Proove :- $(0+1)^*$ is Countable.

b_{11} b_{12} . . . b_{1n} . . .

b_{21} b_{22} - - - b_{2n} - - -

Union of Countable sets is Countable.

Countable Union of Countable sets is Countable.

$\epsilon \cup \{1, 0\} \cup \{01, 10, 00, 11\} \ldots$

1) $(0+1)^*$ is Countable.

2) Number of languages over $(0+1)^*$ is unCountable

$\approx 2^N$ is unCountable

3) Number of machines are Countable.

Date
Page

Undcidablity :-

$\langle M, w \rangle$

$H(\langle M, w \rangle)$ = accept if M accepts w and halts

= rejects otherwise

$A_{TM} = \{ \langle M, w \rangle \mid M \text{ accepts } w \text{ and halt} \}$

Proof :- Suppose that A_{TM} is deciadable,

That means there exists a turing Machine H such that H decides A_{TM}.

H does the following action:

H on input $\langle M, w \rangle$ accepts if M accepts and halt o/w rejects otherwise.

$D\langle M, w \rangle$ → accept if H say reject

→ reject if H says accept

$D\langle D \rangle$ → reject if D accepts $\langle D \rangle$

= accepts if D rejects $\langle D \rangle$

No such D exist, so No such H exist.

$E_{TM} = \{ \langle M \rangle \mid M \text{ is a turing machine and } L(M) = \phi \}$

ATM ~ ETM } Reducibility.

Start with an input of ATM.
Construct an input of ETM.
Prove that a turing machine for ETM will give a turing machine to ATM.

$\langle M, w \rangle \equiv \langle M_1 \rangle \qquad M_1 \xrightarrow{\text{not equal}} \text{Reject}, \quad M_1 \xrightarrow{\text{equal}} M$

How to proove a language is undeciadable

step1 take an input string of a known undecidable problem

step2 Convert their Input to a specific input of your language.

step3 Proove that if a turing machine exists that decides the input in step2, then a turing machine can be designed for the know undeciable problem.

$A_{TM} = \{ \langle M, w \rangle \mid M$ is T.M that accept w and halts $\}$

$E_{TM} = \{ \langle M \rangle \mid M$ is a T.M and $L(M) = \phi \}$

$Eq_{TM} = \{ \langle M_1, M_2 \rangle \mid M_1$ and M_2 are turing machine and $L(M_1) = L(M_2) \}$

Proof:- M_1 is a TM that accepts all strings Σ^*

$M_2 =$ $w' \neq w \in \Sigma^*$ M_2 accept w'
for w, M_2 accept w iff M accept w.

> Compare $\xrightarrow{w \neq w'}$ halt and accept
> Compare $\xrightarrow{w = w'}$ Start M on input w.

$\langle M, w \rangle \Longrightarrow \langle M_1, M_2 \rangle$

of A_{TM} of Eq_{TM}.

Claim Suppose that there exists a T.M H that can decide Eq_{TM}.

Run $\langle M_1, M_2 \rangle$ on H.
if $L(M_1) = L(M_2) \Rightarrow M$ accept w.
if $L(M_1) \neq L(M_2) \Rightarrow M$ does not accept w.

This implies that A_{TM} is decidable, a Contradiction
Therefore such H cannot exist
$\Rightarrow$ Eq_{TM} is undecidable.

Let $L = \{ \langle M, w \rangle \mid M$ halts on some $y \in \Sigma^*$ s.t $|y| \geq |w| \}$

Soln: Reduction from A_{TM}. E_{TM} $\langle M \rangle \Rightarrow L(M) = \phi$
$\Downarrow$
$\langle M', w \rangle$

$(M, \epsilon) \in L \Rightarrow$ there exist a string y, $|y| \geq 0$ s.t M halts and accept y.

$M' = M$
$w' = \epsilon$

$\Rightarrow L(M) \neq \phi$.

$(M, \epsilon) \notin L$ $L(M) = \phi$

Then E_{TM} becomes deciadable which is not true.

Let $L = \{ \langle M, y, q \rangle \mid M$ is a T.M, q is a ~~finite~~ non-hault state, $y \in \Sigma^*$ s.t M enters q on y $\}$

Date
Page

Q Let $L = \{ \langle M \rangle \mid L(M) \text{ is regular} \}$

$\langle M, w \rangle \approx$ if M accept w then $L(M')$ is regular.
else $L(M')$ is non regular.

Q Let $L = \{ \langle M \rangle \mid L(M) \text{ is CFL} \}$

Time - Complexity :-

P : Set of all problems which has polynomial running time.

