

1) Quiz

2) HW



Test 1: 2006 → web page.

Mar 26

Berry's paradox
Kolmogorov complexity
Gödel's theory

$B(n)$ = no. of words to express n .

fixed dictionary of English

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

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

$$B(1048676) = 3 \quad \text{"two power twenty"}$$

$$S = \{ n \mid B(n) < 20 \} \rightarrow \text{finite} = \# \text{ words in dictionary.}$$

$$\bar{S} \in \mathbb{N} \rightarrow \text{infinite.}$$

$b = \min \text{ of } \bar{S} : \text{min } \# \text{ can't be expressed in } < 20 \text{ words.}$

What is flawed? English is not formal lang.

one word → many meanings
express 1 thing → many ways

Prek Java instead of English:

$K(n) = \min \text{ number of bytes in Java to output } n$

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

limit = 10G;

$i = 1$;

while $K(i) \leq \text{limit}$

$i++$

output(i)

$K(i)$

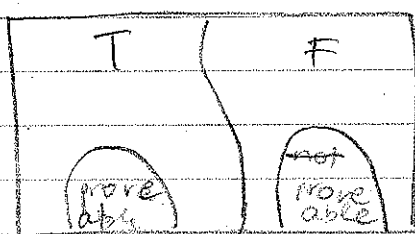
Upper: $\forall n \quad K(n) \leq \log n + c$
 Lower: $K(n) \geq \log n - c$ for most n .

Gödel's Theorem:

→ In every ^{consistent} formal system that's strong enough,
 there is a statement that is true but not provable.

Axiom: statement assumed \rightarrow true.
 Inference rules.

Consistent: possible to prove smth true / false



Hofstadter: Gödel Escher Bach
 math drawing music
 - self-reference

Chaitin

" $K(n) \geq \log n - c$ "

Program P enumerates proofs in F .

$P(F) \rightarrow$ loops all proofs of F in order
 until find a proof:
 $K(n) \geq \log n - c$ for $\log n > |F|$
 output n .

$$\lg n - c \gg |F| + |P| \gg K(n) \gg \lg n - c$$

Jul 2 '09

Kolmogorov complexity of the word x ...

min len of p , p generates x $\{$

+ `||||| | ... ||` \rightarrow `for (i=0; i < 1e6; i++)`
`print('1');`

+ `01...0101...` \rightarrow `for (i=0; i < 5e5; i++)`
`print('01');`

+ `print('010111...01001...')`

- non-random sequences.

$$K(x) \leq \text{len}(x).$$

- random

$$K(x) \sim \text{len}(x).$$

⊕ Barny's paradox

smallest integer that cannot be described in < 100 words.

\rightarrow Exactly this integer is described in ≤ 10 words.

\rightarrow paradox.

computed by a program.

Proof:

`x = 0;`

`found = true;`

`while (!found)`
`{`

`if (K(x) > 100)`

`found = true;`

`else`

`x++;`

`}`

We count the # of char in p .
→ estimate $K(n, 1) \leq 2^n$.
upper-bound

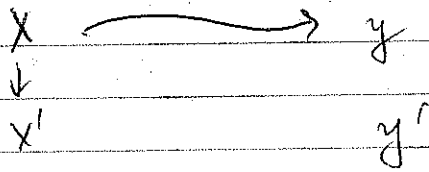
NP : class of problems : non-deterministic problems
→ can't be solved in polynomial time

Informally : You can guess an answer and check it
in poly. time.

Formally : 1) a feasible predicate $c(x, y)$
2) a polynomial

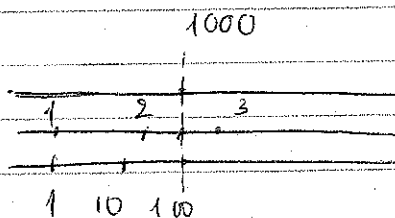
Instances : - x : given
- find y such that $c(x, y)$.
and $\text{len}(y) \leq p_2(\text{len}(x))$.

Reduction :



$$X = (x_1, x_2)$$
$$c(x, y) \equiv y = x_1 x_2$$

slide rule :



$$X \in (X_1, X_2)$$

$$c(X, Y) \equiv y = X_1, X_2$$

$$X' = (X_1', X_2')$$

$$c(X', Y') = y' = X_1' + X_2'$$

$$\begin{array}{ccc} X_1 X_2 & \xrightarrow{?} & X_1 + X_2 \\ \downarrow & & \uparrow \\ \ln X_1, \ln X_2 & \rightarrow & \ln X_1 + \ln X_2 \end{array}$$

- 3 algorithms:

$$\begin{array}{ccc} X & \xrightarrow{?} & y \\ u_1 \downarrow & & u_2 \downarrow \uparrow u_3 \\ X' & \rightarrow & y' \end{array}$$

$$u_1(X_1, X_2) = (\log X_1, \log X_2)$$

if y' is the ^{2nd} solution to the problem &
then $u_3(y')$ solves the 1st problem.

if $c(u_1(X, y'))$ is true,
then $c(X, u_3(y'))$ is also true.

