Alternation-Trading Proofs, Linear Programming, and Lower Bounds

Ryan Williams[*]
Institute for Advanced Study

Abstract

A fertile area of recent research has demonstrated concrete polynomial time lower bounds for solving natural hard problems on restricted computational models. Among these problems are Satisfiability, Vertex Cover, Hamilton Path, MOD_6-SAT, Majority-of-Majority-SAT, and Tautologies, to name a few. The proofs of these lower bounds follow a certain proof-by-contradiction strategy, which we call "resource-trading" or "alternation-trading." An important open problem is to determine how powerful such proofs can possibly be.

We propose a methodology for studying these proofs that makes them amenable to both formal analysis and automated theorem proving. Formalizing the framework, we prove that the search for better lower bounds can often be turned into a problem of solving a large series of linear programming instances. We implement a small-scale theorem prover and report surprising results, which allow us to extract new human-readable time lower bounds for several problems. We also use the framework to prove concrete limitations on the current techniques.

[*]This material is based on work supported in part by NSF grant CCR-0122581 at CMU, and NSF grant CCF-0832797 at Princeton University/IAS. Any opinions, findings and conclusions or recommendations expressed are those of the author.
Email: `ryanw@ias.edu`.

1 Introduction

This work is concerned with proving new limitations on computers by exploiting their capabilities. Many known lower bounds for natural problems follow a pattern that we call a *resource-trading scheme*. Informally speaking, the scheme uses four basic steps:

(1) Assume a hard problem Π can be solved in n^c time with resources R. (Let's abbreviate the class of such problems as $R[n^c]$.) We wish to obtain a contradiction. For example, $R[n^c]$ may be $\mathsf{DTISP}[n^c, \text{poly}(\log n)]$, the set of problems solvable in n^c time and $\text{poly}(\log n)$ space, and Π may be satisfiability (SAT).

(2) Prove a *Speedup Lemma* that "trades time for resources", where $R[t]$ is shown to be in a class $S[o(t)]$, for a more powerful resource S and polynomials t. For example, $S[t]$ may be the class of problems solvable by alternating machines in time t. Nepomnjascii [Nep70] showed that $\text{poly}(\log n)$ space algorithms running in n^k time can be simulated by a Σ_k machine (using k alternations) in $\tilde{O}(n)$ time.

(3) Prove a *Slowdown Lemma* that "trades resources for time", where $S[t]$ is shown to be in $R[t^d]$, for small $d \geq 1$. This typically uses the assumption that $\Pi \in R[n^c]$. For example, if SAT has an n^c time, $\text{poly}(\log n)$ space algorithm, then (by a strong form of the Cook-Levin theorem) it follows that $\mathsf{NTIME}[t]$ has $t^{c+o(1)}$ time, $\text{poly}(\log t)$ space algorithms, and consequently $\Sigma_k\mathsf{TIME}[t]$ has $t^{c^k+o(1)}$ time, $\text{poly}(\log t)$ space algorithms.

(4) Combine (2) and (3) to show $\mathcal{C}[t] \subseteq \mathcal{C}[t^{1-\varepsilon}]$, for some $\varepsilon > 0$ and complexity class $\mathcal{C}[t]$, implying a contradiction with a hierarchy theorem for $\mathcal{C}$. For example, if SAT has an n^c time, $\text{poly}(\log n)$ space algorithm, then $\Sigma_2\mathsf{TIME}[t] \subseteq \mathsf{DTISP}[t^{c^2+o(1)}, \text{poly}(\log t)] \subseteq \Pi_2\mathsf{TIME}[t^{c^2/2}]$, where the first inclusion holds by (3) and the second holds by (2). This contradicts the alternating time hierarchy if $c^2 < 2$. The above is the $n^{\sqrt{2}-\varepsilon}$ lower bound of Lipton and Viglas [LV99].

This scheme has been applied in many settings, dating back to the 70's. A partial list includes:

Time Versus Space. Hopcroft, Paul, and Valiant [HPV77] proved that $\mathsf{SPACE}[n] \not\subseteq \mathsf{DTIME}[o(n \log n)]$ for multitape Turing machines, by proving the "speedup lemma" that $\mathsf{DTIME}[t] \subseteq \mathsf{SPACE}[t/\log t]$ and invoking diagonalization. (Their result was also extended to general models [PR81, HLMW86].)

Determinism vs Nondeterminism for Multitape TMs. A celebrated result of Paul-Pippenger-Szemeredi-Trotter [PPST83] is that $\mathsf{NTIME}[n] \neq \mathsf{DTIME}[n]$ for multitape Turing machines. The key component in the proof is the "speedup lemma" that $\mathsf{DTIME}[t] \subseteq \Sigma_4\mathsf{TIME}[t/\log^* t]$ in the multitape setting.

Deterministic and Nondeterministic Space-Bounded Algorithms. Our model is a random access machine using small space ($n/(\log n)^c$, $n^{1-\varepsilon}$, and $n^{o(1)}$ are typical values). The time lower bounds are for traditional NP-complete problems and problems higher in the polynomial hierarchy [Kan84, For97, LV99, FvM00, FLvMV05, Wil06, Wil08]. The best known deterministic time lower bound for solving SAT with $n^{o(1)}$ space algorithms is $n^{2\cos(\pi/7)-o(1)} \geq n^{1.801}$ [Wil08]. The bound also holds for the counting problem MODm-SAT where m is a composite that is not a prime power. For nondeterministic algorithms using $n^{o(1)}$ space, the best known time lower bound known for natural $\mathsf{co\text{-}NP}$ problems (such as TAUTOLOGY) has been $n^{\sqrt{2}-o(1)}$, by Fortnow and Van Melkebeek [FvM00].

Probabilistic and Quantum Space-Bounded Algorithms. Allender *et al.* [AKRRV01] showed that Maj-Maj-SAT requires $n^{1+\Omega(1)}$ time to solve on unbounded error machines that use $n^{1-\varepsilon}$ space, for $\varepsilon > 0$. Diehl and Van Melkebeek [DvM06] proved that for $k \geq 2$, k-QBF requires $\Omega(n^{k-o(1)})$ time with randomized two-sided error algorithms using $n^{o(1)}$ space. Viola [Vio07] has shown that 3-QBF requires $n^{1+\Omega(1)}$ time on Turing machines with a random access input tape and two-way read-write access to a random bit tape. Van Melkebeek and Watson [vMW07, vM07] have shown how to adapt the result of Adleman *et al.* [ADH97] that $\mathsf{BQP} \subseteq \mathsf{PP}$ to extend Allender *et al.* to a time lower bound for solving Maj-Maj-SAT with quantum algorithms.

General Multidimensional TMs. This model has read-only random access to its input, an $n^{o(1)}$ read-write store, and read-write access to a d-dimensional tape for a fixed $d \geq 1$. This model generalizes several others, and is the most powerful (and physically realistic) model known where we still know non-trivial time lower bounds for SAT. Multidimensional TMs have been studied for many years; for instance, [Lou80, PR81, Gri82, Kan83, MS87, LL90, vMR05, Wil06] proved lower bounds for solving problems in this model, and the best bound for SAT is essentially $\Omega(n^{\sqrt{(d+2)/(d+1)}})$ time in the d-dimensional case.

The lower bound proofs of the above type have been typically *ad hoc*, making it hard to build intuition about them. One gets a sense that the space of all possible proofs might be difficult to systematically study.

1.1 Main Results

We introduce a methodology for reasoning about resource-trading proofs that is also practically implementable for finding short proofs. We argue that for almost all known resource-trading lower bounds, the proofs can be reformulated in a way that the search for new lower bounds becomes a feasible problem that computers can help attack.[1] Informally, the "hard work" in proofs can often be replaced by a series of linear programming problems. Furthermore the framework allows us to prove limitations on what can be proved. These limitations are important since some components of these proofs do not relativize in some sense (cf. Appendix A).

In this paper, this approach is applied in several scenarios. In all cases, the resource being "traded" is alternations, so for the purposes of this work we call the proofs *alternation-trading*.

