Symmetry of information and nonuniform lower bounds

Sylvain Perifel

LIP, ENS Lyon

June 7, 2007

Outline

Two complexity classes

- EXP: set of languages recognized in exponential time by a deterministic Turing machine

$$\mathrm{EXP} = \cup_{k \geq 0} \mathrm{DTIME}(2^{n^k}).$$

Two complexity classes

- EXP: set of languages recognized in exponential time by a deterministic Turing machine

$$\mathrm{EXP} = \cup_{k \geq 0} \mathrm{DTIME}(2^{n^k}).$$

- $\mathrm{P/poly}$: set of languages recognized by a family of polynomial-size boolean circuits (gates $\wedge$, $\vee$ and $\neg$, one circuit per input length)

Two complexity classes

- EXP: set of languages recognized in exponential time by a deterministic Turing machine — uniform

$$\mathrm{EXP} = \cup_{k \geq 0} \mathrm{DTIME}(2^{n^k}).$$

- $\mathrm{P/poly}$: set of languages recognized by a family of polynomial-size boolean circuits (gates $\wedge$, $\vee$ and $\neg$, one circuit per input length) — nonuniform

- Open question: $\mathrm{EXP} \subset \mathrm{P/poly}$?

Two complexity classes

- EXP: set of languages recognized in exponential time by a deterministic Turing machine — uniform

$$\mathrm{EXP} = \cup_{k \geq 0} \mathrm{DTIME}(2^{n^k}).$$

- $\mathrm{P/poly}$: set of languages recognized by a family of polynomial-size boolean circuits (gates $\wedge$, $\vee$ and $\neg$, one circuit per input length) — nonuniform

- Open question: $\mathrm{EXP} \subset \mathrm{P/poly}$?

- Main result: polynomial-time symmetry of information implies $\mathrm{EXP} \not\subset \mathrm{P/poly}$.

Remarks

- $\mathrm{EXP} \neq \mathrm{P/poly}$ (there are undecidable languages in $\mathrm{P/poly}$).

Remarks

- $\mathrm{EXP} \neq \mathrm{P/poly}$ (there are undecidable languages in $\mathrm{P/poly}$).
- $\mathrm{EXP} \neq \mathrm{P}$ (time hierarchy theorem).

Remarks

- $\mathrm{EXP} \neq \mathrm{P/poly}$ (there are undecidable languages in $\mathrm{P/poly}$).
- $\mathrm{EXP} \neq \mathrm{P}$ (time hierarchy theorem).
- Space complexity version:

$$\mathrm{PSPACE} \subset \mathrm{NC/poly}?$$

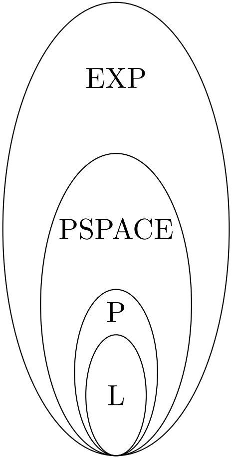
Remarks

- $\mathrm{EXP} \neq \mathrm{P/poly}$ (there are undecidable languages in $\mathrm{P/poly}$).
- $\mathrm{EXP} \neq \mathrm{P}$ (time hierarchy theorem).
- Space complexity version:

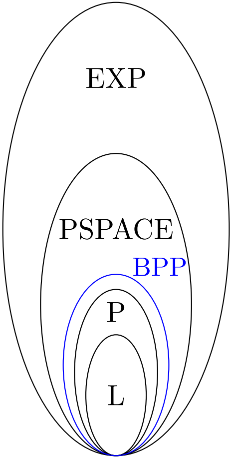
$$\mathrm{PSPACE} \subset \mathrm{NC/poly?}$$

- Even the question "$\mathrm{EXP} \subset \mathrm{L/poly?}$" is open.

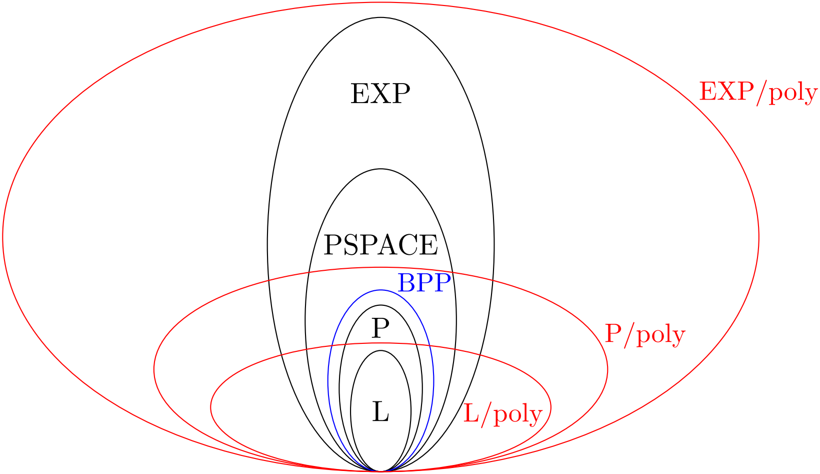
Some classes



Some classes



Some classes



Advices

- A Turing machine can be helped by an advice (one word given for all inputs of same size).

Advices

- A Turing machine can be helped by an advice (one word given for all inputs of same size).
- If $\mathcal{C}$ is a complexity class and $a : \mathbb{N} \to \mathbb{N}$ a function, then $\mathcal{C}/a(n)$ is the set of languages A such that there exists $B \in \mathcal{C}$ and a function $c : \mathbb{N} \to \{0,1\}^*$ satisfying:
  - $\forall n, |c(n)| \leq a(n)$;
  - $\forall x \in \{0,1\}^*$, $x \in A \iff (x, c(|x|)) \in B$.
- "The class $\mathcal{C}$ is helped by the advice $c(|x|)$" (the same for all words of each length).