$$u_3(y') = 10^{y'}$$

$$\begin{array}{ccc} X_1 & X_2 & \\ \downarrow & \downarrow & \\ \log X_1 & \log X_2 & \\ \searrow & \swarrow & \\ \log X_1 + \log X_2 & \rightarrow & \log X_1 \cdot X_2 \end{array}$$

$X_1 \cdot X_2$
 \uparrow 10^x

if $c(x, y)$ then $c'(u_1(x), u_2(y))$.

$$\text{Ex: } at^4 + bt^2 + c = 0 \rightarrow az^2 + bz^2 + c = 0$$
$$z = t^2$$

$$\begin{cases} X = (a, b, c) & : \text{given} \\ y = t & : \text{solution} \end{cases}$$

$$\Rightarrow \begin{cases} X' = (a, b, c) \\ y' = t^2 (\neq) \end{cases}$$

$$\begin{cases} u_2 = \text{square} \\ u_3 = \text{square root} \end{cases}$$

+ A problem is NP hard if every problem from class NP can be reduced to it.

SAT : propositional satisfiability

Given : a Boolean formula $X_1 \dots X_n$.

Quest : find a combination of values that make the formula True.

$$(X_1 \vee \bar{X}_2 \vee X_3) \wedge (\bar{X}_1 \vee X_2 \vee X_3)$$

$$\rightarrow \begin{cases} X_1 = T \\ X_2 = T \\ X_3 = F \end{cases}$$

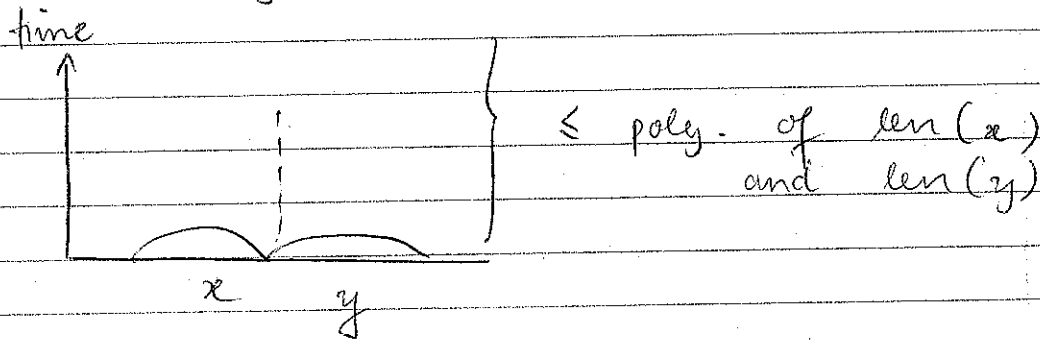
We have an instance of a problem NP

Given : x

$C(x, y)$ poly. time.

We want to find y

$$C(x, y) \text{ is true.}$$
$$\text{len}(y) \leq P_e(\text{len}(x)).$$



S - largest # of possible states

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

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

April 7th

$C(x, y)$

Given: x

Find: $y > t$

$$C(x, y) \wedge \text{len}(y) \leq P_e(\text{len}(x))$$

comp. device

input: x, y

Output: T or F.

Δt - time quantum

Δv - smallest size of the cell.

S - largest states in a cell.

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

B - # of bits to describe an individual state.

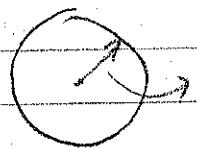
$$2^B \gg S \approx 2^{\log_2 S}$$

$$B \gg \log_2 S$$

$$B = \lceil \log_2 S \rceil$$

S small - s_{it} be b -th bit in the binary descr. of the state S_{it} .

$S_{i,t+1}$: depends on the state of different cells in previous moment of time.



$$r = c \times \Delta t$$

||

light speed

$$N_{\text{neighbor}} \leq \frac{V}{\Delta V} = \frac{4/3 \pi c^3 \cdot \Delta t^3}{\Delta V}$$

speed of light.

$$S_{i,t+1} = f_{i,t} (S_{i,t} \dots S_{j,t})$$

$\leq N_{\text{neigh}}$

$$S_{i,b,t+1} = f_{i,b,t} (S_{i,t} \dots S_{j,t})$$

$t+1$

t

\rightarrow depends on the previous state of cell

⊕ cell; any thing - transistor ...

