Logic and Complexity

Avijeet Ghosh[1]

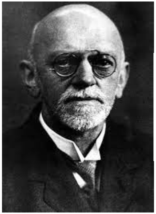
[1]Indian Statistical Institute, Kolkata

Table of Contents

Table of Contents

A System of Proofs

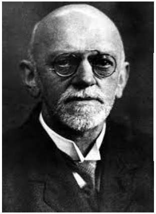
- A Formal System of Mathematical Proofs



?

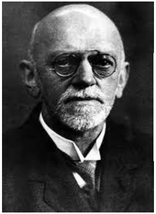
A System of Proofs

- A Formal System of Mathematical Proofs
  - Axioms $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$



?

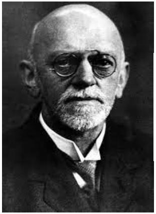
A System of Proofs



- A Formal System of Mathematical Proofs
  - Axioms $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$
  - Truth-preserving rules $\mathcal{I} = \{I_1, \ldots, I_m\}$

$$I_i = \frac{S_1, S_2, \ldots, S_n}{C}$$

A System of Proofs



?

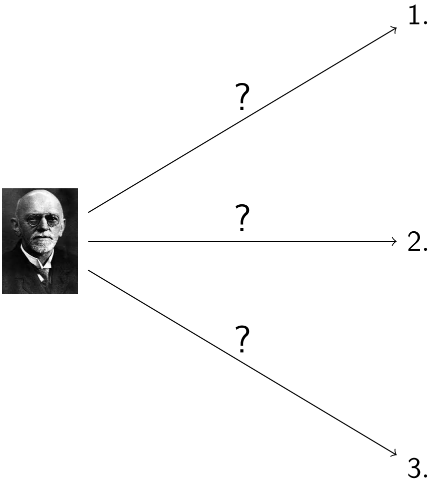
- A Formal System of Mathematical Proofs
  - Axioms $\mathcal{A} = \{A_1, A_2, \ldots, A_k\}$
  - Truth-preserving rules $\mathcal{I} = \{I_1, \ldots, I_m\}$

$$I_i = \frac{S_1, S_2, \ldots, S_n}{C}$$

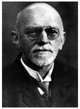
- Objective: Anything true has a proof

$$\begin{tikzcd}
 & & & T & & & \\
 & T_1 \arrow[urr, "I_{k'}"] & & & & T_2 \arrow[ull] & \\
A_i \arrow[ur, "I_{i'}"] & & A_j \arrow[ul] & & A_k \arrow[ur] & & A_\ell \arrow[ul, "I_{j'}"']
\end{tikzcd}$$

Hilbert's Questions



Hilbert's Questions

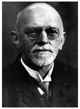


? 1. Complete

? 2. Consistent

? 3. Decidable

Hilbert's Questions



1. Complete X

Some Truths cannot be proven

?

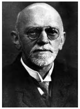
?

2. Consistent

?

3. Decidable

Hilbert's Questions



?

?

?

1. Complete X

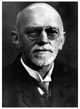
Some Truths cannot be proven

2. Consistent ?

Proving its own consistency beyond system

3. Decidable

Hilbert's Questions



?

?

?

1. Complete X 

Some Truths cannot be proven

2. Consistent ? 

Proving its own consistency beyond system

3. Decidable 🤔 This talk

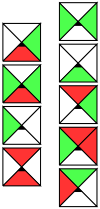
Given a statement and axioms, is there a proof?

Computability

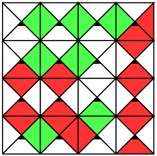
- **NEED:** Given S and $\mathcal{A}$, "procedure" to decide whether there is a proof
    - For any axiom system $(\mathcal{A}, \mathcal{I})$ and any statement S
      $$f(S, \mathcal{A}, \mathcal{I}) = \begin{cases} 1 & \mathcal{A} \vdash_{\mathcal{I}} S \\ 0 & \mathcal{A} \nvdash_{\mathcal{I}} S \end{cases}$$
    - Is f *computable*?
- What is *computability*?
- **Example 1**: $\forall x \in \mathbb{N}, f(x) = x^2 + 2x + 1$
    - $f(13) =?$
- **Example 2**: $\forall x \in \mathbb{N}, \forall y \in \mathbb{N} f(x, y) = x^y + xy + 1$
    - $f(13, 12) =?$
    - **HARDER**