Advices (quizz)

- $\mathrm{P}/0 = ?$

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.
- $\mathrm{P}/2^n = ?$

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.
- $\mathrm{P}/2^n = \mathcal{P}(\{0,1\}^*)$.

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.
- $\mathrm{P}/2^n = \mathcal{P}(\{0,1\}^*)$.
- $\mathrm{P}/1 \subseteq ?$

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.
- $\mathrm{P}/2^n = \mathcal{P}(\{0,1\}^*)$.
- $\mathrm{P}/1$ is uncountable and contains undecidable languages...

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.
- $\mathrm{P}/2^n = \mathcal{P}(\{0,1\}^*)$.
- $\mathrm{P}/1$ is uncountable and contains undecidable languages...
- $\mathrm{P}/\mathrm{poly} = ?$

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.
- $\mathrm{P}/2^n = \mathcal{P}(\{0,1\}^*)$.
- $\mathrm{P}/1$ is uncountable and contains undecidable languages...
- $\mathrm{P/poly} = \cup_{k \geq 0} \mathrm{P}/n^k$ (polynomial-size advice).

  $\mathrm{P/poly}$: conversion advice $\longleftrightarrow$ boolean circuit.

Advices (quizz)

- $\mathrm{P}/0 = \mathrm{P}$.
- $\mathrm{P}/2^n = \mathcal{P}(\{0,1\}^*)$.
- $\mathrm{P}/1$ is uncountable and contains undecidable languages...
- $\mathrm{P/poly} = \cup_{k \geq 0} \mathrm{P}/n^k$ (polynomial-size advice).

  $\mathrm{P/poly}$: conversion advice $\longleftrightarrow$ boolean circuit.
- $\mathrm{EXP} \subset \mathrm{P/poly} \iff \mathrm{EXP/poly} = \mathrm{P/poly}$.

Links with derandomization

- Pseudo-random generators approach: Yao 1982, Nisan & Wigderson 1994

Links with derandomization

- Pseudo-random generators approach: Yao 1982, Nisan & Wigderson 1994
- Impagliazzo & Wigderson 1997: if EXP requires circuits of exponential size, then $\mathrm{BPP} = \mathrm{P}$.
- Babai, Fortnow, Nisan & Wigderson 1993: if $\mathrm{EXP} \not\subset \mathrm{P/poly}$ then BPP has subexponential-time deterministic algorithms.

Links with derandomization

- Pseudo-random generators approach: Yao 1982, Nisan & Wigderson 1994
- Impagliazzo & Wigderson 1997: if EXP requires circuits of exponential size, then $\mathrm{BPP} = \mathrm{P}$.
- Babai, Fortnow, Nisan & Wigderson 1993: if $\mathrm{EXP} \not\subset \mathrm{P/poly}$ then BPP has subexponential-time deterministic algorithms.
- For the other direction, Kabanets & Impagliazzo 2002: if $\mathrm{P} = \mathrm{BPP}$ then NEXP does not have polynomial-size circuits.

The question "$\mathrm{EXP} \subset \mathrm{P/poly}$?"

- Simple diagonalization fails (too many circuits).

The question "$\mathrm{EXP} \subset \mathrm{P/poly}$?"

- Simple diagonalization fails (too many circuits).
- Kannan 1982: $\mathrm{NEXP}^{\mathrm{NP}} \not\subset \mathrm{P/poly}$;
- Schöning 1985: $\mathrm{EXPSPACE} \not\subset \mathrm{P/poly}$.

The question "$\mathrm{EXP} \subset \mathrm{P/poly}$?"

- Simple diagonalization fails (too many circuits).
- Kannan 1982: $\mathrm{NEXP}^{\mathrm{NP}} \not\subset \mathrm{P/poly}$;
- Schöning 1985: $\mathrm{EXPSPACE} \not\subset \mathrm{P/poly}$.
- Homer & Mocas 1995: $\forall c > 0, \mathrm{EXP} \not\subset \mathrm{P}/n^c$.

The question "$\mathrm{EXP} \subset \mathrm{P/poly}$?"

- Simple diagonalization fails (too many circuits).
- Kannan 1982: $\mathrm{NEXP}^{\mathrm{NP}} \not\subset \mathrm{P/poly}$;
- Schöning 1985: $\mathrm{EXPSPACE} \not\subset \mathrm{P/poly}$.
- Homer & Mocas 1995: $\forall c > 0, \mathrm{EXP} \not\subset \mathrm{P}/n^c$.
- Here: symmetry of information $(\mathrm{SI_p}) \Rightarrow \mathrm{EXP} \not\subset \mathrm{P/poly}$;
- Lee & Romashchenko 2004: $(\mathrm{SI_p}) \Rightarrow \mathrm{EXP} \neq \mathrm{BPP}$
  (remark: $\mathrm{BPP} \subset \mathrm{P/poly}$, Adleman 1978).

Advices of size n^c

- Words of $\{0,1\}^n$ are ordered lexicographically $x_1 < x_2 < \cdots < x_{2^n}$.
- We fix an "efficient" universal Turing machine $\mathcal{U}$.

Advices of size n^c

- Words of $\{0,1\}^n$ are ordered lexicographically $x_1 < x_2 < \cdots < x_{2^n}$.
- We fix an "efficient" universal Turing machine $\mathcal{U}$.

Lemma

If $A \in \mathrm{P}/n^c$ then there exists a constant k and a family (p_n) of programs of size $k + n^c$ such that

- $\mathcal{U}(p_n, x) = 1$ *iff* $x \in A$*;*
- $\mathcal{U}(p_n, x)$ *works in polynomial time.*

Advices of size n^c

- Words of $\{0,1\}^n$ are ordered lexicographically $x_1 < x_2 < \cdots < x_{2^n}$.
- We fix an "efficient" universal Turing machine $\mathcal{U}$.

