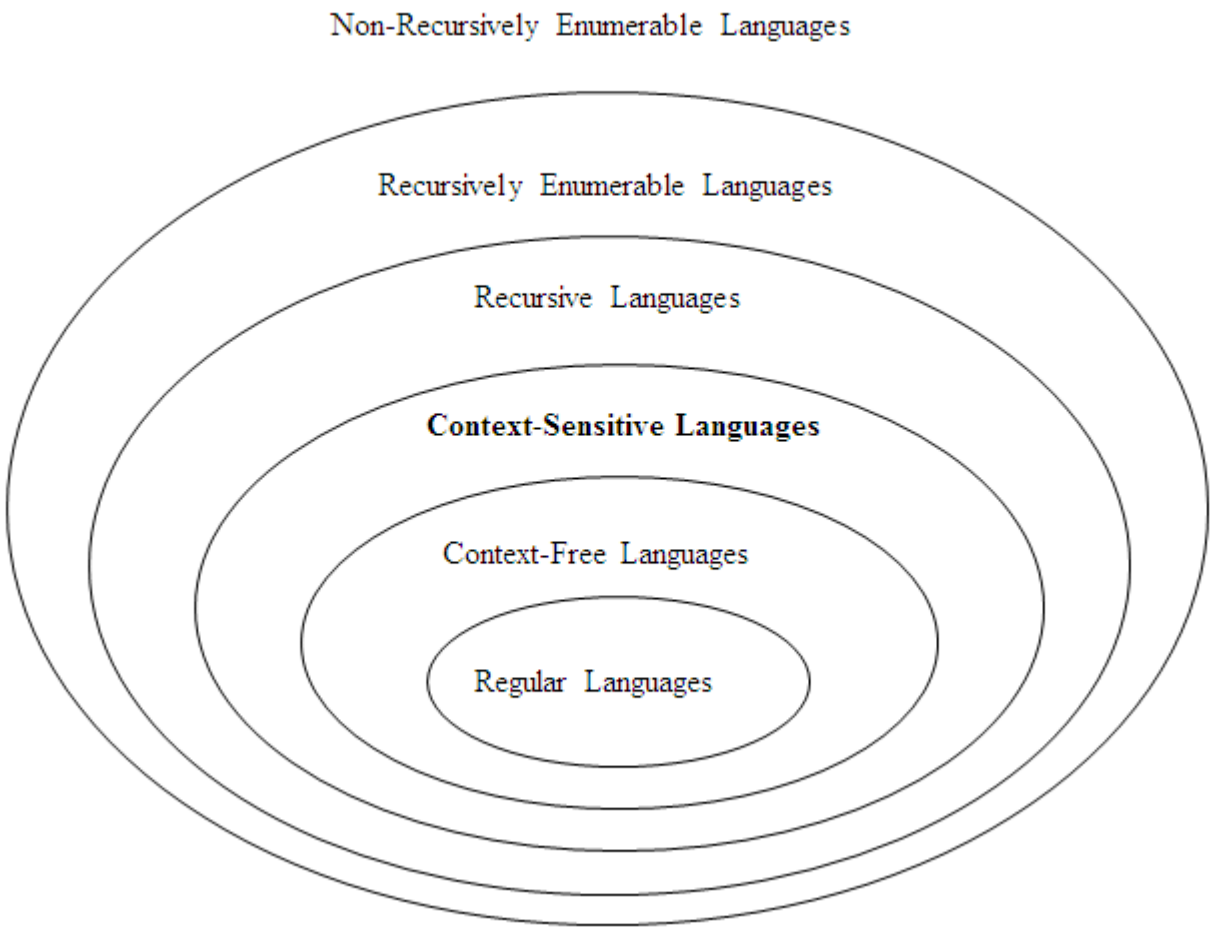
Turing Machines

Recursively Enumerable and Recursive Languages

- **The Hierarchy of Languages:**



- Recursively enumerable languages are also known as *type 0* languages.
- Context-sensitive languages are also known as *type 1* languages.
- Context-free languages are also known as *type 2* languages.
- Regular languages are also known as *type 3* languages.

