The Power of Diagonalization for Separating Complexity Classes

Sam Buss

Yandex (Moscow)
July 21, 2015

A fundamental open problem for computer science is to prove (or, disprove)

$$\mathrm{P} \neq \mathrm{NP},$$

Namely, does non-determinism help computation?

No less fundamental are questions about separating time classes from space classes; e.g.:

$$\mathrm{L} = \mathrm{P}? \qquad \text{and} \qquad \mathrm{P} = \mathrm{PS{\small PACE}}?$$

(L is log space; P is polynomial time; $\mathrm{PS{\small PACE}}$ is polynomial space.)

These latter problems are potentially easier to answer — in the negative —, since

$$\mathrm{L} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PS{\small PACE}}.$$

Q: Why conjecture $\mathrm{P} \neq \mathrm{NP}$?

A1: Because attempts at proving $\mathrm{P} = \mathrm{NP}$ using direct simulation have failed. (!)

A2: Because oracle results give barriers on using diagonalization to separate P and NP. [Baker-Gill-Solovay'1975]

Diagonalization has been useful mostly for proving the time and space hierarchies. For example:

Theorem: $\mathrm{L} \neq \mathrm{PSPACE}$ and $\mathrm{P} \neq \mathrm{DTIME}(2^n)$.
[Hartmanis-Lewis-Stearns'1965; Stearns-Hartmanis'1965].

$\mathrm{DTIME}(2^n)$ denotes $\mathrm{EXPTIME}$ (exponential time).

A barrier to stronger diagonalization is:

Oracle separation: [Baker-Gill-Solovay, 1975] There are oracles collapsing L and NP and oracles collapsing P and $\mathrm{PS{\scriptsize PACE}}$, so any proof of separation must not relativize.

This means that any proof of "$\mathrm{P} \neq \mathrm{NP}$" (or "$\mathrm{P} = \mathrm{NP}$") must use techniques that do not relativize.

This talk will concentrate, however, on the positive aspects of diagonalization, and how diagonalization can be surprisingly strong.

Remark: Other barriers to separating complexity classes include *Natural Proofs* [Razborov-Rudich, 1997], and *Algebrization* [Aaronson-Wigderson, 2008].

This talk: Using diagonalization for:

- Space hierarchy.
- Time hierarchy.
- Nondeterministic time hierarchy.
- Alternation trading proofs, and lower bounds for satisfiability.

Hierarchy of complexity classes:

$$\mathrm{L} \subseteq \mathrm{P} \subseteq \mathrm{NP} \subseteq \mathrm{PSPACE} \subseteq \mathrm{EXPTIME}.$$

Space hierarchy gives: $\mathrm{L} \neq \mathrm{PSPACE}$.
Time hierarchy gives: $\mathrm{P} \neq \mathrm{EXPTIME}$.
No other separations for these classes are known.

Classical uses of self-reference:

I. Gödel incompleteness:

"I am not provable".

II. Halting Problem is undecidable [Turing]:
If recursive enumerable is same as recursive, form a Turing machine M so that

"M halts iff M does not halt".

Classical use of diagonalization:

Diagonalization
- Underlies the use of self-reference.
- Is easier to work with.

For example: To prove not all recursive enumerable sets are recursive: Suppose this fails, and form a recursive predicate $P(i)$ by

$$P(i) \;\Leftrightarrow\; M_i(i) \text{ rejects}$$

M_i is the i-th Turing machine.

This argument uses a universal Turing machine.

Space Hierarchy

Theorem (Hartmanis-Lewis-Stearns'65)

Suppose $s(n) = o(t(n))$. *Then* $\textsc{Space}(s) \neq \textsc{Space}(t)$.

Computational Model:
Turing machines with k tapes, $k \geq 1$, and finite alphabet Γ.
Inputs: Binary strings $x \in \{0,1\}^*$.
Outputs: "Yes" / "No" ("Accept" / "Reject").
Runtimes are stated as a function of the *length* $n = |x|$ of the input string x.

Space is the total number of tape squares (memory) used by the computation. – Does not count size of the input.

Constant factors of speed-up can be achieved with large alphabets, so time/space bounds always use "Big-O" or "little-o" notation.

We assume all space/time bounds are well-behaved (space- and time-constructible).

