Example (1.30)

• Accept string of at least length three that contains a 1 in the third from end

\[ \sum = \{0,1\}; \quad \sum^* 1(0 \cup 1)(0 \cup 1) \]

What makes this difficult for a DFA?

Equivalent DFA takes 8 states. Why 8?
Formal Definition of NFA

- 5 tuple \((Q, \Sigma, \delta, q_0, F)\)

\[\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}\]
\[\delta: Q \times \Sigma_\varepsilon \rightarrow P(Q)\]

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>(\varepsilon)</th>
</tr>
</thead>
<tbody>
<tr>
<td>q_1</td>
<td>{q_1, q_2}</td>
<td>(\emptyset)</td>
<td>(\emptyset)</td>
</tr>
<tr>
<td>q_2</td>
<td>(\emptyset)</td>
<td>{q_3}</td>
<td>{q_3}</td>
</tr>
<tr>
<td>q_3</td>
<td>(\emptyset)</td>
<td>{q_4}</td>
<td>(\emptyset)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>(\varepsilon)</td>
<td></td>
</tr>
</tbody>
</table>

Formal Definition of Computing for NFA

- Given a machine \(M = (Q, \Sigma, \delta, q_0, F)\) and a string \(w = w_1w_2\ldots w_n\) over \(\Sigma\), then \(M\) accepts \(w\) if there exists a sequence of states \(r_0, r_1, \ldots, r_n\) in \(Q\) such that:

\[\neg r_0 = q_0\]
\[\neg \delta (r_i, w_{i+1}) = r_{i+1}, i=0,\ldots,n-1\]
\[\neg r_n \in F\]
### Practice

- Construct a NFA with three states that recognizes \( \{w \mid w \text{ ends with two } 0s \} \)

\[
\Sigma = \{0, 1\}
\]

---

- Construct a NFA with six states \( \{w \mid w \text{ even \# } 0s \text{ OR exactly two } 1s \} \)

\[
\Sigma = \{0, 1\}
\]
### Practice

- **Construct a NFA with four states** \(0^*1^*0^*0\)
  
  **Bonus:** Can this be done with three states?

\[ \Sigma = \{0, 1\} \]

---

### Practice

- **Construct a NFA with seven states:**
  
  \( \{ w \mid w \text{ starts and ends with a different symbol}\} \)

\[ \Sigma = \{0, 1\} \]