

DFA NFA regexp

regular languages

- sequencing
- branching
- repetition

CFG

context-free languages

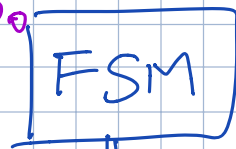
- recursion

Turing machine

recursive / decidable

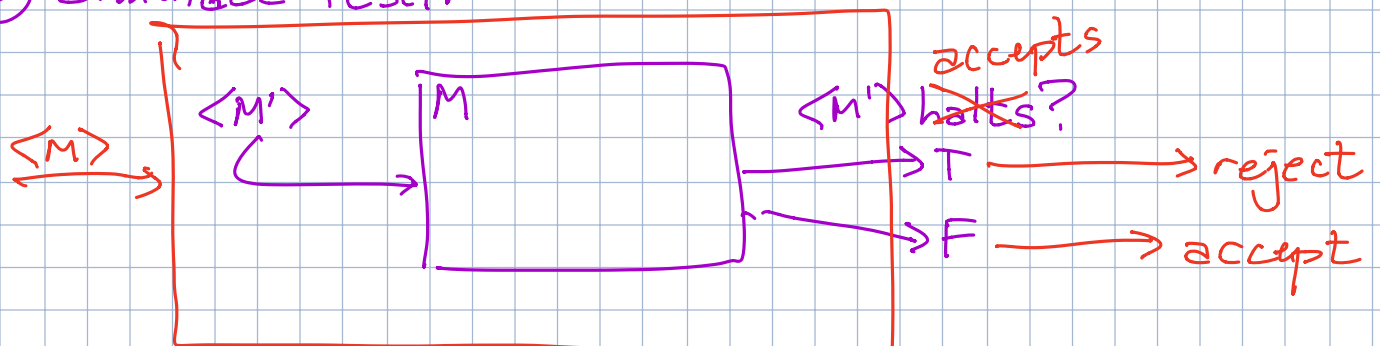
- unbounded memory

374



① Descriptive of "arbitrary" computation

② Simulate itself



$TM = (\Gamma, \square, \Sigma, Q, \text{start}, \text{reject}, \text{accept}, \delta)$

$\Gamma = \text{tape alphabet}$        $\square \in \Gamma$  blank

$\Sigma = \text{input alphabet}$        $\Sigma \subseteq \Gamma \setminus \{\square\}$

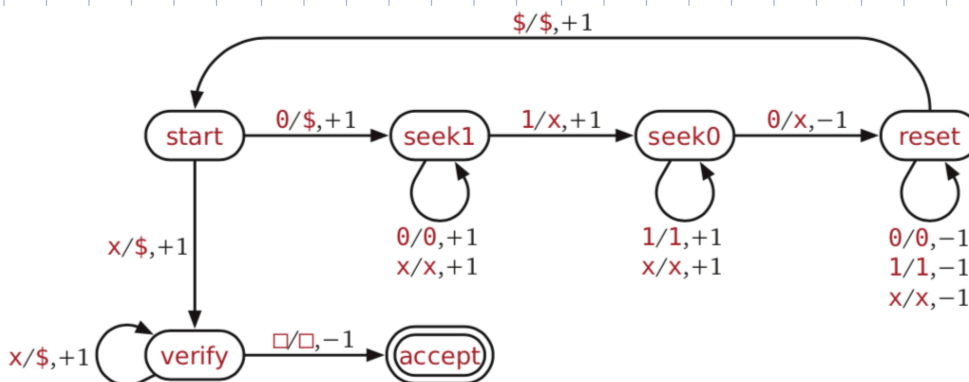
$Q = \text{states}$        $\text{start}, \text{reject}, \text{accept} \in Q$

$\delta : (Q \setminus \{\text{accept}, \text{reject}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{-1, +1\}$

