DFA  NFA  regexp

regular languages
- sequencing
- branching
- repetition

context-Free languages
- recursion
- recursive/decidable
- unbounded memory

CFG

Turing machine

FSM

1. Descriptive of "arbitrary" computation

2. Simulate itself

\[ \langle M \rangle \xrightarrow{C} \langle M' \rangle \xrightarrow{\text{halts?}} \begin{cases} T \rightarrow \text{accept} \\ F \rightarrow \text{reject} \end{cases} \]
\[ TM = (\Gamma, \Delta, \Sigma, Q, \text{start}, \text{reject}, \text{accept}, S) \]

\[ \Gamma = \text{tape alphabet} \quad \square \in \Gamma \text{ blank} \]

\[ \Sigma = \text{input alphabet} \quad \Sigma \subseteq \Gamma \setminus \{\square\} \]

\[ Q = \text{states} \quad \text{start}, \text{reject}, \text{accept} \in Q \]

\[ S : (Q \setminus \{\text{accept, reject}\}) \times \Gamma \to Q \times \Gamma \quad \delta \]

\[ \frac{0^n1^n0^n \mid n \geq 0}{\delta} \]

<table>
<thead>
<tr>
<th>p</th>
<th>a</th>
<th>q</th>
<th>b</th>
<th>\Delta</th>
</tr>
</thead>
<tbody>
<tr>
<td>start</td>
<td>0</td>
<td>seek1</td>
<td>$</td>
<td>+1</td>
</tr>
<tr>
<td>start</td>
<td>x</td>
<td>seek1</td>
<td>x</td>
<td>+1</td>
</tr>
<tr>
<td>seek1</td>
<td>x</td>
<td>seek1</td>
<td>$</td>
<td>+1</td>
</tr>
<tr>
<td>seek0</td>
<td>0</td>
<td>seek1</td>
<td>x</td>
<td>+1</td>
</tr>
<tr>
<td>seek0</td>
<td>1</td>
<td>seek0</td>
<td>x</td>
<td>+1</td>
</tr>
<tr>
<td>seek0</td>
<td>0</td>
<td>seek0</td>
<td>x</td>
<td>+1</td>
</tr>
</tbody>
</table>

**Explanation:**
- **mark first \( \square \) and scan right**
- **looks like we’re done, but let’s make sure**
- **scan rightward for \( 1 \)**
- **mark \( 1 \) and continue right**
- **scan rightward for \( 0 \)**
- **mark \( 0 \) and scan left**
- **scan leftward for \( \$ \)**
- **step right and start over**
- **scan right for any unmarked symbol**
- **success!**

The transition function for a Turing machine that decides the language \( \{0^n1^n0^n \mid n \geq 0\} \).
Random access memory storing [addr, value] on tape

Multiple tapes

Memory
Addr Register
Data Register
Stack!
local vars

FSM

Input

Recursion / function call

Church-Turing thesis: "computable" = computable by a TM