NSF-CBMS Research Conference
Algorithmic Fractal Dimensions

Lecture 4
Dimensions in computational complexity

Jack Lutz
Iowa State University

May 21, 2024
Drake University

Decision Problems

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

Decision Problems

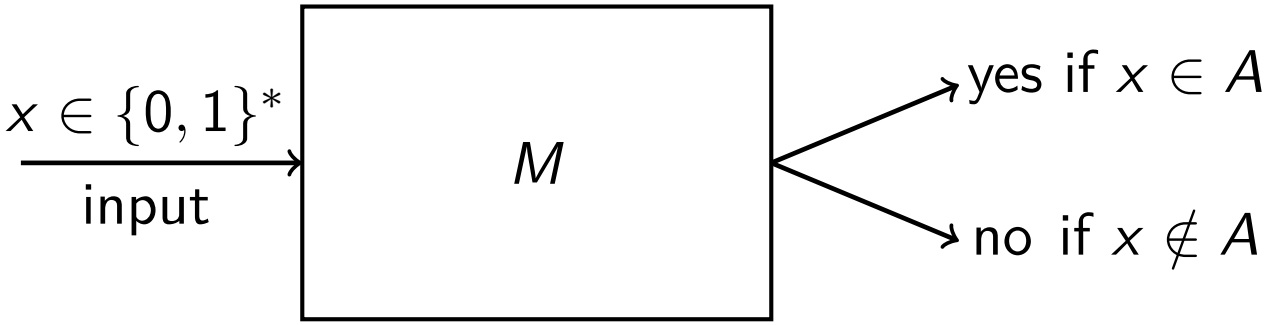
Much of computational complexity theory concerns decision problems, which are typically represented as sets $A \subseteq \{0,1\}^*$.

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 $x \in A$.

Decision Problems

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 $x \in A$.



Decision Problems

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),

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),

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,

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 $t : \mathbb{N} \to \mathbb{N}$,

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 $t : \mathbb{N} \to \mathbb{N}$,

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?

Decision Problems

If the devices are Turing machines and the resource is time, we write TIME(t) for the complexity class of decision problems A for which such a solution exists.

Decision Problems

If the devices are Turing machines and the resource is time, we write TIME(t) for the complexity class of decision problems A for which such a solution exists.

Minor correction: TIME(t) really means TIME($O(t)$).

Decision Problems

If the devices are Turing machines and the resource is time, we write TIME(t) for the complexity class of decision problems A for which such a solution exists.

Minor correction: TIME(t) really means TIME($O(t)$).

SPACE(t) is defined analogously.

Decision Problems

If the devices are Turing machines and the resource is time, we write TIME(t) for the complexity class of decision problems A for which such a solution exists.

Minor correction: TIME(t) really means TIME($O(t)$).

SPACE(t) is defined analogously.

Convention

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

Thus, for example, TIME(n^2) is the class of all decision problems that can be solved in quadratic time.

Complexity Classes

Some famous complexity classes:

Complexity Classes

Some famous complexity classes:

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

Complexity Classes

Some famous complexity classes:

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

$$\mathsf{PSPACE} = \bigcup_{k=1}^{\infty} \mathsf{SPACE}(n^k) \qquad \text{polynomial space}$$

Complexity Classes

Some famous complexity classes:

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

$$\mathsf{PSPACE} = \bigcup_{k=1}^{\infty} \mathsf{SPACE}(n^k) \qquad \text{polynomial space}$$

$$\mathsf{E} = \bigcup_{k=1}^{\infty} \mathsf{TIME}(2^{kn}) \qquad \text{linear-exponential time}$$

Complexity Classes

Some famous complexity classes:

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

$$\mathsf{PSPACE} = \bigcup_{k=1}^{\infty} \mathsf{SPACE}(n^k) \qquad \text{polynomial space}$$

$$\mathsf{E} = \bigcup_{k=1}^{\infty} \mathsf{TIME}(2^{kn}) \qquad \text{linear-exponential time}$$

$$\mathsf{EXP} = \bigcup_{k=1}^{\infty} \mathsf{TIME}(2^{n^k}) \qquad \text{exponential time}$$

Complexity Classes

Some things that we know:

$$\mathrm{P} \subsetneq \mathrm{E} \subsetneq \mathrm{EXP}$$

$$\mathrm{P} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXP}$$

$$\mathrm{PSPACE} \neq \mathrm{E}$$

Complexity Classes

Some things that we know:

$$\mathsf{P} \subsetneq \mathsf{E} \subsetneq \mathsf{EXP}$$

$$\mathsf{P} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXP}$$

$$\mathsf{PSPACE} \neq \mathsf{E}$$

We <u>believe</u> that $\mathsf{P} \subsetneq \mathsf{PSPACE} \subsetneq \mathsf{EXP}$, but we cannot rule out either equality.