$\{0^n 1^n 0^n \mid n \geq 0\}$

$\delta(p, a) = (q, b, \Delta)$	explanation
$\delta(\text{start}, 0) = (\text{seek1}, \$, +1)$	mark first 0 and scan right
$\delta(\text{start}, x) = (\text{verify}, \$, +1)$	looks like we're done, but let's make sure
$\delta(\text{seek1}, 0) = (\text{seek1}, 0, +1)$	scan rightward for 1
$\delta(\text{seek1}, x) = (\text{seek1}, x, +1)$	
$\delta(\text{seek1}, 1) = (\text{seek0}, x, +1)$	mark 1 and continue right
$\delta(\text{seek0}, 1) = (\text{seek0}, 1, +1)$	scan rightward for 0
$\delta(\text{seek0}, x) = (\text{seek0}, x, +1)$	
$\delta(\text{seek0}, 0) = (\text{reset}, x, +1)$	mark 0 and scan left
$\delta(\text{reset}, 0) = (\text{reset}, 0, -1)$	scan leftward for \$
$\delta(\text{reset}, 1) = (\text{reset}, 1, -1)$	
$\delta(\text{reset}, x) = (\text{reset}, x, -1)$	
$\delta(\text{reset}, \$) = (\text{start}, \$, +1)$	step right and start over
$\delta(\text{verify}, x) = (\text{verify}, \$, +1)$	scan right for any unmarked symbol
$\delta(\text{verify}, \square) = (\text{accept}, \square, -1)$	success!

The transition function for a Turing machine that decides the language  $\{0^n 1^n 0^n \mid n \geq 0\}$ .



$$\delta(p, a) = (q, b, \Delta)$$


---


$$\delta(\text{start}, 0) = (\text{seek1}, \$, +1)$$

$$\delta(\text{start}, x) = (\text{verify}, \$, +1)$$


---


$$\delta(\text{seek1}, 0) = (\text{seek1}, 0, +1)$$

$$\delta(\text{seek1}, x) = (\text{seek1}, x, +1)$$


---


$$\delta(\text{seek1}, 1) = (\text{seek0}, x, +1)$$

$$\delta(\text{seek0}, 1) = (\text{seek0}, 1, +1)$$


---


$$\delta(\text{seek0}, x) = (\text{seek0}, x, +1)$$

$$\delta(\text{seek0}, 0) = (\text{reset}, x, +1)$$


---


$$\delta(\text{reset}, 0) = (\text{reset}, 0, -1)$$

$$\delta(\text{reset}, 1) = (\text{reset}, 1, -1)$$


---


$$\delta(\text{reset}, x) = (\text{reset}, x, -1)$$

$$\delta(\text{reset}, \$) = (\text{start}, \$, +1)$$


---


$$\delta(\text{verify}, x) = (\text{verify}, \$, +1)$$

$$\delta(\text{verify}, \square) = (\text{accept}, \square, -1)$$

$$\Rightarrow (\text{start}, \overset{\blacktriangle}{001100})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$01100})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$01100})$$

$$\Rightarrow (\text{seek0}, \overset{\blacktriangle}{\$0x100})$$

$$\Rightarrow (\text{seek0}, \overset{\blacktriangle}{\$0x100})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$0x1x0})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$0x1x0})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$0x1x0})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$0x1x0})$$

$$\Rightarrow (\text{start}, \overset{\blacktriangle}{\$0x1x0})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$\$x1x0})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$\$x1x0})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$\$x1x0})$$

$$\Rightarrow (\text{seek0}, \overset{\blacktriangle}{\$\$xxx0})$$

$$\Rightarrow (\text{seek0}, \overset{\blacktriangle}{\$\$xxx0})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$\$xxxx})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$\$xxxx})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$\$xxxx})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$\$xxxx})$$

