)) \) for all \( p \) in \( S \).

◆ **Intuition:** \( \delta(q, w) \) is the set of states you can reach from \( q \) following a path labeled \( w \).
  And notice that \( \delta(q, a) \) is *not* that set of states, for symbol \( a \).
Example:

Extended Delta

\( \delta(A, \epsilon) = \text{CL}(A) = \{A\} \).

\( \delta(A, 0) = \text{CL}(\{E\}) = \{B, C, D, E\} \).

\( \delta(A, 01) = \text{CL}(\{C, D\}) = \{C, D\} \).

Language of an \( \epsilon \)-NFA is the set of strings \( w \) such that \( \delta(q_0, w) \) contains a final state.
Equivalence of NFA, $\varepsilon$-NFA

- Every NFA is an $\varepsilon$-NFA.
  - It just has no transitions on $\varepsilon$.
- Converse requires us to take an $\varepsilon$-NFA and construct an NFA that accepts the same language.
- We do so by combining $\varepsilon$–transitions with the next transition on a real input.

*Warning:* This treatment is a bit different from that in the text.
Picture of $\epsilon$-Transition Removal

Transitions on $\epsilon$

Transitions on $\epsilon$
Picture of $\varepsilon$-Transition Removal

Text goes from here to here, and performs the subset construction.

Transitions on $\varepsilon$
Picture of $\epsilon$-Transition Removal

We’ll go from here

Transitions on $\epsilon$

To here, with no subset construction

Transitions on $\epsilon$
Equivalence – (2)

- Start with an $\varepsilon$-NFA with states $Q$, inputs $\Sigma$, start state $q_0$, final states $F$, and transition function $\delta_E$.
- Construct an “ordinary” NFA with states $Q$, inputs $\Sigma$, start state $q_0$, final states $F'$, and transition function $\delta_N$. 
Equivalence – (3)

◆ Compute $\delta_N(q, a)$ as follows:

1. Let $S = \text{CL}(q)$.
2. $\delta_N(q, a)$ is the union over all $p$ in $S$ of $\delta_E(p, a)$.

◆ $F' = \text{the set of states } q \text{ such that } \text{CL}(q) \text{ contains a state of } F$.

◆ Intuition: $\delta_N$ incorporates $\varepsilon$-transitions before using $a$ but not after.
Equivalence – (4)

◆ Prove by induction on $|w|$ that

$$CL(\delta_N(q_0, w)) = \delta_E(q_0, w).$$

◆ Thus, the $\epsilon$-NFA accepts $w$ if and only if the “ordinary” NFA does.
Example: $\epsilon$-NFA-to-NFA

Interesting closures: $CL(B) = \{B, D\}$; $CL(E) = \{B, C, D, E\}$

Since closure of $E$ includes $B$ and $C$; which have transitions on 1 to $C$ and $D$.

Since closures of $B$ and $E$ include final state $D$. 

$\epsilon$-NFA

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>$\epsilon$</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>{E}</td>
<td>{B}</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>B</td>
<td>$\emptyset$</td>
<td>{C}</td>
<td>{D}</td>
</tr>
<tr>
<td>C</td>
<td>$\emptyset$</td>
<td>{D}</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>D</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>E</td>
<td>{F}</td>
<td>$\emptyset$</td>
<td>{B, C}</td>
</tr>
<tr>
<td>F</td>
<td>{D}</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>{E}</td>
<td>{B}</td>
</tr>
<tr>
<td>* B</td>
<td>$\emptyset$</td>
<td>{C}</td>
</tr>
<tr>
<td>C</td>
<td>$\emptyset$</td>
<td>{D}</td>
</tr>
<tr>
<td>D</td>
<td>$\emptyset$</td>
<td>$\emptyset$</td>
</tr>
<tr>
<td>* E</td>
<td>{F}</td>
<td>{C, D}</td>
</tr>
<tr>
<td>* F</td>
<td>{D}</td>
<td>$\emptyset$</td>
</tr>
</tbody>
</table>
Summary

- DFA’s, NFA’s, and ε-NFA’s all accept exactly the same set of languages: the regular languages.
- The NFA types are easier to design and may have exponentially fewer states than a DFA.
- But only a DFA can be implemented!