TURING MACHINES

ITEU133

AUTOMATA AND THEORY OF COMPUTATION

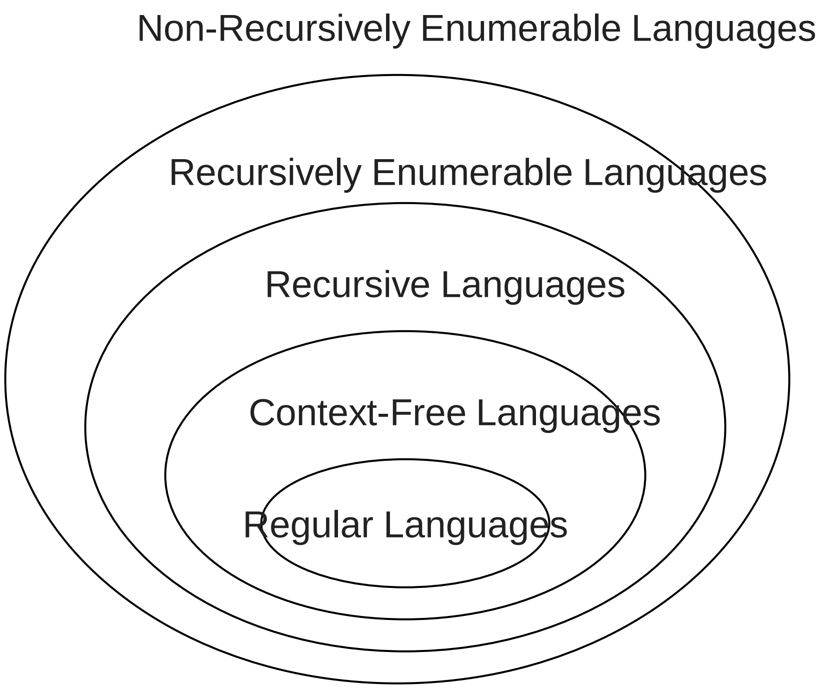
CHURCH–TURING'S THESIS

- Simultaneously mathematicians were working independently on the same problem.

Emil Post	® Production Systems
Alonzo Church	® Lambda Calculus
Noam Chomsky	®Unrestricted Grammars
Stephen Kleene	® Recursive function Theory
Raymond Smullyn	®Formal Systems.

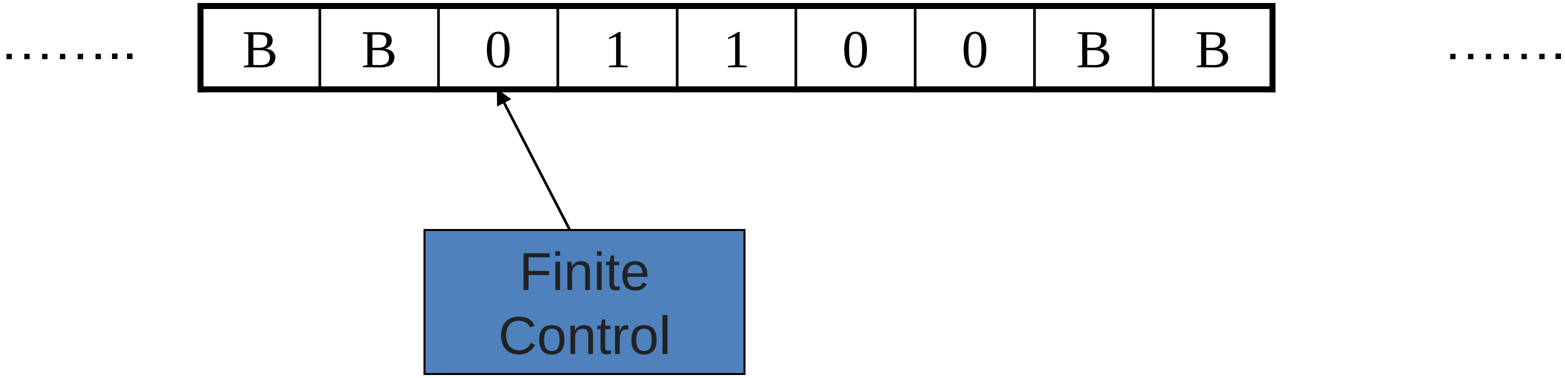
TURING MACHINE

- Generalize the class of CFLs:



- TMs model the computing capability of a general purpose computer, which informally can be described as:
  - Effective procedure
    - Finitely describable
    - Well defined, discrete, “mechanical” steps
    - Always terminates
  - Computable function
    - A function computable by an effective procedure
- TMs formalize the above notion.
- **Church-Turing Thesis:** There is an effective procedure for solving a problem if and only if there is a TM that halts for all inputs and solves the problem.
  - There are many other computing models, but all are equivalent to or subsumed by TMs. *There is no more powerful machine* (Technically cannot be proved).
- DFAs and PDAs do not model all effective procedures or computable functions, but only a subset.