Complexity Classes

Some things that we know:

$$\mathsf{P} \subsetneq \mathsf{E} \subsetneq \mathsf{EXP}$$

$$\mathsf{P} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXP}$$

$$\mathsf{PSPACE} \neq \mathsf{E}$$

We <u>believe</u> that $\mathsf{P} \subsetneq \mathsf{PSPACE} \subsetneq \mathsf{EXP}$, but we cannot rule out either equality.

We should not omit the famous class NP.

Complexity Classes

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

$$\langle x, y \rangle = 0^{|x|} 1 x y$$

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

Complexity Classes

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

$$\langle x, y \rangle = 0^{|x|} 1 x y$$

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 $x \in \exists^q B$.

Complexity Classes

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

$$\langle x, y \rangle = 0^{|x|} 1 x y$$

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 $x \in \exists^q B$.

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

Complexity Classes

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

$$\langle x, y \rangle = 0^{|x|} 1 x y$$

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 $x \in \exists^q B$.

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

We know that $\mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE}$. We believe that these inclusions are proper.

Complexity Classes

Why is the complexity class NP important?

Complexity Classes

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

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

Complexity Classes

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

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

We believe that these important problems C are intractable.

Complexity Classes

Why is the complexity class NP important?

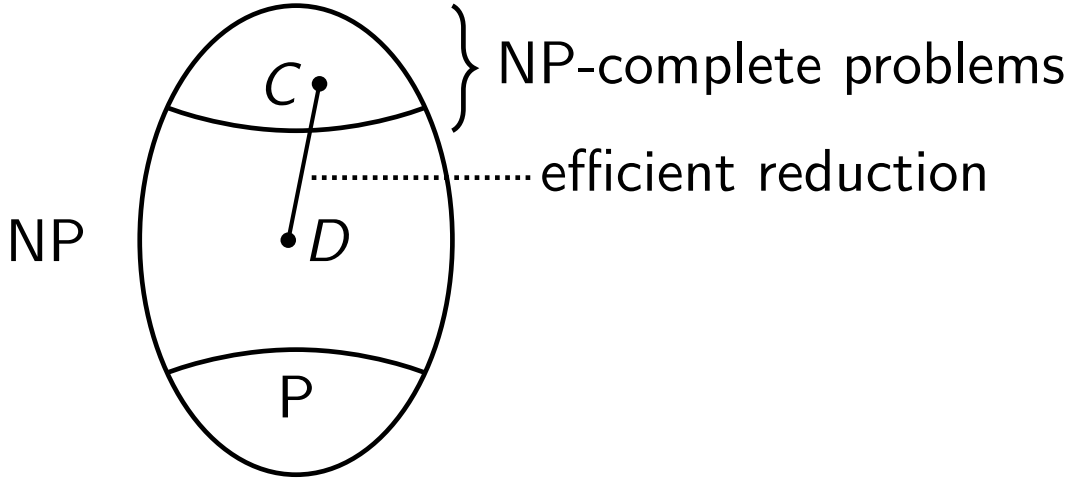
Because hundreds of very important problems in scientific computing can be formulated as decision problems $C \subseteq \{0,1\}^*$ that are NP-complete, meaning that

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

We believe that these important problems C are intractable.