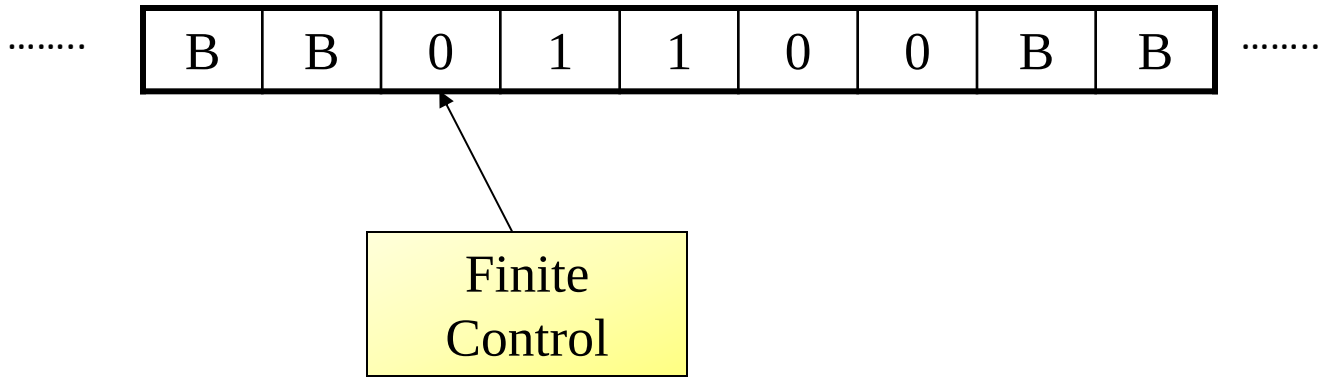
- TMs model the computing capability of a general purpose computer, which informally can be described as:
  - Effective procedure
    - Finitely describable
    - Well defined, discrete, “mechanical” steps
    - Always terminates
  - Computable function
    - A function computable by an effective procedure

- 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).

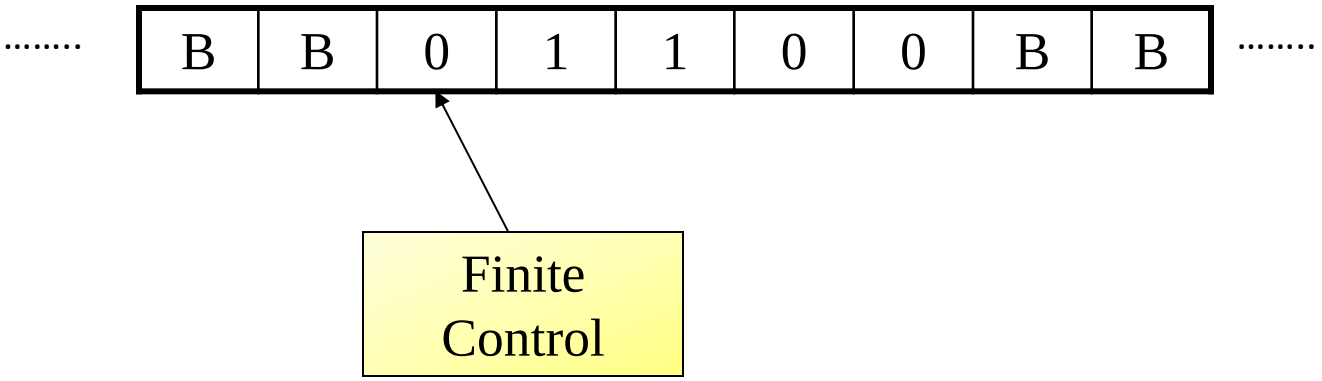
- DFAs and PDAs do not model all effective procedures or computable functions, but only a subset.

Deterministic Turing Machine (DTM)



- Two-way, infinite tape, broken into cells, each containing one symbol.
- Two-way, read/write tape head.
- 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 most input character.
- Finite control (read/write head is part of this control), knows current symbol being scanned, and its current state.

Deterministic Turing Machine (DTM)



- In one move, depending on the current state and the current symbol being scanned, the TM does: (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, but Church-Turing declares equivalence of all.

