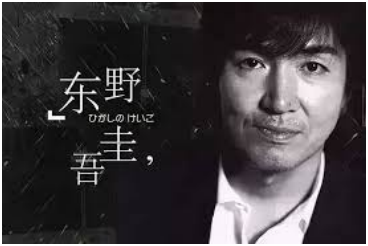
4-11 P and NP

Hengfeng Wei

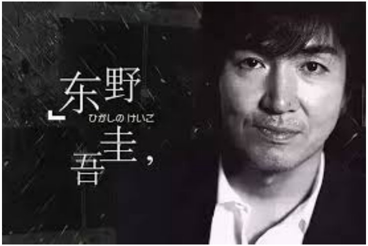
hfwei@nju.edu.cn

May 20, 2019



东野
ひがしの けいご
圭吾,

东野圭吾
嫌疑人X的献身



“对于数学问题，自己想出解答，
和判断别人说的解答是否正确，何者比较简单？”


CONCEPT


decide.

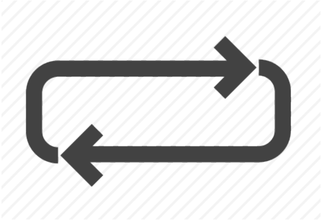

ACCEPT


decide.


ACCEPT


Yes
No
I·DECIDE

Always terminate.



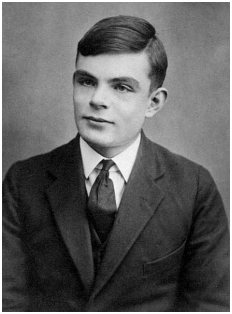
Always terminate.

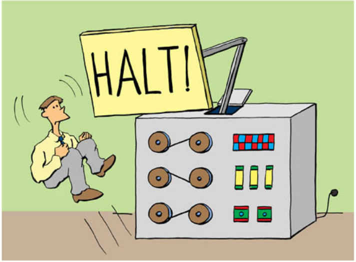
May loop forever for “NO” instance.

Definition (Halting Problem)

Input: An arbitrary program and input

Output: Will the program eventually halt?



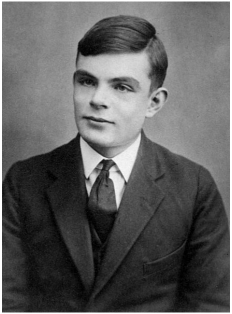


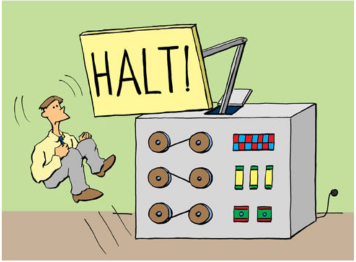
Alan designed the perfect computer

Definition (Halting Problem)

Input: An arbitrary program and input

Output: Will the program eventually halt?





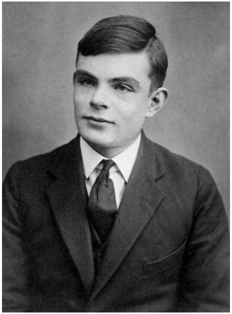
Alan designed the perfect computer

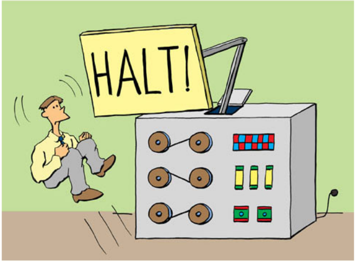
Undecidable

Definition (Halting Problem)

Input: An arbitrary program and input

Output: Will the program eventually halt?





Alan designed the perfect computer

Undecidable

But Acceptable (Semi-decidable)

$$\mathrm{P} = \Big\{ L : L \text{ is decided by a poly. time algorithm} \Big\}$$

$$P = \left\{ L : L \text{ is decided by a poly. time algorithm} \right\}$$

Theorem (Theorem 34.2)

$$P = \left\{ L : L \textit{ is accepted by a poly. time algorithm} \right\}$$

$$\text{P} = \Big\{ L : L \text{ is decided by a poly. time algorithm} \Big\}$$

Theorem (Theorem 34.2)

$$P = \Big\{ L : L \textit{ is accepted by a poly. time algorithm} \Big\}$$

You can safely forget "semi-decidable" in computational complexity theory.

