17/10/2022

Languages of PDA:

1) Acceptance by reaching the final state
2) Acceptance by reaching the final state & empting the stack.
3) Acceptance by empting the final stack.

① $L = \{0^n 1^{3n} \mid n \geq 1\}$

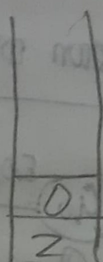
$L = \{0111, 00111111, 000111111111 \ldots\}$

$\delta(q_0, 0, Z) = \{(q_0, 0Z)\}$

$\delta(q_0, 0, 0) = \{(q_0, 000)\}$

$\delta(q_0, \varepsilon, 0) = \{(q_1, 0)\}$

$\delta(q_1, 1, 0) = \{(q_1, \varepsilon)\}$



stack

① $\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$

⇒ Acceptance by reaching the final state.

② $\delta(q_1, \varepsilon, Z) = \{(q_2, \varepsilon)\}$ stay in the same state

⇒ Empting the stack (F doesn't there) (Only 6 tuples) (we don't have final state)

③ $\delta(q_1, \varepsilon, Z) = \{(q_2, \varepsilon)\}$

⇒ Acceptance by reaching the final state & empting the stack.

Acceptance by Final state:

$L_F(M) = \{w \mid (q_0, w, Z_0) \vdash^* (P, \varepsilon, \gamma)\}$

for some p in F & γ in Γ^*
($p \in F$) ($\gamma \in \Gamma^*$)

2) Acceptance by Emptying the stack:

$L_E(M) = \{w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)\}$
for some p in Q
($p \in Q$)

3) Acceptance by Final state and empty stack:

$L(M) = \{w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)$
for some p in F (p should be in Final state)
($p \in F$) ...

18/10/2022

Equivalance of PDA:

-> If we have a PDA then we have a CFG
-> If we have a CFG then we have a PDA
-> CFG and PDA are equivalent.
-> If both are equivalent then both languages are equivalent.

CFG $\rightleftharpoons$ PDA
L(G) $\rightleftharpoons$ L(PDA)

Conversion of CFG to PDA:

$\rightarrow \delta(q_0, \varepsilon, Z) = \{(q_1, SZ)\}$
1) PUSH first symbol on to the stack.
2) PUSH right hand side side of the production into the stack. (S -> α pop the non terminal S at and PUSH α to the stack)
$\delta(q_1, \varepsilon, A) = \{(q, \alpha)\}$ for each $A \to \alpha$ in P

3) Pop operation (Stay in the same state and do the pop operation)

$\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$ for each $a \in T$

$\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$ reach final state.

(Every step stay in the same state either push or pop)

4) Reaching the final state.

$\boxed{\text{Ex:}}$

1) $S \rightarrow aSA \mid a$

$A \rightarrow bB$

$B \rightarrow b$

$G = (N, T, P, S)$

$N = \{S, A, B\}$

$T = \{a, b\}$

$S \rightarrow$ start symbol.

①

(i) $\delta(q_0, \varepsilon, Z) = \{(q_1, SZ)\}$

(ii) ②

$A \rightarrow \alpha$ then push α

$\delta(q_1, \varepsilon, S) = \{(q_1, aSA), (q_1, a)\}$

(iii)

$A \rightarrow bB$

$\delta(q_1, \varepsilon, A) = \{(q_1, bB)\}$

(iv)

$B \rightarrow b$

$\delta(q_1, \varepsilon, B) = \{(q_1, b)\}$

③

$\forall a \in T$

3) pop operation (stay in the same state and do the pop operation)

$\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$ for each $a \in T$

$\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$ reach final stack.

(Every step stay in the same state either push or pop)

4) Reaching the final state.

EX:

1) $S \to aSA \mid a$

$A \to bB$

$B \to b$

$G = (N, T, P, S)$

$N = \{S, A, B\}$

$T = \{a, b\}$

$S \to$ start symbol.

①

(i) $\delta(q_0, \varepsilon, Z) = \{(q_1, SZ)\}$

(ii) ② $A \to \alpha$ then push α

$\delta(q_1, \varepsilon, S) = \{(q_1, aSA), (q_1, a)\}$

(iii) $A \to bB$

$\delta(q_1, \varepsilon, A) = \{(q_1, bB)\}$

(iv) $B \to b$

$\delta(q_1, \varepsilon, B) = \{(q_1, b)\}$

③ $\forall\, a \in T$

$\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$

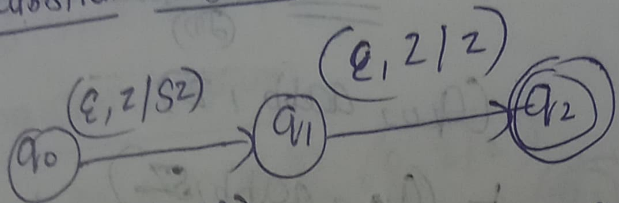
$\delta(q_1, b, b) = \{(q_1, \varepsilon)\}$

④ $\delta(q_1, \varepsilon, z) = \{(q_2, z)\}$

→ Acceptance by reaching the final state.

PDA Tuples: $(Q, \Sigma, \delta, S, Z_0, \Gamma, F)$

Transistion Diagram:



$(\varepsilon, S/aSA)$
$(\varepsilon, S/a)$
$(\varepsilon, a/bB)$
$(\varepsilon, b/b)$
$(a, a/\varepsilon)$
$(b, b/\varepsilon)$

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

$\Sigma = \{a, b\}$

$S \rightarrow q_0$

$Z_0 \rightarrow$ stack start symbol

$\Gamma = \{a, b, z\}$

$F = \{q_2\}$

Validate The PDA:

a	a	b	b

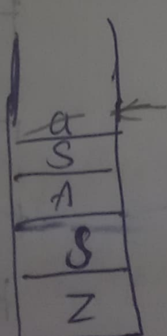
$\delta(q_0, \varepsilon, Z) = \{(q_1 SZ)\}$

$\delta(q_1, \varepsilon, S) = \{(q_1, aSA)\}$

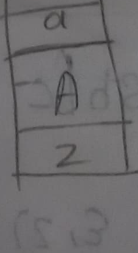
$\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$

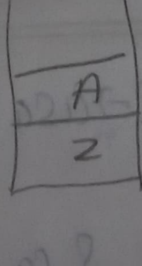
$\delta(q_1, \varepsilon, S) = \{(q_1, a)\}$

$\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$



If terminal is on Top of stack then pop





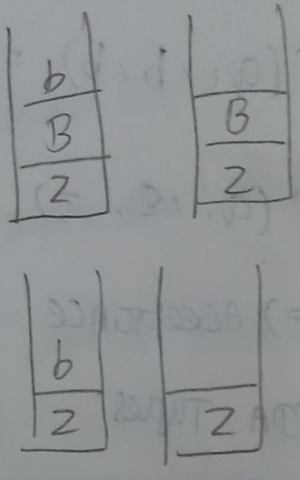
$\delta(q_1, \varepsilon, a) = \{(q, bB)\}$