Deterministic Time-Space Lower Bounds for SAT and Beyond. Aided by results of a computer program, we show that any algorithm solving SAT in $t(n)$ time and $s(n)$ space must have $t \cdot s \geq \Omega(n^{2\cos(\pi/7)-o(1)})$. Previously, the best known result was $t \cdot s \geq \Omega(n^{1.573})$ [FLvMV05]. It has been conjectured that the current framework sufficed to prove a $n^{2-o(1)}$ time lower bound for SAT, against algorithms using $n^{o(1)}$ space.[2] We present strong evidence (from computer search) that the best known $n^{2\cos(\pi/7)-o(1)}$ lower bound [Wil08] is already optimal for the framework. We show that it will be impossible to obtain n^2 with the framework, formalizing a conjecture of [FLvMV05].[3] We also prove lower bounds on QBF$_k$ (quantified Boolean formulas with at most k quantifier blocks), showing it requires $\Omega(n^{k+1-\delta_k})$ time for $n^{o(1)}$ space algorithms, where $\delta_k < 0.2$ and $\lim_{k\to\infty} \delta_k = 0$.[4] These results appear also optimal for the current tools.

Nondeterministic Time-Space Lower Bounds for Tautologies. Adapting the methodology for this problem, a computer program found a very short proof improving upon Fortnow and Van Melkebeek's 8-year old bound. Longer proofs suggested an interesting pattern. Formalizing it, we prove (on paper) an $n^{4^{1/3}-o(1)} \geq n^{1.587}$ time lower bound, and experiments suggest optimality for the framework. After learning of our short proof, Diehl and Van Melkebeek have proven a similar result [DvMW07]. We also show it is not possible to obtain an n^{ϕ} time lower bound, where $\phi = 1.618\ldots$ is the golden ratio. This is surprising since we have known for some time [FvM00] that a n^{ϕ} lower bound is provable for *deterministic* algorithms.

Lower Bounds for Multidimensional TMs. Here the method uncovers highly regular behavior in the best lower bound proofs, regardless of the dimension of the tape. Studying the output of a theorem prover, we extract an $\Omega(n^{r_d})$ time lower bound for the d-dimensional case, where $r_d \geq 1$ is the root of a particular quintic

[1]We note that combinatorial arguments such as Santhanam's time-space lower bound for SAT on multitape Turing machines [San01] do not fall under the alternation-trading paradigm, but they are already known to have different limitations.

[2]I could not find an explicit reference for this conjecture, but I have received several referee reports in the past that state it. Also cf. [LV99] in FOCS'99.

[3]That is, we formalize the statement: "...some complexity theorists feel that improving the golden ratio exponent beyond 2 would require a breakthrough" in Section 8 of [FLvMV05].

[4]Note the QBF$_k$ results appeared in the author's PhD thesis in 2007 but have been unpublished to date.

$p_d(x)$ with coefficients depending on d. For example, $r_1 \approx 1.3009$, $r_2 \approx 1.1887$, and $r_3 \approx 1.1372$. Again, computer search suggests this is the best possible, and we prove that it is impossible to improve the bound for d-dimensional TMs to $n^{1+1/(d+1)}$ with the current tools.

The above lower bounds hold for other NP and $\mathsf{co\text{-}NP}$-hard problems as well, since the only property required is that every set in $\mathsf{NTIME}[n]$ (respectively, $\mathsf{coNTIME}[n]$) has sufficiently efficient reductions to the problem. Furthermore, we stress that this approach is not limited to the above scenarios, and can be applied to the league of problems discussed in Van Melkebeek's surveys [vM04, vM07]. This work promotes a new methodology for proving lower bounds, where prospective lower-bounders formalize their proof rules, write a program to test ideas and generate short proofs, then study the results and extrapolate new results.

1.2 Reduction to Linear Programming

The key to our formulation is that we separate the *discrete choices* in a lower bound proof from the *real-valued choices*. The discrete choices consist of the sequence of rules to apply in a proof, and which complexity class $\mathcal{C}[t]$ to use in the proof by contradiction. We give several simplifications that greatly reduce the number of necessary discrete choices, without loss of generality. Real-valued choices come from selecting t, as well as parameters arising from rule applications. We prove that once the discrete choices are made, the remaining real-valued problem can be expressed as an instance of linear programming. This makes it possible to search for new proofs via computer, and it also gives us a formal handle on the limitations of these proofs.

One cannot easily search over all possible proofs, as the number of discrete choices is still $\sim 2^n/n^{3/2}$ for proofs of n lines (proportional to the nth Catalan number). Nevertheless it is quite feasible to search over all $20+$ line proofs. These searches already reveal highly regular patterns, indicating that certain strategies will be most successful in proving lower bounds; in each case we study, the resulting strategies are different. Following the strategies, we establish new lower bound proofs. Finally, the patterns also suggest how to prove limitations on the proof system.

Important Note: *In the first 12 pages, we can only briefly describe the results and techniques. Please see the Appendices for background information and more details.*

2 Preliminaries

We assume familiarity with the basics of complexity, especially alternation [CKS81]. We use big-Ω notation in the infinitely often sense, so statements like "SAT is not in $O(n^c)$ time" are equivalent to "SAT requires $\Omega(n^c)$ time." All functions are assumed to be constructible within the appropriate bounds. Our default computational model is the random access machine, but particular variants do not affect the results. $\mathsf{DTISP}[t(n), s(n)]$ is the class of languages accepted by a RAM running in $t(n)$ time and $s(n)$ space, simultaneously. For convenience, we define $\mathsf{DTS}[t(n)] := \mathsf{DTISP}[t(n)^{1+o(1)}, n^{o(1)}]$ to avoid negligible $o(1)$ factors.

To properly formalize alternation-trading proofs, we introduce notation for alternating complexity classes which include *input constraints* between alternations. These constraints are critical for the formalism. Define $(\exists\ f(n))^b\mathcal{C}$ to be the class of languages recognized by a machine N that, on input x, writes a $f(n)^{1+o(1)}$ bit string y nondeterministically, copies at most $n^{b+o(1)}$ bits z of the tuple $\langle x, y\rangle$ deterministically (in $O(n^{b+o(1)})$ time), then feeds z as input to a machine from class $\mathcal{C}$. We refer to this behavior by saying that the *class $\mathcal{C}$ is constrained to n^b input*. Define $(\exists\ f(n))\mathcal{C} := (\exists\ f(n))^{\max\{1,(\log f(n))/(\log n)\}}\mathcal{C}$. That is, the default input length is assumed to be $O(f(n)^{1+o(1)} + n^{1+o(1)})$. The class $(\forall\ f(n))^b\mathcal{C}$ is defined similarly (with co-nondeterminism). We say that the existential and universal phases of an alternating computation are *quantifier blocks*, to reflect the notation. Hence a machine of the class $(Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}]$ with $Q_i \in$

$\{\exists, \forall\}$ means that the input to the computation starting at the ith quantifier block is of length $n^{b_i+o(1)}$ for all $i = 1, \ldots, k$, and the input to the DTS computation has length $n^{b_{k+1}+o(1)}$. (Of course, the first quantifier block always has an input of length n.) It is important to keep track of the input lengths to quantifier blocks, since several lower bounds rely on the fact that these inputs can be small in certain interesting cases.

For readers new to this area, we strongly encourage them to read Appendix A, *A Short Introduction to Time-Space Lower Bounds*.

