

Definition: Language B is coNP-complete if:

1. $B \in \text{coNP}$
2. Every A in coNP is poly-time reducible to B (i.e. B is coNP-hard)

$\text{UNSAT} = \{ \phi \mid \text{no assignment of the variables satisfies } \phi \}$

Proof that UNSAT is coNP-complete:

(1) $\text{UNSAT} \in \text{coNP}$

(2) UNSAT is coNP-hard:

Let $A \in \text{coNP}$. We show $A \leq_p \text{UNSAT}$

On input w , transform w into a formula ϕ using Cook-Levin

$$w \in \neg A \Leftrightarrow \phi \in \text{SAT}$$

$$w \in A \Leftrightarrow \phi \in \text{UNSAT}$$

$\text{TAUT} = \{ \phi \mid \text{every assignment of the variables satisfies } \phi \}$

TAUT is coNP-complete

Is $P = NP \cap \text{coNP}$?

FACTORING-DECISION = { (m, a, b) | m has a prime factor between a and b inclusive }

FACTORING-DECISION \in NP \cap coNP

Is **FACTORING-DECISION** NP-complete?

If **FACTORING-DECISION** is NP-complete, then NP = coNP

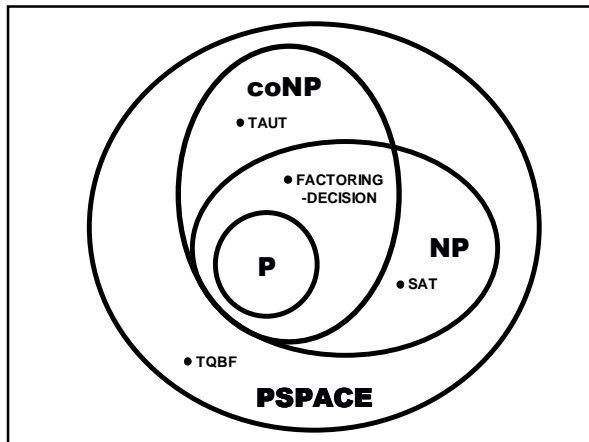
Theorem: **FACTORING-DECISION** \in NP \cap coNP

(1) **FACTORING-DECISION** \in NP

A prime factor of m between a and b is a certificate

(2) **FACTORING-DECISION** \in coNP

The factorization of m is a certificate that (m,a,b) \notin **FACTORING-DECISION**



Definition: $A \in$ coP if and only if $\neg A \in$ P

$$P = \text{co}P$$

Similarly, PSPACE = coPSPACE

Is BPP = coBPP?

Is RP = coRP?

NP-complete problems:

SAT, 3SAT, CLIQUE, HAMPATH, ...

coNP-complete problems:

UNSAT, TAUT, ...

PSPACE-complete problems:

TQBF, GG, ...

(NP \cap coNP)-complete problems:

Nobody knows if they exist

NP, coNP and PSPACE were all defined in terms of specific machines, whereas NP \cap coNP does not have a machine model

ORACLE MACHINES

An oracle is a set B to which the TM may pose membership questions and always receive correct answers after one step of time

M^A denotes the machine M with access to an oracle for A

$P^A = \{ L \mid L \text{ can be decided with a poly-time oracle machine } M \text{ that uses an oracle for } A \}$

P^{SAT} = all languages that can be decided in polynomial time with an oracle for SAT

Is $NP \subseteq P^{\text{SAT}}$?

Is $coNP \subseteq P^{\text{SAT}}$?

Is $NP = NP^{\text{SAT}}$?

Two Boolean formulas ϕ and ψ over the variables x_1, \dots, x_i are equivalent if they have the same value on any assignment to the variables

Are x and $x \vee x$ equivalent?

Are x and $x \vee \neg x$ equivalent?

Are $(x \vee \neg y) \wedge \neg(\neg x \wedge y)$ and $x \vee \neg y$ equivalent?

A Boolean formula is minimal if no smaller formula is equivalent to it

NON-MIN-FORMULA = $\{ \phi \mid \phi \text{ is not minimal} \}$

Theorem: NON-MIN-FORMULA \in NP^{SAT}

Proof:

EQUIV = $\{ (\phi, \psi) \mid \phi \text{ and } \psi \text{ are equivalent} \}$

EQUIV \in coNP

So EQUIV can be decided with an oracle for SAT

NP^{SAT} machine for NON-MIN-FORMULA:

Guess an equivalent smaller formula and test it

Theorem:

(1) An oracle B exists where $P^B = NP^B$

(2) An oracle A exists where $P^A \neq NP^A$

Proof of (1):

Let B = TQBF

Then:

$NP^{TQBF} \subseteq NPSpace \subseteq PSpace \subseteq PTQBF$

(2) An oracle A exists where $P^A \neq NP^A$

For oracle A, define $L_A = \{ w \mid \exists x \in A [|x| = |w|] \}$

Notice that: $L_A \in NP^A$

We show how to construct A so that $L_A \notin P^A$

Let M_1, M_2, \dots be a list of all poly-time oracle TMs

We assume that M_i runs in time at most n^i

We will construct A in stages

Stage i guarantees that M_i^A doesn't decide L_A

All strings
of 0s and 1s

— strings of length 1
— strings of length 2

— strings of length n

In STAGE i :

A finite number of strings have
been assigned to A so far

Pick n greater than the length
of any string in A so far but
such that n^i is smaller than 2^n

Run M_i on 1^n and respond to
its queries as follows:

If M_i queries about a
string y whose status has
already been determined,
respond consistently

Otherwise, respond NO

Theorem:

(1) An oracle B exists where $P^B = NP^B$

(2) An oracle A exists where $P^A \neq NP^A$

WWW.FLAC.WS