Proof sketch for space hierarchy theorem:

Fact: There is a 1-tape **universal Turing machine** U^t so that,
- for any Turing machine M_e using space s, there is $c_e > 0$, s.t.
- $U^t(\langle e, x\rangle)$ uses space $c_e \cdot s$ and outputs $M_e(x)$
— unless $c_e \cdot s > t$.
- U^t aborts if simulating M_e requires space $> t$.

Define the Turing machine N so that $N(\langle e, x\rangle)$ runs $U^t(e, \langle e, x\rangle)$ and outputs the *opposite* answer.

Thus $N \in \textsc{Space}(t)$. But for all $M_e \in \textsc{Space}(s)$ and all sufficiently large x,

$$N(\langle e, x\rangle) \neq U^t(e, \langle e, x\rangle) = M_e(\langle e, x\rangle)$$

So $N \notin \textsc{Space}(s)$. **qed**

Theorem (Hartmanis-Stearns'65)

Let $s(n) \log s(n) = o(t(n))$. Then $\text{TIME}(s) \neq \text{TIME}(t)$.

Proof idea:

Fact: [H-S'65] There is a 2-tape universal Turing machine V^t so that,

- for any Turing machine M_e using time s, there is $c_e > 0$, s.t.
- $V^t(\langle e, x \rangle)$ uses time $c_e \cdot s \cdot \log s$ and outputs $M_e(x)$
  — unless $c_e \cdot s \cdot \log s > t$.
- V^t aborts if simulating M_e requires time $> t$.

Remainder of the proof is similar to before.

Nondeterministic Turing machines

Nondeterministic Turing machines have the ability to "guess". If any guess leads to acceptance, then the Turing machine is said to *accept*.

Formally: A nondeterministic Turing machine has multiple possible moves allowed by its transition function. A configuration is *accepting* iff it is in an accepting state or at least one legal move leads to an accepting configuration.

Satisfiability (SAT) is the canonical NP-complete problem. It is accepted by a nondeterministic, polynomial time Turing machine: the machine guesses and verifies a truth assignment.

Nondeterministic time hierarchy

[Cook'72; Seiferas-Fisher-Meyer'78; Žák'83; Santhanam-Fortnow'11]

Theorem (S-F-M'78; Nondeterministic time hierarchy)

Suppose $s(n+1) = o(t(n))$. *Then* $\mathrm{NTIME}(s) \neq \mathrm{NTIME}(t)$.

To start the proof sketch:

Fact: There is a 2-tape universal non-deterministic Turing machine U^t so that,

- for any nondeterministic M_e using time s, there is $c_e > 0$, s.t.
- $U^t(\langle e, x\rangle)$ uses time $c_e \cdot s$, and accepts iff $M_e(x)$ accepts — unless $c_e \cdot s > t$.
- U^t rejects on paths that use time $> t$

The problem with the previous proof is that with non-determinism, there is no way to output an "opposite" answer, negating the answer takes us from nondeterminism (existential choices) to co-nondeterministic (universal choices).

To avoid this [Žák'83]:

Let $T_e(n)$:= deterministic time to compute $M_e(x)$, $|x| = n$.
That is, $T_e(n) \leq d^{s(n)}$ for some $d > 0$.

Define $N(\langle e, x0^i \rangle)$ to equal (for $|x|$ sufficiently large)

- $U^t(e, \langle e, x0^{i+1} \rangle) = M_e(\langle e, x0^{i+1} \rangle)$, if $t(n) < T_e(|\langle e, x, \rangle|)$.
- $\neg M_e(\langle e, x \rangle)$ otherwise.

The first case is non-deterministic, the second is deterministic.

$N(\langle e, x0^i \rangle) = M_e(\langle e, x0^i \rangle)$ cannot hold for all i.

Thus $N \in \mathrm{NTIME}(t) \setminus \mathrm{NTIME}(s)$. **qed**

Nondeterministic space hierarchy

Theorem

Suppose $s(n) = o(t(n))$*. Then* $\mathrm{NS{\scriptsize PACE}}(s) \neq \mathrm{NS{\scriptsize PACE}}(t)$*.*

The proof is very similar to the proof of the Hartmanis-Lewis-Stearns space hierarchy. However, to negate the output of $U^t(e, \langle e, x\rangle)$, the proof uses the fact that NSPACE is closed under complement [Immerman'87; Szelepcsényi'87].

