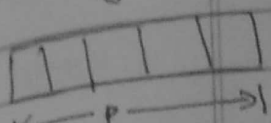
Michael Sipser

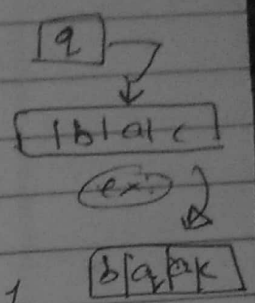
$A_{LBA} = \{\langle M, w\rangle \mid LBA(M) \text{ accepts } w\}$

A_{LBA} is decidable

$|Q| = q$
$|\Gamma| = \gamma$



$q\gamma^{p-1}p \rightarrow$ distinct configurations



Run M on w
→ if M accepts w within $q\gamma^{p-1}p$ steps
accept
→ otherwise reject (looping)

$2q\gamma^{p-1}p + 1$

$E_{LBA} = \{\langle M\rangle \mid LBA(M) \text{ such that } L(M) = \emptyset\}$ undecidable

$\langle M, w\rangle \in A_{TM} \iff \langle B\rangle \in \bar{E}_{LBA}$

B is the LBA such that $L(B) = \{x \mid x \text{ is a computation history of } M \text{ on } w\}$

M on w

start w

b a q c a c

b a a q' a

→ accept

$\delta(q, c) \rightarrow (q', a, R)$

$\delta(q, @) \rightarrow (q', a, R)$ → right

extra symbol

$x \Rightarrow \#x_1\#x_2\#x_3 \cdots \#x_{11}\#x_{12}$

R is a decider
R(⟨B⟩) → accepts ⟨B⟩ if $L(B) = \emptyset$
rejects if $L(B) \neq \emptyset$ $\in A_{TM}$ → ⟨M, w⟩ accepts

---

PCP post correspondence problem

dominos

$$P = \left\{ \begin{bmatrix} t_1 \\ b_1 \end{bmatrix}, \begin{bmatrix} t_2 \\ b_2 \end{bmatrix} \cdots \begin{bmatrix} t_k \\ b_k \end{bmatrix} \right\}$$

$\begin{bmatrix} t_{i_1} \\ b_{i_1} \end{bmatrix} \begin{bmatrix} t_{i_2} \\ b_{i_2} \end{bmatrix} \begin{bmatrix} t_{i_3} \\ b_{i_3} \end{bmatrix} \cdots \begin{bmatrix} t_{i_m} \\ b_{i_n} \end{bmatrix}$

DATE : 28/1/19
PAGE NO. : 2

Total length $t_{i_1} t_{i_2} \cdots t_{i_m} = b_{i_1} b_{i_2} \cdots b_{i_n}$

$y' \rightarrow \left\{ \begin{bmatrix} b \\ ca \end{bmatrix} \begin{bmatrix} a \\ ab \end{bmatrix} \begin{bmatrix} ca \\ a \end{bmatrix} \begin{bmatrix} abc \\ c \end{bmatrix} \right\}$

$\rightarrow \begin{bmatrix} x & \cdots & y \\ x & & y \end{bmatrix}$

$\begin{bmatrix} a \\ ab \end{bmatrix} \begin{bmatrix} b \\ ca \end{bmatrix} \begin{bmatrix} abc \\ c \end{bmatrix} \begin{bmatrix} x \end{bmatrix}$ dominos repeated

Undecidable Problem

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

$t = \# c_1 \# c_2 \# c_3 \# c_4$
$b = \# c_1 \# c_2 \# c_3 \# c_4$

$\begin{bmatrix} t_1 \\ h \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$ $\quad \begin{vmatrix} \# \\ \\ \# q_0 w \end{vmatrix}$ First domino (M, w)

$\delta \in M$

$P = \{ [\,] \cdots [\,] \}$

$\forall a, b \in \Sigma$:

$\begin{vmatrix} q\,a \\ b\,\gamma \end{vmatrix} \leftarrow$ if $\delta(q, a) \rightarrow (\gamma, b, R)$

$\leftarrow \delta(q, a) \Rightarrow (\gamma, b, L)$

$\begin{vmatrix} cq\ a \\ \gamma\epsilon\ b \end{vmatrix}$ $\quad \forall a, b, c \in \Gamma,\ q, \gamma \in Q$

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

DATE : / /
PAGE NO. :