Formal Definition of a DTM

- A DTM is a seven-tuple:

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

Q	A finite set of states
Σ	A finite input alphabet, which is a subset of $\Gamma - \{B\}$
Γ	A finite tape alphabet, which is a strict superset of Σ
B	A distinguished blank symbol, which is in Γ
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 $Q \times \Gamma \rightarrow Q \times \Gamma \times \{L,R\}$

  Intuitively, $\delta(q,s)$ 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:** {w | w is in {0,1}* and w ends with a 0}

0

00

10

10110

Not ε

$Q = \{q_0, q_1, q_2\}$

$\Gamma = \{0, 1, B\}$

$\Sigma = \{0, 1\}$

$F = \{q_2\}$

δ:

	0	1	B
$->q_0$	$(q_0, 0, R)$	$(q_0, 1, R)$	(q_1, B, L)
q_1	$(q_2, 0, R)$	-	-
q_2^*	-	-	-

  - q_0 is the start state and the "scan right" state, until hits B
  - q_1 is the verify 0 state
  - q_2 is the final state

- **Example #2:** $\{0^n1^n \mid n \geq 1\}$

	0	1	X	Y	B
->q_0	(q_1, X, R)	-	-	(q_3, Y, R) *0's finished*	-
q_1	(q_1, 0, R) *ignore1*	(q_2, Y, L)	-	(q_1, Y, R) *ignore2*	- (more 0's)
q_2	(q_2, 0, L) *ignore2*	-	(q_0, X, R)	(q_2, Y, L) *ignore1*	-
q_3	-	- (more 1's)	-	(q_3, Y, R) *ignore*	(q_4, B, R)
q_4*	-	-	-	-	-

- **Sample Computation:** (on 0011), *presume state q looks rightward*

$q_0$0011BB.. |— X$q_1$011
|— X0$q_1$11
|— X$q_2$0Y1
|— q_2X0Y1
|— X$q_0$0Y1
|— XXq_1Y1
|— XXY$q_1$1
|— XXq_2YY
|— Xq_2XYY
|— XXq_0YY
|— XXYq_3Y B…
|— XXYYq_3 BB…
|— XXYYBq_4