3 Time-Space Lower Bounds for SAT

Our study begins with polynomial time-space lower bounds for $\mathsf{NTIME}[n]$ problems such as SAT. We shall describe the approach in some detail here; the other settings assume knowledge of this section. We begin with a formalization of the alternation-trading framework. Alternation-trading proofs apply a sequence of "speedup" and "slowdown" lemmas in some order, with the goal of reaching a contradiction by a time hierarchy theorem. We formalize alternation-trading proofs for DTS classes as follows:[5]

Definition 3.1 *Let $c > 1$. An* alternation-trading proof for c *is a list of complexity classes of the form:*

$$(Q_1\, n^{a_1})^{b_2}(Q_2\, n^{a_2}) \cdots^{b_k} (Q_k\, n^{a_k})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}], \tag{1}$$

where $k \geq 0$, $Q_i \in \{\exists, \forall\}$, $Q_i \neq Q_{i+1}$, $a_i > 0$, and $b_i \geq 1$, for all i. (When $k = 0$, the class is deterministic.) The items of the list are called lines of the proof. *Each line is obtained from the previous line by applying either a* speedup rule *or a* slowdown rule. *More precisely, if the ith line is*

$$(Q_1\, n^{a_1})^{b_2}(Q_2\, n^{a_2}) \cdots^{b_k} (Q_k\, n^{a_k})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}],$$

then the $(i+1)$st line has one of three possible forms:

1. (Speedup Rule 0) $(Q_k\, n^x)^{\max\{x,1\}}(Q_{k+1}\, n^0)^1\mathsf{DTS}[n^{a_{k+1}-x}]$, *when $k = 0$ and $x \in (0, a_{k+1})$.*[6]
2. (Speedup Rule 1) $(Q_1\, n^{a_1})^{b_2}(Q_2\, n^{a_2}) \cdots^{b_k} (Q_k\, n^{\max\{a_k,x\}})^{\max\{x,b_{k+1}\}}(Q_{k+1}\, n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}]$, *for $k > 0$ and any $x \in (0, a_{k+1})$.*
3. (Speedup Rule 2) $(Q_1\, n^{a_1})^{b_2} \cdots^{b_k} (Q_k\, n^{a_k})^{b_{k+1}}(Q_{k+1}\, n^x)^{\max\{x,b_{k+1}\}}(Q_{k+2}\, n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}]$, *for $k > 0$ and any $x \in (0, a_{k+1})$.*
4. (Slowdown Rule) $(Q_1\, n^{a_1})^{b_2}(Q_2\, n^{a_2}) \cdots^{b_{k-1}} (Q_{k-1}\, n^{a_{k-1}})^{b_k}\mathsf{DTS}[n^{c\cdot\max\{a_{k+1},a_k,b_k,b_{k+1}\}}]$, *for $k > 0$.*

An alternation-trading proof shows $(\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c] \implies A_1 \subseteq A_2)$ *if its first line is A_1 and its last line is A_2.*

The definition comes directly from the statements of the Speedup Lemma (Lemma A.1) and Slowdown Lemma (Lemma A.2) for space-bounded computations. (Note the n^0 in the Speedup Lemma corresponds to $\log n \leq n^{o(1)}$, which is negligible.) Speedup Rules 0, 1, and 2 can be easily verified to be syntactic formulations of the Speedup Lemma, where the DTS part of the sped-up computation only reads two guessed

[5]This formalization has *implicitly* appeared in prior work, but not to the degree that we investigate in this paper.

[6]Please note that the $(k+1)$th quantifier is n^0 in order to account for the $O(\log n)$ size of the quantifier.

configurations—so the input it reads is different from the input read by the innermost quantifier block. For instance, Speedup Rule 1 holds, since

$$\begin{aligned}
&(Q_1\, n^{a_1})^{b_2}(Q_2\, n^{a_2}) \cdots^{b_k} (Q_k\, n^{a_k})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}] \\
\subseteq\ &(Q_1\, n^{a_1})^{b_2}(Q_2\, n^{a_2}) \cdots^{b_k} (Q_k\, n^{a_k})^{b_{k+1}}(Q_k\, n^{x})^{\max\{b_{k+1},x\}}(Q_{k+1}\, n^{0})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}] \\
\subseteq\ &(Q_1\, n^{a_1})^{b_2}(Q_2\, n^{a_2}) \cdots^{b_k} (Q_k\, n^{\max\{a_k,x\}})^{\max\{b_{k+1},x\}}(Q_{k+1}\, n^{0})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}].
\end{aligned}$$

Rule 2 is akin to Rule 1, except that it uses opposite quantifiers in its invocation of the Speedup Lemma. The Slowdown Rule works analogously to Lemma A.2. It follows that alternation-trading proofs are sound.

Note Speedup Rules 0 and 2 add two quantifier blocks, Speedup Rule 1 adds only one quantifier, and all three rules introduce a parameter x. By considering "normal form" proofs (defined in the next section), we can show that Rule 2 can always be replaced by applications of Rule 1. For a proof, cf. Appendix C. For this reason we just refer to *the Speedup Rule*, depending on which of Rule 0 or Rule 1 applies.

A Normal Form. Define any class of the form in (1) to be *simple*. Define classes A_1 and A_2 to be *complementary* if A_1 is the class of complements of languages in A_2. Every known (model-independent) time-space lower bound for SAT shows "$\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ implies $A_1 \subseteq A_2$", for some complementary simple classes A_1 and A_2, contradicting a time hierarchy (cf. Theorem A.2). A similar claim holds for nondeterministic time-space lower bounds against tautologies (which prove $\mathsf{NTIME}[n] \subseteq \mathsf{coNTS}[n^c]$ implies $A_1 \subseteq A_2$), for d-dimensional machine lower bounds solving SAT (which prove $\mathsf{NTIME}[n] \subseteq \mathsf{DTIME}_d[n^c]$ implies $A_1 \subseteq A_2$), and for other problems.

We now introduce a *normal form* for alternation-trading proofs. We show that any lower bound provable with complementary simple classes can also be established with a normal form proof. This simplification greatly reduces the degrees of freedom in a proof, as we no longer have to worry about *which* complementary simple classes to choose for the contradiction.

Definition 3.2 *Let $c \geq 1$. An alternation-trading proof for c is in* normal form *if (a) the first and last lines are $\mathsf{DTS}[n^a]$ and $\mathsf{DTS}[n^{a'}]$ respectively, for some $a \geq a'$, and (b) no other lines are DTS classes.*

We show that a normal form proof for c implies that $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$.

Lemma 3.1 *Let $c \geq 1$. If there is an alternation-trading proof for c in normal form having at least two lines, then $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$.*

Theorem 3.1 *Let A_1 and A_2 be complementary. If there is an alternation-trading proof P for c that shows ($\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c] \implies A_1 \subseteq A_2$), then there is a normal form proof for c, of length at most that of P.*

Proofs of Lemma 3.1 and Theorem 3.1 can be found in Appendix D. The important consequence is that we only need to focus on normal form proofs in a proof search. For the remainder of this section, we assume that all alternation-trading proofs under discussion are in normal form.

Proof Annotations. Different lower bound proofs can result in quite different sequences of speedups and slowdowns. A *proof annotation* represents such a sequence.

Definition 3.3 *A* proof annotation *for an alternation-trading proof of ℓ lines is the $(\ell-1)$-bit vector A where for all $i = 1, \ldots, \ell-1$, $A[i] = 1$ (respectively, $A[i] = 0$) if the ith line applies a Speedup Rule (respectively, a Slowdown Rule).*

An $(\ell-1)$-bit proof annotation corresponds to a "strategy" for an ℓ-line proof. For a normal form alternation-trading proof with ℓ lines, it is not hard to show that its annotation A must have $A[1] = 1$, $A[\ell-2] = 0$, and

$A[\ell-1] = 0$. The number of possible proof annotations is closely related to the number of well-balanced strings over parentheses. Recall that the kth Catalan number is $C(k) = \frac{1}{k+1}\binom{2k}{k}$. A well-known fact states that the number of well-balanced strings of length $2k$ is $C(k)$.

Proposition 1 *Let $\ell > 3$ be even. The number of possible annotations for proofs of ℓ lines is $C(\ell/2 - 1)$.*

Hence the number of possible annotations for proofs of ℓ lines is $\Theta(2^{\ell}/\ell^{3/2})$. Note that an annotation *does not* determine a proof entirely, as there are other parameters. (The problem of determining optimal values for these parameters is tackled in the next section.) To illustrate the annotation concept, we give four examples.

