CSC363 Tutorial 10

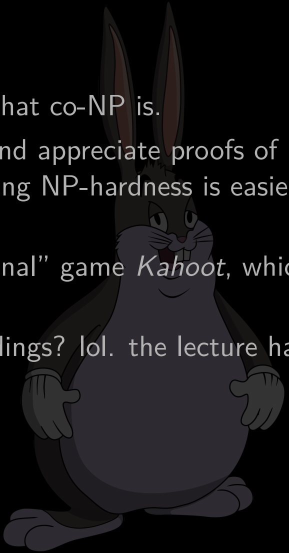
This will take like 40 mins, i'm guessing?

Paul “sushi_enjoyer” Zhang

University of Chungus in Japanese dub

March 30, 2021

Learning objectives this tutorial

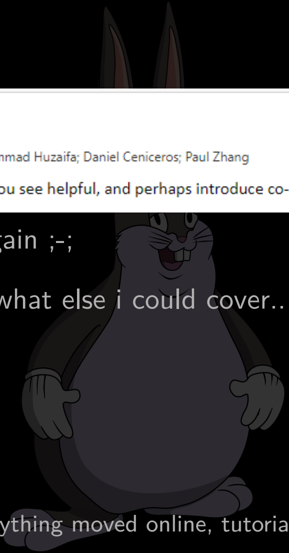


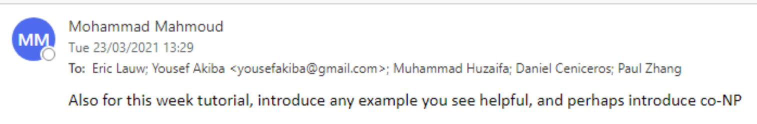
By the end of this tutorial, you should...

- Have an intuitive understanding of what co-NP is.
- Be able to define what NP-hard is, and appreciate proofs of NP-hardness, because ironically proving NP-hardness is easier than proving NP-completeness.
- Be a master of the famous “educational” game *Kahoot*, which is much more enjoyable than amogus.

Big Chungus certified readings: what readings? lol. the lecture hasn’t even happened yet.

AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA




Mohammad Mahmoud
Tue 23/03/2021 13:29
To: Eric Lauw; Yousef Akiba <yousefakiba@gmail.com>; Muhammad Huzaifa; Daniel Ceniceros; Paul Zhang

Also for this week tutorial, introduce any example you see helpful, and perhaps introduce co-NP

oh no... not again ;-;

so uh, i hope youse like kahoot! i dunno what else i could cover... :([1]

[1] last year when i was taking CSC363 and everything moved online, tutorials were just straight up cancelled! D:

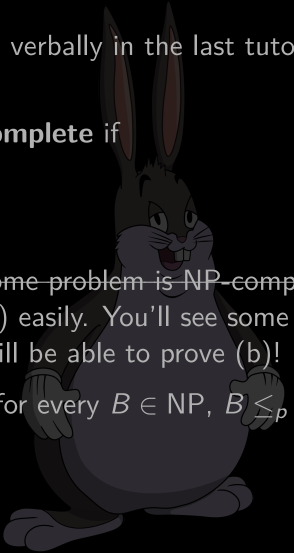
NP-hard 😳

mmm... I briefly mentioned this definition verbally in the last tutorial, but I don't know if any of you remember D:

Recall (maybe): A language A is **NP-complete** if

(a) $A \in \mathrm{NP}$;

(b) For every $B \in \mathrm{NP}$, $B \leq_p A$.



Now, sometimes we ~~get lazy in proving some problem is NP-complete and forget to prove (a)~~ are unable to prove (a) easily. You'll see some examples later. Despite that, we might still be able to prove (b)!

Definition: A language A is **NP-hard** if for every $B \in \mathrm{NP}$, $B \leq_p A$.

NP-hard 😳

mmm… I briefly mentioned this definition verbally in the last tutorial, but I don't know if any of you remember D:

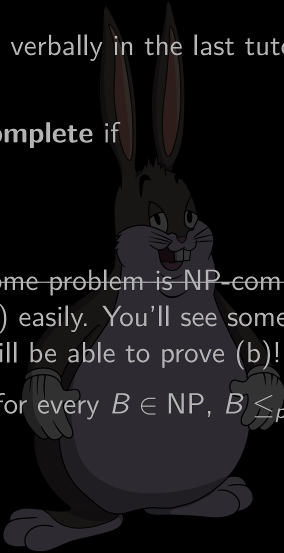
Recall (maybe): A language A is **NP-complete** if

(a) $A \in \mathrm{NP}$;

(b) For every $B \in \mathrm{NP}$, $B \leq_p A$.