Lemma

If $A \in \mathrm{P}/n^c$ then there exists a constant k and a family (p_n) of programs of size $k + n^c$ such that

- $\mathcal{U}(p_n, x) = 1$ *iff* $x \in A$;
- $\mathcal{U}(p_n, x)$ *works in polynomial time.*

Proof

By definition, $x \in A \Longleftrightarrow (x, c(|x|)) \in B$. Then p_n is merely the concatenation of the program for B and of $c(n)$. □

Advices of size n^c (continued)

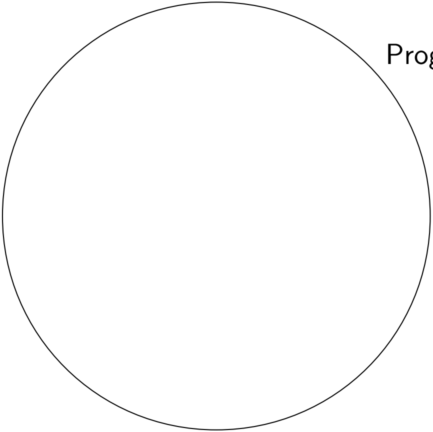
Proposition (warm-up)

For all constants $c_1, c_2 \geq 1$, there exists a sparse language A in $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ but not in $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.

Advices of size n^c (continued)

Proposition (warm-up)

For all constants $c_1, c_2 \geq 1$, there exists a sparse language A in $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ but not in $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.

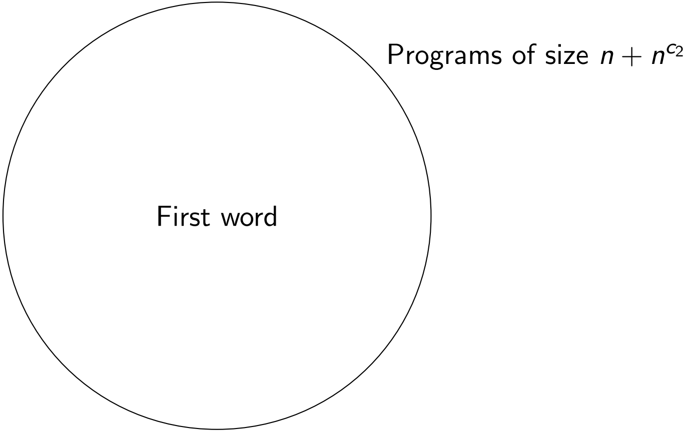


Programs of size $n + n^{c_2}$

Advices of size n^c (continued)

Proposition (warm-up)

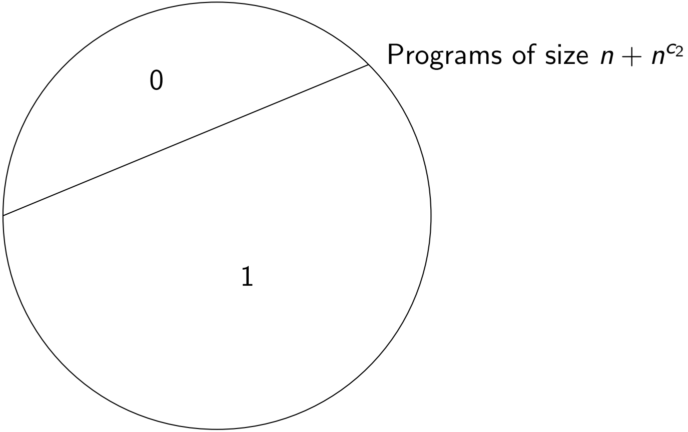
For all constants $c_1, c_2 \geq 1$*, there exists a sparse language* A *in* $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ *but not in* $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$*.*



Advices of size n^c (continued)

Proposition (warm-up)

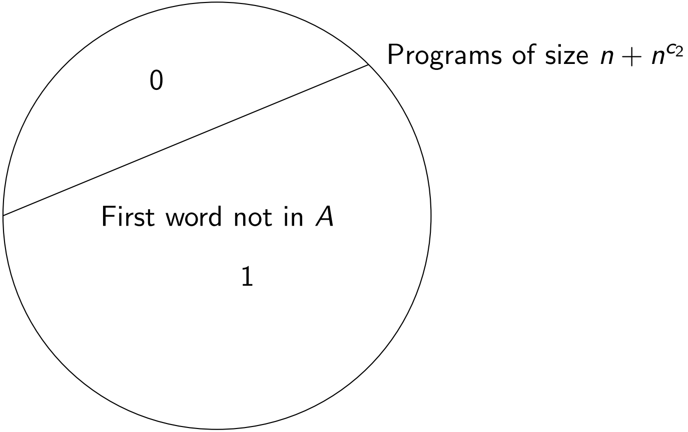
For all constants $c_1, c_2 \geq 1$, there exists a sparse language A in $\mathrm{DTIME}(2^{n^{1+c_1c_2}})$ but not in $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

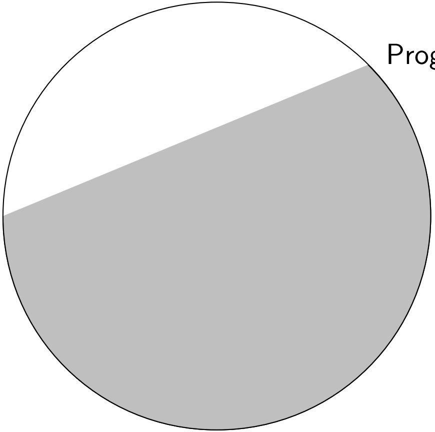
For all constants $c_1, c_2 \geq 1$*, there exists a sparse language* A *in* $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ *but not in* $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