- Lipton-Viglas' $n^{\sqrt{2}}$ lower bound [LV99] (from the Introduction) has the annotation $[1, 0, 0]$.
- The $n^{1.6004}$ bound of the *Short Introduction* (cf. Appendix A) corresponds to $[1, 1, 0, 0, 1, 0, 0]$.
- The n^{ϕ} bound of Fortnow-Van Melkebeek [FvM00] is an inductive proof, corresponding to an infinite sequence of annotations. In normal form, the sequence is: $[1, 0, 0], [1, 1, 0, 0, 0], [1, 1, 1, 0, 0, 0, 0], \ldots$
- The $n^{2\cos(\pi/7)}$ bound [Wil08] has two stages of induction. Let $A = 1, 0, 1, 0, \ldots, 1, 0, 0$, where the '$\ldots$' contain any number of repetitions. The sequence is $[A], [1, A, A], [1, 1, A, A, A], [1, 1, 1, A, A, A, A], \ldots$

  That is, the proof performs many speedups, then a sequence of many slowdown-speedup alternations, then two consecutive slowdowns, repeating this until all the quantifiers have been removed.

3.1 Translation To Linear Programming

Given a (normal form) proof annotation, how can we determine the best proof possible with it? The obstacles are (a) the runtimes of the first and last DTS classes of the proof are free parameters, and (b) each application of a Speedup Rule introduces a parameter x_i. We now show how to reduce an annotation A and $c > 1$ to a linear program that is feasible if and only if there is an alternation-trading proof of $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$ with annotation A. More precisely, the problem of setting parameters can be viewed as an arithmetic circuit evaluation where the circuit has $\max$ gates, addition gates, and input gates that multiply their input by c. Such circuits can be evaluated using a linear program (cf. [Der72]) that minimizes the sum of the gate values.

Let A be an annotation of $\ell - 1$ bits, and let m be the maximum number of quantifier blocks in a line of A; note m is easily computed in linear time. The target LP has variables $a_{i,j}$, $b_{i,j}$, and x_i, for all $i = 0, \ldots, \ell - 1$ and $j = 1, \ldots, m$. The variables $a_{i,j}$ represent the runtime exponent of the jth quantifier block in the class on the ith line, $b_{i,j}$ is the input exponent to the jth quantifier block of the class on the ith line, and for all lines i that use a Speedup Rule, x_i is the choice of x in the Speedup Rule. For example:

- If the kth line of a proof is $\mathsf{DTS}[n^a]$, the corresponding constraints are
$$a_{k,1} = a, \ b_{k,1} = 1, \ (\forall k > 0) \ a_{k,i} = b_{k,i} = 0.$$
- If the kth line of a proof is $(\exists\, n^{a'})^b \mathsf{DTS}[n^a]$, then the constraints are
$$a_{k,0} = a, \ b_{k,1} = b, \ a_{k,1} = a', \ b_{k,1} = 1, \ (\forall k > 1) \ a_{k,i} = b_{k,i} = 0.$$

The objective is to minimize $\sum_{i,j}(a_{i,j} + b_{i,j}) + \sum_i x_i$. The LP constraints depend on the lines of the annotation, as follows.

Initial Constraints. For the 0th and $(\ell - 1)$th lines we have $a_{0,1} \geq a_{\ell-1,1}$, as well as
$$a_{0,1} \geq 1, \ b_{0,1} = 1, \ (\forall\, k > 1) \ a_{0,k} = b_{0,k} = 0, \ \text{ and } \ a_{\ell,1} \geq 1, \ b_{\ell,1} = 1, \ (\forall k > 1) \ a_{\ell,k} = b_{\ell,k} = 0,$$

representing $\mathsf{DTS}[n^{a_{0,1}}]$ and $\mathsf{DTS}[n^{a_{\ell-1,0}}]$, respectively. The 1st line of a proof always applies Speedup Rule 1, having the form $(Q_1 n^x)^{\max\{x,1\}}(Q_2\ n^0)^1\mathsf{DTS}[n^{a-x}]$. So the constraints for the 1st line are:

$$a_{1,1} = a_{0,1} - x_1,\ b_{1,1} = 1,\ a_{1,2} = 0,\ b_{1,2} \geq x_1,\ b_{1,2} \geq 1,\ a_{1,3} = x_3,\ b_{1,3} = 1,$$
$$(\forall\ k:\ 4 \leq k \leq m)\ a_{1,k} = b_{1,k} = 0.$$

The below constraint sets simulate the Speedup and Slowdown Rules:

Speedup Rule Constraints. For the ith line where $i > 1$ and $A[i] = 1$, the constraints are

$$a_{i,1} \geq 1,\ a_{i,1} \geq a_{i-1,1} - x_i,\ b_{i,1} = b_{i-1,1},\ a_{i,2} = 0,\ b_{i,2} \geq x_i,\ b_{i,2} \geq b_{i-1,1},\ a_{i,3} \geq a_{i-1,2},$$
$$a_{i,3} \geq x_i,\ b_{i,3} \geq b_{i-1,2},\ (\forall\ k:\ 4 \leq k \leq m)\ a_{i,k} = a_{i-1,k-1}, b_{i,k} = b_{i-1,k-1}.$$

These constraints express that $\cdots\ ^{b_2}(Q_2\ n^{a_2})^{b_1}\mathsf{DTS}[n^{a_1}]$ in the $(i-1)$th line is replaced with

$$\cdots\ ^{b_2}(Q_2\ n^{\max\{a_2,x\}})^{\max\{x,b_1\}}(Q_1\ n^0)^{b_1}\mathsf{DTS}[n^{\max\{a_1-x,1\}}]$$

in the ith line, where Q_1 is opposite to Q_2.
Slowdown Rule Constraints. For the ith line where $A[i] = 0$, the constraints are

$$a_{i,1} \geq c \cdot a_{i-1,1},\ a_{i,1} \geq c \cdot a_{i-1,2},\ a_{i,1} \geq c \cdot b_{i-1,1},\ a_{i,1} \geq c \cdot b_{i-1,2},\ b_{i,1} = b_{i-1,2}$$
$$(\forall\ k:\ 2 \leq k \leq m-1)\ a_{i,k} = a_{i-1,k+1},\ b_{i,k} = b_{i-1,k+1},\ a_{i,m} = b_{i,m} = 0.$$

These express the replacement of $\cdots\ ^{b_2}(Q_1 n^{a_2})^{b_1}\mathsf{DTS}[n^{a_1}]$ in the $(i-1)$th line with

$$\cdots\ ^{b_2}\mathsf{DTS}[n^{c\cdot\max\{a_1,a_2,b_1,b_2\}}]$$

in the ith line.

This concludes the description of the linear program. To find the largest c that still yields a feasible LP, we can simply binary search for it. The following theorem summarizes the above discussion.

Theorem 3.2 *Given a proof annotation of n lines, the best possible lower bound proof following the annotation can be determined up to n digits of precision, in* poly(n) *time.*

Proof Search Results. Following the above formulation, we wrote proof search routines in Maple. Millions of proof annotations were tried, including all those of previous work, with no success beyond the $2\cos(\pi/7)$ exponent. The best lower bounds followed a highly regular pattern. For a 424 line annotation following the pattern, the optimal exponent was only in the interval $[1.80175, 1.8018)$. For more details, cf. Appendix E. One interesting sequence of annotations from the pattern is

$$1^k 11000(10)0(10)^2 0 \cdots (10)^k 0,$$

for $k \geq 2$. One can prove that this sequence cannot yield any lower bound better than $2\cos(\pi/7)$.

Theorem 3.3 *In the limit (as $k \to \infty$), the maximum lower bound provable with the sequence of annotations $1^k 11000(10)0(10)^2 0(10)^3 0 \cdots (10)^k 0$ for $k \geq 0$ is that SAT cannot be solved in $O(n^{2\cos(\pi/7)-o(1)})$ time and $n^{o(1)}$ space.*

A similar argument applies to all other annotations found by computer. We are led to:

Conjecture 3.1 *There is no alternation-trading proof that* $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$, *for any* $c > 2\cos(\pi/7)$.

Proving the above conjecture seems currently out of reach. We can give a partial result:

Theorem 3.4 *There is no alternation-trading proof that* $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^2]$.

A proof is in Appendix F. At a high level, the proof argues that any minimum length proof of a quadratic lower bound could be shortened, giving a contradiction.