$\delta(q_1, \varepsilon, b) = \{(q, b)\}$

$\delta(q_1, b, B) = \{(q_1, b)\}$

$\delta(q_1, \varepsilon, b) = \{(q_1, \varepsilon)\}$

$\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$



CFG: (Derivation)

aabb

$S \overset{lm}{\Rightarrow} aSA$

$\overset{lm}{\Rightarrow} aaA$

$\overset{lm}{\Rightarrow} aabB$

$\overset{lm}{\Rightarrow} aabb$ //

(After constructing the PDA we need to validate the PDA)

PDA (Instaneous Description (ID))

$(q_0, aabb, Z)$

$\vdash (q_1, aabb, SZ)$

$\vdash (q_1, aabb, aSAZ)$

$\vdash (q_1, aabb, aSAZ)$

$\vdash (q_1, abb, SAZ)$

$\vdash (q_1, abb, aAZ)$

$\vdash (q_1, bb, AZ)$

$\vdash (q_1, bb, bBZ)$

$\vdash (q_1, b, BZ)$

$\vdash (q_1, b, bZ)$

$\vdash (q_2, \varepsilon, Z)$

$q_2 \in F$ accepted //

---

② $S \to aSa \mid bSb \mid c$

① (i) $\delta(q_0, \varepsilon, Z) = \{(q_1, SZ)\}$

PDA validation:

$(q_0, abcba, Z)$

$\vdash (q_1, abcba, SZ)$

$\vdash (q_1, abcba, aSaZ)$

$\vdash (q_1, bcba, SaZ)$

② $\delta(q_1, \varepsilon, S) = \{(q_1, aSa),$
1) $(q_1, bSb), (q_1, c)\}$

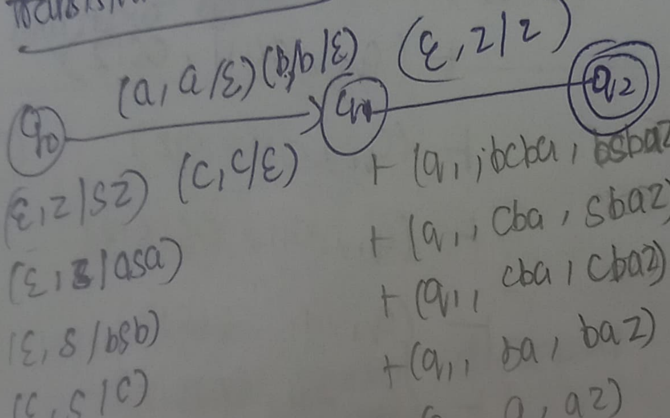
③ $\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$

$\delta(q_1, b, b) = \{(q_1, \varepsilon)\}$

$\delta(q_1, c, c) = \{(q_1, \varepsilon)\}$

④ $\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$

Transition Diagram:



① $S \to 0S1 \mid A$
$A \to 1A0 \mid S \mid \varepsilon$

② $A \to aABC \mid bB \mid a$
$B \to b$
$C \to c$

Validation $\boxed{w = abcba}$

$S \overset{lm}{\Rightarrow} aSa$
$\overset{lm}{\Rightarrow} abSba$
$\overset{lm}{\Rightarrow} abcba$ //

PDA:

$Q = \{q_0, q_1, q_2\}$
$F = \{q_2\}$
Z_0
$\delta_0 \to q_0$
$\Gamma = \{a, b, Z\}$
$\Sigma = \{a, b\}$

$\vdash (q_1, bcba, bSbaZ)$
$\vdash (q_1, cba, SbaZ)$
$\vdash (q_1, cba, cbaZ)$
$\vdash (q_1, ba, baZ)$
$\vdash (q_1, a, aZ)$
$\vdash (q_2, \varepsilon, Z)$ //

---

⑤ $S \to 0S1 \mid A$
$A \to 1A0 \mid S \mid \varepsilon$

① $\delta(q_0, \varepsilon, Z) = \{(q_1, SZ)\}$

② $\delta(q_1, \varepsilon, S) = \{(q_1, 0S1), (q_1, A)\}$

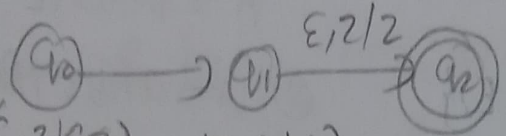
$\delta(q_1, \varepsilon, A) = \{(q_1, 1A0), (q_1, S), (q_1, \varepsilon)\}$

③ $\delta(q_1, 0, 0) = \{(q_1, \varepsilon)\}$

$\delta(q_1, 1, 1) = \{(q_1, \varepsilon)\}$

④ $\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$

Transistion Diagram:



(ε,2/S2) (ε,A/S)

(ε, S/0S1) (ε,A/ε)

(ε, S/A) (0,0/ε)

(ε, A/1A0) (1, 1/ε)

PDA:

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

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

z_0

$S \to q_0$

$\Gamma = \{0,1,2\}$

$F = \{q_0\}$

CFG:

w = 0011

$S \Rightarrow 0S1$

$\Rightarrow 00S11$

$\Rightarrow 00\varepsilon 11$

$\Rightarrow 0011$ //

PDA:

$\delta(q_0, 0011, 2) \vdash (q_1, 0011, S2)$

$\vdash (q_1, 0011, 0S12)$

$\vdash (q_1, 011, S12)$

$\vdash (q_1, 011, 0S1S12)$

$\vdash (q_1, 11, S1S12)$

$\vdash (q_1, 11, A1S12)$

$\vdash (q_1, 11, \varepsilon 1S12)$

$\vdash (q_1, 11, 1S12)$

$\vdash (q_1, 1, S12)$

$\vdash (q_1, 1, A12)$

$\vdash (q_1, 1, \varepsilon 12)$

$\vdash (q_1, 1, 12)$

$\vdash (q_2, \varepsilon, 2)$

$q_2 \in F$ ∴ accepted

20/10/2022

Deterministic PDA's

NPDA – multiple paths are there.

1) $\delta(P, a, B) = (q, \gamma)$

There must be at most one choice of move for any state

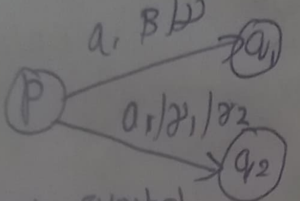
2) $\delta(P, \varepsilon, B)$ is not empty then $\delta(P, a, B)$ must be empty

$a \in \Sigma$, $P \in Q$, $B \in \Gamma$

There must not be a choice between ε and the any input symbol a.

---

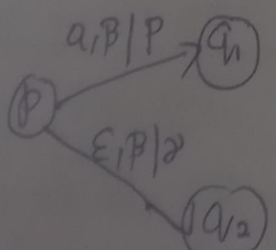
①



Input symbol & Top stack symbol different.
so ⇒ deterministic

---

②



can not make the choice the between a & ε

⇒ Non deterministic.