$$\Rightarrow (\text{start}, \overset{\blacktriangle}{\$\$xxxx})$$

$$\Rightarrow (\text{verify}, \overset{\blacktriangle}{\$\$\$xxx})$$

$$\Rightarrow (\text{verify}, \overset{\blacktriangle}{\$\$\$\$xx})$$

$$\Rightarrow (\text{verify}, \overset{\blacktriangle}{\$\$\$\$\$x})$$

$$\Rightarrow (\text{verify}, \overset{\blacktriangle}{\$\$\$\$\$\square})$$

$$\Rightarrow (\text{accept}, \overset{\blacktriangle}{\$\$\$\$\$\$}) \Rightarrow \text{accept!}$$

$$\Rightarrow (\text{start}, \overset{\blacktriangle}{00100})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$0100})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$0100})$$

$$\Rightarrow (\text{seek0}, \overset{\blacktriangle}{\$0x00})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$0xx0})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$0xx0})$$

$$\Rightarrow (\text{reset}, \overset{\blacktriangle}{\$0xx0})$$

$$\Rightarrow (\text{start}, \overset{\blacktriangle}{\$0xx0})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$\$xx0})$$

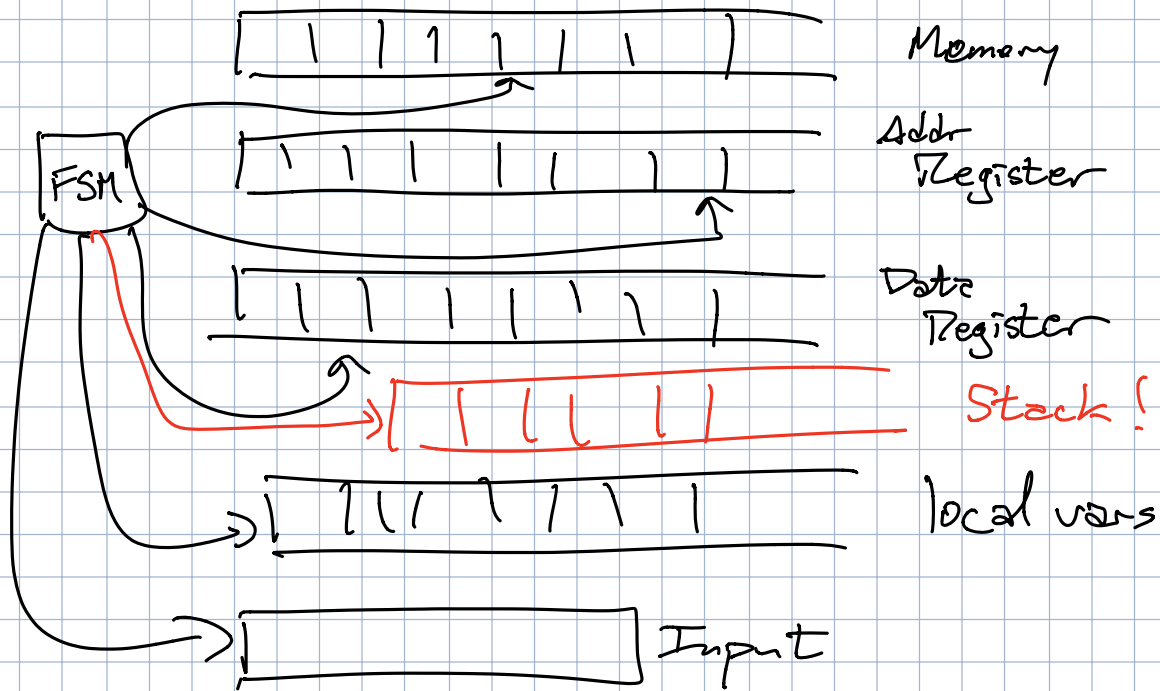
$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$\$xx0})$$

$$\Rightarrow (\text{seek1}, \overset{\blacktriangle}{\$\$xx0}) \Rightarrow \text{reject!}$$

Random access memory

storing [addr, value] on tape

Multiple tapes



Recursion / function call

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