Alternation-trading proofs

The rest of the talk discusses upper and lower bounds on what separations can be obtained with alternation-trading proofs.

Alternation-trading proofs involve iterating the restricted space methods of Nepomnjasci [1970] together with simulations. This is essentially

a sophisticated version of diagonalization.

The best alternation-trading results obtained so-far state that $\textsc{Sat}$ is not computable in simultaneous time n^c and space n^ϵ for certain values of $c > 1$ and of $\epsilon > 0$.

E.g., alternation-trading proofs give partial results towards separating logspace (L) and NP.

Satisfiability

Definition (Satisfiability – Sat)

An instance of satisfiability is a set of clauses.
Each clause is a set of literals.
A *literal* is a negated or nonnegated propositional variable.
Satisfiability (Sat) is the problem of deciding if there is a truth assignment that sets at least one literal true in each clause.

Thm: Satisfiability is NP-complete.

Conjecture: Satisfiability is not polynomial time. ($\mathrm{P} \neq \mathrm{NP}$.)

Why is Satisfiability important?

1. Satisfiability is NP-complete.

2. Many other NP-complete problems are many-reducible to SAT in quasilinear time, that is, time $n \cdot (\log n)^{O(1)}$.

3. For a given non-deterministic machine M, the question of whether $M(x)$ accepts in n steps is reducible to SAT in quasilinear time. [Sharpened Cook-Levin theorem about the NP-completeness of SAT].

Thus SAT is a "canonical" and natural non-deterministic time problem. Lower bounds on algorithms for SAT imply the same lower bounds for many other NP-complete problems.

We now use the Random Access Memory (RAM) model for computation. This gives a very robust notion of linear time computation (the classes $\mathrm{DTIME}(n)$ and $\mathrm{NTIME}(n)$).
"DTIME" / "NTIME" = Deterministic/Nondeterministic time.

Sharpened Cook-Levin Theorem:

Theorem (Schnorr'78; Pippenger-Fischer'79; Robson'79,'91; Cook'88)

There is a $c > 0$ so that, for any language $L \in \mathrm{NTIME}(T(n))$, there is a quasi-linear time, many-one reduction from L to

$$\text{instances of } \mathrm{SAT} \text{ of size } T(n)(\log T(n))^c.$$

In fact, the symbols of the instances of SAT are computable in polylogarithmic time $(\log T(n))^c$.

We now use the Random Access Memory (RAM) model for computation. This gives a very robust notion of linear time computation (the classes $\mathrm{DT{\scriptsize IME}}(n)$ and $\mathrm{NT{\scriptsize IME}}(n)$).
"DTIME"/"NTIME" = Deterministic/Nondeterministic time.

Sharpened Cook-Levin Theorem:

Theorem (Schnorr'78; Pippenger-Fischer'79; Robson'79,'91; Cook'88)

There is a $c > 0$ so that, for any language $L \in \mathrm{NTIME}(T(n))$, there is a quasi-linear time, many-one reduction from L to

$$\textit{instances of } \mathrm{SAT} \textit{ of size } T(n)^{1+o(1)}.$$

In fact, the symbols of the instances of SAT are computable in polylogarithmic time $(\log T(n))^c$.

The "Slowdown" Theorem.

Corollary (Slowdown Theorem)

If $\text{SAT} \in \text{DTIME}(n^c)$*, then* $\text{NTIME}(n^d) \subset \text{DTIME}(n^{c \cdot d + o(1)})$.

The factor $n^{o(1)}$ hides polylogarithmic factors.

Definition

Let $c \geq 1$. $\mathrm{DTS}(n^c)$ is the class of problems solvable in simultaneous deterministic time $n^{c+o(1)}$ and space $n^{o(1)}$.

For instance, Logspace restricted to time n^c.

A series of results by Kannan [1984], Fortnow [1997], Lipton-Viglas, van Melkebeek, Williams, and others gives:

Theorem (Williams, 2007)

Let $c < 2\cos(\pi/7) \approx 1.8019$. *Then* $\textsc{Sat} \notin \mathrm{DTS}(n^c)$.