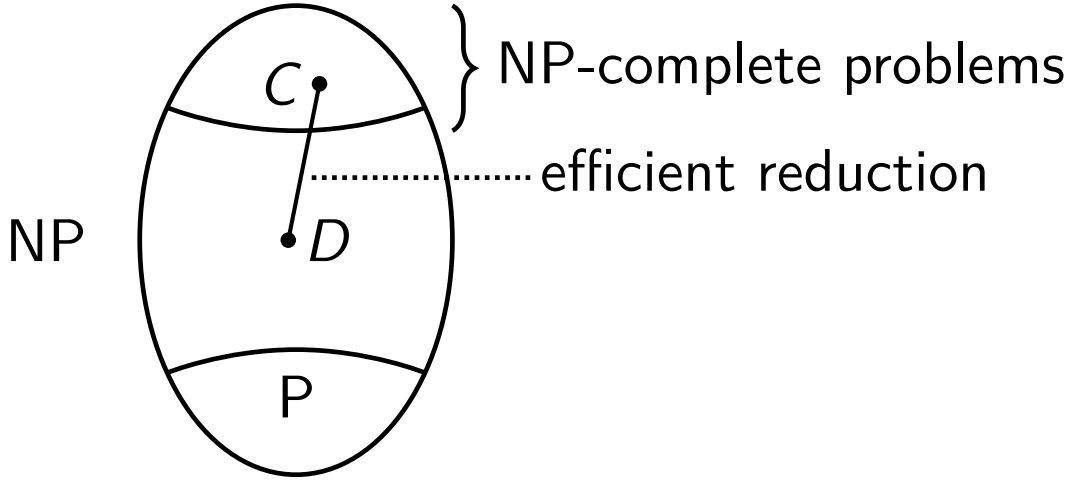
We can prove this if we can find any intractable problem $D \in \mathrm{NP}$.

Complexity Classes



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

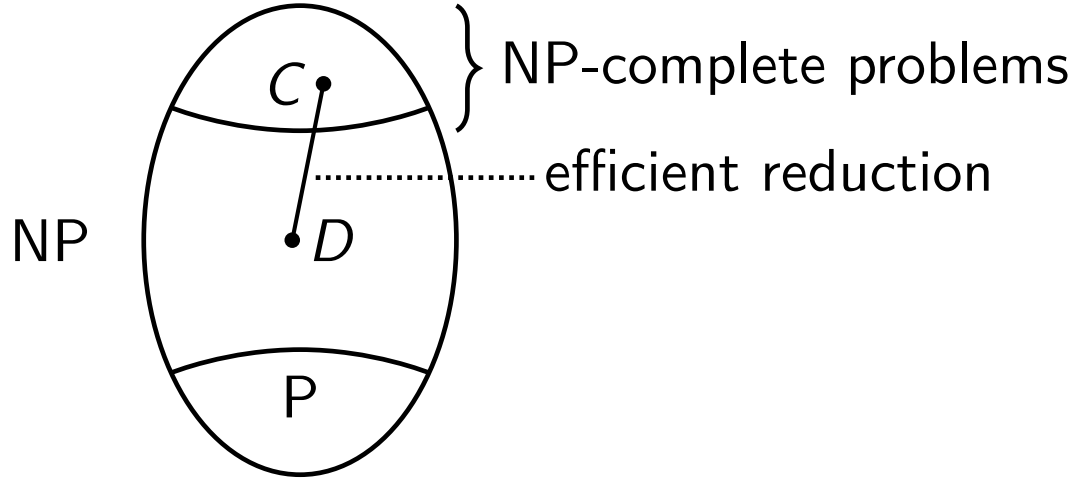
Complexity Classes



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?