$\begin{vmatrix} 0q & a \\ \gamma_0 & b \end{vmatrix} \begin{vmatrix} 1q & a \\ \gamma_1 & b \end{vmatrix} \begin{vmatrix} 2q & a \\ \gamma_2 & b \end{vmatrix}$

$\forall x \in \Gamma$ First Domino

$\begin{bmatrix} x \\ x \end{bmatrix} \begin{bmatrix} \# \\ \# \end{bmatrix}$ $\begin{vmatrix} \# \\ \# q_0 w \end{vmatrix} \rightarrow \begin{vmatrix} \# \\ \# q_0 0100 \end{vmatrix}$ $w = 0100$

$(q_0, 0) \rightarrow (q_1, 1, R)$ $\begin{bmatrix} q_0 & 0 \\ 1 & q_1 \end{bmatrix}$

qaccepting state

#	q_0	0	1	0	0	#	1	q_1	0	0	#	1	0	q_2	0	#	→
#	q_0	0	1	0	0	#	1	q_1	1	0	0	#	~~q_2~~	1	q_2	0	0	#	1	q_3	0	0	q_4	#

$(q_1, 1) \rightarrow (q_2, 0, R)$ $\begin{bmatrix} q_1 & 1 \\ 0 & q_2 \end{bmatrix}$

$(q_2, 0) \rightarrow (q_3, 0, L) \rightarrow \begin{bmatrix} q_2 & 0 \\ q_3 & 0 \end{bmatrix}$ $\Gamma = \{0, 1, \sqcup\}$

$\rightarrow \left\{ \begin{bmatrix} 0 & q_2 & 0 \\ q_3 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & q_2 & 0 \\ q_3 & 1 & 0 \end{bmatrix} \begin{bmatrix} \sqcup & q_2 & 0 \\ q_3 & \sqcup & 0 \end{bmatrix} \right.$

qaccepting state: $\#0 1 q_{accept} 0 \# \rightarrow \# q_{accept} \#$

So after a series of transition we will reach the $\#q_{accept}\#$ state

$\begin{bmatrix} q_{accept} & a \\ q_{accept} \end{bmatrix} \& \begin{bmatrix} a & q_{accept} \\ q_{accept} \end{bmatrix}$ → to vanish the front & back symbols.

#	0	1	q_{accept}	0	#	→	#	q_{accept}	#	#
#	0	1	q_{accept}	0	#	0	1	q_{accept}	#	#	q_{accept}	#	#

$\langle M, w \rangle \in A_{TM} \iff$ match in P

$\forall a, b, c \in \Gamma$

and $i \exists j$

$$P = \left\{ \begin{bmatrix} \# \\ \# \Sigma \cup \Gamma \end{bmatrix} \begin{bmatrix} q_i a \\ b q_j \end{bmatrix} \begin{bmatrix} c q_i a \\ a_j b \end{bmatrix} \right.$$

$$\begin{bmatrix} \# \\ \sqcup \# \end{bmatrix} \begin{bmatrix} \# \\ \# \end{bmatrix} \begin{bmatrix} \gamma \\ \gamma \end{bmatrix} \begin{bmatrix} q_{accept} a \\ q a_{accept} \end{bmatrix} \begin{bmatrix} a q_{accept} \\ q_{accept} a \end{bmatrix}$$

$$\left. \begin{bmatrix} q_{accept} \# \# \\ \# \end{bmatrix} \right\}$$

Decidability of Logical Theories $\vee, \wedge, \rightarrow, \leftrightarrow, R, \neg, \exists, \forall$

Given a Statement $\begin{cases} \text{true / false} \\ \text{Has a proof} \quad \text{Provable / not Provable} \end{cases}$

(Provable Has a Proof)

or not has a proof

mean $A = 5$

$x = \{ \quad \}$ ~~$x \in A$~~

$x \in A \quad x \leq 5$

$y \in A \quad y \geq 5$

$\{ p \rightarrow (q \rightarrow \gamma), p, \neg \gamma \} \rightarrow \neg q$

$\{ p \rightarrow q, p \} \rightarrow q$

$q \rightarrow \gamma \leftrightarrow \neg \gamma \rightarrow \neg q$

$P \to (Q \to R)$

Proof: ① P given

② $(Q \to R)$

③ $\neg R \to \neg Q$ 3 & 4 are equivalent

④ $\neg R$ given

$\neg Q$

P	Q	R	$Q \to R$	$\neg R$	$P \to (Q \to R)$

$\{P \to (Q \to R) \wedge \{P\} \wedge \{\neg R\} \to \{\neg Q\}\} \vdash \neg Q$

↳ ~~Semantic~~ Syntactic entailment

$\{P \to (Q \to R), P, \neg R\} \models \neg Q$ → Semantic analysis.