The Post Correspondence Problem (PCP)

Given a set of Pairs of strings

$$P = \left\{ \left[\frac{t_1}{b_1}\right], \left[\frac{t_2}{b_2}\right], \left[\frac{t_3}{b_3}\right], \dots \left[\frac{t_k}{b_k}\right] \right\}$$

where $t_i, b_i \in \Sigma^*$

Question Does there exist a Sequence $i_1, i_2, \dots i_n$ such that $t_{i_1} t_{i_2} t_{i_3} \dots t_{i_n} = b_{i_1} b_{i_2} b_{i_3} \dots b_{i_n}$.

$i_j \in \{1 \dots k\}$

$$\begin{bmatrix} b \\ ca \end{bmatrix} \begin{bmatrix} a \\ ab \end{bmatrix} \begin{bmatrix} ca \\ a \end{bmatrix} \begin{bmatrix} abc \\ bc \end{bmatrix}$$

1 2 3 4

2 1 3 4

Thm PCP is undecidable.

Thm For a given CFG G, whether G is ambigious.

Proof:- Reduce from PCP.

Let $\begin{bmatrix} t_1 \\ b_1 \end{bmatrix} \cdots \begin{bmatrix} t_k \\ d_k \end{bmatrix}$ are given PCP.

$$L_1 = \{ t_{i_1} t_{i_2} \ldots t_{i_j} c_{i_j} c_{i_{j-1}} \ldots c_{i_1} \}$$

for $1 \leq i_1 \leq k$

$\{c_1 .. c_k\}$ is a set.

$$L_2 = \{ b_{i_1} b_{i_2} \ldots b_{i_j} c_{i_j} c_{i_{j-1}} \ldots c_{i_1} \}$$

$L = L_1 \cup L_2$.

$S \to U \mid V$

$U \to t_i U c_i \mid t_j c_j \quad \forall\, i = 1 \dots k$

$V \to b_i V c_i \mid b_j c_j$

Ch- 6.2
Pg:- 167

PCP

Given cards/dominos with two strings $\begin{bmatrix} t_1 \\ b_1 \end{bmatrix} \begin{bmatrix} t_2 \\ b_2 \end{bmatrix} \cdots \begin{bmatrix} t_k \\ b_k \end{bmatrix}$

Find an arrangement $\begin{bmatrix} t_{i_1} \\ b_{i_1} \end{bmatrix} \begin{bmatrix} t_{i_2} \\ b_{i_2} \end{bmatrix} \cdots \begin{bmatrix} t_{i_n} \\ b_{i_n} \end{bmatrix}$ such that

$t_{i_1} t_{i_2} \dots t_{i_n} = b_{i_1} b_{i_2} \dots b_{i_n}, \; i_j \in \{1 \dots k\}$

MPCP Same input as PCP + One special card/domino $\begin{bmatrix} t_{k+1} \\ b_{k+1} \end{bmatrix}$

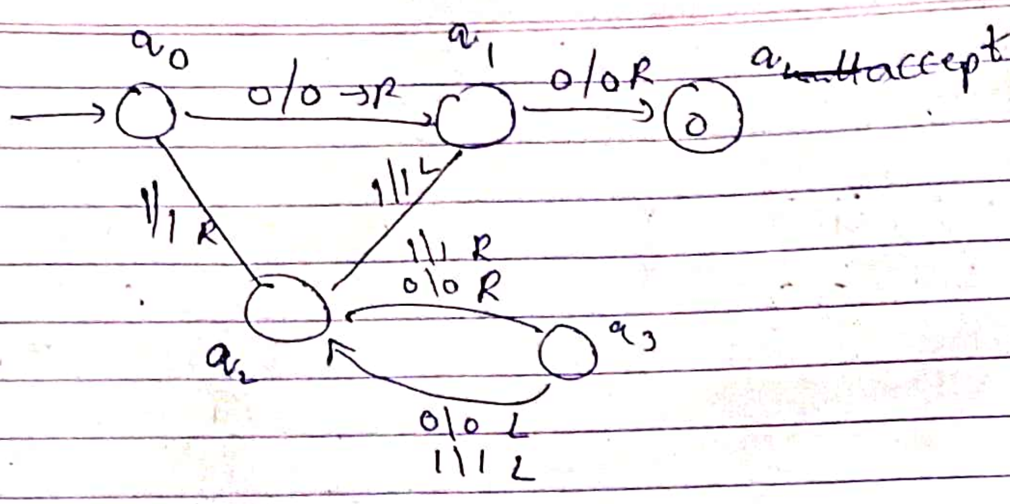
PCP :-

type-1 $\begin{bmatrix} \# \\ \# q_0 w \# \end{bmatrix}$

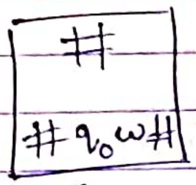
type-2 $\begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} \# \\ \# \end{bmatrix}$

