Much of computational complexity theory concerns decision problems, which are typically represented as sets $A \subseteq \{0, 1\}^*$. 
Much of computational complexity theory concerns decision problems, which are typically represented as sets $A \subseteq \{0, 1\}^*$.

A solution of a decision problem $A \subseteq \{0, 1\}^*$ (also called a language $A \subseteq \{0, 1\}^*$) is a device (Turing machine, circuit, etc.) $M$ that, on input $x \in \{0, 1\}^*$, decides whether $x \in A$. 
Much of computational complexity theory concerns **decision problems**, which are typically represented as sets $A \subseteq \{0, 1\}^*$.  

A **solution** of a decision problem $A \subseteq \{0, 1\}^*$ (also called a **language** $A \subseteq \{0, 1\}^*$) is a device (Turing machine, circuit, etc.) $M$ that, on input $x \in \{0, 1\}^*$, decides whether $x \in A$. 

```
\begin{center}
\begin{tikzpicture}
  \node (M) at (0,0) {$M$};
  \node (input) at (-2,0) {$x \in \{0, 1\}^*$};
  \node (output_y) at (2,0) {yes if $x \in A$};
  \node (output_n) at (2,0) {no if $x \notin A$};
  \draw[->] (M) -- (input) node [midway, below] {input};
  \draw[->] (M) -- (output_y) node [midway, below] {yes if $x \in A$};
  \draw[->] (M) -- (output_n) node [midway, below] {no if $x \notin A$};
\end{tikzpicture}
\end{center}
```
The canonical question in computational complexity:

Given:

- a decision problem $A \subseteq \{0, 1\}^*$;
Decision Problems

The canonical question in computational complexity:

Given:

- a decision problem \( A \subseteq \{0, 1\}^* \);
- a computational resource, e.g.,
  - time (number of steps),
The canonical question in computational complexity:

Given:

- a decision problem $A \subseteq \{0, 1\}^*$;
- a computational resource, e.g.,
  - time (number of steps),
  - space (number of memory cells),
The canonical question in computational complexity:

Given:

- a decision problem $A \subseteq \{0, 1\}^*$;
- a computational resource, e.g.,
  - time (number of steps),
  - space (number of memory cells),
  - number of circuit gates,
Decision Problems

The canonical question in computational complexity:

Given:

- a decision problem $A \subseteq \{0, 1\}^*$;
- a computational resource, e.g.,
  - time (number of steps),
  - space (number of memory cells),
  - number of circuit gates,
  - etc;
- and a resource bound $t : \mathbb{N} \rightarrow \mathbb{N}$,
The canonical question in computational complexity:

Given:
- a decision problem \( A \subseteq \{0, 1\}^* \);
- a computational resource, e.g.,
  - time (number of steps),
  - space (number of memory cells),
  - number of circuit gates,
  - etc;
- and a resource bound \( t : \mathbb{N} \to \mathbb{N} \),

does there exist a device \( M \) that, on every input \( x \in \Sigma^* \), decides whether \( x \in A \) using at most \( t(|x|) \) of the resource?
If the devices are Turing machines and the resource is time, we write $\text{TIME}(t)$ for the complexity class of decision problems $A$ for which such a solution exists.

Minor correction: $\text{TIME}(t)$ really means $\text{TIME}(O(t))$. $\text{SPACE}(t)$ is defined analogously.

Convention: If $t(n) = n^2$, then complexity theorists write $n^2$ for the function $t$. Thus, for example, $\text{TIME}(n^2)$ is the class of all decision problems that can be solved in quadratic time.
If the devices are Turing machines and the resource is time, we write \( \text{TIME}(t) \) for the complexity class of decision problems \( A \) for which such a solution exists.

Minor correction: \( \text{TIME}(t) \) really means \( \text{TIME}(O(t)) \).
If the devices are Turing machines and the resource is time, we write \( \text{TIME}(t) \) for the complexity class of decision problems \( A \) for which such a solution exists.

