smallest integer that cannot be described in <100 words .

Thursday
26th March.

Berry's paradox : let's take a function,

Sol : natural lang is ambiguous .

$B(n)$ = number of $\underbrace{\text{words}}_{\downarrow}$ to express n
fixed dictionary of english

$B(2) = B(\text{two}) = 1$

$B(52) = B(\text{fifty two}) = 2$

$B(1048676) = B(2^{20}) = B(\text{two power twenty}) = 3$

Let's form a set of simple words .

$S = \{ n \mid B(n) < 20 \}$ → a very large set but finite.

$\bar{S}$ (taken over natural number) $= S \setminus \mathbb{N}$ is infinite .

$b = \min$ of $\bar{S}$ (min of any set of +ve integers)

↳ so b is the min no that cannot be expressed with less than twenty words . — but it is expressed. Thus a paradox exists & it is in dictionary. Because English is not a formal language.

So let's express B with a programming lang.

$KB(n) =$ min no of bytes in Java output n
no of words to express n
no input .

$\therefore S = \{ n \mid KB(n) < 5G \}$

$\therefore b = \min$ of $\bar{S}$ → a Java prog of size < 5G cann't print b

To hava a paradox, we should hava prog < 5G size which prints b

The prog is,

```
limit = 5G
i = 1;
while k(i) ≤ limit
    i++;
output (i);
```

$k(i)$ → This is a function to compute size of i

```
k(i)
{
  :
}
```

(This method can not be written or computed)

output (i) → which eventually will print i.

If $k(i)$ can be written in <5G then we will have a contradiction . so, size of $k(i) > 5G$

K(i) is not computable.

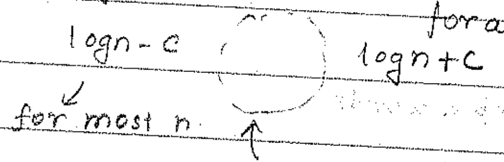
Kolmogorov complexity. KB(n) shortest length of prog
is the size of smallest program to print a number.

HC can be proved with KC.

Those with ~~print~~ pattern have low KC. (it got theoritical limit)

$\forall n.\ K(n) \le \overset{\text{max of } n}{\log n} + C$. (n is the no to print) → complexity of the smallest prog. to print n.

formost n $k(n) \ge \log n - C$



K. random (they are totally incompressable)

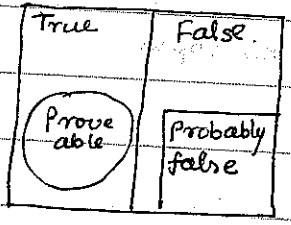
Gödel's theorem.

In every consistent formal system that's strong enough, there is a statement that is true, but not proveable.

(Gödel constructed a stmt true, but not proveable for every consistent formal system,

Formal systems have axioms — in that formal stmts that are true.

Inference rules — Tells about generating new rules.



Formal systems

$1+1=2$
$1+1\ne2$ } A system prove this, is inconsistent.

A system is consistent — cannot prove S True and S false.

A formal system which can express another formal system, then that is strong enough.

Turing understood Gödel proof

Halfstadter Gödel Eschor Bach } they all have self-referencing technique

writing, formal language, drawing, music

Chaitin - reinvented KC again after 4 yrs

"$K(n) \geq \log n - c$" This statement is true but not proveable.

n is komogorov random

Proofs: if we put this as axiom then $|n| > \underset{\text{size of formal system}}{|F|}$

then $K(n)$ doesn't fit in the FS. ∴ $K(n)$ is not proveable in that FS.

Proof 1. $F \rightarrow FS$

$P \rightarrow$ enumerates proof in F

$P(F) \rightarrow$ loops all proofs of F in order.

loop untill find a proof "$K(n) \geq \log n - c$" where

$|n| > |F| + \underbrace{|P|}_{\text{size of prog } P} + c$ ∴ $\log n - c > |F| + |P|$

output n [$|n| = \log n$]

we know $\overset{\log n - c}{|n| - c} > |F| + |P| > K(n) \geq \log n - c$

$\underset{\text{Kom..com}}{K(n)} < |F| + |P|$

Thus we reached to a contradiction.

By $n = 1, 2, \ldots$ we can constrct a large family of sets that follow $K(n) \geq \log n - c$.

Today - Monday

Crawling

Quiz

Review++

Thursday. 2nd April kolmogorov's complexity of the word x.