$X \vdash Y$

$\vdash X \to Y$

$\vdash \{P \to (Q \to R), P, \neg R\} \to \neg Q$

"If a statement/system" (soundness & logical sys)

① If a statement has a proof: Then it is true

② All true statements are provable and it has a proof. {completeness}

propositional logic → $\vee, \wedge, \to, \leftrightarrow$

predicate logic → $\forall \exists$

$X \vdash Y \to X \models Y$ → soundness

$X \models Y \to X \vdash Y$ → completeness

consistency

$\vdash A$ A has a proof

$\vdash \neg A$ Not A has proof

inconsistent 0 = 1 provable

consistent 0 = 1 Not provable

$\vdash A \wedge \vdash \neg A \models (0 = 1)$

inconsistent

$0 \neq 1$ provable Consistent

$0 \neq 1$ Not provable (X)

can't say anything

Logical → not no natural numbers

↓

Mathematical

DATE: / /

PAGE NO.:

checkability algorithmic sense of consistency

$(\langle \phi, \pi \rangle)$, Proof of ϕ

↑

Statement logical formula.

Q $\phi = \forall x \exists y [y = x + x]$ ← model / interpretation

$\mathbb{R}, \mathbb{N}, (S, +)$

Q ϕ

$\phi : \forall x \in] \exists y [x \geq y \vee x \leq y]$

$(R, \geq)$

Th(M)

set of all true statements in this logical formula

$M = (\mathbb{N}, +, \times)$

$(\mathbb{N}, +, \times)$ Godel Numbering

union

$\forall x \exists y [(x \geq y) \vee (x \leq y)]$

∴ Symbol ∪ formula ∪ Proof

$\forall, \neg, \vee, \wedge, \rightarrow, \leftrightarrow, \forall, \exists$

1 2 3 4 5

bijection →

$(,), \{,\}, [,] \longrightarrow \mathbb{N}$

$\alpha(x) = n$

$\alpha(\leftrightarrow) = 5$

for symbols

$x_0, x_1, \ldots$ variable

$R_0^{\leftarrow} \; R_1^{\leftarrow}$ → relations

$\beta(6_1 6_2 6_3 \ldots 6_n)$

↓

for formula

$6_i \in$ Symbols

$6_1 = \forall$

$6_2 = x$

$6_3 = \exists$

$\beta(6_1 6_2 6_3 \cdots 6_n) = 2^{\alpha(6_1)}\, 3^{\alpha(6_1 6_2)}\, 5^{\alpha(6_n)} \cdots p_n^{\alpha(6_n)}$

↳ Prime factorization of a particular number & its unique for every number.

Proof: series of formulas.

$Y(S_1 S_2 S_3 \cdots S_n)$

where $S_i \in$ formulas

$= 2^{\beta(S_1)}\, 3^{\beta(S_2)} \cdots p_n^{\beta(S_n)}$

(bijective Map)

Predicate $P_r(m,n)$ m is the ~~Godel~~ Godel number of a proof of a formula whose Godel number is n.

$n \to 6_1 6_2 \cdots 6_n$

$m \to S_1 S_2 \cdots S_m$.

Provability formula:

$$P(g(x)) = \exists x \left[P_r(x,n) \quad (x = m) \right]$$

(x: godel number of proof; g(x): is a formula; n.)

Properties of Predicate

Okay ① $\vdash X \to \nexists x (\vdash P(g(x))$

② $\vdash P(g(x \to y)) \to [P(g(x)) \to P(g(y))]$

if $x \to y$ (implies) and x is provable then y is provable and has a proof

③ $\vdash P(g(x)) \to \vdash P(g(P(g(x))))$

($P(g(x))$: x is provable; $P(g(P(g(x))))$: is provable; $\vdash$: has a proof)

④ $\nvdash P(g(0=1)) \to$ consistent

$\forall x \exists y [x \geq y \vee x < y]$ | Quantised

↓ free variable

Theorem

Definition

DATE : / /

PAGE NO. :

$[P(1) \wedge \forall n (P(n) \rightarrow P(n+1)] \rightarrow P(n)$

$P(n)$ = Sum of n natural numbers $= \frac{n(n+1)}{2}$

Enumerate All logical formula of a single free variable n.

