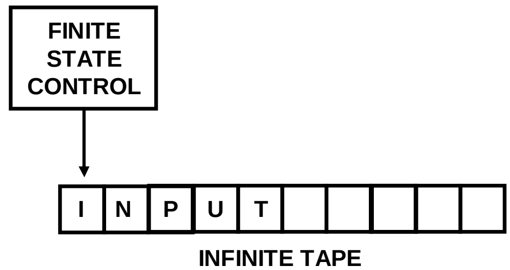
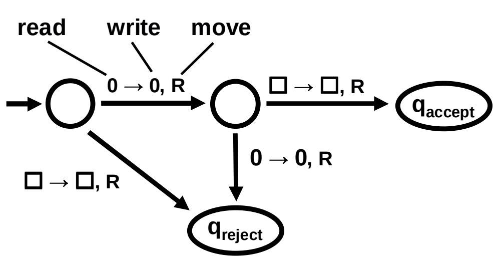
TURING MACHINES

TUESDAY SEP 20

TURING MACHINE





TMs VERSUS FINITE AUTOMATA

TM can both write to and read from the tape

The head can move left and right

The string doesn't have to be read entirely

Accept and Reject take immediate effect

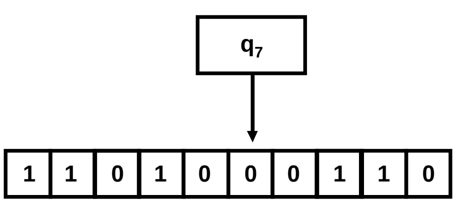
Testing membership in B = { w#w | w ∈ {0,1}* }

Definition: A Turing Machine is a 7-tuple
$T = (Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject})$, where:

- **Q is a finite set of states**
- **Σ is the input alphabet, where $\square \notin \Sigma$**
- **Γ is the tape alphabet, where $\square \in \Gamma$ and $\Sigma \subseteq \Gamma$**
- **$\delta : Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$**
- **$q_0 \in Q$ is the start state**
- **$q_{accept} \in Q$ is the accept state**
- **$q_{reject} \in Q$ is the reject state, and $q_{reject} \neq q_{accept}$**

CONFIGURATIONS

$11010q_7 00110$



A TM recognizes a language if it accepts all and only those strings in the language

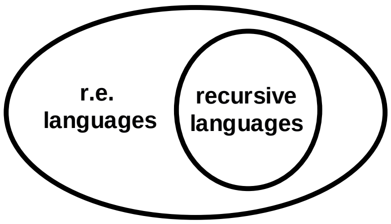
A language is called Turing-recognizable or recursively enumerable if some TM recognizes it

A TM decides a language if it accepts all strings in the language and rejects all strings not in the language

A language is called decidable or recursive if some TM decides it

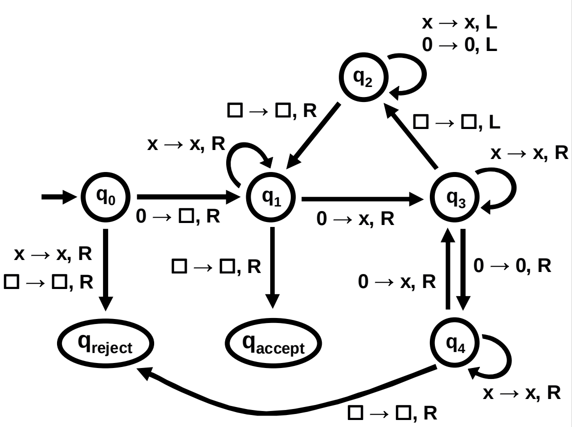
A language is called Turing-recognizable or recursively enumerable if some TM recognizes it

A language is called decidable or recursive if some TM decides it



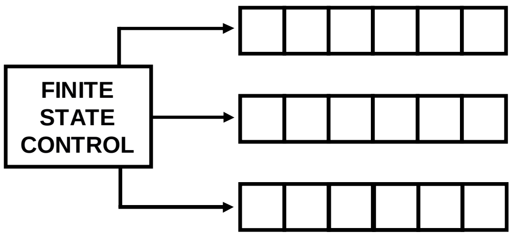
Theorem: If A and ¬A are r.e. then A is recursive

Given TM that recognizes A and TM that recognizes ¬A, we can build a new machine that decides A



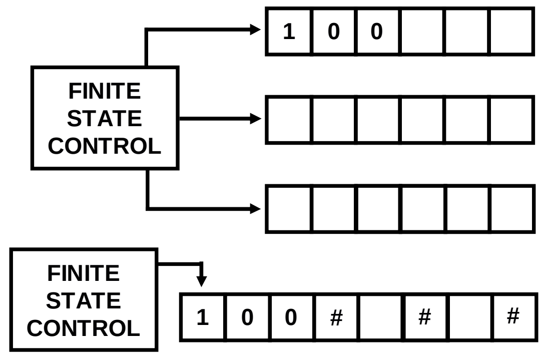
$C = \{a^i b^j c^k \mid ij = k \text{ and } i, j, k \geq 1\}$

MULTITAPE TURING MACHINES



$$\delta : Q \times \Gamma^k \rightarrow Q \times \Gamma^k \times \{L,R\}^k$$

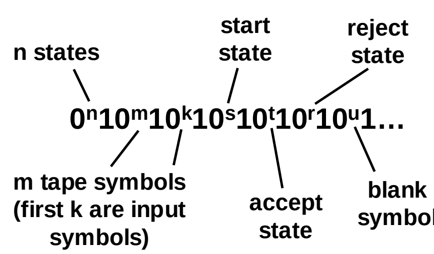
Theorem: Every Multitape Turing Machine can be transformed into a single tape Turing Machine



THE CHURCH-TURING THESIS

Intuitive notion of algorithms
equals
Turing Machines

Can we encode a TM as a string of 0s and 1s?



$$(\,(p,a), (q,b,L)\,) = 0^p10^a10^q10^b10$$

Similarly, we can encode DFAs, NFAs, CFGs, etc into strings of 0s and 1s

So we can define the following languages:

A_{DFA} = { (B,w) | B is a DFA that accepts string w }

A_{NFA} = { (B,w) | B is an NFA that accepts string w }

A_{CFG} = { (G,w) | G is a CFG that generates string w }

A_{DFA} = { (B,w) | B is a DFA that accepts string w }

A_{CFG} = { (G,w) | G is a CFG that generates string w }

WWW.FLAC.WS

Read chapter 3 of the book for next time