Algorithms & Models of Computation

CS/ECE 374, Fall 2020

NP and NP Completeness

Lecture 23

Tuesday, December 1, 2020

LaTeXed: October 27, 2020 13:58

23.1
NP-Completeness: Cook-Levin Theorem

23.1.1
Completeness

NP: Non-deterministic polynomial

Definition 23.1.

A decision problem is in **NP**, *if it has a polynomial time certifier, for all the all the YES instances.*

Definition 23.2.

A decision problem is in **co-NP**, *if it has a polynomial time certifier, for all the all the NO instances.*

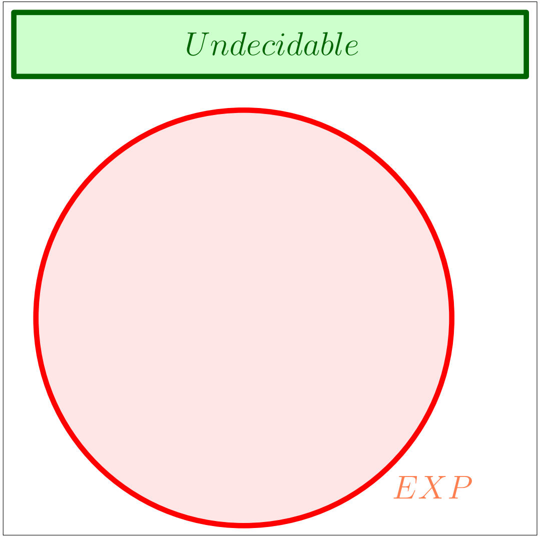
Example 23.3.

1. **3SAT** is in **NP**.
2. But **Not3SAT** is in **co-NP**.

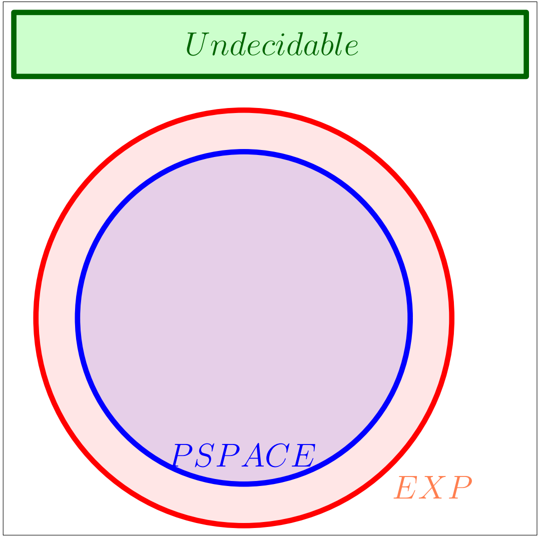
In the beginning...

In the beginning...

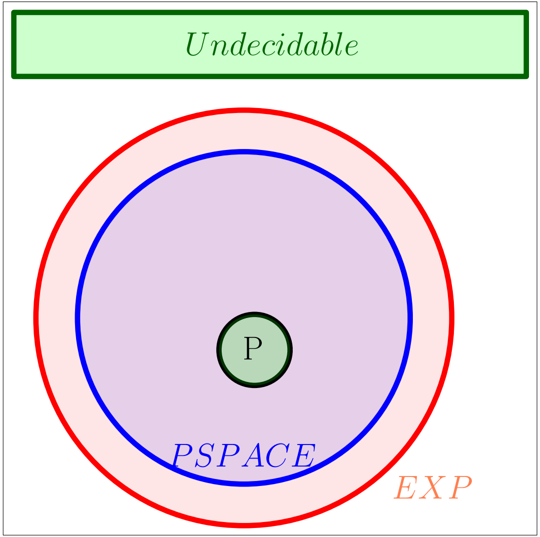
In the beginning...