Complexity Classes

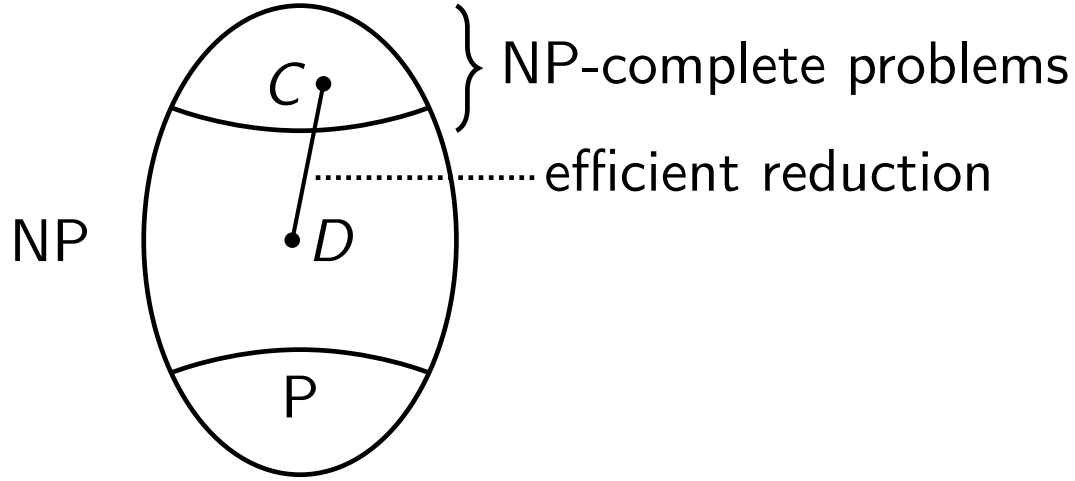


If D is intractable, then so is C, no matter how contrived D may be.

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 $\boldsymbol{C} = \{0,1\}^\omega$ 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 $\boldsymbol{C} = \{0,1\}^\omega$ 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.

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 $w \in \{0,1\}^*$, $w \sqsubsetneq \delta(w)$. That is, δ 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 $\delta : \{0,1\}^* \to \{0,1\}^*$ such that, for all $w \in \{0,1\}^*$, $w \sqsubsetneq \delta(w)$. That is, δ simply adds one or more bits to its input.

The result of a constructor δ 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 $w \in \{0,1\}^*$, $w \sqsubsetneq \delta(w)$. That is, δ simply adds one or more bits to its input.

The result of a constructor δ 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 Δ of functions is

$$R(\Delta) = \{R(\delta) | \delta \in \Delta \text{ is a constructor}\}.$$

Constructors and their Results

In this talk, a resource bound Δ is one of the following sets.

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

Constructors and their Results

In this talk, a resource bound Δ is one of the following sets.

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

Constructors and their Results

In this talk, a resource bound Δ is one of the following sets.

$$\text{all} = \{f | f : \{0,1\}^* \to \{0,1\}^*\}$$
$$\text{comp} = \{f \in \text{all} | f \text{ is computable}\}$$
$$\text{p} = \{f \in \text{all} | f \text{ is computable in } n^{O(1)} \text{ time}\}$$

Constructors and their Results

In this talk, a resource bound Δ is one of the following sets.

$$\text{all} = \{f | f : \{0,1\}^* \to \{0,1\}^*\}$$
$$\text{comp} = \{f \in \text{all} | f \text{ is computable}\}$$
$$\text{p} = \{f \in \text{all} | f \text{ is computable in } n^{O(1)} \text{ time}\}$$
$$\text{qp} = \{f \in \text{all} | f \text{ is computable in } n^{(logn)^{O(1)}} \text{ time}\}$$

Constructors and their Results

In this talk, a resource bound Δ is one of the following sets.

$$\text{all} = \{f | f : \{0,1\}^* \to \{0,1\}^*\}$$
$$\text{comp} = \{f \in \text{all} | f \text{ is computable}\}$$
$$\text{p} = \{f \in \text{all} | f \text{ is computable in } n^{O(1)} \text{ time}\}$$
$$\text{qp} = \{f \in \text{all} | f \text{ is computable in } n^{(logn)^{O(1)}} \text{ time}\}$$

pspace, qpspace

Constructors and their Results

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

Constructors and their Results

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.*

Constructors and their Results

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* $A \subseteq \{0,1\}^*$.

Constructors and their Results

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* $A \subseteq \{0,1\}^*$.
- $R(\text{p}) = \text{E} = \text{TIME}(2^{\text{linear}})$.

Constructors and their Results

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* $A \subseteq \{0,1\}^*$.
- $R(\text{p}) = \text{E} = \text{TIME}(2^{\text{linear}})$.

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

$$poly(|w|) = |w|^{O(1)} = \left(2^{|x|}\right)^{O(1)} = 2^{O(|x|)}.$$

Constructors and their Results

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* $A \subseteq \{0,1\}^*$.
- $R(\text{p}) = \text{E} = \text{TIME}(2^{\text{linear}})$.
- $R(\text{qp}) = \text{EXP}$.

Constructors and their Results

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* $A \subseteq \{0,1\}^*$.
- $R(\text{p}) = \text{E} = \text{TIME}(2^{\text{linear}})$.
- $R(\text{qp}) = \text{EXP}$.
- $R(\text{pspace}) = \text{ESPACE}$.