We also have:

Theorem (B - Williams'12)

The exponent $c = 2\cos(\pi/7)$ *is the best that can be obtained with present-day techniques.*

Nepomnjasci's method

Definition

$$^{b}(\exists n^{c})^{d}\mathrm{DTS}(n^{e})$$

denotes the class of problems taking inputs of length $n^{b+o(1)}$, existentially choosing $n^{c+o(1)}$ bits, keeping in memory a total of $n^{d+o(1)}$ bits (using time $n^{\max\{c,d\}+o(1)}$) which are passed to a deterministic procedure that uses time $n^{e+o(1)}$ and space $n^{o(1)}$.

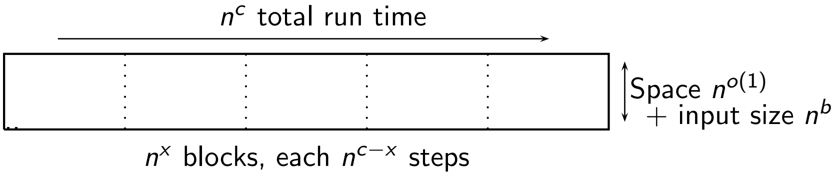
Speedup Theorem (by method of [Nepomnjasci'1970]

$^{b}\mathrm{DTS}(n^{c}) \subseteq {}^{b}(\exists n^{x})^{\max\{b,x\}}(\forall n^{0})^{b}\mathrm{DTS}(n^{c-x})$.

Proof next page....

${}^{b}\mathrm{DTS}(n^c) \subseteq {}^{b}(\exists n^x)^x(\forall n^0)^b\mathrm{DTS}(n^{c-x})$, for $x \geq b$

Proof idea: Split the n^c time computation into n^x many blocks. Existentially guess the memory contents (apart from the input) at each block boundary (**using $\mathbf{n^x \cdot n^{o(1)} = n^{x+o(1)}}$ many bits**), then universally choose one block to verify its correctness (**using $\mathbf{O(\log n) = n^{o(1)}}$ universal binary choices**), and simulate that block's computation (**in $\mathbf{n^{c-x}}$ time**).



Alternation trading proofs [Williams]

An *alternation trading proof* is a proof that $\text{SAT} \notin \text{DTS}(n^c)$, for some fixed $c \geq 1$. It is a proof by contradiction, based on deducing

$$^1\text{DTS}(n^a) \subseteq {}^1\text{DTS}(n^b)$$

for some $a > b$, from the assumption that $\text{SAT} \in \text{DTS}(n^c)$.

The lines of an alternation trading proof are of the form

$$^1(\exists n^{a_1})^{b_2}(\forall n^{a_2})^{b_3} \cdots {}^{b_k}(Qn^{a_k})^{b_{k+1}}\text{DTS}(n^{a_{k+1}}).$$

There are two kinds of inferences: "speedup" inferences that add quntifiers and reduce run time (based on Nepomnjascii) and "slowdown" inferences that remove a quantifier and increase run time (based on the S-P-F-R-C theorem)....

The rules of inferences for alternation trading proofs are:

Initial speedup: $(x \leq a)$

$$^1\mathrm{DTS}(n^a) \subseteq {}^1(\exists n^x)^{\max\{x,1\}}(\forall n^0)^1\mathrm{DTS}(n^{a-x}),$$

Speedup: $(0 < x \leq a_{k+1})$

$$\cdots {}^{b_k}(\exists n^{a_k})^{b_{k+1}}\mathrm{DTS}(n^{a_{k+1}}) \subseteq \cdots {}^{b_k}(\exists n^{\max\{x,a_k\}})^{\max\{x,b_{k+1}\}}(\forall n^0)^{b_{k+1}}\mathrm{DTS}(n^{a_{k+1}-x}),$$

Slowdown: (Under assumption that $\mathrm{SAT} \in \mathrm{DTS}(n^c)$)

$$\cdots {}^{b_k}(\exists n^{a_k})^{b_{k+1}}\mathrm{DTS}(n^{a_{k+1}}) \subseteq \cdots {}^{b_k}\mathrm{DTS}(n^{\max\{cb_k,ca_k,cb_{k+1},ca_{k+1}\}}).$$

and the dual rules.

Example: alternation trading proof.

Let $1 < c < \sqrt{2}$. Then, if $\mathrm{SAT} \in \mathrm{DTS}(n^c)$,

$$\begin{aligned} \mathrm{DTS}(n^2) &\subseteq (\exists n^1)^1 (\forall n^0)^1 \mathrm{DTS}(n^1) \\ &\subseteq (\exists n^1)^1 \mathrm{DTS}(n^c) \\ &\subseteq \mathrm{DTS}(n^{c^2}). \end{aligned}$$

which is a contradiction. Proof uses a speedup-slowdown-slowdown pattern, also denoted **100**.