Minor correction: \( \text{TIME}(t) \) really means \( \text{TIME}(O(t)) \).

\( \text{SPACE}(t) \) is defined analogously.
If the devices are Turing machines and the resource is time, we write $\text{TIME}(t)$ for the complexity class of decision problems $A$ for which such a solution exists.

Minor correction: $\text{TIME}(t)$ really means $\text{TIME}(O(t))$.

$\text{SPACE}(t)$ is defined analogously.

**Convention**

If $t(n) = n^2$, then complexity theorists write $n^2$ for the function $t$.

Thus, for example, $\text{TIME}(n^2)$ is the class of all decision problems that can be solved in quadratic time.
Some famous complexity classes:
Some famous complexity classes:

$$P = \bigcup_{k=1}^{\infty} \text{TIME}(n^k)$$

polynomial time
Some famous complexity classes:

\[ P = \bigcup_{k=1}^{\infty} \text{TIME}(n^k) \]  

\[ \text{PSPACE} = \bigcup_{k=1}^{\infty} \text{SPACE}(n^k) \]  

polynomial time

polynomial space
Complexity Classes

Some famous complexity classes:

\[ P = \bigcup_{k=1}^{\infty} \text{TIME}(n^k) \quad \text{polynomial time} \]

\[ \text{PSPACE} = \bigcup_{k=1}^{\infty} \text{SPACE}(n^k) \quad \text{polynomial space} \]

\[ E = \bigcup_{k=1}^{\infty} \text{TIME}(2^{kn}) \quad \text{linear-exponential time} \]
Some famous complexity classes:

\[
P = \bigcup_{k=1}^{\infty} \text{TIME}(n^k) \quad \text{polynomial time}
\]

\[
PSPACE = \bigcup_{k=1}^{\infty} \text{SPACE}(n^k) \quad \text{polynomial space}
\]

\[
E = \bigcup_{k=1}^{\infty} \text{TIME}(2^{kn}) \quad \text{linear-exponential time}
\]

\[
\text{EXP} = \bigcup_{k=1}^{\infty} \text{TIME}(2^{n^k}) \quad \text{exponential time}
\]
Complexity Classes

Some things that we know:

\[ P \subsetneq E \subsetneq \text{EXP} \]

\[ P \subseteq \text{PSPACE} \subseteq \text{EXP} \]

\[ \text{PSPACE} \neq E \]
Some things that we know:

\[ P \subsetneq E \subsetneq \text{EXP} \]
\[ P \subseteq \text{PSPACE} \subseteq \text{EXP} \]
\[ \text{PSPACE} \neq \text{E} \]

We believe that \( P \subseteq \text{PSPACE} \subseteq \text{EXP} \), but we cannot rule out either equality.
Complexity Classes

Some things that we know:

\[ P \subseteq \mathcal{E} \subseteq \mathcal{EXP} \]
\[ P \subseteq \mathcal{PSPACE} \subseteq \mathcal{EXP} \]
\[ \mathcal{PSPACE} \neq \mathcal{E} \]

We believe that \( P \subsetneq \mathcal{PSPACE} \subsetneq \mathcal{EXP} \), but we cannot rule out either equality.

We should not omit the famous class \( \mathcal{NP} \).
Fix a simple *pairing function* (encoding of two strings into one), e.g.,

\[ \langle x, y \rangle = 0^{|x|}1xy \]

for all \( x, y \in \{0, 1\}^* \).
Fix a simple **pairing function** (encoding of two strings into one), e.g.,

\[ \langle x, y \rangle = 0^{\left| x \right|}1xy \]

for all \( x, y \in \{0, 1\}^* \).