$B_1(n), B_2(n), \ldots B_k(n) \ldots$

$B_k(n) = \neg P(g(B_n(n)))$

$\vdash B_k(n) \leftrightarrow \neg P(g(B_n(n))$ $\quad p \rightarrow p \wedge p \leftarrow p$

$1 - B_k(n) \leftrightarrow \neg P(g(B_n(n))$

$1 - B_k(k) \leftrightarrow \neg P(g(B_k(k))$

1. $\vdash x \rightarrow \vdash P_i(g(x))$
2. $\vdash P(g(x \rightarrow y)) \rightarrow [P(g(x)) \rightarrow P(g(y))]$
3. $\vdash P(g(x)) \rightarrow P(g(P(g(x))))$
4. $\nvdash P(g(0=1))$
5. $\vdash P \leftrightarrow \neg P(g(A))$
6. $\vdash P(g(A)) \rightarrow P(g(\neg A))$
7. $\vdash P(g(A)) \rightarrow P(g(0=1))$
8. $\nvdash \neg A$
9. $\nvdash A$
10. $\nvdash \neg P(g(0=1))$

$B_0(n), B_1(n), B_2(n) \ldots B_j(n) \ldots B_n(n)$

$B_k(n) = \neg P(g(B_n(n))) = p$

$\vdash B_k(n) \leftrightarrow \neg P(g(B_n(n)))$

↓

n is a free variable, $(\mathbb{N}, +, \times)$ has SAn in free

$\vdash B_k(k) \leftrightarrow \neg P(g(B_k(k)))$

$\vdash A \leftrightarrow P(g(A))$ $\quad B_k(k) = A$

~~$\vdash P(g(A)) \rightarrow \neg A$~~

$\vdash P(g(A)) \longrightarrow \neg A$

DATE: / /
PAGE NO:

$\vdash P(g(\neg P(g(A) \longrightarrow \neg A))) \rightarrow [P(g(P(g(A))) \rightarrow P(g(\neg 1))]$

1 $\vdash P(g[P(g(A) \longrightarrow \neg A)])$

$\vdash P(g(P(g(A))) \longrightarrow P(g(\neg A))$
$\vdash P(g(A)) \longrightarrow \vdash P(g(P(g(A))))$

$\vdash P(g(A)) \longrightarrow P(g(\neg A))$
$\vdash A \rightarrow (\neg A \rightarrow (0=1))$ ~~$\vdash A$~~
$\vdash P(g[A \rightarrow (\neg A \rightarrow (0=1))])$
$\rightarrow [P(g(A)) \rightarrow P(g(\neg A \rightarrow (0=1))]$
$\vdash P(g(A)) \rightarrow P(g(\neg A \rightarrow (0=1)))$

$\vdash P(g(A)) \rightarrow [P(g(\neg A)) \longrightarrow P(g(0=1))]$

$x \Leftrightarrow (y \rightarrow z) \Leftrightarrow (x \rightarrow y) \rightarrow (x \rightarrow y)$
$\neg x \vee (y \rightarrow z)$
$\neg x \vee (\neg y \vee z)$

$\neg x \vee y \;(\longrightarrow) \neg x \vee z$
$\neg(\neg x \vee y) \vee (\neg x \vee z)$

$\vdash (P(g(A)) \rightarrow P(g(\neg A))) \rightarrow (P(g(A)) \rightarrow P(g(0=1)))$

$\vdash P(g(A)) \rightarrow P(g(0=1))$

$\vdash A$ $\quad\quad$ $\vdash P(g(A))$ 1
contrapositive $\vdash P(g(0=1))$ contradiction.
of 5 $\vdash \neg A$ $\vdash P(g(A))$ $\vdash P(g(0=1))$

$\neg P(g(A)) \rightarrow A$
$\neg A \rightarrow P(g(A))$

Godel Completeness Theorem

Axiom of Choice → contradicting
→ Russel Prorador

DATE : / /
PAGE NO. :

ZFC Axiom. → Present Mathematics Works on this

Godel Numbering: $(\mathbb{N}, +, \times)$
↳ One statem

$|R| > |\mathbb{N}|$ $\quad$ $|R| > |S| > |\mathbb{N}|$

→ Recursion Theorem — not in exam

$q: \Sigma^* \to \Sigma^*$

$q(w)$ is a description of TM P_w (Prints) which prints "w"

P_w (for any input) $q(w) = \langle P_w \rangle$
Erase tape
prints w

WAP which prints the own program.