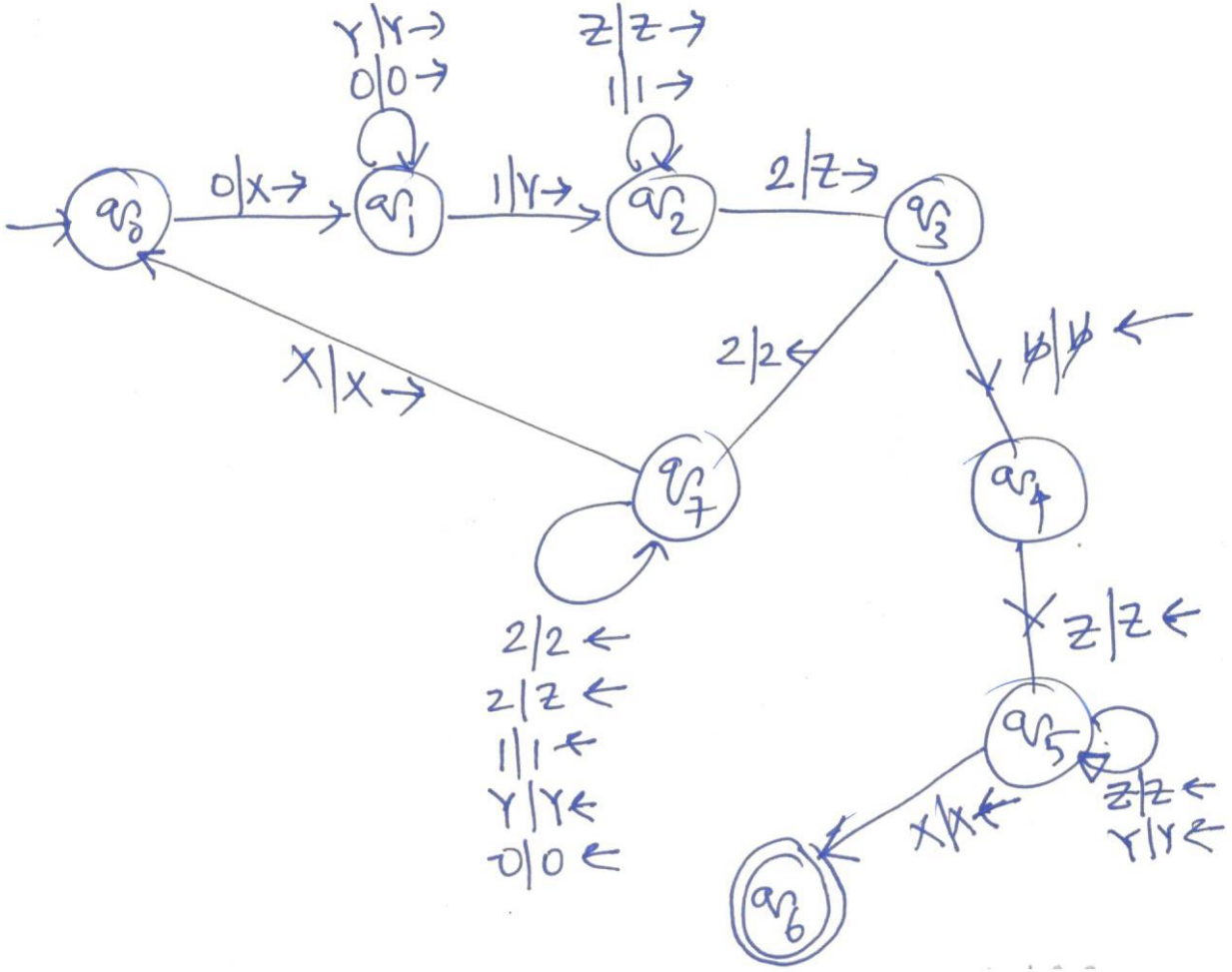
TM

- TM can be seen as a recognizer.
- TM can be seen as a input-output mapper.
- TM can also be seen as an enumerator.
  - Generates the language one string after the other.

Eg: $\Sigma = \{0,1,2\}$, $\Gamma = \{X, Y, Z, B\}$ $L = \{0^n 1^n 2^n / n \geq 1\}$

- But, there is a mistake in this.
- Γ has to be a superset of Σ

Eg. $\Sigma = \{0,1,2\}$, $\Gamma = \{X, Y, Z, \not b\}$ $L = \{0^n 1^n 2^n \mid n \geq 1\}$

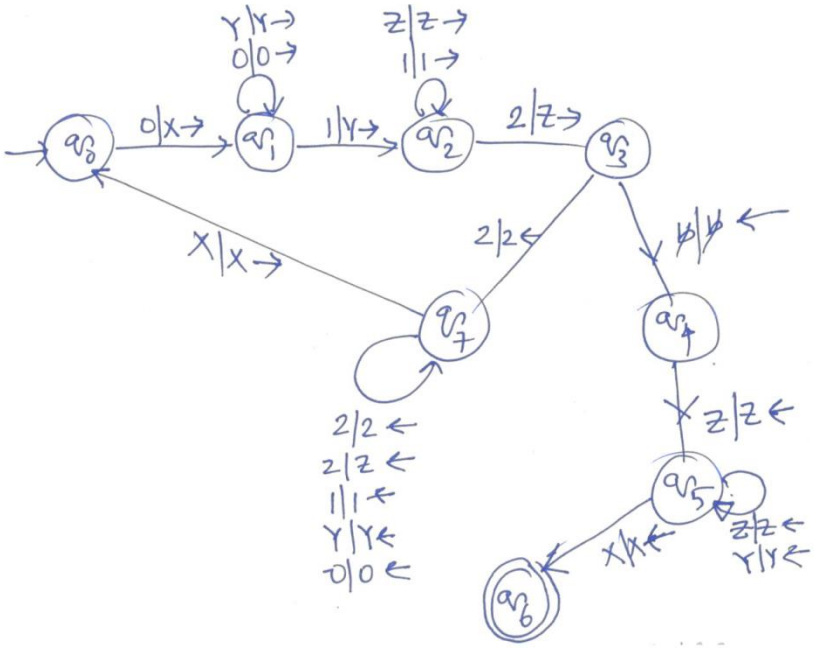


ID

- To describe the steps we use IDs.
- An ID is $\alpha q \beta$
- $\alpha, \beta \in \Gamma^*$
- $\alpha\beta$ is on the tape. Left and right of this are blanks.
- Head is on the first character of β
- The TM is in state q

Step

- $id_1 \vdash id_2$ describes one step
- $\vdash^*$ is reflexive and transitive closure of $\vdash$



$w = 001122$

$q_0 001122 \vdash X q_1 01122 \vdash X0 q_1 1122 \vdash X0Y q_2 122$

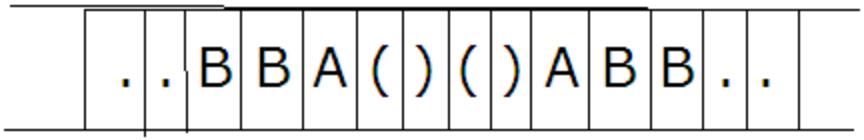
$\vdash X0Y1 q_2 22 \vdash X0Y1Z q_3 2 \vdash X0Y1 q_7 Z2$