Good News. Despite the bad news above, the theorem prover did give us enough insight to prove a new lower bound of $\Omega(n^{2\cos(\pi/7)-o(1)})$ on the time-space product of algorithms solving SAT. Also, these results have also been generalized to quantified Boolean formulas, leading to new lower bounds. For more, cf. Appendices B and M, respectively.

4 Nondeterministic Time-Space Lower Bounds for Tautologies

The problem of proving nondeterministic time-space lower bounds for $\mathsf{co\text{-}NP}$ has also been studied. Fortnow and Van Melkebeek [FvM00] proved that TAUTOLOGY requires $\Omega(n^{\sqrt{2}-o(1)})$ time on a nondeterministic $n^{o(1)}$ space RAM. However, since their initial result, no further improvements had been made. We show how to extend the approach of the previous section to this problem, and find that the best proof annotations look quite different. Here the approach turns out to be successful in finding new proofs.

4.1 The Framework and Linear Programming Translation

Similar to the class DTS, set $\mathsf{NTS}[n^a] := \mathsf{NTISP}[n^a, n^{o(1)}]$ and $\mathsf{coNTS}[n^a] := \mathsf{coNTISP}[n^a, n^{o(1)}]$ for brevity. As in the previous lower bound setting, there are Speedup and Slowdown rules that are applied in some way that contradicts a time hierarchy, although the rules are somewhat different here. In the following, let Q be a string of quantifier blocks, so $Q = (Q_1\, n^{a_1})^{b_2} \cdots (Q_{k-1}\, n^{a_{k-1}})$.

Lemma 4.1 *(Speedup) For* $b \geq 1$, $a \geq 1$, $x \geq 0$, *and* $s \geq 0$,

$$Q^b\mathsf{NTISP}[n^a, n^s] \subseteq Q^b(\exists\, n^{x+s})^{\max\{b,x+s\}}(\forall\, \log n)^{\max\{b,s\}}\mathsf{NTISP}[n^{a-x}, n^s].$$

In particular for $s = o(1)$ *we have* $\mathsf{NTS}[n^a] \subseteq (\exists\, n^x)^{\max\{1,x\}}(\forall\, \log n)^1\mathsf{NTS}[n^{a-x}]$.

Proof. The proof is analogous to Lemma A.1 (the Speedup Lemma for DTISP). □

Lemma 4.2 *(Slowdown) If* TAUTOLOGY *is in* $\mathsf{NTS}[n^c]$ *then*

1. $Q^b(\exists\, n^{a_k})^{b_{k+1}}\mathsf{NTS}[n^{a_{k+1}}] \subseteq Q^b\mathsf{coNTS}[n^{c\cdot\max\{a_k,a_{k+1},b_{k+1},b\}}]$,
2. $Q^b(\forall\, n^{a_k})^{b_{k+1}}\mathsf{coNTS}[n^{a_{k+1}}] \subseteq Q^b\mathsf{NTS}[n^{c\cdot\max\{a_k,a_{k+1},b_{k+1},b\}}]$,
3. $Q^b(\exists\, n^{a_k})^{b_{k+1}}\mathsf{coNTS}[n^{a_{k+1}}] \subseteq Q^b(\exists\, n^{a_k})^{b_{k+1}}\mathsf{NTS}[n^{c\cdot\max\{a_{k+1},b_{k+1}\}}]$, *and*
4. $Q^b(\forall\, n^{a_k})^{b_{k+1}}\mathsf{NTS}[n^{a_{k+1}}] \subseteq Q^b(\forall\, n^{a_k})^{b_{k+1}}\mathsf{coNTS}[n^{c\cdot\max\{a_{k+1},b_{k+1}\}}]$.

The proofs are omitted and are left to the interested reader. To obtain contradictions, one uses the alternating time hierarchy (Theorem A.2) just as in the deterministic case. The above lemmas immediately lead to a natural definition of *alternation-trading proof that* $\mathsf{coNTIME}[n] \subseteq \mathsf{NTS}[n^c] \implies A_1 \subseteq A_2$, for classes A_1 and A_2. Another way to yield a contradiction uses a result similar to Lemma 3.1, which showed that $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ implies $\mathsf{DTS}[n^a] \not\subseteq \mathsf{DTS}[n^{a'}]$ for $a > a'$.

Lemma 4.3 *If* $\mathsf{coNTIME}[n] \subseteq \mathsf{NTS}[n^c]$ *then* $\mathsf{NTS}[n^a] \not\subseteq \mathsf{coNTS}[n^{a'}]$ *for* $a > a'$.

This lemma can be used to motivate a definition of *normal form proof*, and prove that any alternation-trading proof can be converted into normal form.

Definition 4.1 *Let* $c \geq 1$. *An alternation-trading proof that* $(\mathsf{coNTIME}[n] \subseteq \mathsf{NTS}[n^c] \implies A_1 \subseteq A_2)$ *is in* normal form *if* (1) $A_1 = \mathsf{NTS}[n^a]$, $A_2 = \mathsf{coNTS}[n^{a'}]$, *for some* $a \geq a'$, *and* (2) *no other lines are* NTS *or* coNTS *classes.*

Example. If TAUTOLOGY $\in \mathsf{NTS}[n^c]$ then $\mathsf{coNTIME}[n] \subseteq \mathsf{NTS}[n^{c+o(1)}]$ by Theorem A.1, so $\mathsf{NTS}[n^2] \subseteq (\exists\, n)^1(\forall\ \log n)^1\mathsf{NTS}[n]$ by Lemma 4.1 (NTISP Speedup). Applying Lemma 4.2 (NTISP Slowdown) thrice, $(\exists\, n)^1(\forall\ \log n)^1\mathsf{NTS}[n] \subseteq (\exists\, n)^1(\forall\ \log n)^1\mathsf{coNTS}[n^c] \subseteq (\exists\, n)^1\mathsf{NTS}[n^{c^2}] \subseteq \mathsf{coNTS}[n^{c^3}]$. When $c < \sqrt[3]{2} \approx 1.25$, $\mathsf{NTS}[n^a] \subseteq \mathsf{coNTS}[n^{a'}]$ for some $a > a'$, which contradicts Lemma 4.3.

Lemma 4.4 *Let* $c \geq 1$. *If there is an alternation-trading proof that* $(\mathsf{coNTIME}[n] \subseteq \mathsf{NTS}[n^c] \implies A_1 \subseteq A_2)$ *in normal form, and the proof has at least two lines, then* $\mathsf{coNTIME}[n] \not\subseteq \mathsf{NTS}[n^c]$.

Theorem 4.1 *Let* A_1 *and* A_2 *be simple and complementary. If there is an alternation-trading proof* P *that* $(\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c] \implies A_1 \subseteq A_2)$, *then there is a normal form proof for* c, *of length at most that of* P.

One can also define proof annotations for this setting. The vectors corresponding to valid annotations change due to differences in the rules. For example, note $[1, 0, 0, 0]$, $[1, 1, 0, 0, 0, 0, 0]$, and $[1, 0, 1, 0, 0, 0, 0]$ are valid annotations for this setting, the first being the annotation for the above example. From the Speedup and Slowdown Lemmas given above, observe that the operations on exponents are again $\max$, $+$, and multiplication by c. Hence the translation of annotations to linear programming follows a similar strategy as before: we define variables $a_{i,j}$, $b_{i,j}$, x_i for all lines i and possible quantifier blocks j, replace components of the form $\max\{a, a'\} = a''$ with $a'' \geq a$, $a'' \geq a'$, then minimize $\sum a_{i,j} + b_{i,j} + x_i$.

4.2 Proof Search Results, a Time Lower Bound, and a Limitation

The structure of good lower bound proofs for the "TAUTOLOGY versus NTISP" problem turned out to be different from those for the "SAT versus DTISP" problem. The program uncovered interesting new results. For one, Fortnow and Van Melkebeek's $\sqrt{2}$ lower bound is not optimal; the best 11-line proof already gives a 1.419 exponent. For more details, cf. Appendix G. From experiments, we found annotations A_1, A_2, A_3, A_4 (all optimal for their number of lines) with the property that $A_{i+1} = [1, A_i, A_i, 0]$. This naturally suggests a proof by induction where the induction hypothesis is applied twice. We arrived at the following.