$S_{i,2,t}$ $S_{i,1,t}$

adder

$$\begin{array}{r} 101001 = S_{i,t} \\ 010111 = S_{j,t} \\ \hline \dots \textcircled{0} \rightarrow S_{i,t+1} \end{array}$$

④ CNF form

$X_1 \dots X_n$ } literals
 $\bar{X}_1 \dots \bar{X}_n$ }

$(a \vee b \vee c)$ $(a \vee b)$

$$(X_1 \vee \bar{X}_2 \vee X_3) \wedge (\bar{X}_1 \vee X_2) \wedge \dots$$

$$f(x_1, x_2) \equiv (x_1 = x_2)$$

x_1	x_2	f
F	F	T
F	T	F
T	F	F
T	T	T

$$f \equiv (\bar{x}_1 \wedge \bar{x}_2) \vee (x_1 \wedge x_2)$$

At the end,

$F_{1,1,t} \& F_{1,2,t} \& \dots \& F_{n,t}$

$$\text{Time } T \leq P_c(\text{len } x + \text{len } y) \leq P_c(n)$$

$$N \leq \frac{V}{\Delta V} = \frac{4}{3\pi C} \cdot \frac{T^3}{\Delta V}$$

$$(S_{s_1, b_1, t} = X_1) \& (\dots) \dots \& S_{s_T, b_T, T} = 1$$

NP \rightarrow computational device to check

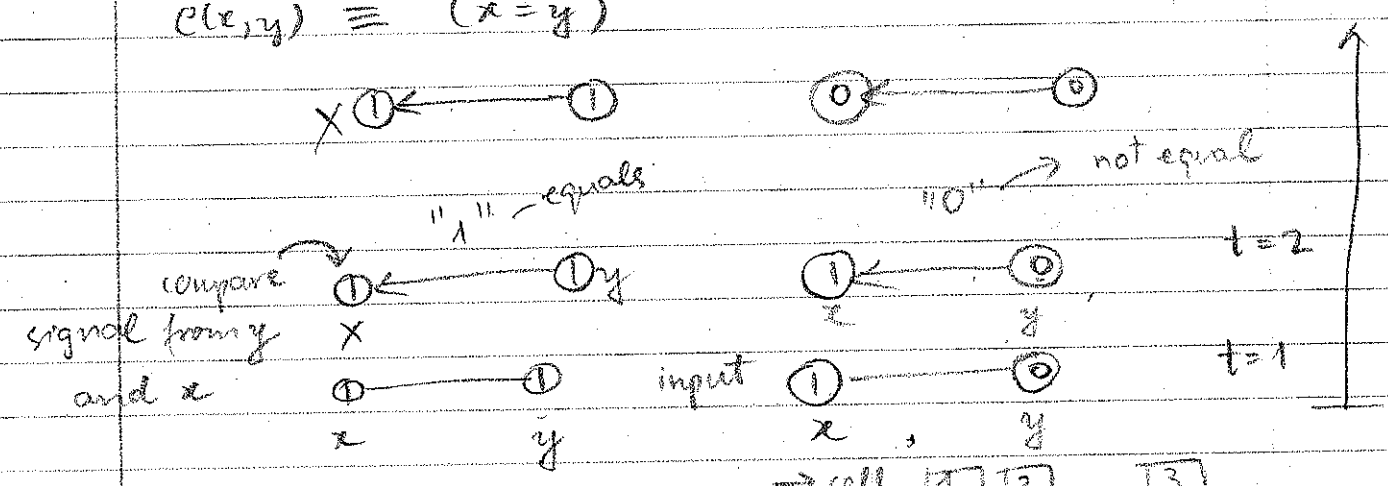
Example :

x - 1 bit

y - 1 bit

$C(x, y) \equiv (x = y)$

equals



\rightarrow cell [1] [2] [3]

3 cells : any part = x, y, 1 wire

State : 1, 0, No signal \rightarrow $S = 3$

\downarrow
2 bits to represent the states

cell x :	00	01	}	S_0
cell y :	00	01		S_1
wire :	no signal	00		S_2
	signal "0"	10		
	signal "1"	11		

ibt

$$S_{112} = 0$$

$$S_{212} = 0$$

$$S_{122} = S_{121}$$

$$S_{222} = S_{221}$$

$$S_{312} = 1$$

$$S_{322} = S_{221}$$

$$\neg S_{112} \& (\bar{S}_{122} \vee S_{222}) \& (S_{122} \vee S_{222}) \&$$

$$(\neg S_{212} \& S_{321} \& (S_{322} \vee S_{221})) \& (S_{322} \vee S_{221}) \\ \& (\neg S_{222} \vee S_{221}) \& (\neg S_{221} \vee S_{222}) \dots$$

$$S_{213} = 0$$

$$S_{223} = S_{222}$$

$$S_{313} = 0$$

$$S_{323} = 0$$

$$S_{113} = 0$$

$$S_{123} = \begin{cases} 1 & \text{if } S_{322} = S_{122} \\ 0 & \text{otherwise} \end{cases}$$

transition

$$\textcircled{P} \rightarrow a = \begin{cases} 1 & \text{if } b=c \\ 0 & \text{else} \end{cases}$$

a	b	c	f	\bar{f}
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0
0	1	1	0	1

$$\bar{f} = (\bar{a} \& \bar{b} \& \bar{c}) \vee (a \& \bar{b} \& c) \vee (a \& b \& \bar{c}) \vee (\bar{a} \& b \& c)$$

$$f = (a \vee b \vee c) \& (a \vee \bar{b} \vee \bar{c}) \& (\bar{a} \vee b \vee \bar{c}) \\ \& (\bar{a} \vee \bar{b} \vee c).$$



: Extra credit

check if $X \rightarrow Y \Leftrightarrow Y \vee \neg X$