type-3 (Right move): $\delta(q, a) = (q', b, R)$

$\begin{bmatrix} q a \\ b q' \end{bmatrix}$

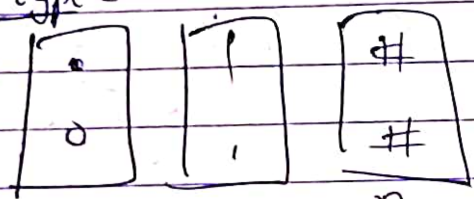


type 1



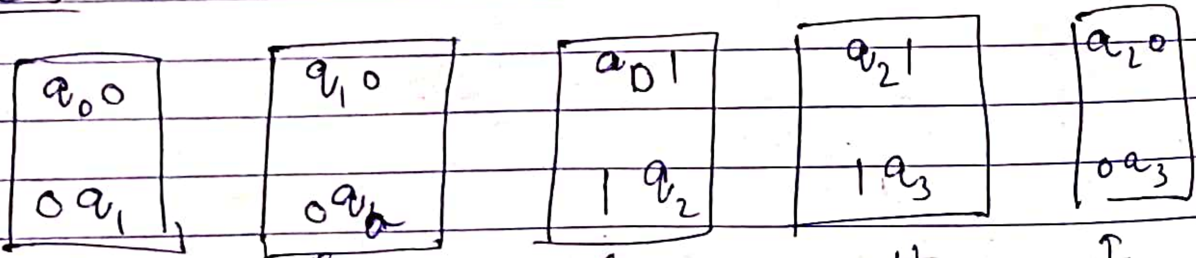
A

type 2



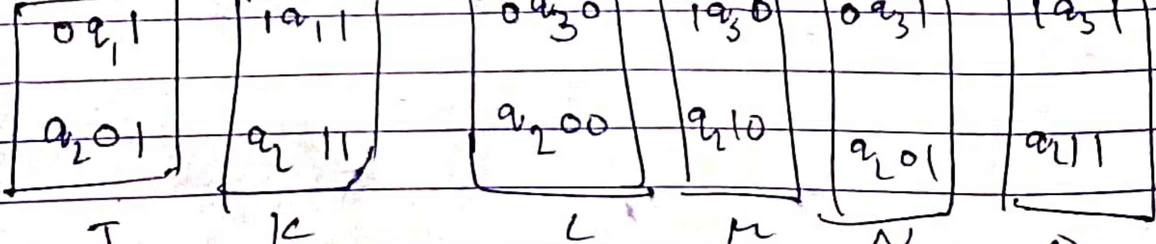
B C D

type 3



E F G H I

type 4 $\delta(q, \bar{a}) = (q', b, L)$



J K L M N O

type 5 $\left(\forall a \in \Sigma \quad \left[\dfrac{q_a a}{q_a}\right] \quad \left[\dfrac{a q_a}{q_a}\right] \right)$

$\left[\dfrac{q_a 0}{q_a}\right]_P \quad \left[\dfrac{0 q_a}{q_a}\right]_R \quad \left[\dfrac{q_a 1}{q_a}\right]_{\cancel{B}\,\phi} \quad \left[\dfrac{1 q_a}{q_a}\right]_S$

type 6

$\left[\dfrac{q_a \# \#}{\#}\right]_T$

If we run $w = 0010$.

$C_1 \quad q_0 0010$

$C_2 \quad 0 q_1 010$

$C_3 \quad 00 q_a 10$

$\# q_0 0010 \# 0 q_1 010 \# 00 q_a 10 \# 00 q_a 0 \# 0 q_a 0 \#$

$\# q_0 0010 \# 0 q_1 010 \# 00 q_a 10 \# 00 q_a 0 \# 0 q_a 0 \# q_0 0 \#$

$q_a 0 \# q_a \# \#$

$q_a \# \#$

Pick A
Pick E
Pick B, C, B, D
Pick B, F
Pick C, B, D
Pick B, B
Pick $\emptyset$, B, D
Pick B, R, B, D
Pick R, B, D
Pick P, D
Pick T

Reduction for MPCP to PCP :-

$$\begin{bmatrix} t_1 \\ b_1 \end{bmatrix} \begin{bmatrix} t_2 \\ b_2 \end{bmatrix} \cdots \begin{bmatrix} t_n \\ b_n \end{bmatrix}$$