---

①

$L = \{a^n b^n \mid n \geq 1\}$

write the NPDA

Solution:

$L = \{ab, aabb, aaabbb \ldots\}$

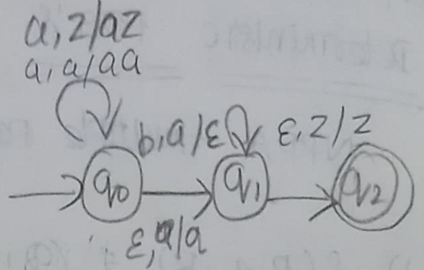
$\delta(q_0, a, z) = \{(q_0, az)\}$

$\delta(q_0, a, a) = \{(q_0, aa)\}$

$\delta(q_0, \varepsilon, a) = \{(q_1, a)\}$

$\delta(q_1, b, a) = \{(q_1, \varepsilon)\}$

$\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$



NPDA:

$Q = \{q_0, q_1, q_2\}$ $\quad \Sigma = \{a, b\}$ ~~Stack~~ $= \{a, b, Z\}$ content (Γ)

$Z_0 \to$ Start stack symbol

$F = \{q_2\}$ $\quad S \to q_0$ (start symbol)

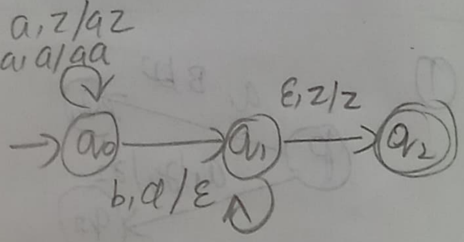
Deterministic PDA:

$\delta(q_0, a, Z) = \{(q_0, aZ)\}$

$\delta(q_0, a, a) = \{(q_0, aa)\}$

$\delta(q_1, b, a) = \{(q_1, \varepsilon)\}$

$\delta(q_1, \varepsilon, Z) = \{(q_2, Z)\}$



① $L = \{a^m b^n c^m \mid m, n \geq 1\}$

Q $L = \{a^m b^n c^m \mid m, n \geq 1\}$

Solution:

$L = \{abc, aabcc, aaabbccc, aabbbcc \ldots\}$

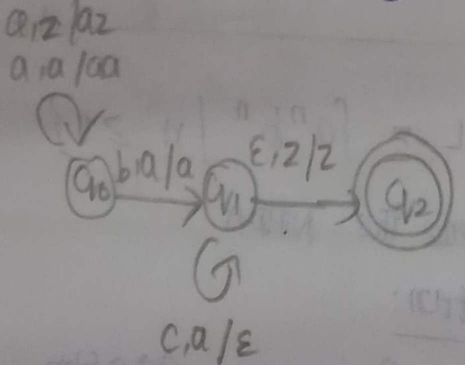
DPDA Transition:

$\delta(q_0, a, Z_0) = \{(q_0, aZ)\}$

$\delta(q_0, a, a) = \{(q_0, aa)\}$

$\delta(q_0, b, a) = \{(q_1, a)\}$

$\delta(q_1, c, a) = \{(q_1, \varepsilon)\}$



$\delta(q_1, c, a) = \{(q_1, \epsilon)\}$

$\delta(q_1, \epsilon, z) = \{(q_2, z)\}$

27/10/2022

Pumping lemma for CF languages:

→ To prove that the language is not a CF languages.

① Let L be a RE 'n' (no of states)

$|w| \geq n$

$w = xyz$

$|xy| > n$

$y \neq \epsilon \quad |y| > 0$

$\forall i \geq 0 \quad x_i y^i z \in L$

for Regular languages.

② L context free language construct a PDA with

'n' no of states

$|w| \geq n$

$w = uvxyz$

1) $|vxy| \leq n$

2) $|vy| > 0 \quad vy \neq \epsilon$

3) For all $i \geq 0$, $uv^i xy^i z$ is in L

Example:

① $L = \{a^n b^n c^n \mid n \geq = 0\}$

Solution

We cannot construct a PDA. So, it is not a CF.

Assume that

L is a CF

$w = a^P b^P c^P$

$|w| \geq n$

$|w| \geq n$

$w = uvxyz$

(i) $|vxy| \leq n$

(ii) $|vy| \neq 0$

(iii) $\forall i$ $uv^i xy^i z \in L$

$w = uvxyz$

$u = a^p$

$vxy = b^p$ $\qquad |vxy| \leq n$

$vy = p^{p-r}$ $\qquad |vy| > 0$

$z = c^p$

$$u\underbrace{v^ixy^i}_{b}z = u\underbrace{vxy}_{b}\underbrace{(vy)}_{b}^{i-1}z$$

$$= u\ vxy\ (vy)^{i-1}\ z$$

$$= a^p\ b^p\ (p^{p-r})^{i-1}\ c^p$$

Put $i=0$

$$= a^p b^p (b^{p-r})^{-1} c^p$$

$$= a^p b^p (b^{-p} b^r) c^p$$

$$= a^p b^r c^p \notin L$$

Put $i=1$

$$= a^p b^p (b^{p-r})^0 c^p$$

$$= a^p b^p c^p \in L$$

Put $i=2$

$$= a^p b^p (b^{p-r})^1 c^p$$

$$= a^p b^p b^p b^{-r} c^p = a^p b^{2p-r} c^p$$

$$= \notin L$$

$\therefore$ The language L is not a CFL.

---

② $L = \{a^k b^j c^k d^j \mid k,j >= 1\}$

Solution:

Let L be a CFL

$|w| \geq n$

$w = a^p b^q c^p d^q$

$w = uvxyz$

$u = a^p$

$vxy = b^p$ $\quad |vxy| \leq n$

$vy = p^{p-r}$ $\quad |vy| > 0$

$z = c^p$

$$uv^ixy^iz = u\,\underbrace{vxy}_{b}\,\underbrace{(vy)}_{b}^{i-1}\,z$$

$$= u\ vxy\ (vy)^{i-1}\ z$$

$$= a^p\ b^p\ (p^{p-r})^{i-1}\ c^p$$

Put $i=0$

$$= a^p b^p (b^{p-r})^{-1}\ c^p$$

$$= a^p b^p (b^{-p}\ b^r)\ c^p$$

$$= a^p b^r c^p \notin L$$

Put $i=1$

$$= a^p b^p (b^{p-r})^0\ c^p$$

$$= a^p b^p c^p \in L$$

Put $i=2$

$$= a^p b^p (b^{p-r})^1\ c^p$$

$$= a^p b^p\ b^p b^{-r}\ c^p = a^p b^{2p-r} c^p$$

$$= \notin L$$

$\therefore$ The language L is not a CFL.

---

② $L = \{a^k b^j c^k d^j \mid k, j >= 1\}$

Solution:

Let L be a CFL

$|w| \geq n$

$w = a^p b^q c^p d^q$

$u = a^p$ $w = uvxyz$

$vxy = b^q$

$vy = b^{q-r}$ $|vxy| \leq n$

$z = c^p d^q$ $|vy| > 0$

$uv^i xy^i z = u\, vxy\, (vy)^{j-1} z$

$= a^p b^q (b^{q-r})^{j-1} c^p d^q$

put $i = 0$

$= a^p b^q (b^{q-r})^{-1} c^p d^q$

$= a^p \cancel{b^q}\, \cancel{b^{-q}}\, b^r c^p d^q$

$= a^p b^r c^p d^q \notin L$

put $i = 1$

$= a^p b^q (b^{q-r})^0 c^p d^q$

$= a^p b^q c^p d^q \in L$

put $j = 2$

$= a^p b^q (b^{q-r})^1 c^p d^q$

$= a^p b^q b^q b^{-r} c^p d^q = a^p b^{2q-r} c^p d^q \notin L$

$\therefore$ The language L is not a CFL.