If \( B \subseteq \{0, 1\}^* \) is a decision problem and \( q : \mathbb{N} \rightarrow \mathbb{N} \), then we define the decision problem

\[ \exists^q B = \left\{ x \in \{0, 1\}^* \mid (\exists w \in \{0, 1\}^{\leq q(|x|)})\langle x, w \rangle \in B \right\} \]

Terminology: \( w \) is a **witness** that **testifies** that \( x \in \exists^q B \).
Complexity Classes

Fix a simple **pairing function** (encoding of two strings into one), e.g.,

\[
\langle x, y \rangle = 0^{|x|}1xy
\]

for all \(x, y \in \{0, 1\}^*\).

If \(B \subseteq \{0, 1\}^*\) is a decision problem and \(q : \mathbb{N} \rightarrow \mathbb{N}\), then we define the decision problem

\[
\exists^q B = \left\{ x \in \{0, 1\}^* \mid (\exists w \in \{0, 1\}^{\leq q(|x|)}) \langle x, w \rangle \in B \right\}
\]

Terminology: \(w\) is a **witness** that **testifies** that \(x \in \exists^q B\).

\[
\text{NP} = \left\{ \exists^n^k B \mid k \in \mathbb{N} \text{ and } B \in \text{P} \right\}
\]
Fix a simple **pairing function** (encoding of two strings into one), e.g.,

\[ \langle x, y \rangle = 0^{\|x\|}1xy \]

for all \( x, y \in \{0, 1\}^* \).

If \( B \subseteq \{0, 1\}^* \) is a decision problem and \( q : \mathbb{N} \to \mathbb{N} \), then we define the decision problem

\[ \exists^q B = \left\{ x \in \{0, 1\}^* | (\exists w \in \{0, 1\}^{\leq q(|x|)}) \langle x, w \rangle \in B \right\} \]

**Terminology:** \( w \) is a **witness** that **testifies** that \( x \in \exists^q B \).

\[ \text{NP} = \left\{ \exists^n B | k \in \mathbb{N} \text{ and } B \in \text{P} \right\} \]

We **know** that \( \text{P} \subseteq \text{NP} \subseteq \text{PSPACE} \). We **believe** that these inclusions are proper.
Why is the complexity class NP important?

Because hundreds of very important problems in scientific computing can be formulated as decision problems \( C \subseteq \{0, 1\}^* \) that are NP-complete, meaning that \( C \in \text{NP} \), and every problem \( A \in \text{NP} \) is efficiently reducible to \( C \).

We believe that these important problems \( C \) are intractable. We can prove this if we can find any intractable problem \( D \in \text{NP} \).
Why is the complexity class NP important?

Because hundreds of very important problems in scientific computing can be formulated as decision problems $C \subseteq \{0, 1\}^*$ that are NP-complete, meaning that

- $C \in \text{NP}$, and
- every problem $A \in \text{NP}$ is efficiently reducible to $C$. 
Why is the complexity class NP important?

Because hundreds of very important problems in scientific computing can be formulated as decision problems $C \subseteq \{0, 1\}^*$ that are NP-complete, meaning that

- $C \in \text{NP}$, and
- every problem $A \in \text{NP}$ is efficiently reducible to $C$.

We believe that these important problems $C$ are intractable.
Why is the complexity class NP important?

Because hundreds of very important problems in scientific computing can be formulated as decision problems $C \subseteq \{0, 1\}^*$ that are NP-complete, meaning that

- $C \in \text{NP}$, and
- every problem $A \in \text{NP}$ is efficiently reducible to $C$.

We believe that these important problems $C$ are intractable.

We can prove this if we can find any intractable problem $D \in \text{NP}$.
If $D$ is intractable, then so is $C$, no matter how contrived $D$ may be.
If $D$ is intractable, then so is $C$, no matter how contrived $D$ may be.

But what do these complexity classes have to do with algorithmic fractal dimensions?
If $D$ is intractable, then so is $C$, no matter how contrived $D$ may be.

But what do these complexity classes have to do with algorithmic fractal dimensions?