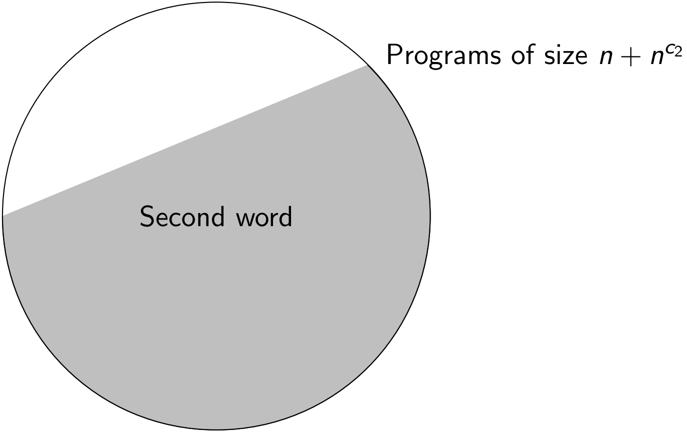
For all constants $c_1, c_2 \geq 1$, *there exists a sparse language* A *in* $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ *but not in* $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

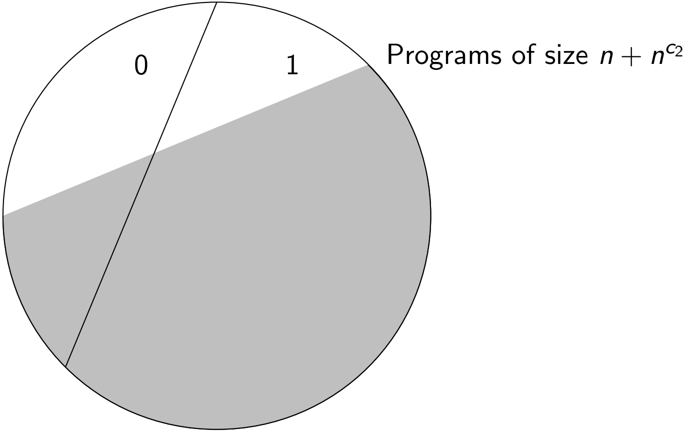
For all constants $c_1, c_2 \geq 1$, *there exists a sparse language* A *in* $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ *but not in* $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

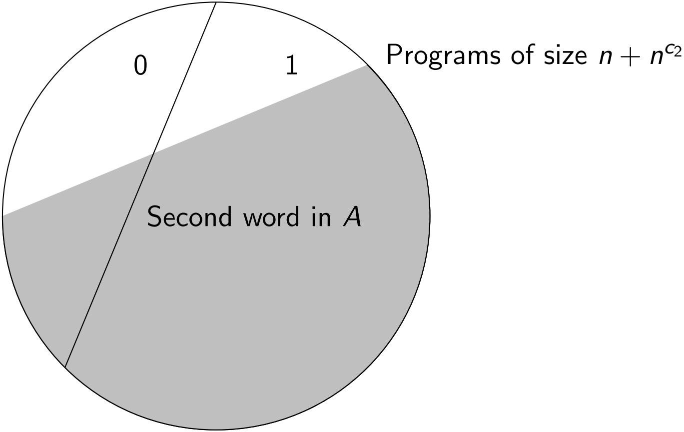
For all constants $c_1, c_2 \geq 1$, there exists a sparse language A in $\mathrm{DTIME}(2^{n^{1+c_1c_2}})$ but not in $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

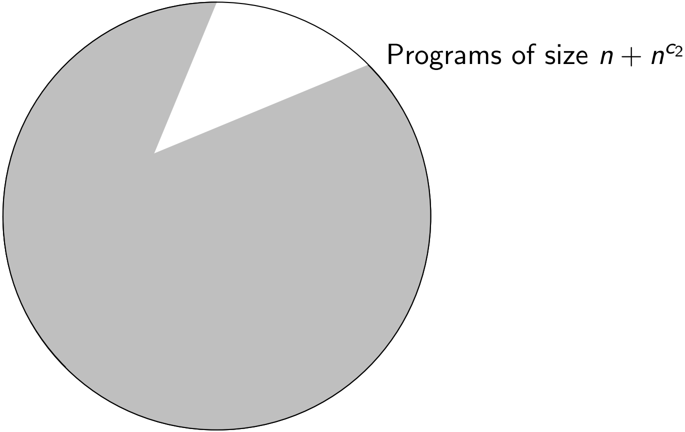
For all constants $c_1, c_2 \geq 1$, there exists a sparse language A in $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ but not in $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

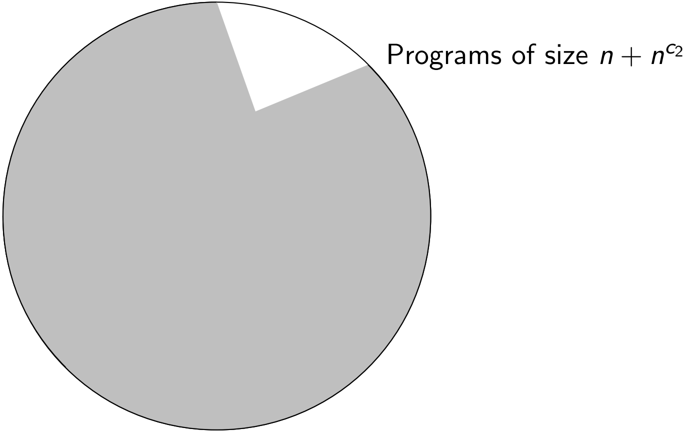
For all constants $c_1, c_2 \geq 1$, *there exists a sparse language* A *in* $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ *but not in* $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

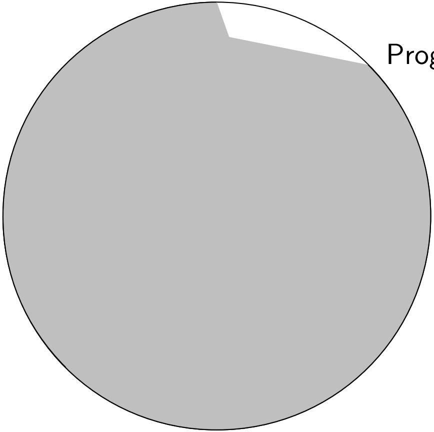
For all constants $c_1, c_2 \geq 1$, *there exists a sparse language* A *in* $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ *but not in* $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.