$\vdash^* q_7 X0Y1Z2 \vdash X q_0 0Y1Z2 \vdash^* XXYYZZ q_3 B$

$\vdash^* X q_5 XYYZZ \vdash q_6 XXYYZZ$

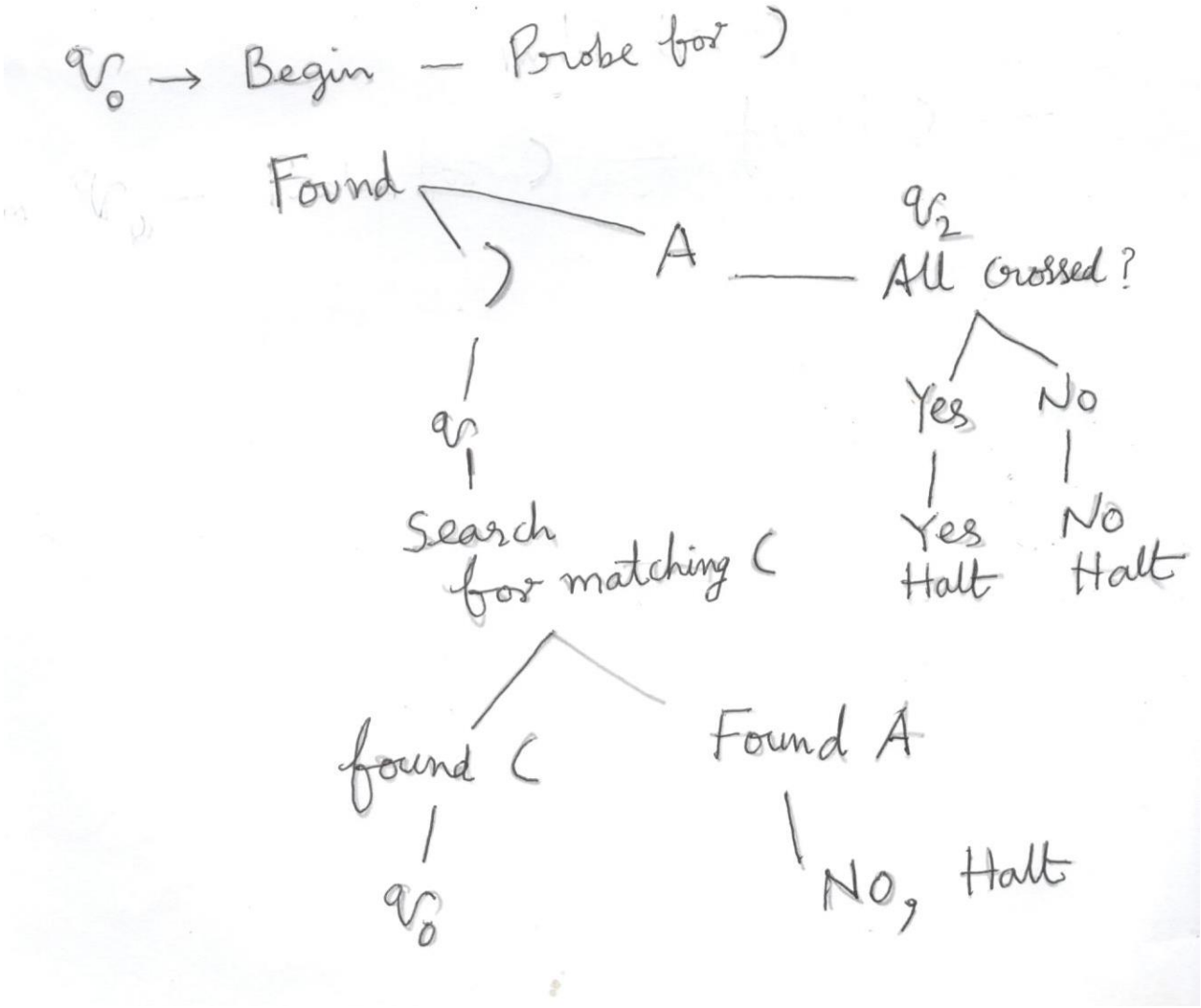
TM that gives output

- Example:
- $\Sigma = \{(,)\}$
- L is set of well formed parentheses.
- $\Gamma = \{A, X, B, Y, N\}$
- We use B or $\not b$ to mean the blank
- To simplify, we assume the given string is embedded between As