Identifying each decision problem $A \subseteq \{0, 1\}^*$ with its characteristic sequence $\chi_A \in \{0, 1\}^\omega$ makes the Cantor space $C = \{0, 1\}^\omega$ the set of all decision problems.
Complexity Classes

If $D$ is intractable, then so is $C$, no matter how contrived $D$ may be.

But what do these complexity classes have to do with algorithmic fractal dimensions?

Identifying each decision problem $A \subseteq \{0,1\}^*$ with its characteristic sequence $\chi_A \in \{0,1\}^\omega$ makes the Cantor space $C = \{0,1\}^\omega$ the set of all decision problems.

$\therefore$ All the above complexity classes are countable subsets of Cantor space!
Constructors and their Results

We know how to define dimensions in Cantor space. To define dimensions (or measure, or Baire category) in complexity classes, we need the following notion.

A constructor is a function \( \delta : \{0, 1\}^* \rightarrow \{0, 1\}^* \) such that, for all \( w \in \{0, 1\}^* \), \( w \mathbin{\mathcal{R}} \delta(w) \). That is, \( \delta \) simply adds one or more bits to its input.

The result of a constructor \( \delta \) is the unique sequence \( R(\delta) \in \{0, 1\}^\omega \) such that, for all \( n \in \mathbb{N} \), \( \delta_n(\lambda) \sqsubseteq R(\delta) \).
Constructors and their Results

We know how to define dimensions in Cantor space. To define dimensions (or measure, or Baire category) in complexity classes, we need the following notion.

A **constructor** is a function $\delta : \{0, 1\}^* \to \{0, 1\}^*$ such that, for all $w \in \{0, 1\}^*$, $w \sqsupseteq \delta(w)$. That is, $\delta$ simply adds one or more bits to its input.
Constructors and their Results

We know how to define dimensions in Cantor space. To define dimensions (or measure, or Baire category) in complexity classes, we need the following notion.

A constructor is a function \( \delta : \{0, 1\}^* \rightarrow \{0, 1\}^* \) such that, for all \( w \in \{0, 1\}^* \), \( w \nsubseteq \delta(w) \). That is, \( \delta \) simply adds one or more bits to its input.

The result of a constructor \( \delta \) is the unique sequence \( R(\delta) \in \{0, 1\}^\omega \) such that, for all \( n \in \mathbb{N} \),

\[
\delta^n(\lambda) \sqsubseteq R(\delta).
\]
Constructors and their Results

We know how to define dimensions in Cantor space. To define dimensions (or measure, or Baire category) in complexity classes, we need the following notion.

A **constructor** is a function \( \delta : \{0, 1\}^* \rightarrow \{0, 1\}^* \) such that, for all \( w \in \{0, 1\}^* \), \( w \not\equiv \delta(w) \). That is, \( \delta \) simply adds one or more bits to its input.

The **result** of a constructor \( \delta \) is the unique sequence \( R(\delta) \in \{0, 1\}^\omega \) such that, for all \( n \in \mathbb{N} \),

\[
\delta^n(\lambda) \sqsubseteq R(\delta).
\]

The **result class** of a set \( \Delta \) of functions is

\[
R(\Delta) = \{ R(\delta) | \delta \in \Delta \text{ is a constructor} \}.
\]
Constructors and their Results

In this talk, a resource bound $\Delta$ is one of the following sets.

$$\text{all} = \{ f | f : \{0, 1\}^* \rightarrow \{0, 1\}^* \}$$
In this talk, a **resource bound** $\Delta$ is one of the following sets.

$$\text{all} = \{ f \mid f : \{0, 1\}^* \to \{0, 1\}^* \}$$

$$\text{comp} = \{ f \in \text{all} \mid f \text{ is computable} \}$$
Constructors and their Results

In this talk, a resource bound $\Delta$ is one of the following sets.

$$\text{all} = \{ f | f : \{0, 1\}^* \to \{0, 1\}^* \}$$

$$\text{comp} = \{ f \in \text{all} | f \text{ is computable} \}$$

$$\text{p} = \{ f \in \text{all} | f \text{ is computable in } n^{O(1)} \text{ time} \}$$
In this talk, a **resource bound** $\Delta$ is one of the following sets.

