A play of tennis

- game, set, match
- love, 15, 30, 40, game
- winning also requires at least two scores more
- deuce, advantage-in, advantage-out
A tennis automaton

love
A tennis automaton

![Diagram showing a tennis score system with love, 15-love, love-15, and o nodes connected by arrows.](image-url)
A tennis automaton
A tennis automaton
A tennis automaton
A tennis automaton
A tennis automaton
Adding a sink state
A tennis automaton (cont.)

- single initial state
- zero or more final states
- many labeled transitions
- for each state and symbol exactly one transition
accepted strings like \textit{sososs} and \textit{soossosoooo}
Which string is *not* accepted by the tennis automaton?

A. sossoososoo
B. ssssoos
C. ooossssss
D. all are accepted
Deterministic finite automaton

DFA $D = (Q, \Sigma, \delta, q_0, F)$

- $Q$ finite set of states
- $\Sigma$ finite alphabet
- $\delta : Q \times \Sigma \rightarrow Q$ transition function
- $q_0 \in Q$ initial state
- $F \subseteq Q$ set of final states
The tennis example
set of states \{ love, 15–love, love–15, \ldots \}

\text{game-in, game-out, deuce, adv-in, adv-out}

alphabet \{ s, o \}

transitions love \xrightarrow{s} 15–love, love \xrightarrow{o} love–15, \ldots

initial state love

set of final states \{ game-in, game-out \}
one-step and multi-step yield

- configuration \((q, w)\) for state \(q\) and string \(w\)
configuration \((q, w)\) for state \(q\) and string \(w\)

one-step yield

\[(q, w) \vdash_D (q', w') \text{ iff } \exists a : w = aw', \ \delta(q, a) = q'\]
one-step and multi-step yield

- configuration \((q, w)\) for state \(q\) and string \(w\)
- one-step yield
  \[(q, w) \vdash_D (q', w')\] iff \(\exists a: w = aw', \delta(q, a) = q'\)
- multi-step yield
  \[(q, w) \vdash^*_D (q', w')\]
one-step and multi-step yield

- configuration \((q, w)\) for state \(q\) and string \(w\)

- one-step yield
  \[
  (q, w) \xrightarrow{D} (q', w') \iff \exists a: w = aw', \ \delta(q, a) = q'
  \]

- multi-step yield
  \[
  (q, w) \xrightarrow{D}^* (q', w') \iff \\
  \exists n \geq 0 \exists w_0, \ldots, w_n \in \Sigma^* \exists q_0, \ldots, q_n \in Q: \\
  (q, w) = (q_0, w_0), \\
  (q_{i-1}, w_{i-1}) \xrightarrow{D} (q_i, w_i), \text{ for } i = 1..n \\
  (q_n, w_n) = (q', w')
  \]
one-step and multi-step yield

- configuration \((q, w)\) for state \(q\) and string \(w\)

- one-step yield
  \[(q, w) \vdash_D (q', w')\] iff \(\exists a: w = aw', \delta(q, a) = q'\)

- multi-step yield
  \[(q, w) \vdash^*_D (q', w')\] iff
  \[\exists n \geq 0 \exists w_0, \ldots, w_n \in \Sigma^* \exists q_0, \ldots, q_n \in Q:
  (q, w) = (q_0, w_0),
  (q_{i-1}, w_{i-1}) \vdash_D (q_i, w_i), \text{ for } i = 1 \ldots n
  (q_n, w_n) = (q', w')\]

\[(q, w) = (q_0, w_0) \vdash_D (q_1, w_1) \vdash_D \ldots \vdash_D (q_n, w_n) = (q', w')\]
for suitable \(n, w_0, \ldots, w_n, q_0, \ldots, q_n\)
Another example DFA

- $(q_0, abaa) \vdash (q_1, baa) \vdash (q_0, aa) \vdash (q_1, a) \vdash (q_2, \varepsilon)$
- $(q_0, bbaa) \vdash (q_0, baa) \vdash (q_0, aa) \vdash (q_1, a) \vdash (q_2, \varepsilon)$
- $(q_1, aa) \vdash^* (q_2, \varepsilon)$ and $(q_1, aaaa) \vdash^* (q_2, \varepsilon)$
- $(q_0, aab) \vdash^* (q_3, \varepsilon)$, $(q_0, baab) \vdash^* (q_3, \varepsilon)$, and $(q_0, baaaabaabb) \vdash^* (q_3, \varepsilon)$
\[ \mathcal{L}(D) = \{ w \in \Sigma^* \mid \exists q \in F: (q_0, w) \xrightarrow{*} (q, \varepsilon) \} \]
\[ \mathcal{L}(D) = \{ w \in \Sigma^* \mid \exists q \in F: (q_0, w) \xrightarrow{*} (q, \varepsilon) \} \]

accepted language \{ w \in \{a, b\}^* \mid w \text{ has a substring } aab \}
Question

Which language is the language accepted by this automaton?