Advices of size n^c (continued)

Proposition (warm-up)

For all constants $c_1, c_2 \geq 1$*, there exists a sparse language* A *in* $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ *but not in* $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$*.*



Programs of size $n + n^{c_2}$

Advices of size n^c (continued)

Proposition (warm-up)

For all constants $c_1, c_2 \geq 1$, there exists a sparse language A in $\mathrm{DTIME}(2^{n^{1+c_1 c_2}})$ but not in $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.

Proof

We build A by input sizes and word by word. Let $t(n) = 2^{n^{1+c_1 c_2}}$ and $a(n) = n + n^{c_2}$. Let us fix n and define $A^{=n}$:

$$x_1 \in A \iff \begin{array}{l} \text{for at least half of the programs } p \text{ of size } \leq a(n), \\ \mathcal{U}^{t(n)}(p, x_1) = 0. \end{array}$$

(at least half of the programs give the wrong answer for x_1).

Let V_1 be the set of programs giving the right answer for x_1.

Advices of size n^c (proof continued)

$$x_2 \in A \iff \begin{array}{l} \text{for at least half of the programs } p \in V_1, \\ \mathcal{U}^{t(n)}(p, x_2) = 0. \end{array}$$

(at least half of the remaining programs are wrong on x_2).

Advices of size n^c (proof continued)

$$x_2 \in A \iff \begin{array}{l}\text{for at least half of the programs } p \in V_1, \\ \mathcal{U}^{t(n)}(p, x_2) = 0.\end{array}$$

(at least half of the remaining programs are wrong on x_2).

and so on. . .

Advices of size n^c (proof continued)

$$x_2 \in A \iff \begin{array}{l} \text{for at least half of the programs } p \in V_1, \\ \mathcal{U}^{t(n)}(p, x_2) = 0. \end{array}$$

(at least half of the remaining programs are wrong on x_2).

and so on...

$$x_k \in A \iff \begin{array}{l} \text{for at least half of the programs } p \in V_{k-1}, \\ \mathcal{U}^{t(n)}(p, x_k) = 0. \end{array}$$

The process stops when V_k is empty, that is, for $k = n + n^{c_2}$. We decide that $x_j \notin A$ for $j > k$.

Advices of size n^c (proof continued)

- A is sparse (at most $n + n^{c_2}$ elements of size n);

Advices of size n^c (proof continued)

- A is sparse (at most $n + n^{c_2}$ elements of size n);
- $A \notin \mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$: any program with any advice makes at least one mistake;

Advices of size n^c (proof continued)

- A is sparse (at most $n + n^{c_2}$ elements of size n);
- $A \notin \mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$: any program with any advice makes at least one mistake;
- $A \in \mathrm{DTIME}(2^{n^{1+c_1 c_2}})$. □

Some consequences

Corollary

For all constant $c > 0$, $\mathrm{EXP} \not\subset \mathrm{P}/n^c$ *and*
$\mathrm{PSPACE} \not\subset (\cup_k \mathrm{DSPACE}(\log^k n)/n^c)$.

Some consequences

Corollary

For all constant $c > 0$, $\mathrm{EXP} \not\subset \mathrm{P}/n^c$ *and* $\mathrm{PSPACE} \not\subset (\cup_k \mathrm{DSPACE}(\log^k n)/n^c)$.

Corollary

For all $c > 0$, $\mathrm{PP} \not\subset \mathrm{DTIME}(n^c)/(n - \log n)$.

Proof idea

It $t(n) = n^c$, deciding whether "for at least half of the programs $p \in V_{k-1}$, $\mathcal{U}^{t(n)}(p, x_k) = 0$" is a PP problem.

Therefore we can decide A with a PP oracle. □

Some consequences

Corollary

For all constant $c > 0$, $\mathrm{EXP} \not\subset \mathrm{P}/n^c$ *and* $\mathrm{PSPACE} \not\subset (\cup_k \mathrm{DSPACE}(\log^k n)/n^c)$.

Corollary

For all $c > 0$, $\mathrm{PP} \not\subset \mathrm{DTIME}(n^c)/(n - \log n)$.

Theorem (Vinodchandran 2004)

For any fixed $c > 0$, PP *does not have circuits of size* n^c.

Kolmogorov complexity

- Plain Kolmogorov complexity:
  $C(x|y) = \min\{|p| \ : \ \mathcal{U}(p, y) = x\}$.

Kolmogorov complexity

- Plain Kolmogorov complexity:
  $C(x|y) = \min\{|p| \ : \ \mathcal{U}(p, y) = x\}$.

- Resource-bounded Kolmogorov complexity: $\mathcal{U}$ is required to run within a time bound t

$$C^t(x|y) = \min\{|p| \ : \ \mathcal{U}^t(p, y) = x\}.$$

Kolmogorov complexity

- Plain Kolmogorov complexity:
  $C(x|y) = \min\{|p| \, : \, \mathcal{U}(p, y) = x\}$.