- $\text{all} = \{ f | f : \{0, 1\}^* \rightarrow \{0, 1\}^* \}$
- $\text{comp} = \{ f \in \text{all} | f \text{ is computable} \}$
- $p = \{ f \in \text{all} | f \text{ is computable in } n^{O(1)} \text{ time} \}$
- $q_p = \{ f \in \text{all} | f \text{ is computable in } n^{(\log n)^{O(1)}} \text{ time} \}$
In this talk, a **resource bound** $\Delta$ is one of the following sets.

$$\text{all} = \{ f | f : \{0, 1\}^* \to \{0, 1\}^* \}$$

$$\text{comp} = \{ f \in \text{all} | f \text{ is computable} \}$$

$$p = \{ f \in \text{all} | f \text{ is computable in } n^{O(1)} \text{ time} \}$$

$$\text{qp} = \{ f \in \text{all} | f \text{ is computable in } n^{(\log n)^{O(1)}} \text{ time} \}$$

pspace, qpspace
Recall that we identify each decision problem \( A \subseteq \{0, 1\}^* \) with its characteristic sequence \( \chi_A \in \{0, 1\}^\omega \).
Recall that we identify each decision problem $A \subseteq \{0, 1\}^*$ with its characteristic sequence $\chi_A \in \{0, 1\}^\omega$.

**Lemma**

- $R(\text{all}) = \mathcal{P}(\{0, 1\}^*)$ is the set of all decision problems.
Recall that we **identify** each decision problem $A \subseteq \{0, 1\}^*$ with its characteristic sequence $\chi_A \in \{0, 1\}^\omega$.

**Lemma**

- $R(\text{all}) = \mathcal{P}(\{0, 1\}^*)$ is the set of all decision problems.
- $R(\text{comp}) = \text{DEC}$ is the set of all decidable sets $A \subseteq \{0, 1\}^*$. 

Recall that we identify each decision problem \( A \subseteq \{0, 1\}^* \) with its characteristic sequence \( \chi_A \in \{0, 1\}^\omega \).

**Lemma**

- \( R(\text{all}) = \mathcal{P}(\{0, 1\}^*) \) is the set of all decision problems.
- \( R(\text{comp}) = \text{DEC} \) is the set of all decidable sets \( A \subseteq \{0, 1\}^* \).
- \( R(\text{p}) = E = \text{TIME}(2^{\text{linear}}) \).
Recall that we identify each decision problem $A \subseteq \{0, 1\}^*$ with its characteristic sequence $\chi_A \in \{0, 1\}^\omega$.

**Lemma**

- $R(\text{all}) = \mathcal{P}(\{0, 1\}^*)$ is the set of all decision problems.
- $R(\text{comp}) = \text{DEC}$ is the set of all decidable sets $A \subseteq \{0, 1\}^*$.
- $R(p) = E = \text{TIME}(2^{\text{linear}})$.

*Note: If $x \in \{0, 1\}^*$ is the first string not decided by a prefix $w$ of $\chi_A$, then*

$$\text{poly}(|w|) = |w|^{O(1)} = (2^{|x|})^{O(1)} = 2^{O(|x|)}.$$
Recall that we identify each decision problem \( A \subseteq \{0, 1\}^* \) with its characteristic sequence \( \chi_A \in \{0, 1\}^\omega \).