Definition (NP)

$$L \in \mathrm{NP}$$

$$\iff$$

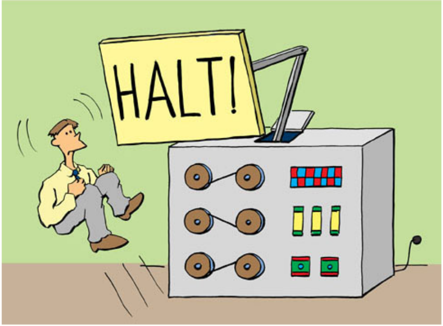
$\exists$ poly. time *verifier* $V(x, c)$ such that $\forall x \in \{0,1\}^*$:

$$x \in L \iff \exists c \in \{0,1\}^*, V(x, c) = 1.$$

NP-problems has short certificates that are easy to verify.

$$\exists L : L \notin \text{NP}?$$

$\exists L : L \notin \text{NP}?$



Alan designed the perfect computer

$\exists L : L \notin \mathrm{NP} \land L$ is decidable?

Theorem (Deterministic Time Hierarchy Theorem (1965))

$$f(n) \log f(n) = o\big(g(n)\big) \implies DTIME\big(f(n)\big) \subsetneq DTIME\big(g(n)\big)$$

$$\exists L : L \notin \text{NP} \land L \text{ is decidable?}$$

Theorem (Deterministic Time Hierarchy Theorem (1965))

$$f(n) \log f(n) = o(g(n)) \implies DTIME(f(n)) \subsetneq DTIME(g(n))$$

$$\text{P} \subsetneq \text{EXP}$$

Theorem (Non-deterministic Time Hierarchy Theorem (Cook, 1972))

$$f(n+1) = o(g(n)) \implies NTIME(f(n)) \subsetneq NTIME(g(n))$$

$$\exists L : L \notin \mathrm{NP} \land L \text{ is decidable?}$$

Theorem (Deterministic Time Hierarchy Theorem (1965))

$$f(n) \log f(n) = o(g(n)) \implies DTIME(f(n)) \subsetneq DTIME(g(n))$$

$$\mathrm{P} \subsetneq \mathrm{EXP}$$

Theorem (Non-deterministic Time Hierarchy Theorem (Cook, 1972))

$$f(n+1) = o(g(n)) \implies NTIME(f(n)) \subsetneq NTIME(g(n))$$

$$\mathrm{NP} \subsetneq \mathrm{NEXP}$$

$\exists L : L \notin \text{NP} \wedge L$ is decidable?

$$\exists L : L \notin \text{NP} \wedge L \text{ is decidable?}$$

"Equivalence of Regular Expressions with Squaring" is NEXP-complete:

$$e_1 \cup e_2, \quad e_1 \cdot e_2, \quad e^2$$

Closure of NP (CLRS 34.2-4)

NP is closed under $\cup, \cap, \cdot, *$.

$$L_1 \in \text{NP}, L_2 \in \text{NP} \implies L = L_1 \circ L_2 \in \text{NP}$$

Theorem

NP is closed under "$*$".

$$c = c_1\#c_2\#\ldots\#c_k\#m_1\&m_2\&\ldots\&m_{k-1}$$

$A^*(x, y) : \forall 1 \le k \le |x|$

$$\begin{aligned} c =& c_1\#c_2\#\ldots\#c_k\#m_1\&m_2\&\ldots\&m_{k-1} \\ & \bigwedge \wedge_{i=1}^{i=k} A(x_i, c_i) \end{aligned}$$

$$x \in L^* \iff \exists c, A(x, c) = 1$$

NP-hard and NP-complete

$$\forall L \in \text{NP}, L \leq_p L' \implies L' \text{ is NP-hard}$$

$$\text{NP-complete} = \text{NP} \cap \text{NP-hard}$$

CLRS 34.5-6

HAM-PATH is NP-complete

$$\text{HAM-CYCLE} \le_p \text{HAM-PATH}$$

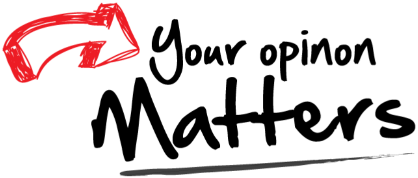
$$\le_p\text{: split } v \text{ into } v_1, v_2\text{; add } s, t, (s, v_1), (v_2, t)$$

Question:

$$\text{HAM-PATH} \le_p \text{HAM-CYCLE}$$

$$\le_p\text{: add } v'; (v', v), \forall v \in V$$


THANK YOU



Office 302

Mailbox: H016

hfwei@nju.edu.cn