Now, sometimes we ~~get lazy in proving some problem is NP-complete and forget to prove (a)~~ are unable to prove (a) easily. You'll see some examples later. Despite that, we might still be able to prove (b)!

Definition: A language A is **NP-hard** if for every $B \in \mathrm{NP}$, $B \leq_p A$.

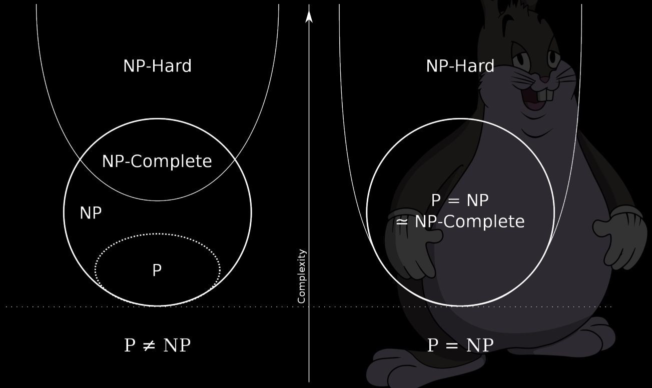


NP-hard 😳

Now, sometimes we ~~get lazy in proving some problem is NP-complete and forget to prove (a)~~ are unable to prove (a) easily. You'll see some examples later. Despite that, we might still be able to prove (b)!

Definition: A language A is **NP-hard** if for every $B \in \mathrm{NP}$, $B \leq_p A$.

Task: speedrun proving NP-complete $\subseteq$ NP-hard fullmarks%.



NP-hard 😳

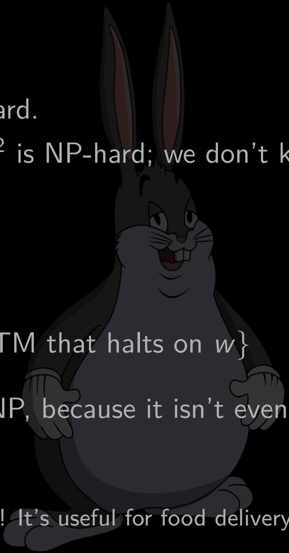
Intuitively, NP-hard problems are "at least as hard as the hardest problems in NP".

Here are some NP-hard problems:

- Every NP-complete problem is NP-hard.
- The **Travelling Salesman Problem**[2] is NP-hard; we don't know if it's in NP or not.
- Super Mario Brothers is NP-Hard.
- The Halting Problem

  $$\mathrm{HP} = \{(M, w) : M \text{ is a TM that halts on } w\}$$

  is NP-hard, but is known to not be NP, because it isn't even computable!

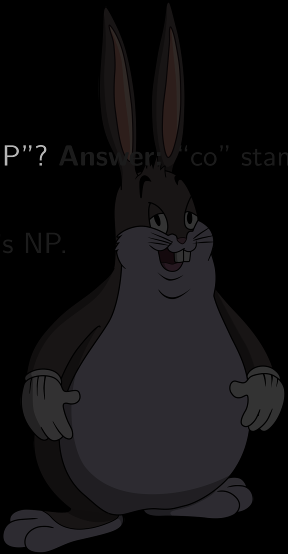


[2] If you haven't heard of it before, search it up! It's useful for food delivery n stuff maybe 🍣

co-NP

Task: What does “co” stand for in “co-NP”? **Answer:** “co” stands for ~~considering dropping cs~~ complement.

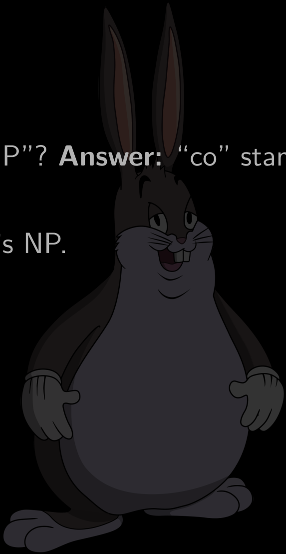
Definition: A language A is **co-NP** if $\overline{A}$ is NP.



co-NP

Task: What does "co" stand for in "co-NP"? **Answer:** "co" stands for ~~considering dropping cs~~ complement.

Definition: A language A is **co-NP** if $\overline{A}$ is NP.



co-NP

Definition: A language A is **co-NP** if $\overline{A}$ is NP.