- Resource-bounded Kolmogorov complexity: $\mathcal{U}$ is required to run within a time bound t

$$C^t(x|y) = \min\{|p| \, : \, \mathcal{U}^t(p, y) = x\}.$$

- Typical time bound: polynomial or exponential. There could also be a space bound.

Links Kolmogorov/nonuniform complexity

Characteristic string $\chi^n \in \{0,1\}^{2^n}$ of $A^{=n}$:

$$\chi^n_i = 1 \iff x_i \in A^{=n}.$$

Lemma

Suppose that for all n and some $1 \le i \le 2^n$ we have

$$C^{ir(n)}(\chi^n[1..i]) > n + a(n).$$

Then $A \notin \mathrm{DTIME}(r(n))/a(n)$.

Links Kolmogorov/nonuniform complexity

Characteristic string $\chi^n \in \{0,1\}^{2^n}$ of $A^{=n}$:

$$\chi_i^n = 1 \iff x_i \in A^{=n}.$$

Lemma

Suppose that for all n and some $1 \leq i \leq 2^n$ we have

$$C^{ir(n)}(\chi^n[1..i]) > n + a(n).$$

Then $A \notin \mathrm{DTIME}(r(n))/a(n)$.

Proof

If $A \in \mathrm{DTIME}(r(n))/a(n)$ then $\chi^n[1..i]$ is computed in time $ir(n)$ with a program of size $a(n) + O(1)$. □

Symmetry of information

- Symmetry of information (Levin, Kolmogorov): given x and y, x contains as much information on y as y on x

$$C(y) - C(y|x) \simeq C(x) - C(x|y).$$

Symmetry of information

- Symmetry of information (Levin, Kolmogorov): given x and y, x contains as much information on y as y on x

$$C(y) - C(y|x) \simeq C(x) - C(x|y).$$

- The (equivalent) version we will use:

$$C(x, y) \simeq C(x) + C(y|x).$$

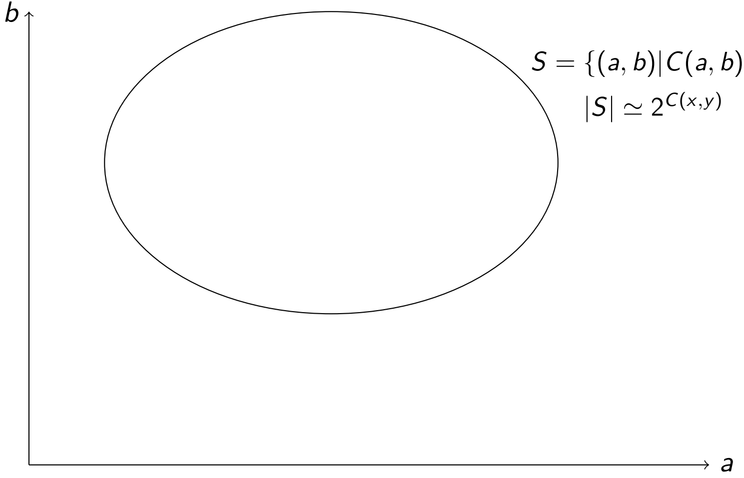
$\leq$: easy direction $\geq$: hard direction.

Symmetry of information (idea of the proof)

$$C(x, y) \geq C(x) + C(y|x).$$

Symmetry of information (idea of the proof)

$$C(x, y) \geq C(x) + C(y|x).$$

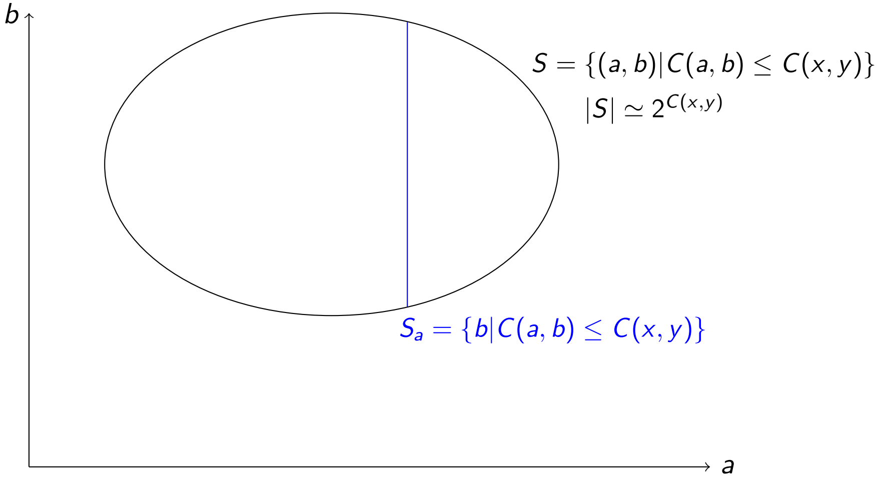


$S = \{(a, b) | C(a, b) \leq C(x, y)\}$

$|S| \simeq 2^{C(x,y)}$

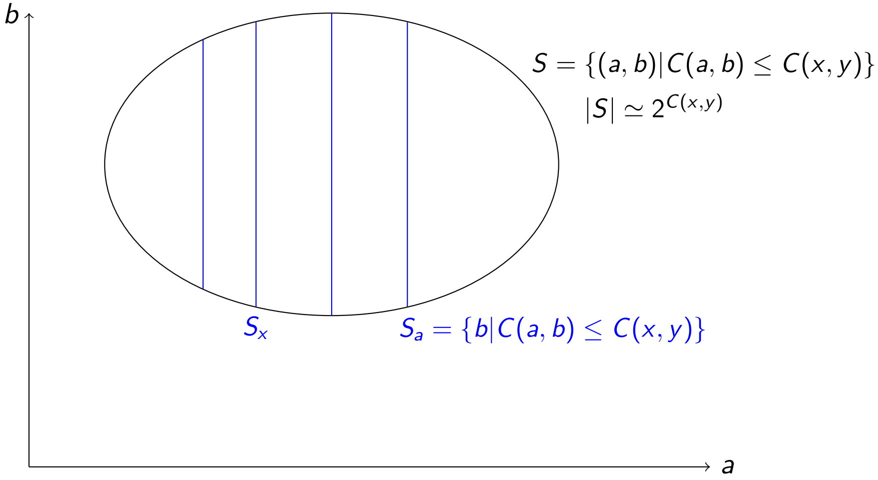
Symmetry of information (idea of the proof)