Theorem 4.2 TAUTOLOGY *requires* $n^{\sqrt[3]{4}-o(1)}$ *time for nondeterministic algorithms using* $n^{o(1)}$ *space.*

The proof is in Appendix H; it is an induction corresponding to an infinite sequence of annotations. The theorem's annotations and parameter settings for the first four steps of the induction are precisely those chosen by the search program for A_1, A_2, A_3, and A_4; in this sense, the formal proof corresponds with the best results from computer search. As experiments indicated that the sequence A_1, A_2, A_3, *etc.* is essentially the best one can do, we believe the lower bound is optimal for this framework.

Conjecture 4.1 *There is no alternation trading proof that* $\mathsf{coNTIME}[n] \not\subseteq \mathsf{NTS}[n^c]$, *for any* $c > \sqrt[3]{4}$.

Some interesting limitation can be proved for alternation-trading proofs. Namely, unlike the case of time lower bounds for SAT, no golden ratio lower bound can be achieved in this setting. The proof is in Appendix I.

Theorem 4.3 *There is no alternation trading proof that* $\mathsf{coNTIME}[n] \not\subseteq \mathsf{NTS}[n^c]$, *for any* $c \geq \phi \approx 1.618$.

5 Lower Bounds for Multidimensional TMs

Next, we consider lower bounds for a multidimensional machine model, which subsumes both the small-space random access model and off-line one-tape Turing machine models. Our approach yields new lower bounds here as well. To recall, the machine model has a read-only/random-access input tape, a read-write/random access storage of $n^{o(1)}$ bits, and a d-dimensional tape that is read-write with sequential (two-way) access.

5.1 The Framework and Linear Programming Translation

Define $\mathsf{DTIME}_d[t(n)]$ to be the class of languages recognized by d-dimensional one-tape machines in $O(t(n))$ time. Lower bound proofs for these machines have different structure from the first two: one speedup rule simulates a d-dimensional machine by a nondeterministic (or co-nondeterministic) machine with a small space bound. More precisely, let Q represent a string of quantifier blocks, so $Q = (Q_1\ n^{a_1})^{b_2} \cdots (Q_k\ n^{a_{k-1}})$.

Lemma 5.1 (DTIME_d to DTISP) *Let* $Q_{k+1} \in \{\exists, \forall\}$. *Then for all* $0 < s \leq a$ *and* $b \geq 1$,

$$Q^b\mathsf{DTIME}_d[n^a] \subseteq Q^b(Q_{k+1}n^{a-s})^{\max\{a-s,b\}}\mathsf{DTISP}[n^a, n^{ds}].$$

The proof of Lemma 5.1 guesses a short crossing sequence that divides the tape into n^s blocks, each of which can be simulated in $O(n^{ds})$ space, cf. [MS87, vMR05]. We omit its proof here.

Lemma 5.1 lets us use the Speedup Lemma for space-bounded machines (Lemma A.1) to prove class inclusions. Another Slowdown Lemma is required; its proof follows the lines of earlier results. Again, let Q be a string of quantifier blocks.

Lemma 5.2 (Slowdown for DTIME_d) *Suppose* $\mathsf{NTIME}[n] \subseteq \mathsf{DTIME}_d[n^c]$. *Then for* $a_i, b_i \geq 1$, $e \leq a_{k+1}$,

$$Q^{b_k}(Q_{k+1}\ n^{a_k})^{b_{k+1}}\mathsf{DTIME}_d[n^{a_{k+1}}] \subseteq Q^{b_k}\mathsf{DTIME}_d[n^{c\cdot\max\{b_k,b_{k+1},a_k,a_{k+1}\}}], \textit{ and}$$
$$Q^{b_k}(Q_{k+1}\ n^{a_k})^{b_{k+1}}\mathsf{DTISP}[n^{a_{k+1}}, n^e] \subseteq Q^{b_k}\mathsf{DTIME}_d[n^{c\cdot\max\{b_k,b_{k+1},a_k,a_{k+1}\}}].$$

We also use a standard time hierarchy theorem:

Lemma 5.3 *For* $a > a'$, $\mathsf{DTIME}_d[n^a] \not\subseteq \mathsf{DTIME}_d[n^{a'}]$.

Example. In 1983, Kannan [Kan83] proved that $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTIME}_1[n^{\sqrt[4]{3/2}}]$, using a weaker version of Lemma 5.1. Reproducing his argument:

$$\begin{aligned}
\mathsf{DTIME}_1[n^{3/2}] &\subseteq (\exists\, n)\mathsf{DTISP}[n^{3/2}, n^{1/2}] \quad \text{by } \mathsf{DTIME}_1 \text{ to } \mathsf{DTISP} \text{ (Lemma 5.1)}\\
&\subseteq (\exists\, n)(\forall\ \log n)\mathsf{DTISP}[n, n^{1/2}] \quad \text{by the Speedup Lemma for } \mathsf{DTISP} \text{ (Lemma A.1)}\\
&\subseteq (\exists\, n)\mathsf{DTIME}_1[n^c] \quad \text{by the Slowdown Lemma for } \mathsf{DTIME}_d \text{ (Lemma 5.2)}\\
&\subseteq \mathsf{DTIME}_1[n^{c^2}] \quad \text{by Slowdown for } \mathsf{DTIME}_d
\end{aligned}$$

A contradiction follows from $c < \sqrt{3/2}$ and Lemma 5.3. But SAT $\in \mathsf{DTIME}_1[n^c]$ implies $\mathsf{NTIME}[n] \subseteq \mathsf{DTIME}_1[n^{c+o(1)}]$ (Theorem A.1), so SAT cannot be solved in $O(n^{\sqrt{3/2}-\varepsilon})$ time on a 1-D TM. This is the SAT lower bound proved by Van Melkebeek and Raz [vMR05].

Corresponding notions of alternation-trading proofs and annotations can be defined here as well. The simple classes here have either a $\mathsf{DTISP}[t,s]$ or $\mathsf{DTIME}_d[t]$ phase at the end of their descriptions. There are two possible Speedup Lemmas for a class with a DTISP phase: one introduces only a single quantifier, and another introduces two. However, just as before, we can prove the second Speedup is unnecessary. Hence the proof annotations for this setting can be bit vectors, for the two rules applicable at each step. If the deterministic class is a DTIME_d class, a "speedup" means that we apply the DTIME_d to DTISP Lemma. If the deterministic class is DTISP, a "speedup" means that we apply the Speedup Lemma. For example, the annotation for above example is $[1,1,0,0]$. The structure of valid proof annotations changes accordingly.

One can also define a normal form proof that begins with $\mathsf{DTIME}_d[n^a]$ and ends with $\mathsf{DTIME}_d[n^{a'}]$, where $a' \leq a$. Such proofs imply lower bounds due to Lemma 5.3. Every alternation-trading lower bound proof against DTIME_d has a corresponding normal form proof, by an argument analogous to Theorem 4.1. Finally, the translation to linear programming is similar, except that new variables s_i are introduced for the space exponent of the DTISP class in line i (for all relevant i). These s_i are additional free parameters.

5.2 Time Lower Bound and Proof Limitation

The best annotation found (for all dimensions d) was the 66 line annotation

$$[1,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,0,$$
$$1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,1,1,0,0],$$

which leads to proofs that $\mathsf{NTIME}[n]$ is not in any of $\mathsf{DTIME}_1[n^{1.3009}]$, $\mathsf{DTIME}_2[n^{1.1875}]$, and $\mathsf{DTIME}_3[n^{1.1343}]$. In fact, all the best annotations found had the form $[1\ 1\ 1\ (0\ 1\ 1)^k\ 0\ (0\ 1\ 1)^\ell\ 0\ 0]$. (More details are in Appendix J.) The annotations suggest a proof where one has an inductive lemma capturing the $(0\ 1\ 1)^*$ behavior, then applies the lemma twice to a proof of six more lines. This strategy leads to:

Theorem 5.1 *SAT $\notin \mathsf{DTIME}_d[n^c]$, for all $c < r_d$ where $r_d \geq 1$ is a root of*

$$p_d(x) = (2d+1)(d+1)^2x^5 - 2(d+1)(2d+1)x^4 - d^2x^3 + 2(d+1)x^2 - ((2d+1)(d+1)^2+1)x + d(d+1).$$

The proof is in Appendix K. It establishes an inductive Speedup Lemma that efficiently simulates DTIME_d in $\Sigma_2\mathsf{TIME}$ assuming $\mathsf{NTIME}[n] \subseteq \mathsf{DTIME}_d[n^c]$, then applies the lemma twice in the lower bound proof. We again conjecture that the above lower bound is optimal for alternation-trading proofs. We can prove that no $n^{1+1/d}$ lower bound is possible for d-dimensional TMs. The proof is in Appendix L.