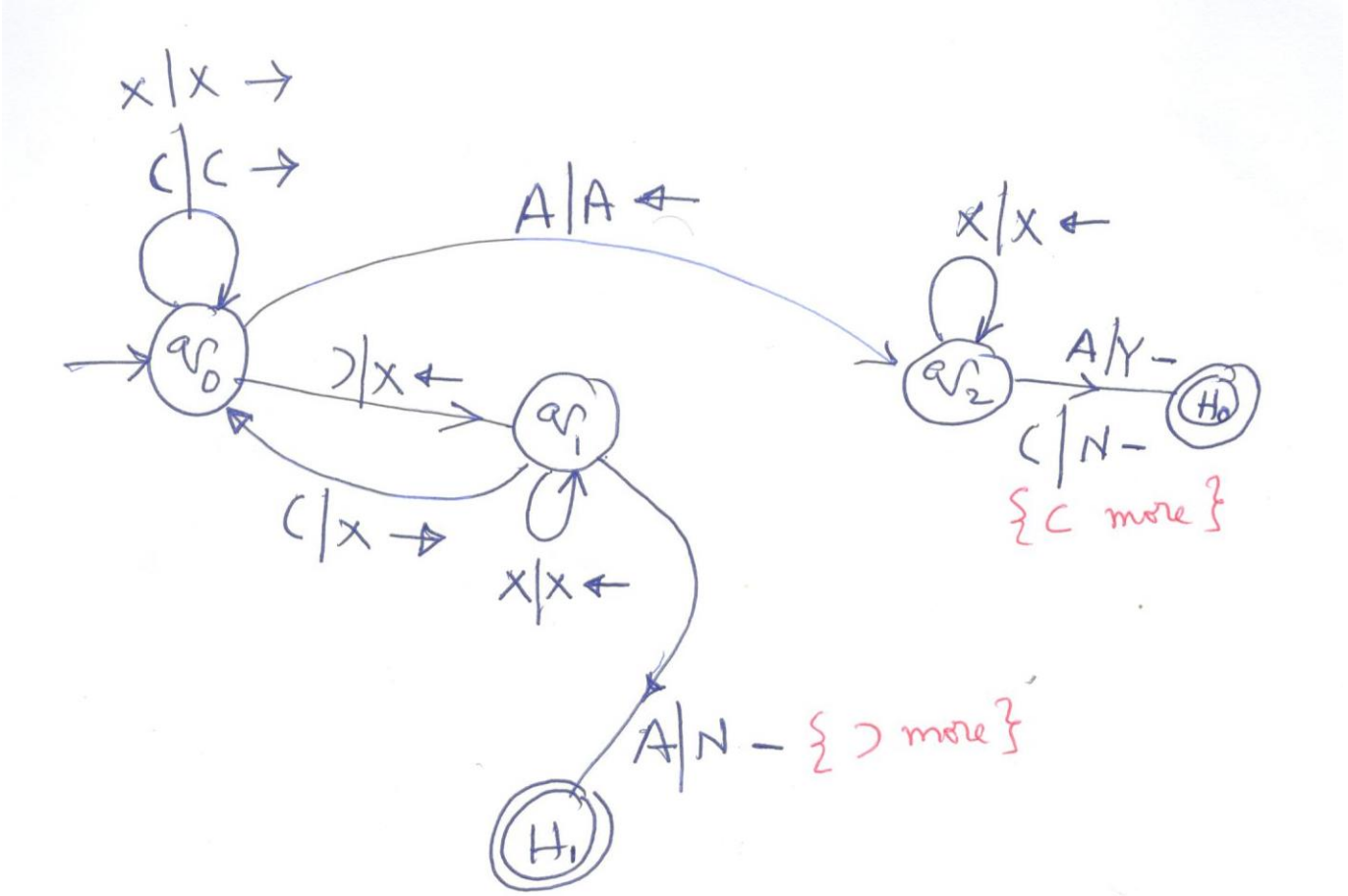


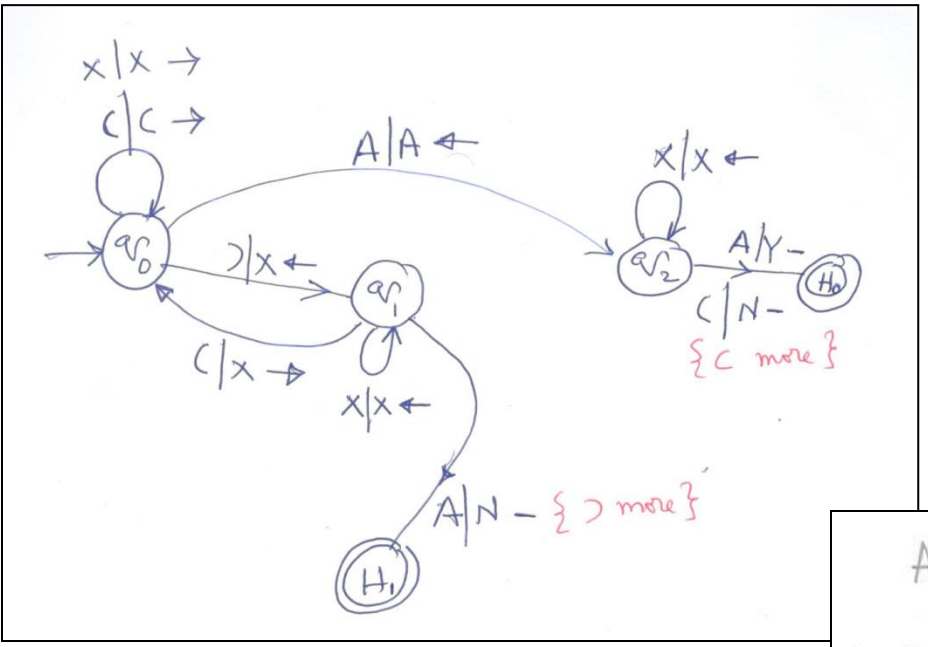
- The TM halts with a Y to mean Yes or a N to mean No.
- $Q = \{ q_0, q_1, q_2, H_0, H_1\}$
- $F = \{H_0, H_1 \}$

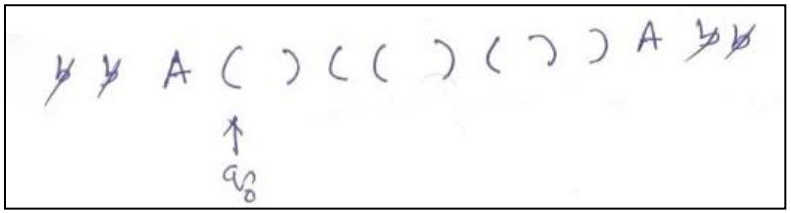
Idea


q_0 → Begin — Probe for)
Found
)
A
q_2
All crossed?
Yes
No
q_3
Search for matching (
Yes Halt
No Halt
found (
Found A
q_0
No, Halt

The transition diagram







$A\,q_0()(()())A \vdash A(q_0)(()())A$

$\vdash A\,q_1(X(()())A \vdash A\,X\,q_0\,X(()())A$

$\vdash A\,X\,X\,q_0(()())A \vdash A\,X\,X(q_0()())A$

$\vdash A\,X\,X\,(q_0)())A \vdash A\,X\,X\,(q_1(X())A$

$\vdash A\,X\,X(X\,q_0\,X())A \vdash^{*} A\,X\,X(X\,X(q_0))A$

$\vdash A\,X\,X(X\,X\,q_1(X))A \vdash A\,X\,X(X\,X\,X\,q_0\,X)A$

$\vdash A\,X\,X(X\,X\,X\,X\,q_0)A \vdash A\,X\,X(X\,X\,X\,q_1\,X\,X\,A$

$\vdash^{*} A\,X\,X\,q_1(X\,X\,X\,X\,X\,A \vdash A\,X\,X\,X\,q_0\,X\,X\,X\,X\,X\,A$

$\vdash^{*} A\,X\,X\,X\,X\,X\,X\,q_0\,A \vdash A\,X\,X\ldots X\,q_2\,X\,A$

$\vdash^{*} q_2\,A\,X\,X\ldots X\,A \vdash H_0\,Y\,X\,X - X\,A$

Halt o/p

The language accepted by a TM M

- Let M be a TM, M = (Q, Σ, Γ, δ, q_0, B, F)
- $L(M) = \{w \in \Sigma^* | q_0 w \vdash^* \alpha p \beta$, where $p \in F$, and $\alpha, \beta \in \Gamma^*\}$.
- Note that, input string getting exhausted is not present (in contrast to the acceptance criterion by PDAs, DFAs, etc).
- The language accepted by a TM is also said the language recognized by a TM.