---

(3) $L = \{a^n b^{n+1} c^{n+2} \mid n >= 0\}$

Solution:

Let L be a CFL

$|w| \geq n$

$w = a^p b^q c^r$ $\boxed{w = uvxyz}$

$u = a^p$

$vxy = b^q$

$vy = b^{q-r}$ $|vxy| \leq n$

$z = c^r$ $|vy| > 0$

(1) $L = \{a^n b^{n+1}\}$

CFL

→ Pushing the 1st a push 2a's

→ Next push 1 a

(2) $L = \{a^i b^j c^j d^i\}$ $i, j \geq 1$

$uv^ixy^iz = uvxy(vy)^{i-1}z$

$= a^p b^q (b^{q-r})^{i-1} c^r$

Put $\boxed{i=0}$

$= a^p b^q (b^{q-r})^{0-1} c^r$

$= a^p \cancel{b^q} \cancel{b^{-q}} b^r c^r$

$= a^p b^r c^r \notin L$

$\boxed{i=1}$

$= a^p b^q (b^{q-r})^{1-1} c^r$

$= a^p b^q c^r \in L$

$\boxed{i=2}$

$= a^p b^q (b^{q-r})^{2-1} c^r$

$= a^p b^q (b^{q-r})^1 c^r$

$= a^p b^q b^q b^{-r} c^r$

$= a^p b^{2q-r} c^r \notin L$

∴ The language L is not a a CFL

28/10/2022

Unit - 4

Turing machines (TM)

Type 0 Rel - Turing machine

Type 1 CSL - Linear bound Automata

Type 2 CFL - Pushdown automata

Type 3 RL - Finite automata.

$RL \subseteq CFL \subset CSL \subseteq Rel$

→ For some problem we cannot create a Turing machine. That is called unsolvable or undecidable problem.

$uv^i xy^i z = uvxy(vy)^{i-1} z$

$= a^p b^q (b^{q-r})^{i-1} c^r$

Put $\boxed{i=0}$

$= a^p b^q (b^{q-r})^{0-1} c^r$

$= a^p \cancel{b^q} \cancel{b^{-q}} b^r c^r$

$= a^p b^r c^r \notin L$

$\boxed{i=1}$

$= a^p b^q (b^{q-r})^{1-1} c^r$

$= a^p b^q c^r \in L$

$\boxed{i=2}$

$= a^p b^q (b^{q-r})^{2-1} c^r$

$= a^p b^q (b^{q-r})^1 c^r$

$= a^p b^q b^q b^{-r} c^r$

$= a^p b^{2q-r} c^r \notin L$

$\therefore$ The language L is not a a CFL

28/10/2022

Unit - 4

Turing machines (TM)

Type 0	Rel	-	Turing machine
Type 1	CSL	-	Linear bound Automata
Type 2	CFL	-	Pushdown automata
Type 3	RL	-	Finite automata.

$RL \subseteq CFL \subseteq CSL \subseteq Rel$

→ For some problem we cannot create a Turing machine. That is called unsolvable or undecidable problem.

→ problems solved by a Turing machine are called solvable or decidable problem.

* In FA & PDA pointer moves from left to right and have a single infinite end (right)

Turing Machine:

→ It have infinite ~~tab~~ at both sides left & right.

→ pointer can move left direction & right direction.

→ Tap is a memory. we can replace a one tap symbol with another symbol.

→ Initially Tap has blank symbol.

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

Q - A finite set of states

Γ - A set of tape symbols

q_0 - start state

B - The blank symbol in Γ

δ - Transition function.

$\underset{CS}{Q} \times \underset{TS}{\Gamma} \rightarrow \underset{NS}{Q} \times \underset{RTS}{\Gamma} \times \underset{move}{\{L, R\}}$ left, right

Σ - Input symbol

F - A set of final states

Example:

$\delta(q, x) = (p, y, L)$

q - current state

x - current tape symbol

p - next state

y - replace the another tape symbol.

L - left move

Representation of TM:

1) Instantaneous Descriptions
2) Transition Table
3) Transition Diagram

Instantaneous description (ID): ⇒ (I = M or I =)

$x_1 x_2 \cdots x_{i-1}\, q\, x_i\, x_{i+1} \cdots x_n$

$q \rightarrow$ is the current state

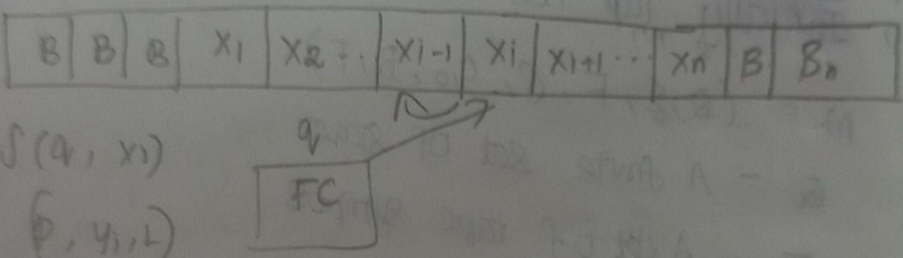
$x_1 x_2 \cdots x_n \rightarrow$ is the portion of the tape between the leftmost and the rightmost nonblank symbols

$n \rightarrow$ Tape length.

$\underbrace{B\,B\,B}_{\text{Blank symbols}}\, x_1\, x_2 \cdots x_n\, \underbrace{B\,B\,B}$

Moves of a TM

⇒ $\delta(q, x_i) = (P, y_i, L)$



$\delta(q, x_i)$

(P, y_i, L)

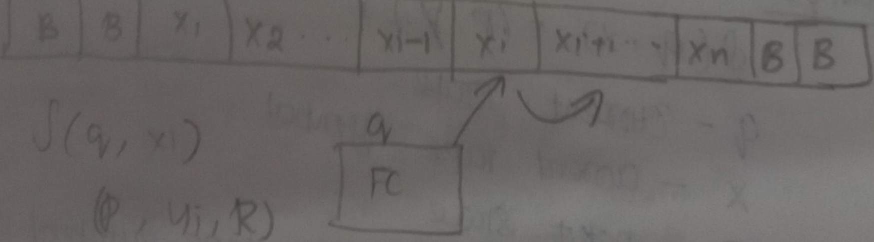
$x_1 x_2 \cdots x_{i-1}\, \underline{q}\, \underline{x_i}\, x_{i+1} \cdots x_n \vdash$

$x_1 x_2 \cdots x_{i-2}\, \underline{P}\, x_{i-1}\, \underline{y_i}\, x_{i+1} \cdots x_n$

Right move

$\delta(q_i, x_i) = (P, y_i, R_i)$

$x_1 x_2 \cdots x_{i-1}\, q\, x_i\, x_{i+1} \cdots x_n \vdash$



$\delta(q, x_i)$

(P, y_i, R)

$x_1 x_2 \cdots x_{i-1}\, \underline{q}\, \underline{x_i}\, x_{i+1} \cdots x_n \vdash$

$x_1 x_2 \cdots x_{i-1}\, \underline{y_i}\, \underline{P}\, x_{i+1} \cdots x_n$

---

① $L = \{0^n 1^n \mid n \geq 1\}$

construct a Turing machine for this language?