**Lemma**

- \( R(\text{all}) = \mathcal{P}(\{0, 1\}^*) \) is the set of all decision problems.
- \( R(\text{comp}) = \text{DEC} \) is the set of all decidable sets \( A \subseteq \{0, 1\}^* \).
- \( R(p) = E = \text{TIME}(2^{\text{linear}}) \).
- \( R(qp) = \text{EXP} \).
Recall that we identify each decision problem $A \subseteq \{0, 1\}^*$ with its characteristic sequence $\chi_A \in \{0, 1\}^\omega$.

**Lemma**

- $R(\text{all}) = \mathcal{P}(\{0, 1\}^*)$ is the set of all decision problems.
- $R(\text{comp}) = \text{DEC}$ is the set of all decidable sets $A \subseteq \{0, 1\}^*$.
- $R(\text{p}) = E = \text{TIME}(2^{\text{linear}})$.
- $R(\text{qp}) = \text{EXP}$.
- $R(\text{pspace}) = \text{ESPACE}$.
Recall that we **identify** each decision problem $A \subseteq \{0, 1\}^\ast$ with its characteristic sequence $\chi_A \in \{0, 1\}^\omega$.

### Lemma
- $R(\text{all}) = \mathcal{P}(\{0, 1\}^\ast)$ is the set of all decision problems.
- $R(\text{comp}) = \text{DEC}$ is the set of all decidable sets $A \subseteq \{0, 1\}^\ast$.
- $R(\text{p}) = E = \text{TIME}(2^{\text{linear}})$.
- $R(\text{qp}) = \text{EXP}$.
- $R(\text{pspace}) = \text{ESPACE}$.
- $R(\text{qpspace}) = \text{EXPSPACE}$.
Technical Note. Many of our functions will be of the form $f : D \rightarrow [0, \infty)$, where $D$ is some discrete domain like $\{0, 1\}^*$ or $\{0, 1\}^* \times \mathbb{N}$. Such a function is $\Delta$-computable if there is a function $\hat{f} : D \times \mathbb{N} \rightarrow \mathbb{Q} \cap [0, \infty)$ such that

- for all $x \in D$ and $r \in \mathbb{N}$,

$$|\hat{f}(x, r) - f(x)| \leq 2^{-r}$$
Technical Note. Many of our functions will be of the form \( f : D \rightarrow [0, \infty) \), where \( D \) is some discrete domain like \( \{0, 1\}^* \) or \( \{0, 1\}^* \times \mathbb{N} \). Such a function is \( \Delta \)-computable if there is a function \( \hat{f} : D \times \mathbb{N} \rightarrow \mathbb{Q} \cap [0, \infty) \) such that

- for all \( x \in D \) and \( r \in \mathbb{N} \),

\[
|\hat{f}(x, r) - f(x)| \leq 2^{-r}
\]

and

- \( \hat{f} \in \Delta \), with \( r \) coded in unary and \( \hat{f}(x, r) \) coded in binary.
Notation If $\Delta$ is a resource bound and $X \subseteq \{0, 1\}^\omega$, then

$$G_\Delta(X) = \left\{ s \in [0, \infty) \mid \text{there is a } \Delta\text{-computable } s\text{-gale } d \text{ such that } X \subseteq S^\infty[d] \right\}$$

and

$$G^{str}_\Delta(X) = \left\{ s \in [0, \infty) \mid \text{there is a } \Delta\text{-computable } s\text{-gale } d \text{ such that } X \subseteq S^\infty_{str}[d] \right\}$$
**Notation** If $\Delta$ is a resource bound and $X \subseteq \{0, 1\}^\omega$, then

$$
G_{\Delta}(X) = \left\{ s \in [0, \infty) \mid \text{there is a } \Delta\text{-computable } \text{s-gale } d \text{ such that } X \subseteq S^{\infty}[d] \right\}
$$

and

$$
G_{\Delta}^{str}(X) = \left\{ s \in [0, \infty) \mid \text{there is a } \Delta\text{-computable } \text{s-gale } d \text{ such that } X \subseteq S^{\infty}_{str}[d] \right\}
$$

We saw yesterday that

$$
dim_H(X) = \inf G_{all}(X)
$$

and

$$
dim_P(X) = \inf G_{all}^{str}(X).
$$

Let $\Delta$ be a resource bound, and let $X \subseteq \{0, 1\}^\omega$. The $\Delta$-dimension of $X$ is

$$\dim_\Delta(X) = \inf G_\Delta(X).$$

Let $\Delta$ be a resource bound, and let $X \subseteq \{0, 1\}^\omega$.