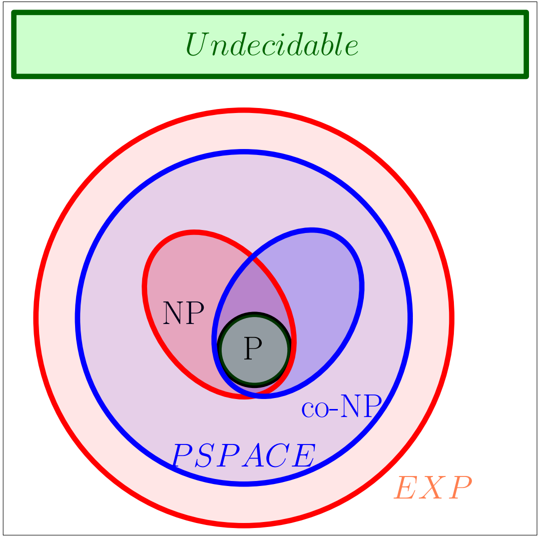
In the beginning...



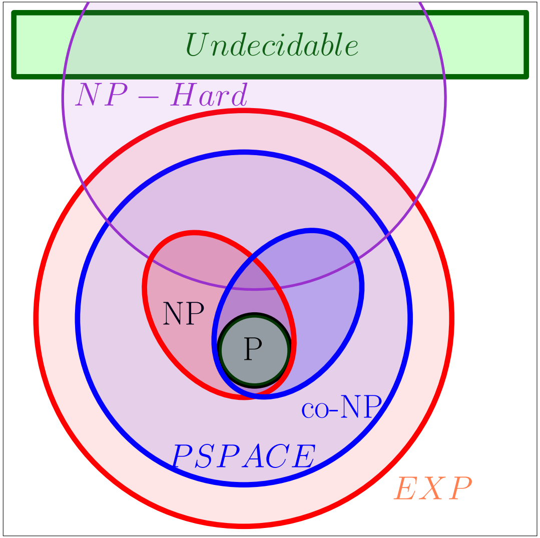
In the beginning...



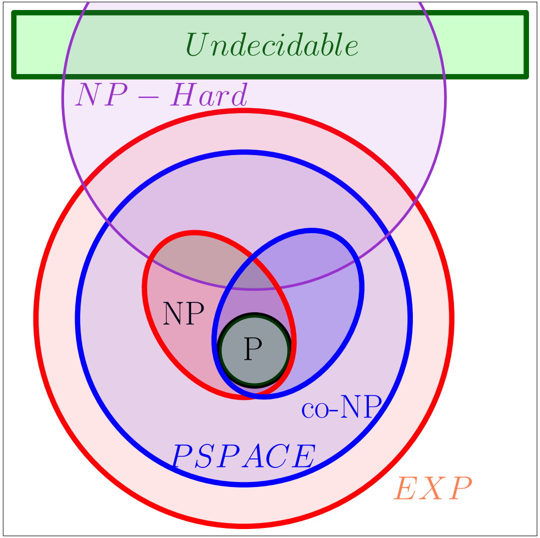
In the beginning...



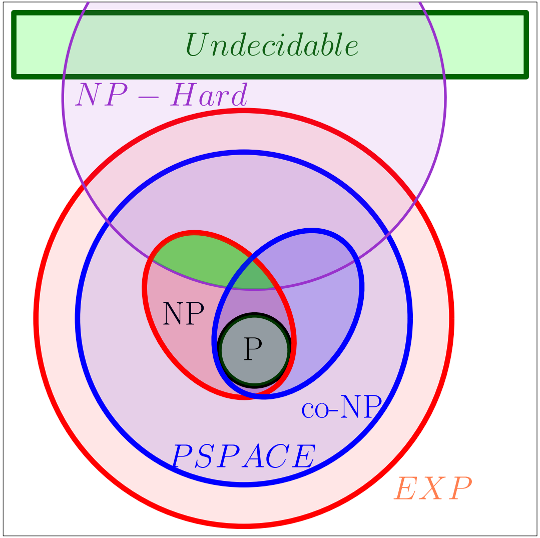
In the beginning...



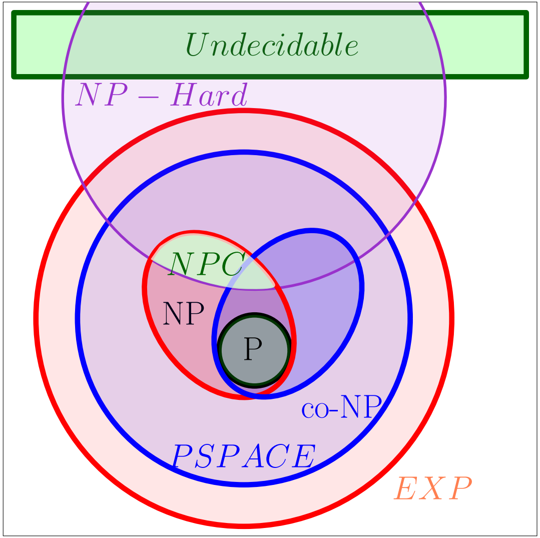
In the beginning...