This proves:

Theorem (Lipton-Viglas, 1999)

$\mathrm{SAT} \notin \mathrm{DTS}(n^{\sqrt{2}})$.

Better results can be found with more alternations.

Theorem (Fortnow, van Melkebeek, et. al)

$\textsc{Sat} \notin \mathrm{DTS}(n^c)$, *where* $c < \phi \approx 1.618$, *the golden ratio.*

The optimal refutation with seven inferences derives:

Theorem (Williams)

$\textsc{Sat} \notin \mathrm{DTS}(n^{1.6})$.

This proof uses the pattern of inferences: **1100100**, where "**1**" denotes a speedup and "**0**" denotes a slowdown.

Theorem (Williams)

Let $c < 2\cos(\pi/7) \approx 1.801$*. Then* $\textsc{Sat} \notin \mathrm{DTS}(n^c)$.

This used proofs of the following $\mathbf{1}/\mathbf{0}$ patterns:

$$\mathbf{1}^n(\mathbf{10})^*(\mathbf{0}(\mathbf{10})^*)^n.$$

Based on using Maple to (unsuccessfully) search for better refutations, these were conjectured by Williams to be the best possible refutations.

Theorem (Buss-Williams'12)

There are alternation trading proofs of $\mathrm{SAT} \notin \mathrm{DTS}(n^c)$ *for exactly the values* $c < 2\cos(\pi/7)$.

Remark: If $\mathrm{SAT} \notin \mathrm{DTS}(n^c)$ for all $c > 1$, then $\mathrm{L} \neq \mathrm{NP}$, something thought to be hard to prove.

So this theorem implies some kind of limit on diagonalization for proving separations towards:

"L versus NP?"

... but only under current proof methods.

Time-Space Tradeoff Lower Bounds

Definition

$\mathrm{DTISP}(n^c, n^\epsilon)$ is the class of problems decidable in deterministic time $n^{c+o(1)}$ and space $n^{\epsilon+o(1)}$.

The notion of alternation trading proofs can be expanded to give proofs that $\textsc{Sat} \notin \mathrm{DTISP}(n^c, n^\epsilon)$ for various values $1 \le c < 2\cos(\pi/7)$ and $0 < \epsilon < 1$.

This is done by giving alteration trading proofs of

$$\mathrm{DTISP}(n^{\alpha c}, n^{\alpha\epsilon}) \subseteq \mathrm{DTISP}(n^{\beta c}, n^{\beta\epsilon})$$

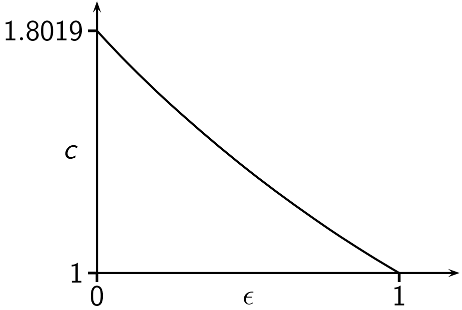
for some $\alpha > \beta > 0$.

Using computer-based search (C++), aided by theorems about pruning the search for alternation trading proofs:

Theorem (B - Williams'12)

The following pairs are the optimal values c and ϵ for which there are alternating trading proofs that $\text{SAT} \notin \text{DTISP}(n^c, n^\epsilon)$.

ϵ	c
0.001	1.80083
0.01	1.79092
0.1	1.69618
0.25	1.55242
0.5	1.34070
0.75	1.15765
0.9	1.06011
0.99	1.00583
0.999	1.00058



These values for c and ϵ are better than prior known lower bounds.

Open problems

- Find a closed form solution for the optimal $\mathrm{DTISP}(n^c, n^\epsilon)$ proofs. Even, find a simple characterization of how to construct the optimal proofs without resorting to a brute-force (pruned) search.
- There are many other flavors of alternation trading proofs, for instance for nondeterministic algorithms for tautologies. One could try giving proofs that the known alternation trading proofs are optimal.
- Most interesting: Try to find *new* principles that go beyond the presently known speedup and slowdown inferences, to give improved lower bound proofs.

Thank you!