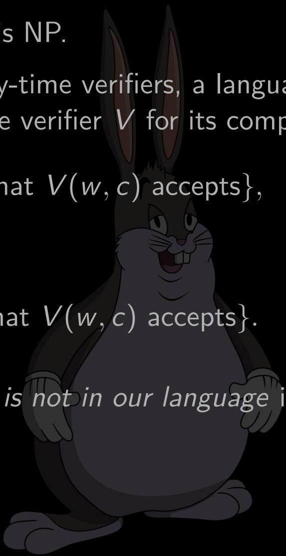
Since NP is the set of languages with poly-time verifiers, a language A is in co-NP if and only if there is a poly-time verifier V for its complement:

$$\overline{A} = \{w : \text{there exists } c \text{ such that } V(w, c) \text{ accepts}\},$$

or equivalently,

$$A = \{w : \text{there is no } c \text{ such that } V(w, c) \text{ accepts}\}.$$

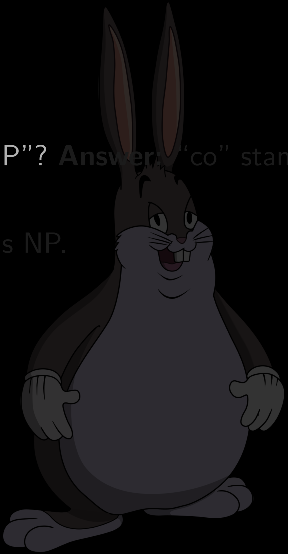
This means we can *verify that something is not in our language* in polynomial time.



co-NP

Task: What does "co" stand for in "co-NP"? **Answer:** "co" stands for ~~considering dropping cs~~ complement.

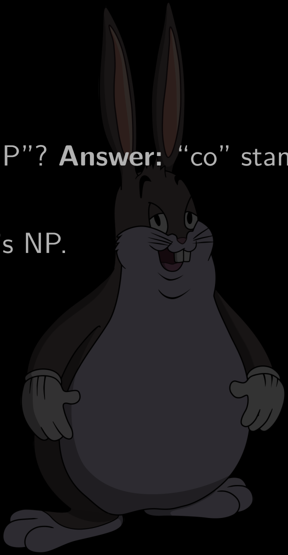
Definition: A language A is **co-NP** if $\overline{A}$ is NP.



co-NP

Task: What does “co” stand for in “co-NP”? **Answer:** “co” stands for ~~considering dropping cs~~ complement.

Definition: A language A is **co-NP** if $\overline{A}$ is NP.



co-NP

Definition: A language A is **co-NP** if $\overline{A}$ is NP.

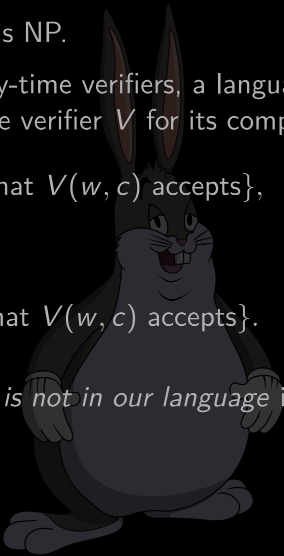
Since NP is the set of languages with poly-time verifiers, a language A is in co-NP if and only if there is a poly-time verifier V for its complement:

$$\overline{A} = \{w : \text{there exists } c \text{ such that } V(w, c) \text{ accepts}\},$$

or equivalently,

$$A = \{w : \text{there is no } c \text{ such that } V(w, c) \text{ accepts}\}.$$

This means we can *verify that something is not in our language* in polynomial time.



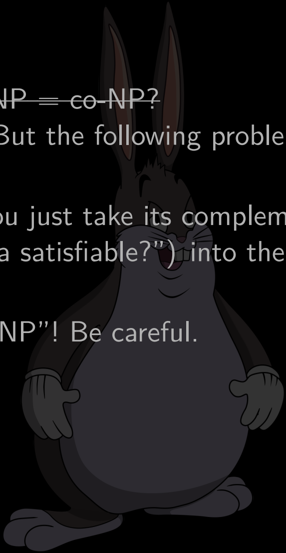
co-NP

~~**Task:** Answer the following question: is NP = co-NP?~~

Actually, we don't know if NP = co-NP. But the following problems are co-NP!

> pretty much any problem in NP, if you just take its complement. e.g. turn SAT (which asks "is this formula satisfiable?") into the problem "is this formula unsatisfiable?"

NOTE: "co-NP" is not the same as "not NP"! Be careful.



yahoot time

(gotta fill time somehow, there isn't much to cover today though!)

winner gets (imaginary) $10 sushi juice coupons. when i open a sushi juice store, you will be able to redeem those coupons.