In the beginning...



In the beginning...



“Hardest” Problems

Question

What is the hardest problem in **NP**? How do we define it?

Towards a definition

1. Hardest problem must be in **NP**.
2. Hardest problem must be at least as “difficult” as every other problem in **NP**.

NP-Complete Problems

Definition 23.4.

A problem X is said to be **NP-Complete** if

1. $X \in \mathbf{NP}$, and
2. (Hardness) For any $Y \in \mathbf{NP}$, $\mathbf{Y} \leq_P \mathbf{X}$.

Solving **NP-Complete** Problems

Proposition 23.5.

Suppose $\boldsymbol{X}$ is **NP-Complete**. *Then $\boldsymbol{X}$ can be solved in polynomial time* $\iff$ **P** $=$ **NP**.

Proof.

$\Rightarrow$ Suppose $\boldsymbol{X}$ can be solved in polynomial time

0.1 Let $\boldsymbol{Y} \in$ **NP**. We know $\mathbf{Y} \leq_P \mathbf{X}$.

0.2 We showed that if $\mathbf{Y} \leq_P \mathbf{X}$ and $\boldsymbol{X}$ can be solved in polynomial time, then $\boldsymbol{Y}$ can be solved in polynomial time.

0.3 Thus, every problem $\boldsymbol{Y} \in$ **NP** is such that $\boldsymbol{Y} \in \boldsymbol{P}$.

0.4 $\implies \boldsymbol{NP} \subseteq \boldsymbol{P}$.

0.5 Since **P** $\subseteq$ **NP**, we have **P** $=$ **NP**.

$\Leftarrow$ Since **P** $=$ **NP**, and $\boldsymbol{X} \in$ **NP**, we have a polynomial time algorithm for $\boldsymbol{X}$. □

NP-Hard Problems

Definition 23.6.

A problem X is said to be **NP-Hard** if

1. (Hardness) For any $Y \in$ **NP**, we have that $\mathbf{Y} \leq_P \mathbf{X}$.

An **NP-Hard** problem need not be in **NP**!

Example: Halting problem is **NP-Hard** (why?) but not **NP-Complete**.

Consequences of proving **NP-Completeness**

If X is **NP-Complete**

1. Since we believe $\mathbf{P} \neq \mathbf{NP}$,
2. and solving X implies $\mathbf{P} = \mathbf{NP}$.

X is unlikely to be efficiently solvable.

At the very least, many smart people before you have failed to find an efficient algorithm for X.
(This is proof by mob opinion — take with a grain of salt.)

Consequences of proving **NP-Complete**ness

If $\boldsymbol{X}$ is **NP-Complete**

1. Since we believe $\mathbf{P} \neq \mathbf{NP}$,
2. and solving $\boldsymbol{X}$ implies $\mathbf{P} = \mathbf{NP}$.

$\boldsymbol{X}$ is unlikely to be efficiently solvable.

At the very least, many smart people before you have failed to find an efficient algorithm for $\boldsymbol{X}$.
(This is proof by mob opinion — take with a grain of salt.)

Consequences of proving **NP-Complete**ness

If X is **NP-Complete**

1. Since we believe $\mathbf{P} \neq \mathbf{NP}$,
2. and solving X implies $\mathbf{P} = \mathbf{NP}$.

X is unlikely to be efficiently solvable.

At the very least, many smart people before you have failed to find an efficient algorithm for X.

(This is proof by mob opinion — take with a grain of salt.)

Consequences of proving **NP-Complete**ness

If X is **NP-Complete**

1. Since we believe $\mathbf{P} \neq \mathbf{NP}$,
2. and solving X implies $\mathbf{P} = \mathbf{NP}$.

X is unlikely to be efficiently solvable.

At the very least, many smart people before you have failed to find an efficient algorithm for X.
(This is proof by mob opinion — take with a grain of salt.)

THE END

...

(for now)