AB

$A = P_{\langle B \rangle}$ → A is the TM which writes the
$B\{M\}$: compute $q(\langle M \rangle)$ description of B
print $q(\langle M \rangle)\langle M \rangle$ and halt

$\langle S \rangle = \langle AB \rangle$
$B(\langle B \rangle) = q(\langle B \rangle)\langle B \rangle$
$= \langle A \rangle \langle B \rangle$

int main() { ——— $\langle B \rangle$ ———
char f = "int main() { char f=%c%s%c; printf(f,34,f,34); }";
A printf(f,34,f,34); ← B
}

From Beginning

1 The Church-Turing Thesis

DATE : / /
PAGE NO. :

Turing Machine - proposed by Alan Turing (1936)
→ Unlimited & Unrestricted memory

Differences b/w finite automata & turing machine
① A turing machine can both write on the tape and read from it.
② The Read-Write head can move both to the left & to the right
③ The tape is infinite ④ The special states for rejecting & accepting take effect immediately

$\forall T: \Sigma^* \times \Sigma^* \longrightarrow \Sigma^*$
$\exists R: \Sigma^* \longrightarrow \Sigma^*$

$R(w) = T(\langle R \rangle, w)$
$R(t) = T(\langle R \rangle, t)$
$= \langle R \rangle t$

Recursion Th → compute with the description.

2 i/p.
Give me any TM → T ← I can compute on → R ←
s.t if I take an $\langle R \rangle$ → description of R

* T = UTM → takes description of TM & input

$T(s,t) = st$ → concatenation

$T(a,b)$ = TM : b simulates a
↳ $R(\langle M \rangle) = T(\langle R \rangle, \langle M \rangle) = M(\langle R \rangle)$
↓ TM

Proof of Recursion Theorem??

$A = P\langle BT \rangle$
$B\langle BT \rangle$
$q(\langle BT \rangle)\langle BT \rangle$
$\langle A \rangle \langle BT \rangle$

Exam ***

contradiction.

R(t)
{
get $\langle R \rangle$
$T(R,t)$
}

{$\langle M, w \rangle$ TM M accepts w}

A_{TM} is undecidable using Recursion Theorem.

Assume decidable & H is a decider.

R(w)
{ ↳ get $\langle B \rangle$
Run H on $\langle B, w \rangle$ and do opposite
}

DATE : / /
PAGE NO. :

Kolmogorov Complexity

comparison

$S_1 = 011|0|00111$

$S_2 = 011|01|011011$

↳ (011) 'copy times'

Complexity of S_2

→ Given a ~~bin~~ binary string x, minimum description of string x, denoted by $d(x)$. Δt $\exists$ a string $\langle M, w \rangle$, if M runs on w then it halts and put x in tape, write

$d(x) = \langle M \rangle w = \langle M, w \rangle$

$K(x) = |d(x)|$

$M(w) = x$

↓

Minimum description string (01..)
from this we can generate x

$d(x) = \langle M, w \rangle \quad K(x) = |d(x)|$

$x = 011\,0\,11\;\;011 \longrightarrow i$ times

$d(x) = |011| + |\langle M \rangle| + |\log_2 i|$

$K(x) \leq |x| + c$ ⇒ Same code of Turing machine for computation

↓

without c, just put $|x|$ in tape & halt

Same encoding

$K(xx) \quad K(x)$

↳ x concat x

Min length

$M(w) = x$

$d(x) = \langle M, w \rangle$

$K(x) = |d(x)|$

String → 0110 11 011

k (0110

DATE: / PAGE NO.:

$d(x) = (M, 011, i \text{ times})$

$k(x) = |d(x)| \rightarrow 3 + \lg i + |\langle M \rangle|$ (length)