$L = \{01, 0011, 000111, 00001111 \cdots\}$

$\boxed{w = 0011}$

$B q_0 0011$

$\vdash x q_1 011$

$\vdash x 0 q_1 11$

$\vdash x q_2 0 y 1$

$\vdash q_2 x 0 y 1$

$\vdash x q_3 0 y 1$

$\vdash x x q_1 y 1$

$\vdash x x y q_1 1$

$\vdash x x q_2 y y$

$\vdash x q_2 x y y$

$\vdash x x q_3 y y$

$\vdash x x y q_4 y$

$\vdash x x y y q_4 B$

$\vdash x x y y q_5 B$

$q_5 \in F$ $\therefore$ The string is accepted

$w = 000111$

$\vdash x q_1 00111$

$\vdash x 0 q_1 0111$

$\vdash x 0 0 q_1 111$

$\vdash x 0 q_2 0 y 11$

$\vdash x q_2 0 0 y 11$

$\vdash q_2 x 0 0 y 11$

$\vdash q_2 x 0 0 y 11$

$\vdash x q_3 0 0 y 11$

$\vdash x x q_1 0 y 11$

$\vdash x x 0 q_1 y 11$

$\vdash x x 0 y q_1 11$

$\vdash x x 0 q_2 y y 1$

$\vdash x x q_2 0 y y 1$

$\vdash x q_2 x 0 y y 1$

$\vdash x x q_3 0 y y 1$

$\vdash x x x q_1 y y 1$

$\vdash x x x y q_1 y 1$

$\vdash x x x y y q_1 1$

$\vdash x x x y q_2 y y$

$\vdash x x x q_2 y y y$

$\vdash x x q_2 x y y y$

$\vdash x x x q_3 y y y$

$\vdash x x x y q_4 y y$

$\vdash x x x y y q_4 y$

$\vdash x x x y y y q_4 B$

$\vdash x x x y y y q_5 B$

$q_5 \in F$ $\therefore$ The string is accepted.

31/10/2022

Designing (construction of) a Turing machine:

Steps:

1) forward direction:

$\delta(q_0, 0) = (q_1, X, R)$

$\delta(q_1, 0) = (q_1, 0, R)$

$\delta(q_1, Y) = (q_1, Y, R)$

$\delta(q_1, 1) = (q_2, Y, L)$

2) Backward direction:

$\delta(q_2, Y) = (q_2, Y, L)$

$\delta(q_2, 0) = (q_2, 0, L)$

$\delta(q_2, X) = (q_0, X, R)$

3) Terminate:

$\delta(q_0, Y) = (q_3, Y, R)$

$\delta(q_3, Y) = (q_3, Y, R)$

$\delta(q_3, B) = (q_4, B, L)$

Turing machine:

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

$Q = \{q_0, q_1, q_2, q_3, q_4\}$

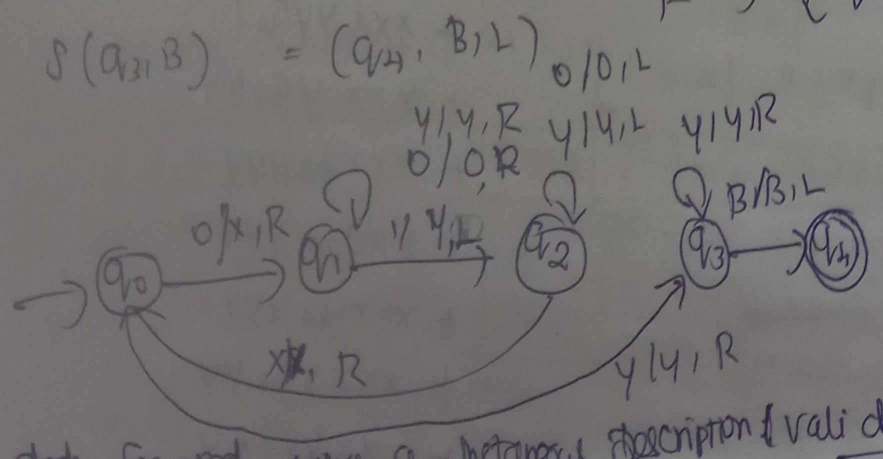
$\Sigma = \{a, b\}$

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

$q_0 \rightarrow$ Initial state

$B \rightarrow$ Blank symbol

$F \rightarrow \{q_4\}$



Check for word using a instantaneous description / validation of a TM:

$w = 0011$

$\underset{q_0}{0}011 \rightarrow X0\underset{q_1}{1}1 \rightarrow X\underset{q_0}{0}Y1 \rightarrow XX\underset{q_2}{Y}Y \rightarrow XXYY\underset{q_3}{B} \quad q_4 \rightarrow$

∴ It is accepted.

① $L = \{a^n b^n c^n, n \geq 1\}$ construct a Turing Machine for this problem?

Solution:

$L = \{abc, aabbcc, aaabbbccc, \dots\}$

$\boxed{w = aabbcc}$

1) **forward direction:**

$w = aaabbbccc$

$\delta(q_0, a) = (q_1, x, R)$

$\delta(q_1, a) = (q_1, a, R)$

$\delta(q_1, b) = (q_2, y, R)$ $\quad \delta(q_1, y) = (q_1, y, R)$

$\delta(q_2, b) = (q_2, z, R)$ $\quad \delta(q_2, z) = (q_2, z, R)$

$\delta(q_2, c) = (q_4, z, L)$

2) **Backward direction:**

$\delta(q_3, z) = (q_3, z, L)$

$\delta(q_3, z) = (q_3, z, L)$

$\delta(q_3, y) = (q_3, y, L)$

$\delta(q_3, x) = (q_0, x, R)$

$\delta(q_2, a) = (q_3, a, L)$

3) **Terminate:**

$\delta(q_0, y) = (q_4, y, R)$

$\delta(q_4, z) = (q_4, z, R)$

$\delta(q_4, B) = (q_5, B, L)$

$\delta(q_4, y) = (q_4, y, R)$

Transition Diagram:

TM:

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

$Q = \{$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, x, y, B\}$

$q_0 \rightarrow$ Initial state

$B \rightarrow$ Blank symbol

$F = \{ \quad \}$

check for the validity:

$w = aabbcc$

$\vdash q_0 aabbcc$

$\vdash x q_1 abbcc$

$\vdash x a q_1 bbcc$

$\vdash x a y q_2 bcc$

$\vdash x a y b q_2 cc$

$\vdash x a y b q_3 bc$

$\vdash x a y q_3 bbc$

$\vdash x a q_3 y bbc$

$\vdash x q_3 a y b z c$

$\vdash q_3 x a y b z c$

$\vdash x q_0 a y b z c$

$\vdash x x q_1 y b z c$

$\vdash x x y q_1 b z c$

$\vdash x x y y q_2 z c$

$\vdash x x y y z q_2 c$

$\vdash x x y y z q_3 z$

$\vdash xxyyq_3zz$

$\vdash xxyq_3yzz$

$\vdash xxq_3yyzz$

$\vdash xq_3xyyzz$

$\vdash xxq_0yyzz$ $\quad q_5 \in F$

$\vdash xxyq_4yzz$ $\quad \therefore$ It is accepted.