Constructors and their Results

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* $A \subseteq \{0,1\}^*$.
- $R(\text{p}) = \text{E} = \text{TIME}(2^{\text{linear}})$.
- $R(\text{qp}) = \text{EXP}$.
- $R(\text{pspace}) = \text{ESPACE}$.
- $R(\text{qpspace}) = \text{EXPSPACE}$.

Constructors and their Results

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

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

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

Constructors and their Results

Technical Note. Many of our functions will be of the form $f : D \to [0, \infty)$, where D is some discrete domain like $\{0,1\}^*$ or $\{0,1\}^* \times \mathbb{N}$. Such a function is Δ-computable if there is a function $\hat{f} : D \times \mathbb{N} \to \mathbb{Q} \cap [0, \infty)$ 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.

Dimensions in Complexity Classes

Notation If Δ is a resource bound and $X \subseteq \{0,1\}^\omega$, then

$$\mathcal{G}_\Delta(X) = \left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \text{there is a } \Delta\text{-computable} \\ s\text{-gale } d \text{ such that } X \subseteq S^\infty[d] \end{array} \right\}$$

and

$$\mathcal{G}^{str}_\Delta(X) = \left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \text{there is a } \Delta\text{-computable} \\ s\text{-gale } d \text{ such that } X \subseteq S^\infty_{str}[d] \end{array} \right\}$$

Dimensions in Complexity Classes

Notation If Δ is a resource bound and $X \subseteq \{0,1\}^\omega$, then

$$\mathcal{G}_\Delta(X) = \left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \text{there is a } \Delta\text{-computable} \\ s\text{-gale } d \text{ such that } X \subseteq S^\infty[d] \end{array} \right\}$$

and

$$\mathcal{G}^{str}_\Delta(X) = \left\{ s \in [0,\infty) \;\middle|\; \begin{array}{l} \text{there is a } \Delta\text{-computable} \\ s\text{-gale } d \text{ such that } X \subseteq S^\infty_{str}[d] \end{array} \right\}$$

We saw yesterday that

$$\dim_H(X) = \inf \mathcal{G}_{\mathsf{all}}(X)$$

and

$$\dim_P(X) = \inf \mathcal{G}^{str}_{\mathsf{all}}(X).$$

Dimensions in Complexity Classes

Definition (J. Lutz 2000, 2003).

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

1. The Δ-dimension of X is

$$\dim_\Delta(X) = \inf \mathcal{G}_\Delta(X).$$

Dimensions in Complexity Classes

Definition (J. Lutz 2000, 2003).

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

1. The Δ-dimension of X is

$$\dim_\Delta(X) = \inf \mathcal{G}_\Delta(X).$$

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

$$\dim(X|R(\Delta)) = \dim_\Delta(X \cap R(\Delta)).$$

Dimensions in Complexity Classes

Definition (Athreya, Hitchcock, J. Lutz, Mayordomo 2007).

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

1. The Δ-strong dimension of X is

$$\mathrm{Dim}_\Delta(X) = \inf \mathcal{G}_\Delta^{str}(X).$$

Dimensions in Complexity Classes

Definition (Athreya, Hitchcock, J. Lutz, Mayordomo 2007).

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

1. The Δ-strong dimension of X is

$$\mathrm{Dim}_\Delta(X) = \inf \mathcal{G}_\Delta^{str}(X).$$

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

$$\mathrm{Dim}(X|R(\Delta)) = \mathrm{Dim}_\Delta(X \cap R(\Delta)).$$

Dimensions in Complexity Classes

Observations

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

Dimensions in Complexity Classes

Observations

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

   E.g., $\dim_{\mathsf{p}}(E) = 1$. Diagonalize against p-computable 1-gales.

Dimensions in Complexity Classes

Observations

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

   E.g., $\dim_{\mathsf{p}}(E) = 1$. Diagonalize against p-computable 1-gales.