Computability

- Proper Tiling:
  - Given a set of tiles $\mathcal{T}$



  - A Proper tiling on a finite grid:



- $f(\mathcal{T}) = \begin{cases} 1 & \text{Proper tiling covering infinite grid} \\ 0 & \text{No such proper tiling} \end{cases}$

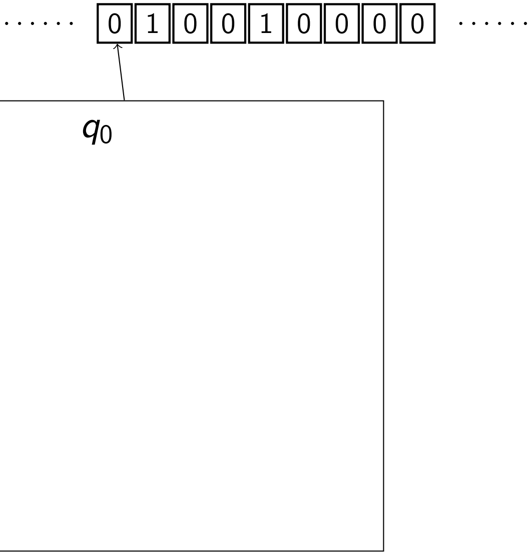
Model of Computability

- $f(x) = x^2 + 2x + 1$.
- $f(13) = 13^2 + (2x13) + 1$
- Local rules:
  - Rule of multiplication
  - Rule of addition
- Local states:
  1. Initial $13^2 + (2x13) + 1$
  2. All multiplication done $169 + 26 + 1$
  3. All addition done 196
  4. Final value 196
- **NEED: MODEL** Anything that model can compute is computable
- Turing Machine, Lambda Calculus

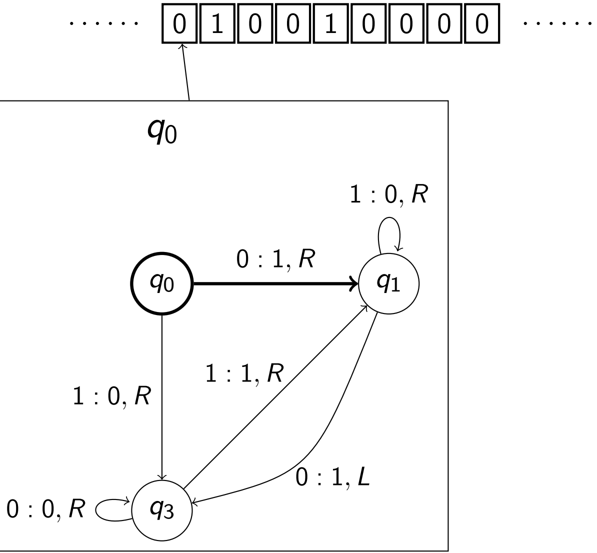
The Turing Machine

$$\cdots\cdots \quad \boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{0}\boxed{0} \quad \cdots\cdots$$