$\min \text{len} \{ p : p \text{ generates } x \}$

$\underbrace{111\ldots1}_{10^6 \text{ times}}$

```
for(i=0; i<1e6; i++)
    print("1");
```

→ from this, $K(11\ldots1) \leq 29$ estimate of upper bound

produce a program & count the no of letters & finds the size.

non-random sequences : $K(x) \ll \text{len}(x)$

random seq : $K(x) \approx \text{len}(x)$

Berry's paradox : smallest integer that can't be computed by a prog < 100 words.

proof : K(X)

```
  x = 0 ;
 found = true ;
while ( !found )
{ if ( K(X) > 100 )
      found = true ;
  else
      x++ ;
}
```

[polynomial time → best approximation of feasibility]

NP ≡ class of problems

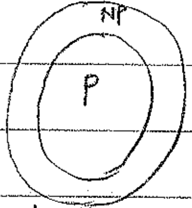
Informally : once u guessed an ans , you can check it in poly time.

Formally : ① a feasible predicate $C(X, Y)$

② a polynomial P_ℓ.

Instances : - given X

- find y s.t. $C(X, Y)$ and $len(Y) \leq P_\ell(len(X))$



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

Reduction :-

$$\begin{tikzcd} X \arrow[r, "?"] \arrow[d, "\log"] & y \\ x' \arrow[r] & y' \arrow[u] \end{tikzcd}$$

(one prob is reduced to ~~another prob~~)

Say, $X = (x_1, x_2)$

[say mul prob reduced to add]

$C(X, Y) \equiv y = x_1 . x_2$

[$\log(a.b) = \log(a) + \log(b)$]

↓

From logarithm table we can find a.b if we know a & b from log table.

$X' = (x_1', x_2')$

$C(x', y') = y' = x_1' + x_2'$

$$\begin{tikzcd} x_1, x_2 \arrow[r] \arrow[d] & x_1 . x_2 \\ \log x_1, \log x_2 \arrow[r] & \log x_1 + \log x_2 \end{tikzcd}$$

$= \log(x_1 . x_2)$

$x_1 x_2 = 10^{\log(x_1 . x_2)}$

$$\begin{tikzcd}
X \arrow[r] \arrow[d, "U_1", "\ln"'] & y \arrow[d, shift right, "U_2"'] \\
x' \arrow[r] & y' \arrow[u, shift right, "U_3"']
\end{tikzcd}$$

$x' = U_1(x)$

$U_3(y') = 10^{y'}$

Algorithms: (should be feasible).

① $U_1(x_1, x_2) = (\log x_1, \log x_2)$

③ if y' is solution to the 2nd problem, then $U_3(y')$ solves the 1st problem ;

→ $\boxed{\text{if } C(U_1(x), y') \text{ is true then } C(x, U_3(y')) \text{ is also true}}$

② if $C(x,y)$ then $C'(U_1(x) U_2(y))$

[$ax^4 + bx^2 + c = 0$

say, $y = x^2$

reduced to, $ay^2 + by + c = 0$

$ax + b = 0$

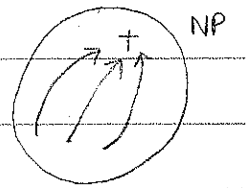
Given: a, b

Find: x

Property,

$C(x,y) \equiv ax + b = 0$]

NP hard ≡ A problem is called NP hard if every problem from the class NP can be reduced to it.



SAT propositional satisfiability :-

Given a Boolean formula: $x_1 \ x_2 \dots x_n$ & V ¬ && || !

Qs: find a combination of values that make the value true.

$(x_1 \vee \bar{x}_2 \vee x_3) \ \& \ (\bar{x}_1 \vee x_2 \vee x_3)$

is T for $x_1 = T, x_2 = T, x_3 = F$

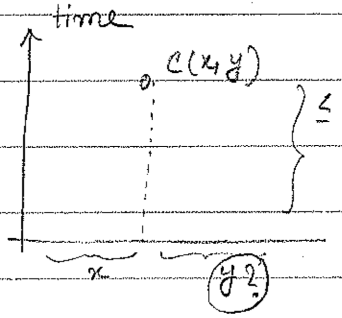
we have an instance of a problem from NP.

Given: x

$C(x,y)$ poly. time

We want to find y ; $C(x,y)$ is true, $len(y) \le P_l(len(x))$

Proof : Reduced to propositional satisfiability.



} ≤ polynomial of len(x) and len(y)

Cells → say memory cells, - - -

smallest size of cells ≥ ΔV

S → largest no of possible states.

time

Δt →

(state of computer can be described by)

$S_{i,t}$ — state of i-th cell at t-th moment.

$S_{i,b,t}$ — b-th bit in i-th cell at t-th moment.