1. The $\Delta$-dimension of $X$ is

$$\dim_{\Delta}(X) = \inf G_{\Delta}(X).$$

2. The dimension of $X$ in $R(\Delta)$ is

$$\dim(X|R(\Delta)) = \dim_{\Delta}(X \cap R(\Delta)).$$
Definition (Athreya, Hitchcock, J. Lutz, Mayordomo 2007).

Let $\Delta$ be a resource bound, and let $X \subseteq \{0, 1\}^\omega$.

1. The $\Delta$-strong dimension of $X$ is

$$\text{Dim}_\Delta(X) = \inf G_\Delta^{str}(X).$$
Definition (Athreya, Hitchcock, J. Lutz, Mayordomo 2007).

Let $\Delta$ be a resource bound, and let $X \subseteq \{0, 1\}^\omega$.

1. The $\Delta$-strong dimension of $X$ is

   \[ \dim_{\Delta}(X) = \inf \mathcal{G}_{\Delta}^{str}(X). \]

2. The strong dimension of $X$ in $R(\Delta)$ is

   \[ \dim(X|R(\Delta)) = \dim_{\Delta}(X \cap R(\Delta)). \]
Observations

1. \( \dim(R(\Delta) | R(\Delta)) = \text{Dim}(R(\Delta) | R(\Delta)) = 1. \)
Observations

1. \( \dim(R(\Delta)|R(\Delta)) = \text{Dim}(R(\Delta)|R(\Delta)) = 1. \)

   E.g., \( \dim_p(E) = 1. \) Diagonalize against p-computable 1-gales.
Dimensions in Complexity Classes

Observations

1. \( \dim(R(\Delta)|R(\Delta)) = \operatorname{Dim}(R(\Delta)|R(\Delta)) = 1. \)
   
   E.g., \( \dim_p(E) = 1. \) Diagonalize against \( p \)-computable 1-gales.

2. Stability
   
   \[ \dim_\Delta(X \cup Y) = \max\{\dim_\Delta(X), \dim_\Delta(Y)\}. \]
   
   In fact, for "\( \Delta \)-countable unions",

   \[ \dim_\Delta\left(\bigcup_{k=0}^{\infty} X_k\right) = \sup\{\dim_\Delta(X_k) | k \in \mathbb{N}\}. \]
Observations (continued)

\[\begin{align*}
\dim_p(X) & \geq \dim_{qp}(X) \geq \dim_{comp}(X) \geq \dim_H(X) \\
\dim(X|E) & \geq \dim(X|\text{EXP}) \geq \dim(X|\text{DEC})
\end{align*}\]
Observations (continued)

3. \[
\dim_p(X) \geq \dim_{qp}(X) \geq \dim_{comp}(X) \geq \dim_H(X)
\]
   \[
   \dim(X|E) \geq \dim(X|\text{EXP}) \geq \dim(X|\text{DEC})
\]

4. For each \( k \in \mathbb{N} \),

   \[
   \dim(\text{TIME}(2^{kn})|E) = \dim(\text{TIME}(2^{n^k})|\text{EXP}) = 0.
   \]
Dimensions in Complexity Classes

Observations (continued)

3. \[ \dim_{\mathcal{P}}(X) \geq \dim_{\mathcal{QP}}(X) \geq \dim_{\mathcal{COMP}}(X) \geq \dim_{H}(X) \]

\[ \dim(X|E) \geq \dim(X|\text{EXP}) \geq \dim(X|\text{DEC}) \]

4. For each \( k \in \mathbb{N} \),

\[ \dim(\text{TIME}(2^{kn})|E) = \dim(\text{TIME}(2^{n^k})|\text{EXP}) = 0. \]

similarly for \( \text{Dim} \).
We conclude today with just one sample application.
We conclude today with just one sample application.