A. \( \{a, b, aba, bab\} \)
B. \( \{ a(bb)^n \mid n \geq 0 \} \cup \{ b(aa)^n \mid n \geq 0 \} \)
C. \( \{ w \in \{a, b\}^* \mid \#_a(w) \text{ is odd} \} \)
D. \( \{ w \in \{a, b\}^* \mid \#_a(w) + \#_b(w) \text{ is odd} \} \)
Path sets

DFA $D$, state $q$

$$\text{pathset}_D(q) = \{ w \in \Sigma^* \mid (q_0, w) \vdash_D^* (q, \varepsilon) \}$$
Path sets

DFA $D$, state $q$

$$\text{pathset}_D(q) = \{ w \in \Sigma^* \mid (q_0, w) \xrightarrow{\ast}_D (q, \varepsilon) \}$$

- $\text{pathset}_D(\text{even}) = \{ a^n \mid n \geq 0, n \text{ even} \}$
- $\text{pathset}_D(\text{odd}) = \{ a^n \mid n \geq 0, n \text{ odd} \}$
Yet another example DFA

\[ L = \{ w \in \{0, 1\}^* \mid w \text{ has no substring } 11 \} \]

<table>
<thead>
<tr>
<th>state</th>
<th>pathset</th>
<th>regular expressions</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_0 )</td>
<td>no substring 11 and no last symbol 1</td>
<td>( 0^* (10^+)^* )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>no substring 11 and last symbol 1</td>
<td>( 0^* (10^+)^* 1 )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>substring 11</td>
<td>( (0 + 1)^* 11 (0 + 1)^* )</td>
</tr>
</tbody>
</table>

regular expressions will be explained later
Another example DFA (rev.)

```
<table>
<thead>
<tr>
<th>state</th>
<th>pathset</th>
</tr>
</thead>
<tbody>
<tr>
<td>ee</td>
<td>{ w \mid #_a(w) \text{ even}, #_b(w) \text{ even} }</td>
</tr>
<tr>
<td>oe</td>
<td>{ w \mid #_a(w) \text{ odd}, #_b(w) \text{ even} }</td>
</tr>
<tr>
<td>eo</td>
<td>{ w \mid #_a(w) \text{ even}, #_b(w) \text{ odd} }</td>
</tr>
<tr>
<td>oo</td>
<td>{ w \mid #_a(w) \text{ odd}, #_b(w) \text{ odd} }</td>
</tr>
</tbody>
</table>
```

\[
\mathcal{L}(D) = \text{pathset}_D(eo) \cup \text{pathset}_D(oe) \\
= \{ w \mid \#_a(w) \text{ odd}, \#_b(w) \text{ even} \} \cup \{ w \mid \#_a(w) \text{ even}, \#_b(w) \text{ odd} \} \\
= \{ w \mid \#_a(w) + \#_b(w) \text{ odd} \} \\
= \{ w \mid |w| \text{ odd} \}
\]
Language accepted by DFA (revisited)

accepted language \( \{ w \in \{a, b\}^* \mid w \text{ has a substring } aab \} \)

<table>
<thead>
<tr>
<th>state</th>
<th>pathset</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q_0 )</td>
<td>( { w \mid \text{no substring } aab, \text{ not ending in } a } )</td>
</tr>
<tr>
<td>( q_1 )</td>
<td>( { w \mid \text{no substring } aab, \text{ ending in } a, \text{ but not in } aa } )</td>
</tr>
<tr>
<td>( q_2 )</td>
<td>( { w \mid \text{no substring } aab, \text{ ending in } aa } )</td>
</tr>
<tr>
<td>( q_3 )</td>
<td>( { w \mid \text{substring } aab } )</td>
</tr>
</tbody>
</table>
Why does the following not represent a DFA?

A. The alphabet has more than 2 letters.
B. It accepts the empty string $\varepsilon$.
C. It has a transition relation, but not a transition function.
D. It does represent a DFA.