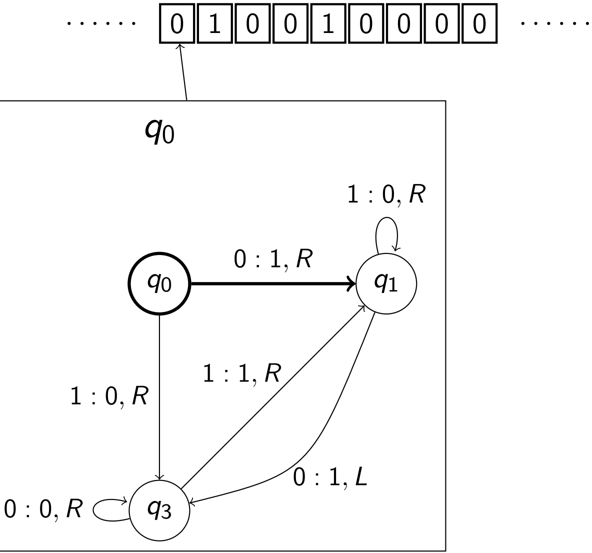
The Turing Machine

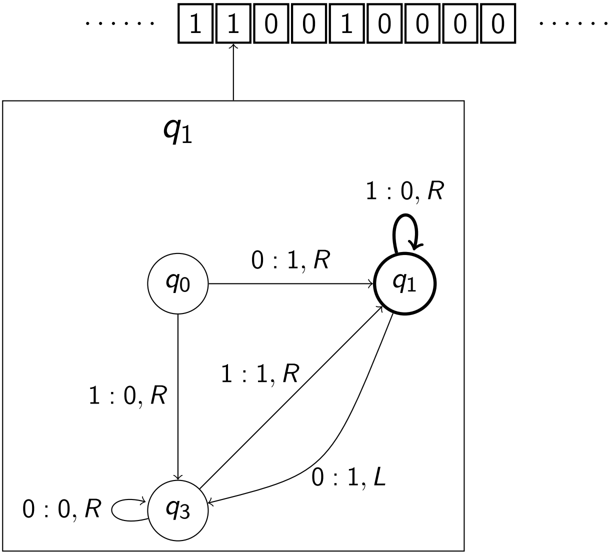


The Turing Machine

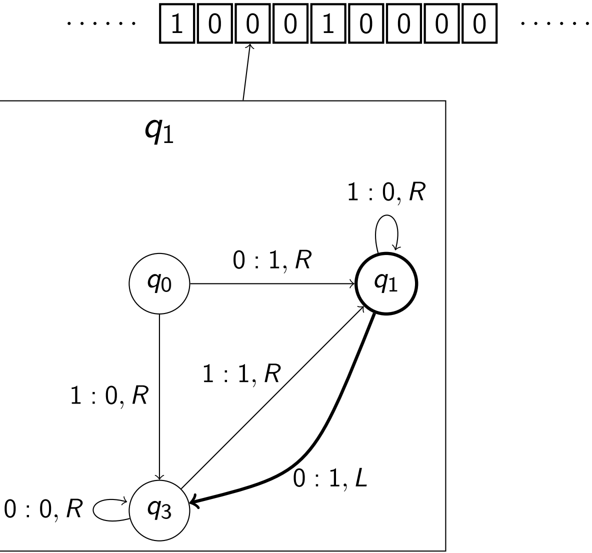


The Turing Machine

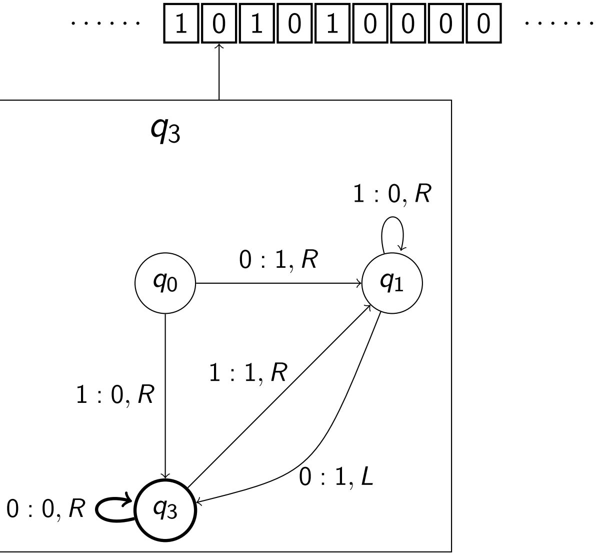



...... 1 1 0 0 1 0 0 0 0 ......
q_1
1 : 0, R
0 : 1, R
q_0
q_1
1 : 1, R
1 : 0, R
0 : 1, L
0 : 0, R
q_3