$k(xx)$ $k(x)$

(1.) $N(\omega) \rightarrow$ Put x in tape

(2.) xx $|d(x)| = |\langle N, \omega \rangle|$ $N(\omega) = x$

$k(x)$

$k(x) + c$

$$k(xx) \leq k(x) + c$$

$$k(xy) \leq 2k(x) + k(y) + c$$

It's hard to identify where x ends and y starts

$x = 01101$

0011110011

x, y

↓

0011110011|01 y

↓ $|x| \, |x| \, y$

↓

$2\log_2 k(x)$

$\therefore 2\log_2 k(x) + k(x) + k(y) + c$

$$k(x) > 2 \lg k(x)$$

This is also undecidable problem:–

$L = \{(x, k) \mid k(x) = k\}$ undecidable

Suppose it is decidable

$L = \{(x, k) \mid K(x) = k\}$ undecidable

binary string ↓ ↳ integer

By contradiction, Assume that L is decidable ∃ a program P which decides L

$K(y) \leq |q|$

$k = K(y) \leq |q| = |P| + \log_2 k + C$

$k > C' \log_2 k + C'|P| + C'C$

$C' > 1$

q:
1. for all binary strings 'y' in the lexicographic order we will check $P(y, k)$
2. As we get first y such that $K(y) = k$
3. P/Py

$\therefore C' \log k + C'|P| + C'C < k = K(y) < |q| = |P| + \log k + C$

$K(y) = \min\{\langle n, w \rangle\}$

→ Contradiction

$K(x) \leq |x| + C$

$K(x) \leq |x| - C$

$M(y) = x$ halt

$\langle M, y \rangle \leq |x| - C$

C- compressible binary string
in compressible by C

C+1 Compressible → C compressible

y first string which has $K(y) = k$

binary strings of length n TPT: incompressible by c

DATE: / /
PAGE NO.:

$y \leftarrow x$.

$|y| = n-c$ v $|y| = n-(c+i)$ $0 \leq$

↓

means x is c+i compressible

Minimum description of string x $y = \langle M, w \rangle$

$0 \leq i \leq n-c$

Possible y

$\therefore 2^0 + 2^1 + 2^2 \cdots 2^{n-c}$

atmost $\Rightarrow 2^{n-c+1} - 1$

$y \rightarrow x$ (each y is generating a unique binary string x)

Atleast $\geq 2^n - 2^{n-c+1} + 1$ is incompressible by c

equivalent of T.M

quiz:
decidable tm.
Just do Sipser book

How to prove
" induction
" contradiction
" well ordering
Extremal Principal
" deduction
" construction

A property of binary strings holds for almost all strings

binary strings of True = $|\{x \in \{0,1\}^* \mid f(x) = \text{True}\}|$

binary strings of False = $|\{x \in \{0,1\}^* \mid f(x) = \text{False}\}|$ $\leq 2^{n-c}$

$\lim_{n \to \infty}$ # binary strings of false / $2^n \rightarrow 0$

For

For ~~all~~ $\forall b > 0$ and for all sufficiently long binary string x, such that $f(x) = \text{False}$

then x is compressible by b.

M (i: binary integer)

{ generate all binary string s in lexicographic order
$f(s) = \text{false}$; prints
}

~~# binary string f false upto length n~~ $\left| \begin{matrix} \{0, 00, 01, 001 \\ 1 \quad 2 \quad 3 \quad 4 \\ i = 3 \end{matrix} \right.$

$$\lim \frac{\text{\# binary string } f \text{ False upto length } n}{\text{Total No. of binary strings upto length } n} = 0$$

$$< \frac{1}{2^n}$$

$d(s) = \langle M, i \rangle$

$|d(s)| \leq |M| + |i|$

↳ length of encoding of M

description

$$\frac{\text{\# of binary strings } f \text{ False upto length } n}{\text{\# of binary string of length } n} < \left(\right)$$

$\{0, 00, 001, 0001, \ldots S_i \ldots\}$

(i)

$\{0, 1, 00, 01, 10, 11\} \mid 001$

$i < 6$

$i < \frac{2^{n+1} - 1}{2}$ *

$d(s) = \langle M, i \rangle$

$|d(s)| = |\langle M \rangle| + |\langle i \rangle|$

$K(s) \leq d(s)$

$K(s) \leq |\langle M \rangle| + \frac{2^{n+1}-1}{2^x}$

$i < \frac{2^{n+1}-1}{2^n} < 2^{n+1-x}$

$\leq |\langle M \rangle| \doteq b$

$\leq |s| - b$

$|\langle M \rangle| + (n+1-x) \leq |s| - b$

$x > |s| + b + |\langle M \rangle| + n + 1$

n=no

b compressible

$\forall b > 0 \quad \forall x \in \{0,1\}^* \wedge |x| > n_0 \quad \wedge f(x) = false$

Then x is b compressible

f is a property of binary strings which holds for ~~all~~ almost all binary strings.