Theorem 5.2 *There is no alternation trading proof that SAT $\notin \mathsf{DTIME}_d[n^c]$, for any $c \geq 1 + 1/d$.*

6 Discussion

We introduced a methodology for reasoning about lower bounds in the alternation-trading framework. This gives an elegant and general way to attack lower bound problems via computer, and lets us establish concrete limitations on known techniques. We now have a better understanding of what these techniques can and cannot do, and a tool for addressing future problems. Previously, the problem of setting parameters to get a good lower bound was an highly technical exercise. This work should reduce the load on further research: once a new speedup or slowdown lemma is found, one only needs to find the relevant linear programming formulation to begin understanding its power. We end with two open problems.

1. *Establish tight limitations for alternation-trading proofs.* That is, show that the best possible alternation-trading proofs match the ones we have provided. Empirical results are sometimes met with skepticism, so it is critical to verify the limitations with formal proof. We have managed to prove non-trivial limitations, and it seems likely that the ideas in those can be extended.

2. *Discover ingredients that add signficantly to the framework.* Here there are several possible avenues. One is to find new separation results that lead to new contradictions. Another is to find improved Speedup and/or Slowdown Lemmas. The Slowdown Lemmas are the "blandest" of the ingredients, in that they are the most elementary (and they relativize). For instance, it may be possible to give an better Speedup Lemma by proving that DTS is contained *infinitely often* in a faster alternating time class, and use an *almost-everywhere* time hierarchy [GHS91, ABHH93] to obtain a contradiction.

   Finally, combinatorial methods have led to several impressive time-space lower bounds. For example, Ajtai [Ajt02] and Beame *et al.* [BJS01] have proven time-space lower bounds for branching programs; Gurevich and Shelah [GS88] gave a problem in $\mathsf{NTISP}[n, \log n]$ but not $\mathsf{DTISP}[n^{1+a}, n^b])$ when $b + 2a < 1/2$. Is it possible to incorporate combinatorial methods into the alternation-trading framework?

7 Acknowledgements

I am grateful to Dieter van Melkebeek, Ryan O'Donnell, Manuel Blum, and Steven Rudich for their invaluable feedback on my PhD thesis, which included some preliminary results on this work. I also thank Scott Aaronson for useful discussions about irrelativization, and anonymous referees for their comments.

References

[ADH97] L. Adleman, J. DeMarrais, and M. Huang. Quantum computability. *SIAM Journal on Computing* 26:1524-1540, 1997.

[ABHH93] E. Allender, R. Beigel, U. Hertrampf, and S. Homer. Almost-Everywhere Complexity Hierarchies for Nondeterministic Time. *Theor. Comput. Sci.* 115(2):225–241, 1993.

[AKRRV01] E. Allender, M. Koucky, D. Ronneburger, S. Roy, and V. Vinay. Time-Space Tradeoffs in the Counting Hierarchy. In *Proceedings of IEEE Conference on Computational Complexity (CCC)*, 295–302, 2001.

[Ajt02] M. Ajtai. Determinism versus Nondeterminism for Linear Time RAMs with Memory Restrictions. *J. Computer and System Sciences* 65:2–37, 2002.

[BJS01] P. Beame, T. S. Jayram, and M. E. Saks. Time-Space Tradeoffs for Branching Programs. *J. Computer and System Sciences* 63(4):542–572, 2001.

[CKS81] A. K. Chandra, D. Kozen, and L. J. Stockmeyer. Alternation. *JACM* 28(1):114–133, 1981.

[Coo88] S. A. Cook. Short Propositional Formulas Represent Nondeterministic Computations. *Information Processing Letters* 26(5): 269-270, 1988.

[Der72] C. Derman. *Finite State Markov Decision Processes.* Academic Press, 1972.

[DvM06] S. Diehl and D. van Melkebeek. Time-Space Lower Bounds for the Polynomial-Time Hierarchy on Randomized Machines. *SIAM J. Computing* 36: 563-594, 2006.

[DvMW07] S. Diehl, D. van Melkebeek, and R. Williams. An Improved Time-Space Lower Bound for Tautologies. Manuscript, 2007.

[Fis74] M. J. Fischer. Lecture Notes on Network Complexity. Yale Technical Report 1104, June 1974. URL: `http://cs-www.cs.yale.edu/homes/fischer/pubs/tr1104.ps.gz`

[For97] L. Fortnow. Nondeterministic Polynomial Time Versus Nondeterministic Logarithmic Space: Time-Space Tradeoffs for Satisfiability. In *Proceedings of IEEE Conference on Computational Complexity (CCC)*, 52–60, 1997.

[FvM00] L. Fortnow and D. van Melkebeek. Time-Space Tradeoffs for Nondeterministic Computation. In *Proceedings of IEEE Conference on Computational Complexity (CCC)*, 2–13, 2000.

[FLvMV05] L. Fortnow, R. Lipton, D. van Melkebeek, and A. Viglas. Time-Space Lower Bounds for Satisfiability. *JACM* 52(6):835–865, 2005.

[GHS91] J. G. Geske, D. T. Huynh, and J. I. Seiferas. A Note on Almost-Everywhere-Complex Sets and Separating Deterministic-Time-Complexity Classes. *Inf. Comput.* 92(1):97–104, 1991.

[Gri82] D. Yu. Grigor'ev. Time complexity of multidimensional Turing machines. *J. Mathematical Sciences* 20(4):2290–2295, 1982.

[GS88] Y. Gurevich and S. Shelah. Nondeterministic Linear-Time Tasks May Require Substantially Nonlinear Deterministic Time in the Case of Sublinear Work Space. *JACM* 37(3):674–687, 1990.

[HLMW86] J. Y. Halpern, M. C. Loui, A. R. Meyer, and D. Weise. On Time versus Space III. *Mathematical Systems Theory* 19(1):13–28, 1986.

[HPV77] J. Hopcroft, W. Paul, and L. Valiant. On time versus space. *JACM* 24(2):332–337, 1977.

[Kan83] R. Kannan. Alternation and the power of nondeterminism. In *Proceedings of ACM Symposium on Theory of Computing (STOC)*, 344–346, 1983.

[Kan84] R. Kannan. Towards Separating Nondeterminism from Determinism. *Mathematical Systems Theory* 17(1):29–45, 1984.

[LV99] R. J. Lipton and A. Viglas. On the Complexity of SAT. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, 459–464, 1999.

[LL90] M. Liskiewicz and K. Lorys. Fast Simulations of Time-Bounded One-Tape Turing Machines by Space-Bounded Ones. *SIAM J. Computing* 19(3):511–521, 1990.

[Lou80] M. C. Loui. Simulations among multidimensional Turing machines. Ph.D. Thesis, Massachusetts Institute of Technology TR-242, 1980.

[MS87] W. Maass and A. Schorr. Speed-Up of Turing Machines with One Work Tape and a Two-Way Input Tape. *SIAM J. Computing* 16(1):195–202, 1987.

[vM04] D. van Melkebeek. Time-Space Lower Bounds for NP-Complete Problems. In G. Paun, G. Rozenberg, and A. Salomaa (eds.), *Current Trends in Theoretical Computer Science* 265–291, World Scientific, 2004.

[vM07] D. van Melkebeek. A Survey of Lower Bounds for Satisfiability and Related Problems. To appear in *Foundations and Trends in Theoretical Computer Science*, 2007.