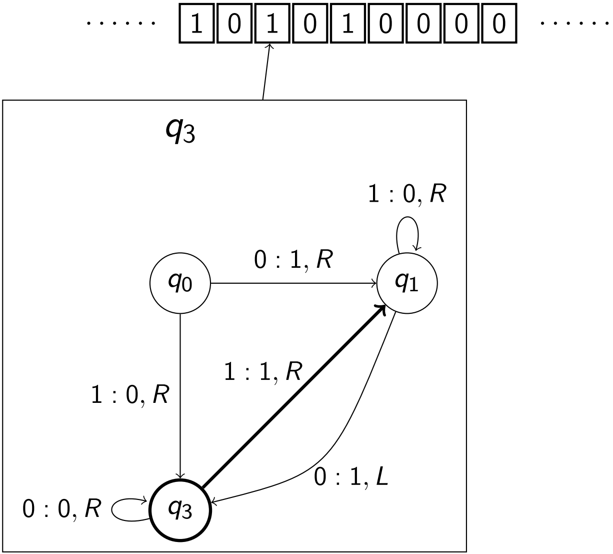
The Turing Machine

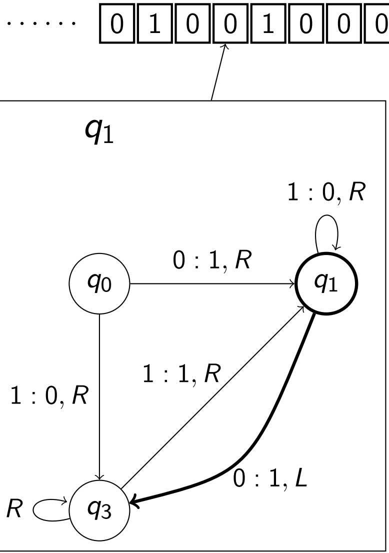


The Turing Machine



The Turing Machine





...... 0 1 0 0 1 0 0 0 0 ......
q_1
$1 : 0, R$
$0 : 1, R$
q_0
q_1
$1 : 1, R$
$1 : 0, R$
$0 : 1, L$
$0 : 0, R$
q_3

The Turing Machine

- $\mathcal{M} = \langle Q, q_0 \in Q, \Sigma, \delta, F \subseteq Q \rangle$
  - Q: finite set of states
  - Σ: alphabet
  - $\delta : Q \times \Sigma \rightarrow Q \times \Sigma \cup \{R, L\}$ transition rule
- Decision functions: $f(x \in \{0,1\}^{\star}) \in \{0,1\}$
- **Other MODELS**: Church's Lambda Calculus

Church-Turing Thesis

Any computable function can be computed by a Turing Machine/λ-Calculus.

Complexity Theory

- **Model:** Turing Machine (TM)
- Bounding resource usage of computation wrt input size n
  - **Time**: Steps a TM takes
  - **Space**: Tape space used
  - **Asymptotical**: $2\mathcal{C}(n) + 3 \equiv 55\mathcal{C}(n) + 7$
    $2^{\mathcal{C}(n)} \not\equiv 3^{\mathcal{C}(n)}$
- Difficulty between classes of problems (Decision functions)
- Decision functions gives Language
  - Language of f $\mathcal{L}_f \subseteq \Sigma^\star$
  - $f(x) = 1 \equiv \mathcal{M}_f(x)$ accepts $\equiv x \in \mathcal{L}_f$

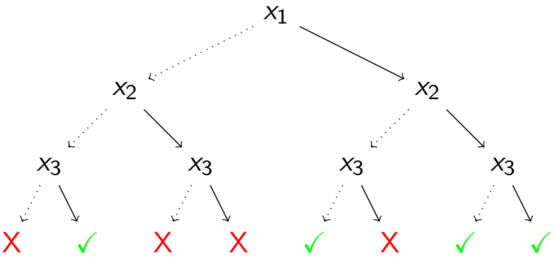
Polynomial Complexity

Consider the following problems:

- Given a tuple of n integers, **are they SORTED**?
    - $\langle 23, 45, 79, 127 \rangle$
    - Easily solvable
    - Compare adjacent pair (constant steps c)
    - Check for all adjacent pairs ($\leq n$)
    - Total steps $\sim cn$
- Given a propositional formula φ, **is it SATISFIABLE**?
    - $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee \neg x_1) \wedge (x_3 \vee x_1 \vee x_2)$
    - Harder to solve
    - If **SATISFIABLE**, what is the certificate?
    - How big?
    - If certificate given, how much time to verify?