$f(x \in \{0,1\}^*) \rightarrow \{T, F\}$

$\lim_{n \to \infty} \frac{\text{No. of binary strings upto length n when f is False}}{\text{Set}} \rightarrow 0$

n ↳ $2^{n+1}-1$

Upper bound for this fraction

$n \geq n_0 \quad < \frac{1}{2^x}$

M(i: integer)

{ ~~Find ith binary~~ Generate all binary string 's' in lexicographic order & find the ith binary string for which f is false and print it's. }

}

$d(s) = \langle M, i \rangle \quad \left| \{ x \in \{0,1\}^* \; |x| = n \wedge f(x) = false \} \right| < \frac{1(2^{n+1}-1)}{2^x}$

↳ $< 2^{n+1-x}$ (binary)

$|d(s)| = |\langle M \rangle| + i$ → decimal integer

$\leq |\langle M \rangle| + n + 1 - x$

$K(s) \leq |\langle M \rangle| + n + 1 - x$

For b compressible

$$K(s) \leq |s| - b$$

$$\therefore |\langle M \rangle| + n + 1 - x = |s| - b$$

$$x = |\langle M \rangle| + b + 1$$

Upto Midsem → Decidable Problems

Decidable Problems

set of decidable problems

Space can be reused
Time once gone can't come back

Time(n) = { }

Set of all languages which take n [linear] to compute

P = (polynomial) time
run by deterministic TM

NP = non-deterministic TM

$P \stackrel{?}{=} NP$

We can convert a Non-deterministic TM into a Deterministic TM → conversion takes non-polynomial time

Conjecture : things not have been Prove

correctness of algorithm :- Loop invariant

it will maintain through out the process

if $n = k^c$ →

↳ convert it to binary then $O(2^{\log_2 n})$

linear to exponential time

iitk →

NDTM - Deterministic TM M

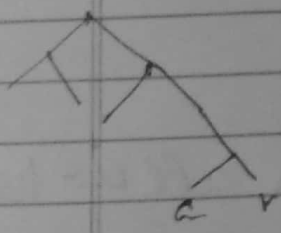
↳ M(w) accept w (NP)

∃ a branch whose leaf is a → accept

∀ branch/leaf ∅ → reject

↳ CONP

complement



HAMPATH = $\{\langle G, s, t\rangle \mid$ ∃ There is a Hamiltonian Path from s to t in G$\}$

NTM {

(i) guess $p_1 p_2 \cdots p_m$ is a permutation of vertices in G

Permutation of V

(ii) if may have covered all v in V | if not reject

(iii) No repetition of vertices

(iv) $p_1 = s$ & $p_m = t$

(v) if p_i & p_{i+1} are connected or not

}

NP = 2 i/p TM M(w, c) → w ∈ HAMPATH

c = guess

↳ certificate of w → Polynomial length

Running time will be measured as

$O(P(|w|))$

Clique = {(G, K): G has a clique of size K or subgraph of K}

Verifier(w, c){
check with the help of (c)
⇔ w ∈ L(verifier)
}

NTM M(w) V(w, c)

L(M)
↓
any accepted by M.

{guess a string which all as the accept to w
V(w, c);
}Run M;

Q = { set of satisfiable Logical Formula}

{guess a satisfying assignment
check (True / False)
}

for verifier.
Pol Pass this assignment as certificate

∀ Polynomial time
∃ NP

BAU B-210

DATE : / /
PAGE NO. :

$\Phi = \{\text{Unsatisfiable } \Phi\} \rightarrow$ co-NP

Prime $\rightarrow [L \in NP \cap co\text{-}NP]$ Which Language

Prime $= \{x \text{ is prime}\} \in P$

Pratt Certificate

Fermat Little Theorem

$gcd(a, n) = 1$

$a^{n-1} \pmod n = 1$