$$C(x, y) \geq C(x) + C(y|x).$$



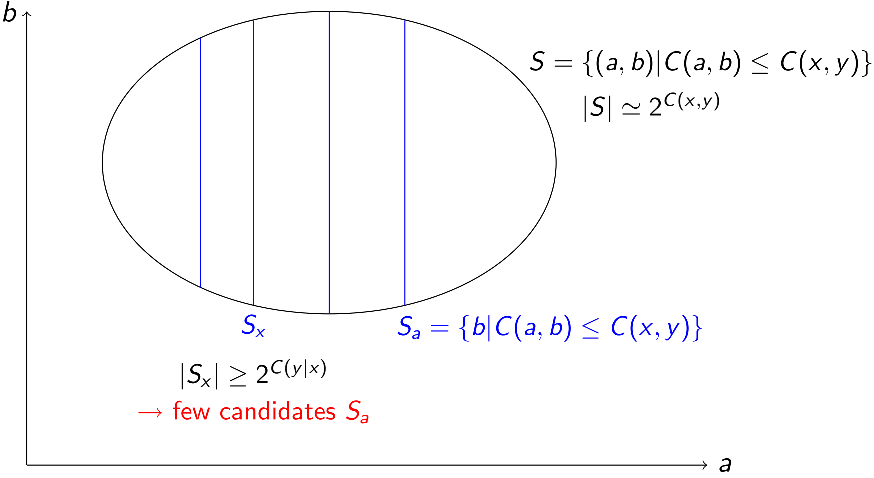
Symmetry of information (idea of the proof)

$$C(x, y) \geq C(x) + C(y|x).$$



Symmetry of information (idea of the proof)

$$C(x, y) \geq C(x) + C(y|x).$$



Symmetry of information (idea of the proof)

$$C(x, y) \geq C(x) + C(y|x).$$

Exponential time bounds $\rightarrow$ OK.

Symmetry of information (idea of the proof)

$$C(x, y) \geq C(x) + C(y|x).$$

Exponential time bounds $\rightarrow$ OK.

Polynomial-time symmetry of information: easy direction still holds; hard direction is open!
(true if $\mathrm{P} = \mathrm{NP}$, Longpré & Watanabe 1995).

Symmetry of information

Hypothesis ($\mathrm{SI_p}$)

There exist a polynomial q and a constant $\alpha > 1/2$ such that for all t and all words x, y of size n:

$$C^t(x,y) \geq \alpha(C^{tq(n)}(x) + C^{tq(n)}(y|x)).$$

Symmetry of information

Hypothesis $(\mathrm{SI_p})$

There exist a polynomial q and a constant $\alpha > 1/2$ such that for all t and all words x, y of size n:

$$C^t(x, y) \geq \alpha(C^{tq(n)}(x) + C^{tq(n)}(y|x)).$$

Remark: stronger time bounds than the usual ones
$tq(n)$ instead of $q(t)$.

Symmetry of information

Hypothesis $(\mathrm{SI_p})$

There exist a polynomial q and a constant $\alpha > 1/2$ such that for all t and all words x, y of size n:

$$C^t(x,y) \geq \alpha(C^{q(t)}(x) + C^{q(t)}(y|x)).$$

Remark: stronger time bounds than the usual ones
$tq(n)$ instead of $q(t)$.

Symmetry of information

Hypothesis $(\mathrm{SI_p})$

There exist a polynomial q and a constant $\alpha > 1/2$ such that for all t and all words x, y of size n:

$$C^t(x, y) \geq \alpha(C^{tq(n)}(x) + C^{tq(n)}(y|x)).$$

Remark: stronger time bounds than the usual ones
$tq(n)$ instead of $q(t)$.

Iterations of $(\mathrm{SI_p})$

Lemma

Suppose $(\mathrm{SI_p})$ *holds.*
Let $u_1, \ldots, u_n$ *be words of size* s. *Let* $m = ns$. *Suppose there exists* k *such that for all* $j \leq n$,

$$C^{tq(m)^{\log n}}(u_j | u_1, \ldots, u_{j-1}) \geq k.$$

Then $C^t(u_1, \ldots, u_n) \geq n^{\log(2\alpha)} k$.

Iterations of $(\mathrm{SI_p})$

Lemma

Suppose $(\mathrm{SI_p})$ *holds.*
Let $u_1, \ldots, u_n$ *be words of size* s. *Let* $m = ns$. *Suppose there exists* k *such that for all* $j \leq n$,

$$C^{tq(m)^{\log n}}(u_j | u_1, \ldots, u_{j-1}) \geq k.$$

Then $C^t(u_1, \ldots, u_n) \geq n^{\log(2\alpha)} k$.

Proof sketch

$$C^t(u_1, \ldots, u_n) \geq \alpha(C^{tq(m)}(u_1, \ldots, u_{n/2}) +$$
$$C^{tq(m)}(u_{n/2+1}, \ldots, u_n | u_1, \ldots, u_{n/2})). \quad \square$$

Polynomial-size advices — the idea

- In EXP, impossible to diagonalize over all advices of polynomial size
- $\rightarrow$ we cut the advices into blocks of size n and diagonalize over these blocks;

Polynomial-size advices — the idea

- In EXP, impossible to diagonalize over all advices of polynomial size
- $\rightarrow$ we cut the advices into blocks of size n and diagonalize over these blocks;
- then we “glue” these blocks together thanks to $(\mathrm{SI_p})$.