$\vdash xxyyq_4zz$

$\vdash xxyyzq_4z$

$\vdash xxyyzzq_4B$

$\vdash xxyyzzq_5B$

1/11/2022

Language Acceptance of TM:

Let $M = \{Q, T, \Sigma, \delta, q_0, B, F\}$ be a TM

The languge accepted by M is

$L(M) = \{w \mid w \in \Sigma^*$ and $\underbrace{q_0 w}_{\text{Initial ID}} \vdash^* \alpha p \beta$ with $p \in F,\ \alpha, \beta \in \underbrace{\Gamma^*}_{\text{combination of Tape symbols}}\}$

Computable Functions:

→ Turing machine have the high capacity.

→ The problems solved by TM can be solved by a modern day computers also.

Turing-computatable Functions:

① $F: \Sigma^* \to \Sigma^*$ Total function

For this type Deterministic TM exist

Ex: $F(x) = x^2$

$$q_0 xB \vdash \underbrace{h}_{\text{final state}} F(x)xB$$

② $F: \underbrace{\Omega}_{\text{Language}} \to \Sigma^*$ Partial function

$$q_0 xB \vdash h\ F(x, B)$$

$\boxed{L(M) = \Omega}$

① F : $F(x) = x+1$

x	$x+1$
1	2
2	3
3	4
⋮	⋮

→ Represent a Natural Number as a unary number.

N	unary representation
1	0
2	00
3	000

whenever found a 0 after that put a Blank symbol that should be replaced by a 00 etc.

x = input = 2
w = 00

$\vdash q_0 00B$
$\vdash 0 q_0 0B$
$\vdash 00 q_0 B$
$\vdash 000 q_1 B$
$q_1 \in F$ ∴ it is accepted

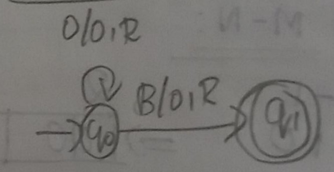
1) forward direction:

$\delta(q_0, 0) = (q_0, 0, R)$
$\delta(q_0, B) = (q_1, 0, R)$

Transition Table:

State	Tape symbols: 0	Tape symbols: B
q_0	$(q_0, 0, R)$	$(q_1, 0, R)$
q_1	–	–

Transition Diagram:



② addition:

$F(x, y) = x+y$

2+3 = 5
00 000   00000

$\boxed{001000}$

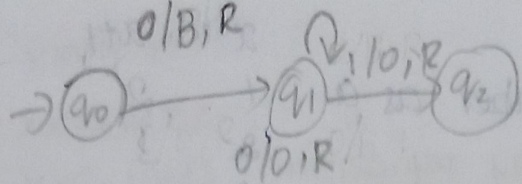
separator

1) replace 1 with 0
2) replace first or last 0 with B

TM:

1) forward direction:

$\delta(q_0, 0) = (q_1, B, R)$
$\delta(q_1, 0) = (q_1, 0, R)$
$\delta(q_1, 1) = (q_2, 0, R)$



Transition table

(3)

State	0	1	B
q_0	(q_1,B,R)		
q_1	$(q_1,0,R)$	$(q_2,0,R)$	
q_2			

x = Input (2)

w = 001000

$\vdash q_0 001000$

$\vdash B q_1 01000$

$\vdash B0 q_1 1000$

$\vdash B00 q_2 000$

∴ accepted

∴ $q_2 \in F$

Since it is

⇒ ∴ accepted

$q_0 001000$

$B q_1 01000$

$B0 q_1 1000$

$B00 q_2 000$

2|11|2022

② subtraction M−N:

Input 3 − 1 ⇒ 2
00010 ⇒ $\boxed{00}$

00010
B0010
B0011
BB011B
BB0BBB
B00BB

Input 2−1 ⇒ 1
0010 ⇒ $\boxed{0}$

0010
B010
B011
BB11
BBBB
B0BB

Instantanious Description: (ID)

W = 0010

$\vdash q_0 0010$

$\vdash B q_1 010$

$\vdash B0 q_1 10$

$\vdash B01 q_2 0$

$\vdash B0 q_3 11$

$\vdash B01 q_4 1$

$\vdash B01 q_5 B$

$\vdash$

Input

1 - 2

0100

↓

B 11

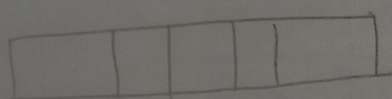
$\boxed{B B}$

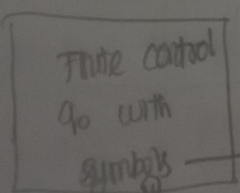
For proper subtraction we don't have a negative symbols.

Programming Techniques of Turing machine:

1) Storage in the Finite control
2) Multiple Tracks
3) Checking off symbols
4) Subroutines. (similar to function)

Storage in Finite control :





→ [ ] represented as

Storage value

Ex. q_0, q_1

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

$[q_0, 0], [q_0, 1], [q_1, 0], [q_1, 1]$ $[q_0, B], [q_1, B]$

$L = 01^* + 10^*$

$L = \{0110, 001110, 1000 \ldots\}$

0111 ↑

$[q_0, B]$

$[q_1, 0]$

$([q_1, 0], 1)$

$([q_1, 0], B)$

1000 ↑

$[q_0, B]$

$[q_1, 1]$

$([q_1, 1], 0)$

$([q_1, 1], B)$

$([q_1, 0], B) \Rightarrow ([q_1, B], B, L)$

$([q_1, 1], B) \Rightarrow ([q_1, B], B, L)$

Multiple Tracks:

B	1	0	1	$
B	B	B	b	B
1	0	1	1	B

Finite control

- → Finite at one end
- → Infinite at another one end
- → It has only one pointer. That pointer points to the same cell positions for all others.

Checkocking off symbols:

① $L = \{wcw^R / w \in \Sigma^*\}$ (even & odd palindrome)

B	B	B	B	✓	✓ ...
a	a	b	c	a	b ...

extra Track

✓	✓	✓	✓
B	B	B	B
a	a	b	b

⇒ If we have Blank in both the tapes at that time stop the process

$\delta(q_0, [B, a]) \vdash (q_1, [\checkmark, a, R])$

$\delta(q_1, [B, b]) \vdash (q_2, [\checkmark, b, R])$

$\delta(q_2, [\checkmark, a]) \vdash (q_2, [\checkmark, a, R])$

$\delta(q_2, [\checkmark, b]) \vdash (q_2, [\checkmark, b, R])$

⇒ palindrome whatever you have in the memory that should be in the last symbol.

Forward direction:

$\delta([q_1, B], [B, d]) = ([q_2, d], [\checkmark, d], R)$

$\delta([q_2, d], [B, e]) = ([q_2, d], [B, e], R)$

$\delta([q_2, d], [B, C]) = ([q_3, d], [B, C], R)$

$\delta([q_3, d], [\checkmark, e]) = ([q_3, d], [\checkmark, e], R)$ already read symbols

$\delta([q_3, d], [B, d]) = ([q_4, B], [\checkmark, d], L)$ match the symbols

Backward direction:

$\delta([q_4, B], [\checkmark, d]) = ([q_4, B], [\checkmark, d], L)$

$\delta([q_4, B], [B, C]) = ([q_5, B], [B, C], L)$

$\delta([q_5, B], [B, d]) = ([q_6, B], [B, d], L)$

$\delta([q_5, B], [B, d]) = ([q_5, B], [B, d], L)$ Not necessary

$\delta([q_5, B], [\checkmark, d]) = ([q_5, B], [\checkmark, d], R)$

$\left.\begin{array}{l} d = a/b \\ e = a/b \end{array}\right\}$

checking off symbols Terminate:

$\delta([q_5, B], [\checkmark, d]) = ([q_7, B], [\checkmark, d], R)$

$\delta([q_7, B], [B, C]) = ([q_8, B], [B, C], R)$

$\delta([q_8, B], [\checkmark, d]) = ([q_8, B], [\checkmark, d], R)$

$\delta([q_8, B], [B, B]) = ([q_9, B], [\ , B], L)$

---

② $(a^n b^n \mid \geq 1)$

H.W

3|11|2022

$L = \{wcw \mid w \in \{a,b\}^*\}$

$\Sigma = \{a, b, c\}$

It canot construct using normal turing machine.