Special Card

$$\begin{bmatrix} *t_1 \\ *b_1* \end{bmatrix} \begin{bmatrix} *t_2 \\ b_2* \end{bmatrix} \begin{bmatrix} *t_3 \\ b_3* \end{bmatrix} \cdots \begin{bmatrix} \quad \end{bmatrix} \Bigg| \begin{bmatrix} *\# \\ \# \end{bmatrix}, \begin{bmatrix} *t_1 \\ b_1* \end{bmatrix}$$

Tutorial

1) $\langle M_1, M_2 \rangle$, M_1, M_2 are DTM.

whether $L(M_1) \cap L(M_2) \neq \emptyset$

Sol) Start with A_{TM}. $\langle M, w \rangle$ whether M accepts w

M_1 is a TM that on any input $x \in \Sigma^*$, halts immediately
Then $L(M_1) = \Sigma^*$

M_2 $\#x\#$, M on w.
On any given input $x \in \Sigma^*$, accepts if M accept w.

~~$\Rightarrow$ If M_2 accepts~~

Date
Page

→ If M accepts w iff M_2 accepts Σ^*
→ M accepts w iff $L(M_1) \cap L(M_2) \neq \phi$.
This is the reduction from A_{TM} to L.

⇒ If a turing Machine exists for ATM then only $L(M_1) \cap L(M_2) \neq \phi$ is undecidability

2) a) Reduction from ATM.

input of ATM → $\langle M, w \rangle$

$$M' \begin{cases} Q' & Q \cup q'_{accept} \\ \Sigma' & \Sigma \cup \{\$\} \text{ when } \$ \notin \Sigma \\ \delta' & \delta \cup \{\delta'(q_{accept}, a) = (q_{accept}, \$, R) \\ q'_0 & = q_0. \end{cases}$$

M accepts w iff M' ~~acc~~ prints \$.

If a turing machine exists for $\langle M', \$ \rangle$ then ATM Could be Solved.

b) Let ATM. (M, w) → Case 1)

$$M' = \begin{cases} Q' & = Q \cup \{q'_{accept}\} \cup \{q''\} \\ \Sigma' & = \Sigma \cup \$ \text{ when } \$ \notin \Sigma' \\ \delta' & = \delta \cup \{\delta'(q_{accept}, a) = (q''_1, \$, R) \\ q'_0 & = q_0 \quad \{\delta'(q'', a) = (q_1, \$, L) \\ & \quad \{\delta'(q_1, \$) = (q_2, \$, R) \end{cases}$$

$\delta'(q_{accept}, a) = (q'', \$, R)$
$\delta'(q_1'', a) = (q_1, \$, L)$
$\delta'(q_1, \$) = (q_2, \$, R)$
$\delta'(q_1, \$) = (q_2, \$, \cancel{R}\,L)$
$\vdots$
$\delta'(q_n, \$) = (q'_{accept}, \$, R)$

3) $\langle M \rangle$ is deciable.
$\Sigma = m, \quad Q = n$

ab q a a c d # # #

$Z = m^{12} \cdot n(12+1) \longrightarrow$ Upper bond.

$C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow C_4 \rightarrow C_5 \rightarrow C_3 \rightarrow C_Z$.

4)

Introduction to NP Completeness :- (~~Difficult Problems in~~ CS).

Travelling Salesman problem.

$n! \in \Omega(2^n)$

The class P :- A language is said to be P if there exists a deterministic turing machine that decides the problem in the polynomial of the input size.

The class NP :- A language is said to be in NP if there exists a non-deterministic turing machine that can decide L in Polynomial time.

Eg of P :- $\{a^n b^n \mid n > 0\}$

$O(n^2)$

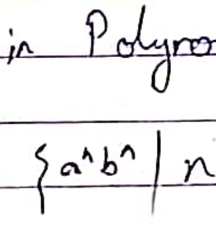
Optimize $O(n \log n)$.

Eg of NP :- $\{0^p \mid p \text{ is prime}\}$

$O(n)$ is non-deterministic machine.

$P \subseteq NP$

$P \neq NP$, $P = NP$ } Not known.



NP hard :- A Problem is said to be N-P hard if every problem of NP can be reducible to M in polynomial time.

$\Rightarrow$ M is as hard as any problem in NP.

NP-Complete :- A ~~Problem~~ language is N-P Complete

(1) if it is in NP.

(2) it is NP-hard.