Polynomial-size advices — the idea

- In EXP, impossible to diagonalize over all advices of polynomial size
- $\rightarrow$ we cut the advices into blocks of size n and diagonalize over these blocks;
- then we "glue" these blocks together thanks to $(\mathrm{SI_p})$.

- Other point of view: thanks to $(\mathrm{SI_p})$, build a characteristic string of high Kolmogorov complexity.

Main result

Theorem

If $(\mathrm{SI_p})$ *holds, then* $\mathrm{EXP} \not\subset \mathrm{P/poly}$.

Main result

Theorem

If $(\mathrm{SI_p})$ *holds, then* $\mathrm{EXP} \not\subset \mathrm{P/poly}$.

Outline of the proof: feedback with previously defined segments.

Proof

We build A by input sizes and word by word. Let $t(n) = n^{\log^3 n}$.
Let us fix n and define $A^{=n}$:

$$x_1 \in A \Longleftrightarrow \begin{array}{l} \text{for at least half of the programs } p \text{ of size } \leq n, \\ \mathcal{U}^{t(n)}(p, x_1) = 0. \end{array}$$

(at least half of the programs give the wrong answer for x_1).

Main result

Theorem

If $(\mathrm{SI_p})$ *holds, then* $\mathrm{EXP} \not\subset \mathrm{P/poly}$.

Outline of the proof: feedback with previously defined segments.

Proof

We build A by input sizes and word by word. Let $t(n) = n^{\log^3 n}$.
Let us fix n and define $A^{=n}$:

$$x_1 \in A \Longleftrightarrow \begin{array}{l}\text{for at least half of the programs } p \text{ of size } \leq n, \\ \mathcal{U}^{t(n)}(p, x_1) = 0.\end{array}$$

(at least half of the programs give the wrong answer for x_1).

Let V_1 be the set of programs giving the right answer for x_1.

Proof (continued)

We go on like this, discarding half of the remaining programs at each step, until x_n:

$$x_n \in A \iff \begin{array}{l} \text{for at least half of the programs } p \in V_{n-1}, \\ \mathcal{U}^{t(n)}(p, x_n) = 0. \end{array}$$

We call $u^{(1)}$ the n first bits of the characteristic string of $A^{=n}$ just defined.

Proof (continued)

Then:

$$x_{n+1} \in A \iff \begin{array}{l}\text{for at least half of the programs } p \text{ of size } \leq n,\\ \mathcal{U}^{t(n)}(p, u^{(1)}, x_{n+1}) = 0.\end{array}$$

(at least half of the programs are wrong on x_{n+1}, even with the advice $u^{(1)}$).

Proof (continued)

Then:

$$x_{n+1} \in A \iff \begin{array}{l} \text{for at least half of the programs } p \text{ of size } \leq n, \\ \mathcal{U}^{t(n)}(p, u^{(1)}, x_{n+1}) = 0. \end{array}$$

(at least half of the programs are wrong on x_{n+1}, even with the advice $u^{(1)}$).

Keep going: call V_1 the set of programs that where right at the preceding step.

$$x_{n+2} \in A \iff \begin{array}{l} \text{for at least half of the programs } p \in V_1, \\ \mathcal{U}^{t(n)}(p, u^{(1)}, x_{n+2}) = 0. \end{array}$$

Proof continued

And so on, until the next segment $u^{(2)}$ of size n is defined. Then:

$$x_{2n+1} \in A \iff \begin{array}{l} \text{for at least half of the programs } p \text{ of size } \leq n, \\ \mathcal{U}^{t(n)}(p, u^{(1)}, u^{(2)}, x_{2n+1}) = 0. \end{array}$$

(at least half of the programs give the wrong answer for x_{2n+1}, even with the advice $u^{(1)}, u^{(2)}$).

We define $n^{\log n}$ segments of size n and decide that $x_j \notin A^{=n}$ for $j > n \times n^{\log n}$.

Proof continued

- $A \notin \mathrm{P/poly}$ because for all j, $C^{t(n)}(u^{(j)} | u^{(1)}, \ldots, u^{(j-1)}) \geq n - 1$. Thus by iteratively applying $(\mathrm{SI_p})$, $C^{t}(\chi^{n}[1..n^{1+\log n}]) \geq n^{\Omega(\log n)}$.
- $A \in \mathrm{EXP}$. $\square$

Proof continued

- $A \notin \mathrm{P/poly}$ because for all j, $C^{t(n)}(u^{(j)}|u^{(1)}, \ldots, u^{(j-1)}) \geq n - 1$. Thus by iteratively applying $(\mathrm{SI_p})$, $C^t(\chi^n[1..n^{1+\log n}]) \geq n^{\Omega(\log n)}$.
- $A \in \mathrm{EXP}$. □

Corollary

If $(\mathrm{SI_p})$ *holds, then there exists a constant* $c > 0$ *such that*

$$\mathrm{BPP} \subseteq \mathrm{DTIME}(2^{\log^c n}).$$

Conclusion

- $(\mathrm{SI_p})$ is a central (and hard) question: if true, then $\mathrm{EXP} \not\subset \mathrm{P/poly}$; if false, then $\mathrm{P} \neq \mathrm{NP}\ldots$
- What about the usual version of $(\mathrm{SI_p})$ (with time bound $q(t)$ instead of $tq(n)$)?
- Can we obtain unconditionnal results by using variants of Kolmogorov complexity ? (for instance CAMD, a version based on the class AM).

Outline

1. Complexity classes
2. Advices of size n^c
3. Symmetry of information
4. Polynomial-size advices