B	B	B	B	B
a	b	c	a	b

$[q_1, B], [B, a]$

ab c ab

B B

B B B B
a b c a b

B B B B
a b c a b

B B B B
a b(B) c a b

{If 2 tracks have both B's the track then terminate it}

(After reach the c then shift in the state)

7|11|2022

SUBROUTINES

1) Similar to function call.
2) any state we can move to the state.
3) do it repeatedly.

Multiplication:

=) Repeated addition. (we are going to copy 'n' 0's to the resultant part 'm' no. of times)

Eg: 2x2 = 4

m n
001 00

=) we are going to copy 'm' numbers & 'n' numbers.

Into the result.

0010010000

00100

B

B0100 00

B01XX
0
B0100
BB1X0 $\rightarrow$ 0
BB1XX $\rightarrow$ 00
BB1XX $\rightarrow$ 0000

(In 'm' part we dont have any pending 0's end it)

Transition Table for subroutine multiplication copy:

States	0	1	X	B
q_1	(q_2, X, R)	$(q_4, 1, L)$		$(q_2, 0, L)$
q_2	$(q_2, 0, R)$	$(q_2, 1, R)$	(q_1, X, L)	
q_3	$(q_3, 0, L)$	$(q_3, 1, L)$	$(q_4, 0, L)$	
q_4		$(q_5, 1, R)$		

Inputs:
$Q \times Q$ 00100 using instantaneous description:

$\delta(q_0, 0) = (q_6, 0, R)$
$\delta(q_6, 1) = (q_1, 1, R)$
$\delta(q_0, 0) = (q_6, B, R)$

$q_0$001001 $\vdash$ B$q_6$01001
$\vdash$ B0$q_6$1001
$\vdash$ B01$q_1$001
$\vdash$ B01X$q_2$01
$\vdash$ B01X0$q_2$1
$\vdash$ B01X01q_2B
$\vdash$ B01X0$q_3$10
$\vdash$ B01X$q_3$010
$\vdash$ B01q_3X010
$\vdash$ B01X$q_1$010
$\vdash$ B01XX$q_2$10
$\vdash$ B01XX1$q_2$0
$\vdash$ B01XX10q_2B
$\vdash$ B01XX1$q_3$00
$\vdash$ B01XX$q_3$100
$\vdash$ B01Xq_3X100

B01X01$q_3$0
B01X0$q_3$10

$\vdash$ B01XX$q_1$100
$\vdash$ B01Xq_4X100
$\vdash$ B01q_4X0100
$\vdash$ B0$q_4$100100

$q_4 \in F$
$\therefore$ accepted.

8/11/2022

Extensions of the Turing Machine:-

(i) Multiple tape turing machine:

* Each tape is connected to the finite control with read / write head.
* Each head can point any location (different location)

Eg: x+y

(ii) Two-way infinite Turing Machine:-

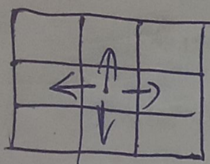
* The tape is infinite in both direction.

(iii) multiple head turing machine:

* uses single tape and multiple head.
* In any state only one head can write or move.
* The head all sense the scanned symbols and move or write independently.

(iv) 2D Tape TM:

* Tape : an infinite 2D grid



(v) Random Access TM:

* It has a fixed no. of registers.
* One way infinite tape and a program counter.

(vi) non-deterministic TM:

$$\delta : Q \times T \rightarrow 2^{Q \times T \times \{L, R\}}$$

* It has more than one options (transistion)

Unit 5

Undecidability

A TM is a 7 tuple

$M = \{Q \cup \{h_a, h_r\}, \Sigma, \Gamma, \delta, q_0, B, T\}$

halt on accept — halt on reject

Δ - Blank symbols

Recursive (R) and recursively enumerable (REL) language:

undecidability → TM halts for accept
↘ TM may or may not halt for reject

decidability → TM halts

(i) decidable problem
(ii) semi decidable problem
(iii) undecidable problem.
(iv) Non recursive

9/11/2022

Non recursive:

When the language is not recursive then it is called Non recursive.

→ For higher level languages the ~~turing~~ turing machine doesn't exist.

Decidable problem:

RL, CFL, CSL, recursive languages are solvable. So, it is decidable.

Undecidable problem:

turing machine exist may or maynot be halt (reject)

turing machine doesn't exist

Non recursive languages:

i) REL

ii) Turing machine doesn't exist.

<u>Decidable problem:</u>

<u>Ex:</u> All problems which are solvable by the Turing machine

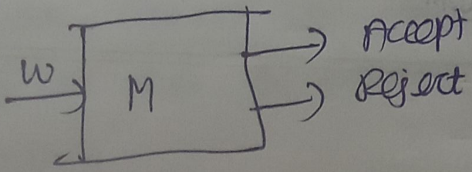
<u>Undecidable problem:</u>

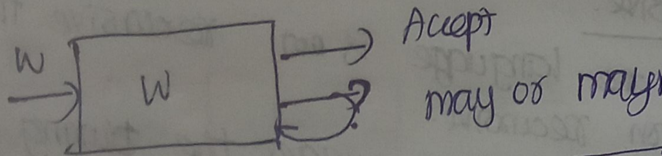
<u>Ex:</u> i) Halting problem — REL

ii) Diagonalization language (higher level) (NREL)

<u>closure properties of REL & Recursive languages:</u>

1) Recursive languages are closed under complementation.

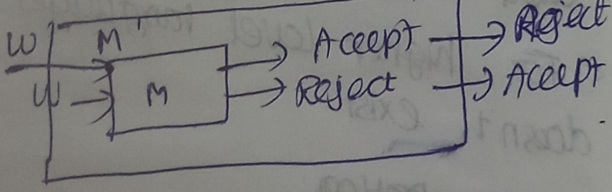




Assume,

L → L'

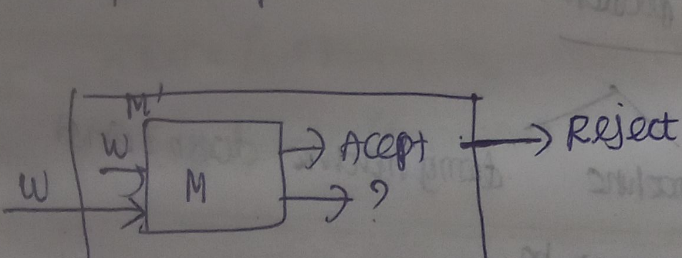
M → M'



2) REL are closed under complementation.