2. Stability

$$\dim_{\Delta}(X \cup Y) = \max\{\dim_{\Delta}(X), \dim_{\Delta}(Y)\}.$$

In fact, for "Δ-countable unions",

$$\dim_{\Delta}(\bigcup_{k=0}^{\infty} X_k) = \sup\{\dim_{\Delta}(X_k)|\ k \in \mathbb{N}\}.$$

Dimensions in Complexity Classes

Observations (continued)

3.

$$\begin{array}{ccccccc} \dim_{\mathsf{p}}(X) & \geq & \dim_{\mathsf{qp}}(X) & \geq & \dim_{\mathsf{comp}}(X) & \geq & \dim_H(X) \\ \vee\!| & & \vee\!| & & \vee\!| & & \\ \dim(X|\mathsf{E}) & & \dim(X|\mathsf{EXP}) & & \dim(X|\mathsf{DEC}) & & \end{array}$$

Dimensions in Complexity Classes

Observations (continued)

3.

$$\begin{array}{ccccccc} \dim_{\mathsf{p}}(X) & \geq & \dim_{\mathsf{qp}}(X) & \geq & \dim_{\mathsf{comp}}(X) & \geq & \dim_H(X) \\ \vee\!| & & \vee\!| & & \vee\!| & & \\ \dim(X|\mathsf{E}) & & \dim(X|\mathsf{EXP}) & & \dim(X|\mathsf{DEC}) & & \end{array}$$

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

$$\dim(\mathsf{TIME}(2^{kn})|\mathsf{E}) = \dim(\mathsf{TIME}(2^{n^k})|\mathsf{EXP}) = 0.$$

Dimensions in Complexity Classes

Observations (continued)

3.

$$\begin{array}{ccccccc} \dim_{\mathsf{p}}(X) & \geq & \dim_{\mathsf{qp}}(X) & \geq & \dim_{\mathsf{comp}}(X) & \geq & \dim_H(X) \\ \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \rotatebox{90}{$\leq$} & & \\ \dim(X|\mathsf{E}) & & \dim(X|\mathsf{EXP}) & & \dim(X|\mathsf{DEC}) & & \end{array}$$

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

$$\dim(\mathsf{TIME}(2^{kn})|\mathsf{E}) = \dim(\mathsf{TIME}(2^{n^k})|\mathsf{EXP}) = 0.$$

similarly for Dim.

Dimensions in Complexity Classes

We conclude today with just one sample application.

Dimensions in Complexity Classes

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 $A \in$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

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

Dimensions in Complexity Classes

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 $A \in$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.

Dimensions in Complexity Classes

Known Facts

- Selman 1979: No p-selective set can be $\leq_m^{\mathrm{p}}$-hard for EXP.

Dimensions in Complexity Classes

Known Facts

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

Dimensions in Complexity Classes

Known Facts

- Selman 1979: No p-selective set can be $\leq_m^{\mathrm{p}}$-hard for EXP.
  If P$\neq$ NP, then no p-selective set can be $\leq_m^{\mathrm{p}}$-hard for NP.
- Burham & Longpré 1996, Wang 1996:
  No p-selective set A can be weakly $\leq_m^{\mathrm{p}}$-hard for EXP, i.e, must have $\mu(P_m(A)|\mathrm{EXP}) = 0$.

Dimensions in Complexity Classes

Known Facts

- Selman 1979: No p-selective set can be $\leq_m^{\mathrm{p}}$-hard for EXP.
  If P$\neq$ NP, then no p-selective set can be $\leq_m^{\mathrm{p}}$-hard for NP.
- Burham & Longpré 1996, Wang 1996:
  No p-selective set A can be weakly $\leq_m^{\mathrm{p}}$-hard for EXP, i.e, must have $\mu(P_m(A)|\mathrm{EXP}) = 0$.

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

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

Dimensions in Complexity Classes

Known Facts

- Selman 1979: No p-selective set can be $\leq_m^{\mathrm{p}}$-hard for EXP.
  If P$\neq$ NP, then no p-selective set can be $\leq_m^{\mathrm{p}}$-hard for NP.
- Burham & Longpré 1996, Wang 1996:
  No p-selective set A can be weakly $\leq_m^{\mathrm{p}}$-hard for EXP, i.e, must have $\mu(P_m(A)|\mathrm{EXP}) = 0$.

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

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

Hence,
$\dim(\mathrm{NP}|\mathrm{EXP}) > 0 \Rightarrow$ no qp-selective set can be $\leq_m^{\mathrm{p}}$-hard for NP.

Thank you!

Thanks to CBMS and to NSF grants 2329555 and 1900716.