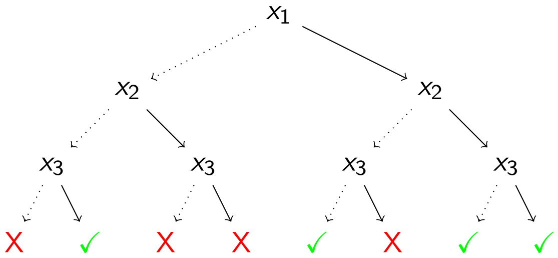
Non-determinism

- $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee \neg x_1) \wedge (x_3 \vee x_1 \vee x_2)$
- TM steps:



Non-determinism

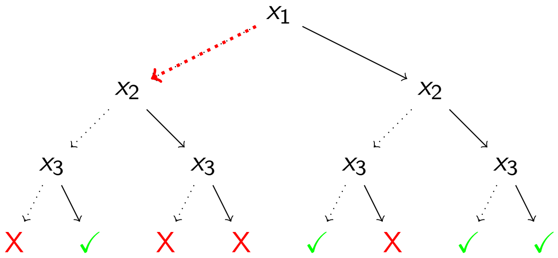
- $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee \neg x_1) \wedge (x_3 \vee x_1 \vee x_2)$
- Lucky TM steps:



- This Lucky TM is *quick* iff small certificate, quick verify
- This Lucky TM is called **Non-deterministic**

Non-determinism

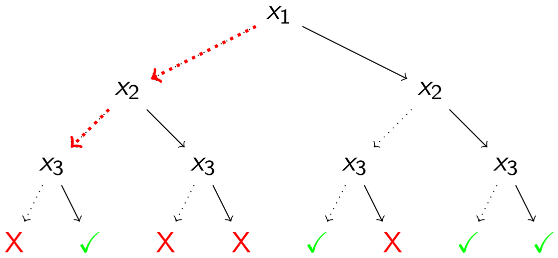
- $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee \neg x_1) \wedge (x_3 \vee x_1 \vee x_2)$
- Lucky TM steps:



- This Lucky TM is *quick* iff small certificate, quick verify
- This Lucky TM is called **Non-deterministic**

Non-determinism

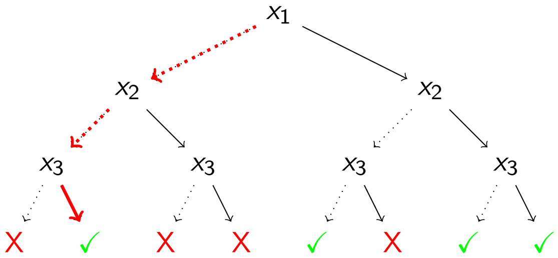
- $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee \neg x_1) \wedge (x_3 \vee x_1 \vee x_2)$
- Lucky TM steps:



- This Lucky TM is *quick* iff small certificate, quick verify
- This Lucky TM is called **Non-deterministic**

Non-determinism

- $\varphi = (x_1 \vee \neg x_2) \wedge (x_2 \vee \neg x_3 \vee \neg x_1) \wedge (x_3 \vee x_1 \vee x_2)$
- Lucky TM steps:



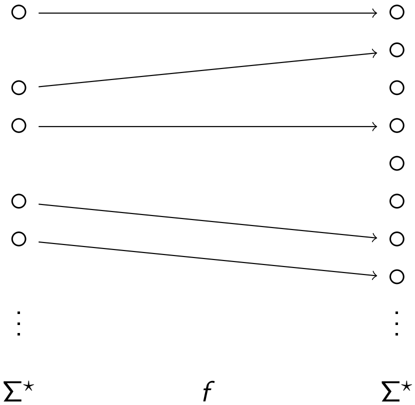
- This Lucky TM is *quick* iff small certificate, quick verify
- This Lucky TM is called **Non-deterministic**

Complexity Classes

- $\mathcal{L} \in$ P:= Steps$(\mathcal{M}_{\mathcal{L}}(x)) \leq O(|x|^c)$
- $\mathcal{L} \in$ NP:= Steps$(\mathcal{M}_{\mathcal{L}}(x)) \leq O(|x|^c)$, $\mathcal{M}_{\mathcal{L}}$ is Non-deterministic
- P $\subset_?$ NP
- $\mathcal{L} \in$ EXPTIME:= Steps$(\mathcal{M}_{\mathcal{L}}(x)) \leq O(2^{|x|^c})$
- $\mathcal{L} \in$ NEXPTIME:= Steps$(\mathcal{M}_{\mathcal{L}}(x)) \leq O(2^{|x|^c})$, $\mathcal{M}_{\mathcal{L}}$ is ND
- P $\subseteq_?$ NP $\subseteq_?$ EXPTIME $\subseteq_?$ NEXPTIME
- P $\subset$ EXPTIME and NP $\subset$ NEXPTIME