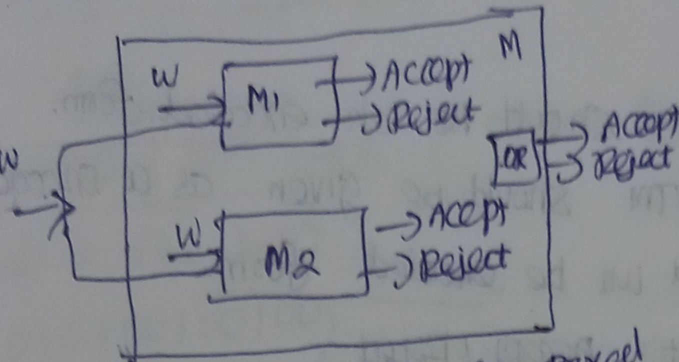
L → M

L' → M'



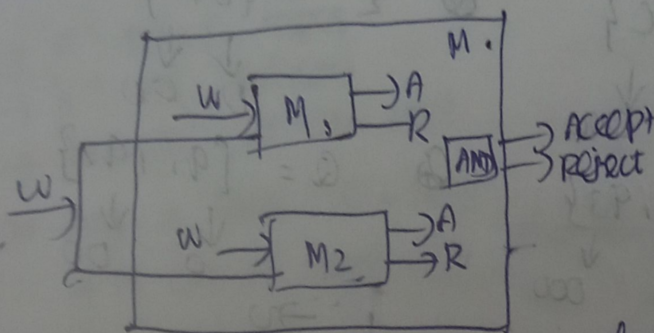
It is not closed under complementation.

3) Recursive languages are closed under union.



It is a recursive language, so proved

4) Recursive languages are closed under Intersection

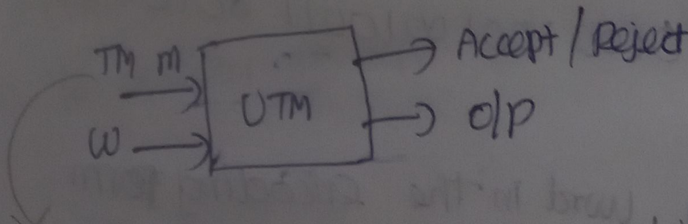


Recursive languages are closed under
U, complementation, Intersection

REL are closed under
↑ complementation, U, Intersection.

Universal Turing machine (UTM):

=) computers are generalized
=) So, TM should be genalized one.
=) universal Turing machine can solve any problem.
=) It takes 2 inputs (original TM & the word w)



If M is produces a output word w then UTM produces a output also.

If TM is accepted then UTM produces a Accept state

TM rejects the w then UTM produces a reject state.

→ Input to the TM should be an encoded form.

→ word to the UTM should be given as a encoded form.

→ Then the output will be encoded form.

→ Then the result Accept / Reject

Encoded form:

① $\Sigma = \{a, b, c\}$ → 0, 00, 000

② $\Sigma = \{0, 1\}$ → 0, 00

③ $Q = \{q_1, q_2, q_3\}$ → 0, 00, 000

④ $Q = \{q_1, q_2\}$ → 0, 00

L → 0

R → 00

⑤ $\Sigma = \{a, b\}$ → 0, 00

⑥ $\Sigma = \{q_0, q_1, q_2\}$ → 0, 00, 000

$0^i 1 0^j 1 0^k 1 0^l 1 0^m$

curr. state, input symbol, next state, replacing symbol, direction

⑦ $\delta(q_0, a) = (q_2, b, R)$

0100 1 00 1 00 1 00

⑧ $\delta(q_2, 1) = (q_3, q_2, R)$

0010 1 000 100 100

1 & 1's the separator for combining & transitions

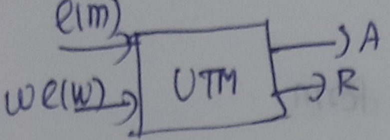
Ans 0100 1001 0010011 00101 000100100.

① w = aba

01001 word in the encoding form.

10/11/2020

Organization of a Turing Machine



/ output the word

$(q_1, a) = (q_2, b, R)$

010100100100 } 010100100100 111 010
aa
010

⇒ Use the concept of Multiple Tape Turing Machine.
- → 'n' no. of tapes
- → 'n' of pointers
- → infinite at one end

⇒ UTM has 4 tape & 4 head.
4th one is used for future use.

→ ① head ① tape
Used for input for UTM.

→ ② tape ② head (Initially word in encoded form)
Tape of M

→ ③ tape ③ head (Initially the Initial state in encoded form)
State of M (Used the finite control to identify the current state)

→ ④ tape ④ head

⇒ 0ⁱ is in ③ tape 0ʲ in ② tape then ok move to the transition in tape ① If it accept move to the next or stop it (reject)

⇒ If it is accept ③ tape 0ⁱ will be replaced by ok.

Theorem: UTM Exist:

Universal language:
The language which is accepted by the UTM.

⇒ The UTM check the tape ① whether the input is done properly. If any errors stops the process in it.

⇒ In tape ② w in encoded form.

⇒ In tape ③ Initial state of M in encoded form.

⇒ In tape ④ others.

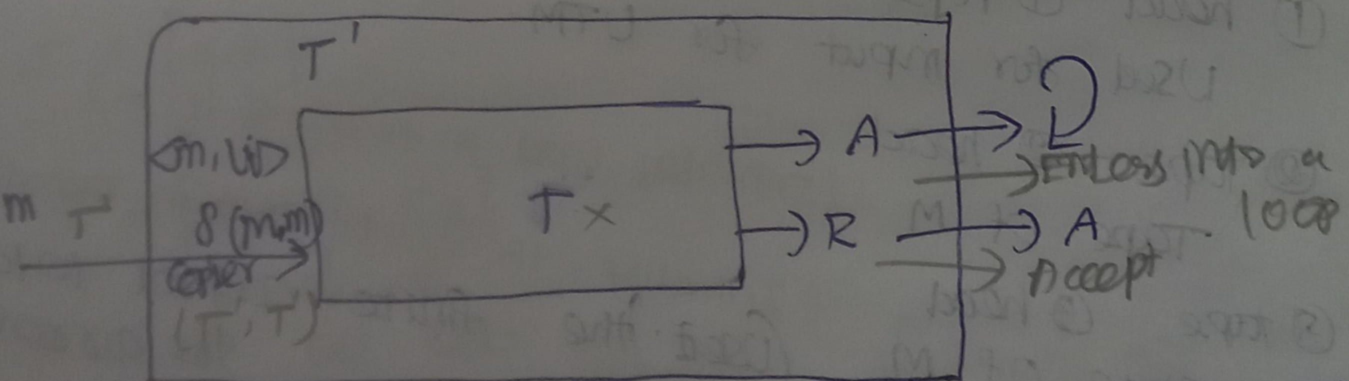
⊛⑧ Universal Turing machine.

Decidable & undecidable problem:

* Acceptance of DFA
* Halting problem
* Busy server problem.

The Halting problem (Rel):

① Whether the turing machine is halt ? or not?



with this contradiction no such Turing machine exist.

It comes under Rel. Because, it accept state accept is the output. But it is in Reject then it is not in a accept state.