Symmetry of information and nonuniform lower bounds

Sylvain Perifel

March 7, 2007

Outline

1. Introduction and notations
2. Advices of size n^c
3. Symmetry of information
4. Polynomial-size advices

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 $\mathrm{P/poly}$).
- $\mathrm{EXP} \not\subseteq \mathrm{P}$ (time hierarchy theorem).

Remarks

- $\mathrm{EXP} \neq \mathrm{P/poly}$ (there are undecidable languages in $\mathrm{P/poly}$).
- $\mathrm{EXP} \not\subseteq \mathrm{P}$ (time hierarchy theorem).
- Space complexity version:

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

Remarks

- $\mathrm{EXP} \neq \mathrm{P/poly}$ (there are undecidable languages in $\mathrm{P/poly}$).
- $\mathrm{EXP} \not\subseteq \mathrm{P}$ (time hierarchy theorem).
- Space complexity version:

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

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

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 $a : \mathbf{N} \to \mathbf{N}$ a function, then $\mathcal{C}/a(n)$ is the set of languages A such that there exists $B \in \mathcal{C}$ and a function $c : \mathbf{N} \to \{0,1\}^*$ satisfying:
  - $\forall n, |c(n)| \leq a(n)$;
  - $\forall x \in \{0,1\}^*$, $x \in A \iff (x, c(|x|)) \in B$.

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 $a : \mathbf{N} \to \mathbf{N}$ a function, then $\mathcal{C}/a(n)$ is the set of languages A such that there exists $B \in \mathcal{C}$ and a function $c : \mathbf{N} \to \{0,1\}^*$ 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 (continued)

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

Advices (continued)

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

Advices (continued)

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

Advices (continued)

- $\mathrm{P}/2^n = \mathcal{P}(\{0,1\}^*)$.
- Even $\mathrm{P}/1$ contains undecidable languages...
- $\mathrm{P}/\mathrm{poly} = \cup_{k \geq 0} \mathrm{P}/n^k$ (polynomial-size advice).
- P/poly: conversion advice $\longleftrightarrow$ boolean circuit.

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: $\mathrm{NEXP}^{\mathrm{NP}} \not\subset \mathrm{P/poly}$;
- Schnöning 1985: $\mathrm{EXPSPACE} \not\subset \mathrm{P/poly}$.

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

- Simple diagonalization fails (too many circuits).
- Kannan 1982: $\mathrm{NEXP}^{\mathrm{NP}} \not\subset \mathrm{P/poly}$;
- Schnöning 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: $\mathrm{NEXP}^{\mathrm{NP}} \not\subset \mathrm{P/poly}$;
- Schnöning 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 (SI) $\Rightarrow \mathrm{EXP} \not\subset \mathrm{P/poly}$;
- Lee & Romashchenko 2004: (SI) $\Rightarrow \mathrm{EXP} \not\subseteq \mathrm{BPP}$ (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

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

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 $\mathrm{DTIME}(2^{n^{c_1}})/n^{c_2}$.

Proof.

We build A by input sizes and word by word. Let $t(n) = 2^{n^{1+c_1c_2}}$ 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 a wrong answer for x_1).

Advices of size n^c (continued)

Proposition

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 a 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 k, $\mathrm{PP} \not\subset \mathrm{DTIME}(n^k)/(n - \log n)$.

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

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

- Symmetry of information: $C(x, y) \simeq C(x) + C(y|x)$.
  $\leq$: easy direction $\geq$: hard direction.

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

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

- Symmetry of information: $C(x, y) \simeq C(x) + C(y|x)$.
  $\leq$: easy direction $\qquad$ $\geq$: hard direction.
- 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 (SI)

There exists a polynomial q such that for all p and all words x, y, z of size $|x| + |y| + |z| = n$:

$$C^{p(n)}(x, y|z) \geq C^{p(n)q(n)}(x|z) + C^{p(n)q(n)}(y|x, z) - O(\log n).$$

Symmetry of information

Hypothesis (SI)

There exists a polynomial q such that for all p and all words x, y, z of size $|x| + |y| + |z| = n$:

$$C^{p(n)}(x, y|z) \geq C^{p(n)q(n)}(x|z) + C^{p(n)q(n)}(y|x, z) - O(\log n).$$

Remark: stronger version than the usual one
$p(n)q(n)$ instead of $q(p(n))$.

Iterations of (SI)

Lemma

Suppose (SI) holds.
Let $u_1, \ldots, u_n$ *be words of size* s *and let* z *be another word. Let* $m = ns + |z|$*. Suppose there exists* k *such that for all* $j \leq n$*,*

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

Then $C^t(u_1, \ldots, u_n | z) \geq nk - (n-1)O(\log m)$*.*

Iterations of (SI)

Lemma

Suppose (SI) holds.
Let $u_1, \ldots, u_n$ *be words of size* s *and let* z *be another word. Let* $m = ns + |z|$. *Suppose there exists* k *such that for all* $j \leq n$,

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

Then $C^t(u_1, \ldots, u_n | z) \geq nk - (n-1)O(\log m)$.

Proof.

Show by induction on n that $\forall z$, if $(\forall j, C(u_j | u_1, \ldots, u_{j-1}, z) \geq k)$ then $C(u_1, \ldots, u_n | z) \geq nk - (n-1)O(\log m)$.

$$C^t(u_1, \ldots, u_n | z) \geq C^{tq(m)}(u_1, \ldots, u_{n/2} | z) +$$

$$C^{tq(m)}(u_{n/2+1}, \ldots, u_n | u_1, \ldots, u_{n/2}, z) - O(\log m). \quad \square$$

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 there exist infinitely many n and $1 \leq i \leq 2^n$ satisfying

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

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 there exist infinitely many n and $1 \leq i \leq 2^n$ satisfying

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

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 $a(n) + O(1)$. $\square$

Polynomial-size advices — the idea

- $\mathcal{U}$ will return χ^n instead of recognizing each word.

Polynomial-size advices — the idea

- $\mathcal{U}$ will return χ^n instead of recognizing each word.
- In EXP, impossible to diagonalize over all advices of polynomial size

Polynomial-size advices — the idea

- $\mathcal{U}$ will return χ^n instead of recognizing each word.

- 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

- $\mathcal{U}$ will return χ^n instead of recognizing each word.

- 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 back thanks to (SI).

Main result

Theorem

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

Main result

Theorem

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

Proof.

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

$x_1 \in A \Longleftrightarrow$ for at least half of the programs p of size $\leq n$, the first bit of $\mathcal{U}^{t(n)}(p)$ is 0.

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

Main result

Theorem

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

Proof.

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

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

(at least half of the programs give a 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 as before, discarding half of the remaining programs at each step:

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

Proof continued

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

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

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

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

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

Proof continued

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

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

Proof continued

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

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

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

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

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

Proof continued

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

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

Conclusion

- Good idea to study (SI): 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 (SI) (with time bound $q(p(n))$ instead of $q(n)p(n)$)?
- Hope: unconditionnal results by using CAMD (a version of Kolmogorov complexity based on the class AM).

Outline

1. Introduction and notations
2. Advices of size n^c
3. Symmetry of information
4. Polynomial-size advices