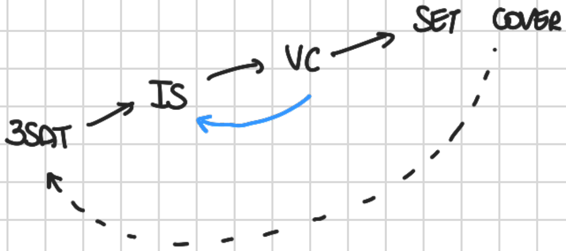
NOTE BOOK

Rokshana Ahmed

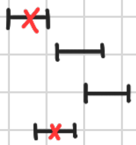


What is P?

DEF: P is the class of decision problems X such that $\exists \gamma \geq 0$ and an algorithm A s.t. for each input I to problem X having n bits, algorithm A determines in time $O(n^{\gamma})$ (in polytime) if I is a "YES"-instance, or a "no"-instance, of problem X.

A decision problem X could be:

- does there exist an interval scheduling with $\geq n/2$ intervals?



"CERTIFICATES"

Sometimes, it is hard to solve a problem, but it is easy to check if a solution is valid:

- 3-SAT is an example
  (we don't know how to find a solution; but, checking whether a solution - a truth assignment - is valid is an easy task)
  $(x_1 \vee \bar{x}_2 \vee x_3)$
- VC is another example
  (we can easily check whether K vertices form a VC)

These are called "YES-CERTIFICATES"

So let us define the concept of certifier

DEF: Let $\gamma \geq 0$, then an algorithm B, that takes as input two strings I (the instance) and C (the certificate) of, respectively, $n = |I|$ bits and $|C| \leq O(n^\gamma)$ bits is an efficient certifier for problem X if

1. the runtime of B is $O(n^\gamma)$,
2. I is a yes-instance of X IFF $\exists$ certificate C s.t. $B(I,C) = TRUE$,
3. I is a no-instance of X IFF $\forall$ 〃 C, $B(I,C) = FALSE$

DEF: NP is the class of problems for which an efficient certifier exists.

L: $P \subseteq NP$

P: Pick arbitrary a problem X in P. Then, $\exists$ algorithm A that solves X in polytime. We aim to show that $X \in NP$. To do so, we need to give an (efficient) certifier B for X. Let us just set $B = A$ (throw away the certificate and just solve the instance).

Non-deterministic Machine

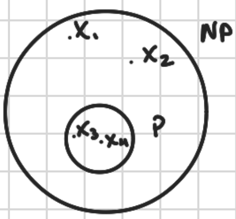
LET $X = [0] * poly(n)$

$X = GUESS(\ )$

$X = [0,0,0..]$ $\quad X = [1,0,0...]$ $\quad X = [0,1,0...]$

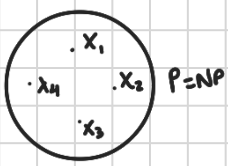
Q: is $P = NP$?

IF $P \neq NP$



?

IF $P = NP$



DEF: If $x \in NP$, and $\forall y \in NP: y \leq_p X$, then X is "NP-Complete".

US / URSS
COOK - LEVIN THEOREM: 3-SAT is NP-Complete.

$(\forall y \in NP: Y \leq_p \text{3-SAT})$

COR: $P = NP$ IFF $\exists x \in P$ s.t. X is NP-Complete

(EXP TIME - COMPLETE)

COR: VC is NP-Complete

" IS " "

" SC " "

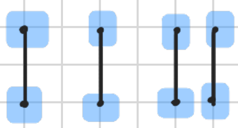
APPROXIMATION ALGORITHMS

APPROX - VC $(G(V,E))$:

- $S \leftarrow \emptyset$
- WHILE $E \neq \emptyset$: ($n/2$ ITERATIONS)
  - PICK ANY EDGE $\{u, v\} \in E$ $O(1)$
  - $S \leftarrow S \cup \underline{\{u, v\}}$ $O(1)$
  - REMOVE FROM E ALL THE EDGES COVERED BY u, OR BY v, OR BY BOTH u AND v. $(deg(u) + deg(v))$
- RETURN S

$\sum_{u} deg(n) \leq 2|E|$

$O(m+n)$



T: APPROX-VC returns a VC that is not larger than twice the smallest VC. (APPROX-VC is a 2-APPROX)