Hardness and Completeness

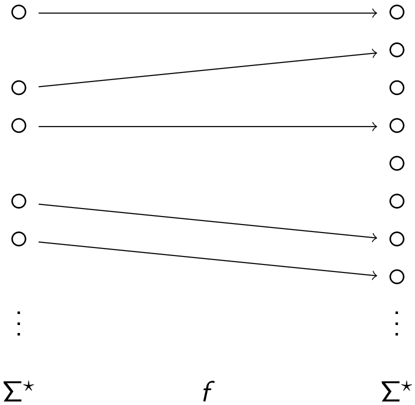
- $\mathcal{L}_1 \geq_{hard} \mathcal{L}_2$



Hardness and Completeness

- $\mathcal{L}_1 \geq_{hard} \mathcal{L}_2$

  $\mathcal{L}_2$ can be computed using $\mathcal{M}_{\mathcal{L}_1}$

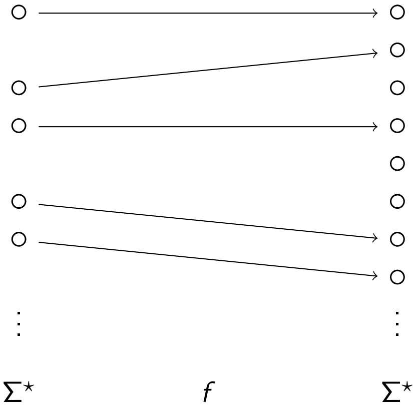


$x \in \mathcal{L}_2$ iff $f(x) \in \mathcal{L}_1$

Hardness and Completeness

- $\mathcal{L}_1 \geq_{hard} \mathcal{L}_2$

  $\mathcal{L}_2$ can be computed using $\mathcal{M}_{\mathcal{L}_1}$
- *Reducing* $\mathcal{L}_2$ to $\mathcal{L}_1$.



$x \in \mathcal{L}_2$ iff $f(x) \in \mathcal{L}_1$

$\mathcal{L}_2$ is reduced to $\mathcal{L}_1$

Completeness

- $\mathcal{L}_1 \geq_{\mathrm{P}} \mathcal{L}_2$: Reduction f is poly computable
- $\mathcal{L} \in \mathcal{C} - \textit{complete}$
  - $\mathcal{L} \in \mathcal{C}$
  - $\mathcal{L} \geq_{\mathrm{P}} \mathcal{L}'$ for any $\mathcal{L}' \in \mathcal{C}$

Cook-Levin

Propositional SAT is $\mathrm{NP} - \textit{complete}$

- A $\mathcal{C} - \textit{complete}$ problem is one of the hardest in $\mathcal{C}$

Model-Checking Hierarchy

- What is model-checking?
  - Given model and a formula, is formula true in model?
  - Proposition Logic: $\langle 10010\ldots01\rangle \models \varphi$?
  - FOL: $\langle \mathcal{D}, \mathcal{I}, \Pi\rangle \models \varphi$?
- Model-checking Prop $\varphi \in \mathrm{P} \quad (\Sigma_0^{\mathrm{P}})$
- Add existential quantifier

  Model-check $\exists x_1 \exists x_2 \ldots \exists x_n \varphi \in \mathrm{NP} \quad (\Sigma_1^{\mathrm{P}})$
- Add universal quantifier

  Model-check $\forall x_1 \forall x_2 \ldots \forall x_n \varphi \in co - \mathrm{NP} \quad (\Pi_1^{\mathrm{P}})$
- Note: $\mathrm{P} \subseteq \mathrm{NP} \cap co - \mathrm{NP}$

  Infact, more generally...

The Polynomial Hierarchy

- $\Sigma_n^{\mathrm{P}} = \exists X_1 \ldots \exists X_n \Pi_{n-1}^{\mathrm{P}}$
- $\Pi_n^{\mathrm{P}} = \exists X_1 \ldots \exists X_n \Sigma_{n-1}^{\mathrm{P}}$
- $\Sigma_n^{\mathrm{P}} \cup \Pi_n^{\mathrm{P}} \subseteq \Sigma_{n+1}^{\mathrm{P}} \cap \Pi_{n+1}^{\mathrm{P}}$
- What tops it?