SAT $x_1, x_2, x_3 \ldots$

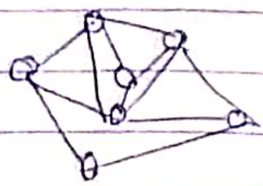
$$\phi = (\bar{x}_1 \vee x_2) \wedge (x_1 \vee x_8 \vee x_8 \vee x_7) \wedge (x_1 \vee x_2 \vee \bar{x}_3)$$

$$\phi = C_1 \wedge C_2 \wedge C_3 \wedge \ldots C_n$$

$C_1 \downarrow$ Clause

3 - SAT :- Each clause has only 3 literals

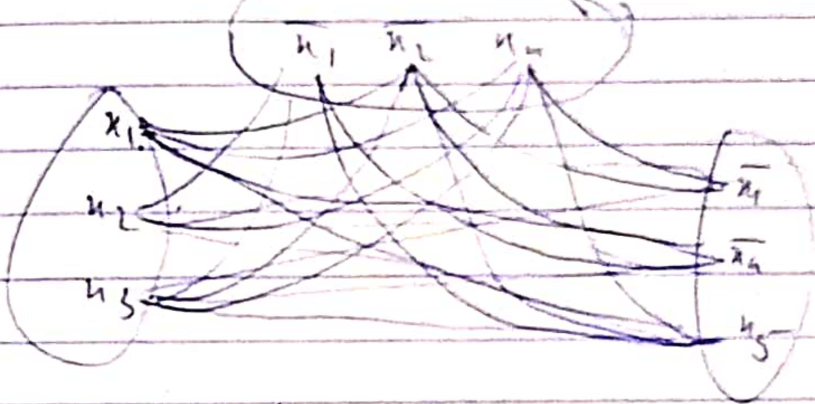
Clique :- $\langle G, k \rangle$



3SAT $x_1, x_2, \ldots x_n$

$\phi = C_1 \wedge C_2 \wedge C_3 \ldots C_n$

$= (x_1 \vee x_2 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee x_4) \wedge (\bar{x}_1 \vee \bar{x}_4 \vee x_5)$



Undecidability Vs NP-Completeness.

Undecidability	NP-Completeness
1) One prob. is undecidable (ATM)	1) One prob is NP-Complete (Proved by Cook SAT)
2) Undecidable proof	2) NP-Completeness proof :-
Reduction from known undecidable problem to unknown problem.	Reduction from known NP-Complete problem to an unknown problem. Reduction should be polynomial time.

$(a \cup b) \wedge (a \cup \bar{b}) = A$ $\quad (a \vee b) \wedge (a \vee \bar{b})$

$a \vee (b \wedge \bar{b})$ $\quad$ $\cancel{a \wedge a}$ $\quad$ $a \vee (b \vee \bar{b})$

SAT $\quad \phi = (x_1) \wedge (x_2 \vee x_3 \vee \bar{x}_4 \vee \bar{x}_1) \wedge (x_1 \vee x_2) \wedge \ldots$

3 SAT $\quad \phi = (x_1 \vee x_2 \vee x_3) \wedge (x_5 \vee \bar{x}_6 \vee x_8) \ldots$

Reduction from SAT

$\phi = C_1 \wedge C_2 \wedge \ldots C_m$

$\phi' = T_1 \wedge T_2 \wedge T_3 \ldots T_n$

Case 1) C_i has 1 variable, Introduce z_1 and z_2

$C_i \to x_2$

$T_i = (x_2 \vee z_1 \vee z_2) \wedge (x_2 \vee \bar{z}_1 \vee z_2) \wedge (x_2 \vee z_1 \vee \bar{z}_2) \wedge (x_2 \vee \bar{z}_1 \vee \bar{z}_2)$

Case 2) C_i has 2 variables, Introduce z

$T_i : (C_i \vee z) \wedge (C_i \vee \bar{z})$

Case 3) C_i has atleast 4 variables, Introduce $(m_1, m_2, \ldots m_{k-3})$

$C_i = l_1 \vee l_2 \vee \ldots l_k$

$t_i = (l_1 \vee l_2 \vee m_1) \wedge (l_3 \vee \bar{m}_1 \vee m_2) \wedge (l_4 \vee \bar{m}_2 \vee m_3) \wedge (l_5 \vee \bar{m}_3 \vee m_4)$

$\ldots (l_{k-1} \vee l_k \vee \bar{m}_{k-3})$

d) Knapsack prob. is not Polynomial time. It is N-P hard.

The above expression is true can be shown by taking that C_j is true for some j

then in the expression t_i we will make all the m_i's before that j^{th} expression should be true and from the j^{th} term we will assign m_i's false. This would garantee that a sol. exists.
Similarly we will show how t_i we get false for all possible values of m_i.

Subset Sum :-

$$S = \{a_1, a_2, \ldots a_n\}$$

W.

3-SAT :-