DETERMINISTIC TURING MACHINE (DTM)



- Two-way, infinite tape, broken into cells, each containing one symbol.
- Two-way, read/write tape head.
- Finite control, i.e., a program, containing the position of the read head, current symbol being scanned, and the current state.
- An input string is placed on the tape, padded to the left and right infinitely with blanks, read/write head is positioned at the left end of input string.
- In one move, depending on the current state and the current symbol being scanned, the TM 1) changes state, 2) prints a symbol over the cell being scanned, and 3) moves its' tape head one cell left or right.
- Many modifications possible.

FORMAL DEFINITION OF A DTM

- A DTM is a seven-tuple:

$$\mathbf{M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)}$$

Q A finite set of states

Γ A finite tape alphabet

B A distinguished blank symbol, which is in Γ

Σ A finite input alphabet, which is a subset of Γ– {B}

q_0 The initial/starting state, q_0 is in Q

F A set of final/accepting states, which is a subset of Q

δ A next-move function, which is a *mapping* (i.e., may be undefined) from **Q x Γ –> Q x Γ x {L,R}**

Intuitively, δ(q,s) specifies the next state, symbol to be written, and the direction of tape head movement by M after reading symbol s while in state q.

Example #1: $\{0^n1^n \mid n >= 1\}$

	0	1	X	Y	B
q_0	(q_1, X, R)	-	-	(q_3, Y, R)	-
q_1	$(q_1, 0, R)$	(q_2, Y, L)	-	(q_1, Y, R)	-
q_2	$(q_2, 0, L)$	-	(q_0, X, R)	(q_2, Y, L)	-
q_3	-	-	-	(q_3, Y, R)	(q_4, B, R)
q_4	-	-	-	-	-

Sample Computation: (on 0011)

W = 1 1 1 0 1 1

$$
\begin{aligned}
Q &= \{q_0, q_1, q_2, q_3, q_4\}, \\
F &= \{q_4\}, \\
\delta(q_0, 1) &= (q_0, 1, R), \\
\delta(q_0, 0) &= (q_1, 1, R), \\
\delta(q_1, 1) &= (q_1, 1, R), \\
\delta(q_1, \square) &= (q_2, \square, L), \\
\delta(q_2, 1) &= (q_3, 0, L), \\
\delta(q_3, 1) &= (q_3, 1, L), \\
\delta(q_3, \square) &= (q_4, \square, R).
\end{aligned}
$$

$$\delta(q_0, 1) = (q_0, x, R),$$
$$\delta(q_0, \square) = (q_1, \square, L),$$
$$\delta(q_1, x) = (q_2, 1, R),$$
$$\delta(q_2, 1) = (q_2, 1, R),$$
$$\delta(q_2, \square) = (q_1, 1, L),$$
$$\delta(q_1, 1) = (q_1, 1, L),$$
$$\delta(q_1, \square) = (q_3, \square, R),$$

W = 1 1

$Q=\{q0, q1, q2, q3, q4, q_{acc}, q_{rej}\}$

$\Gamma=\{a, b, X, Y, \#\}$

$\Sigma = \{a, b, \}$ s=q0 B=#

δ is given by

W=baabab

$\delta(q0, \#) = (q_{acc}, \#, R)$	$\delta(q0, a) = (q2, X, R)$	$\delta(q0, b) = (q3, X, R)$
$\delta(q1, Y) = (q1, Y, R)$	$\delta(q1, a) = (q2, X, R)$	$\delta(q1, b) = (q3, X, R)$
	$\delta(q1, \#) = (q_{acc}, \#, R)$	
$\delta(q2, a) = (q2, a, R)$	$\delta(q2, Y) = (q2, Y, R)$	$\delta(q2, b) = (q4, Y, L)$
$\delta(q3, b) = (q3, b, R)$	$\delta(q3, Y) = (q3, Y, R)$	$\delta(q3, a) = (q4, Y, L)$
$\delta(q4, a) = (q4, a, L)$	$\delta(q4, b) = (q4, b, L)$	$\delta(q4, Y) = (q4, Y, L)$
	$\delta(q4, X) = (q1, X, R)$	