**Definition (Selman 1979, adapting Jockusch 1968).**

A decision problem $A \subseteq \{0, 1\}^*$ is \textbf{p-selective}, and we write $A \in \text{p-SEL}$, if there is a polynomial-time algorithm that, given an ordered pair $(x, y)$ of strings $x, y \in \{0, 1\}^*$, outputs a string $z \in \{x, y\}$ such that

$$\{x, y\} \cap A \neq \emptyset \Rightarrow z \in A.$$
We conclude today with just one sample application.

**Definition (Selman 1979, adapting Jockusch 1968).**

A decision problem $A \subseteq \{0, 1\}^*$ is p-selective, and we write $A \in \text{p-SEL}$, if there is a polynomial-time algorithm that, given an ordered pair $(x, y)$ of strings $x, y \in \{0, 1\}^*$, outputs a string $z \in \{x, y\}$ such that

$$\{x, y\} \cap A \neq \emptyset \Rightarrow z \in A.$$

The qp-selective sets are defined analogously.
Known Facts

- Selman 1979: No p-selective set can be $\leq_p^m$-hard for EXP.
Known Facts

- Selman 1979: No p-selective set can be $\leq_m^p$-hard for EXP.
  
  If $P \neq NP$, then no p-selective set can be $\leq_m^p$-hard for NP.
Known Facts

- Selman 1979: No p-selective set can be $\leq_p^m$-hard for EXP. If $P \neq NP$, then no p-selective set can be $\leq_p^m$-hard for NP.

- Burham & Longpré 1996, Wang 1996: No p-selective set $A$ can be weakly $\leq_p^m$-hard for EXP, i.e, must have $\mu(P_m(A)|EXP) = 0$. 

Theorem (J. Lutz, N. Lutz, & Mayordomo 2023).

$\dim(P_m(qp\text{-SEL})) \mid EXP = 0$.

Hence, $\dim(NP \mid EXP) > 0 \Rightarrow$ no qp-selective set can be $\leq_p^m$-hard for NP.
**Known Facts**

- **Selman 1979**: No p-selective set can be $\leq^p_m$-hard for EXP. If $P \neq NP$, then no p-selective set can be $\leq^p_m$-hard for NP.

- **Burham & Longpré 1996, Wang 1996**: No p-selective set $A$ can be weakly $\leq^p_m$-hard for EXP, i.e., must have $\mu(P_m(A)|\text{EXP}) = 0$.

**Theorem (J. Lutz, N. Lutz, & Mayordomo 2023).**

$$\dim(P_m(\text{qp-SEL}))|\text{EXP}) = 0.$$
Known Facts

- Selman 1979: No p-selective set can be \( \leq^p_m \)-hard for EXP.
  If \( P \neq NP \), then no p-selective set can be \( \leq^p_m \)-hard for NP.

- Burham & Longpré 1996, Wang 1996:
  No p-selective set \( A \) can be weakly \( \leq^p_m \)-hard for EXP, i.e., must have \( \mu(P_m(A)|EXP) = 0 \).

Theorem (J. Lutz, N. Lutz, & Mayordomo 2023).

\[
\text{dim}(P_m(\text{qp-SEL})|\text{EXP}) = 0.
\]

Hence,
\[
\text{dim}(NP|\text{EXP}) > 0 \Rightarrow \text{no qp-selective set can be } \leq^p_m \text{-hard for NP.}
\]
Thank you!

Thanks to CBMS and to NSF grants 2329555 and 1900716.