Recursively enumerable

- If $w \in L(M)$, then M accepts/recognizes the string w.
- If, $w \notin L(M)$, then M may or may not halt.
  - No transition means M halts and rejects.
- We say $L(M)$ for a given TM M is recursively enumerable (RE).

Recursive languages

- Recursive languages (R) is a subset of RE.
- We say $L(M)$ for a TM M is recursive, if for any given input string w, M halts.
- That is, if $w \in L(M)$, M halts in an accepting (final) state.
- Else, M halts in a non-final state.
- M never goes in to an infinite loop.

RE Vs. R

RE

- A TM M recognizes.

R

- A TM M decides.

RE Vs. R

RE

- A TM M recognizes.
- If $L \in RE - R$, then complement of L, that is $\bar{L} \notin RE$.

R

- A TM M decides.
- $L \in R \Leftrightarrow \bar{L} \in R$

RE Vs. R

RE

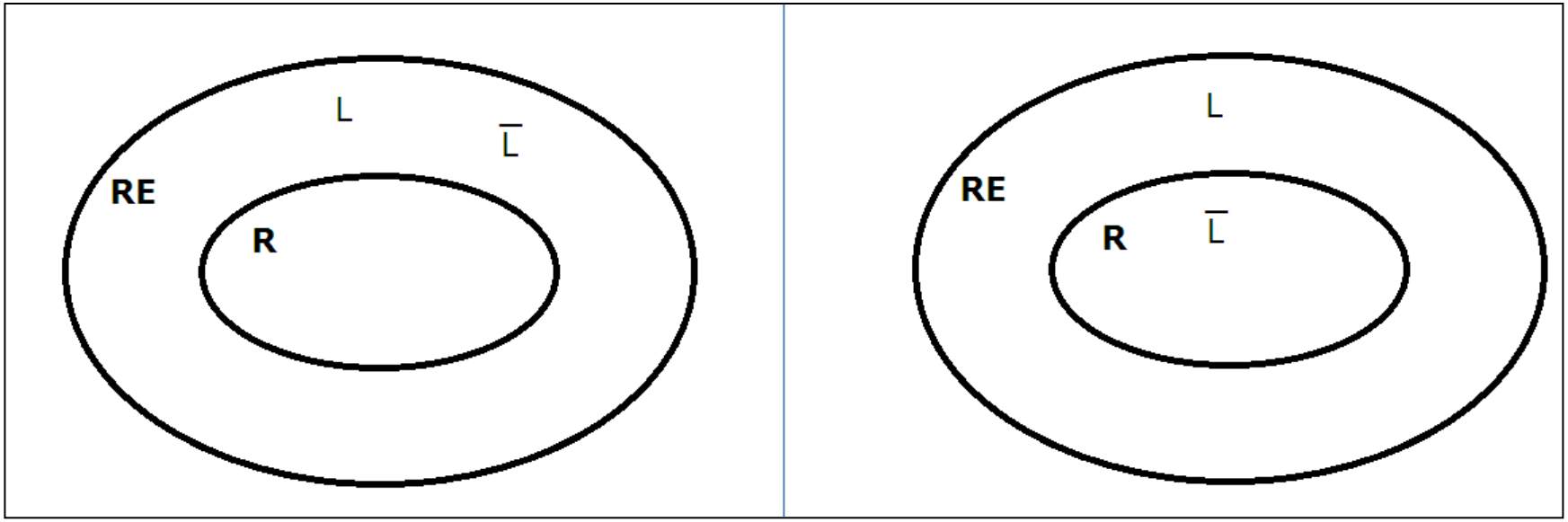
- A TM M recognizes.
- If $L \in RE - R$, then complement of L, that is $\bar{L} \notin RE$.

R

- A TM M decides.
- $L \in R \Leftrightarrow \bar{L} \in R$

$$L \in RE \text{ and } \bar{L} \in RE \quad \Rightarrow \quad L \in R$$

Not Possible



Possible