[vMR05] D. van Melkebeek and R. Raz. A Time Lower Bound for Satisfiability. *Theoretical Computer Science* 348(2-3):311–320, 2005.

[vMW07] D. van Melkebeek and T. Watson. A quantum time-space lower bound for the counting hierarchy. Technical Report 1600, Department of Computer Sciences, University of Wisconsin-Madison, 2007.

[Nep70] V. Nepomnjascii. Rudimentary predicates and Turing calculations. *Soviet Math. Doklady* 11:1462–1465, 1970.

[PR81] W. Paul and R. Reischuk. On time versus space II. *J. Computer and System Sciences* 22:312–327, 1981.

[PPST83] W. Paul, N. Pippenger, E. Szemeredi, and W. Trotter. On determinism versus nondeterminism and related problems. In *Proceedings of IEEE Symposium on Foundations of Computer Science (FOCS)*, 429–438, 1983.

[PF79] N. Pippenger and M. J. Fischer. Relations Among Complexity Measures. *JACM* 26(2):361–381, 1979.

[San01] R. Santhanam. Lower bounds on the complexity of recognizing SAT by Turing machines. *Information Processing Letters* 79(5):243–247, 2001.

[Sch78] C. Schnorr. Satisfiability is quasilinear complete in NQL. *JACM* 25(1):136–145, 1978.

[Tou01] I. Tourlakis. Time-Space Tradeoffs for SAT on Nonuniform Machines. *J. Computer and System Sciences* 63(2):268–287, 2001.

[Vio07] E. Viola. On approximate majority and probabilistic time. In *Proceedings of the 22th IEEE Conference on Computational Complexity (CCC)*, 2007.

[Wil06] R. Williams. Inductive Time-Space Lower Bounds for SAT and Related Problems. *J. Computational Complexity* 15:433–470, 2006.

[Wil07] R. Williams. Algorithms and Resource Requirements for Fundamental Problems. Ph.D. Thesis, Carnegie Mellon University, CMU-CS-07-147, August 2007.

[Wil08] R. Williams. Time-Space Tradeoffs for Counting NP Solutions Modulo Integers. *J. Computational Complexity* 17(2), 2008.

Guide to the Appendices

- Appendix A is a short introduction to the techniques used in this work and related ones.
- Appendix B proves that for any time t and space s SAT algorithm, $ts = \Omega(n^c)$ for all $c < 2\cos(\pi/7)$.
- Appendix C proves that Speedup Rule 2 is redundant, i.e., it can be simulated by Speedup Rule 1 in the proof system.
- Appendix D proves some properties that allow us to restrict ourselves to considering normal form proofs.
- Appendix E analyzes results of a computer proof search for SAT time-space lower bounds.
- Appendix F proves that no quadratic time lower bound for SAT in the $n^{o(1)}$ space setting can be proved with alternation-trading proofs.
- Appendix G analyzes results of a computer proof search for Tautology time-space lower bounds (on nondeterministic machines).
- Appendix H proves a $n^{\sqrt[3]{4}}$ time lower bound, based on results of Appendix F.
- Appendix I proves that an $n^{\phi} \approx n^{1.618}$ time lower bound for Tautology is not possible.
- Appendix J analyzes results of a computer proof search for SAT lower bounds on multidimensional TMs.
- Appendix K proves lower bounds based on the results of Appendix I.
- Appendix L proves that an $n^{1+1/d}$ time lower bound for SAT on d-dimensional TMs is not possible with alternation-trading.
- Appendix M proves new lower bounds for QBF$_k$, inspired by the results of computer searches.

A A Short Introduction to Time-Space Lower Bounds

Here we give a brief overview of the tools that have been used to prove time-space tradeoff lower bounds. We focus on deterministic time lower bounds for satisfiability for algorithms using $n^{o(1)}$ space, as the other relevant lower bound problems use analogous tools and the $n^{o(1)}$ space case is especially simple to work with.

It is known that satisfiability of Boolean formulas in conjunctive normal form (SAT) is a complete problem under tight reductions for a small nondeterministic complexity class. Define NQL as *nondeterministic quasi-linear time, i.e.*

$$\mathsf{NQL} := \bigcup_{c \geq 0} \mathsf{NTIME}[n \cdot (\log n)^c] = \mathsf{NTIME}[n \cdot poly(\log n)].$$

Theorem A.1 (Cook [Coo88], Schnorr [Sch78], Tourlakis [Tou01], Fortnow *et al.* [FLvMV05]) *SAT is NQL-complete, under reductions in quasi-linear time and $O(\log n)$ space simultaneously, for both multitape and random access machine models. Moreover, each bit of the reduction can be computed in $O(poly(\log n))$ time and $O(\log n)$ space in both machine models.*

Let $\mathcal{C}[t(n)]$ represent a time $t(n)$ complexity class under one of the three models:

- deterministic RAM using time t and $t^{o(1)}$ space,
- co-nondeterministic RAM using time t and $t^{o(1)}$ space,
- d-dimensional Turing machine using time t.

The above theorem implies that if $\mathsf{NTIME}[n] \nsubseteq \mathcal{C}[t]$, then SAT $\notin \mathcal{C}[t]$, modulo polylogarithmic factors.

Corollary A.1 *If $\mathsf{NTIME}[n] \nsubseteq \mathcal{C}[t(n)]$, then there is a $k > 0$ such that SAT $\notin \mathcal{C}[t(n) \cdot (\log t(n))^k]$.*

Hence we want to prove $\mathsf{NTIME}[n] \nsubseteq \mathcal{C}[n^c]$ for a large constant $c > 1$. For the purposes of time lower bounds for small space algorithms, we work with $\mathcal{C}[n^c] = \mathsf{DTS}[n^c] = \mathsf{DTISP}[n^c, n^{o(1)}]$. Van Melkebeek and Raz [vMR05] observed that a similar corollary holds for any problem Π such that SAT reduces to Π under highly efficient reductions, *e.g.* VERTEX COVER, HAMILTON PATH, 3-SAT, and MAX-2-SAT. It follows that identical time lower bounds hold for these problems as well.

Speedups, Slowdowns, and Contradictions. Given that our goal is to prove $\mathsf{NTIME}[n] \nsubseteq \mathsf{DTS}[n^c]$, how can we go about this? In an alternation-trading proof, we assume that $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ and attempt to establish a contradiction, by applying two lemmas in such a way that a time hierarchy is violated. One lemma (called the "speedup lemma") takes a $\mathsf{DTS}[t]$ class and places it in an alternating class with runtime $o(t)$; the other (called the "slowdown lemma") takes an alternating class with runtime t and places it in a class with one less alternation and runtime $O(t^c)$.

Lemma A.1 (Speedup Lemma) *Let $a \geq 1$, $e \geq 0$ and $0 \leq x \leq a$. Then*

$$\mathsf{DTISP}[n^a, n^e] \subseteq (Q_1\ n^{x+e})^{\max\{1,x+e\}}(Q_2\ \log n)^{\max\{1,e\}}\mathsf{DTISP}[n^{a-x}, n^e],$$

for $Q_i \in \{\exists, \forall\}$ where $Q_1 \neq Q_2$. In particular,

$$\mathsf{DTS}[n^a] \subseteq (Q_1\ n^x)^{\max\{1,x\}}(Q_2\ \log n)^1\mathsf{DTS}[n^{a-x}].$$

Proof. Let M be a random access machine using n^a time and n^e space. To get a simulation of M having type $(\exists\ n^{x+e})^{\max\{1,x+e\}}(\forall\ \log n)^{\max\{1,e\}}\mathsf{DTISP}[n^{a-x}, n^e]$, the simulation $N(x)$ existentially guesses a sequence of configurations $C_1, \ldots, C_{n^x}$ of $M(x)$. It then appends the initial configuration to the beginning of the sequence and the accepting configuration to the end of the sequence. Then $N(x)$ universally guesses a $i \in \{0, \ldots, n^x\}$, erases all configurations except C_i and C_{i+1}, then simulates $M(x)$ starting from C_i, accepting if and only if the C_{i+1} is reached within n^{a-x} steps. It is easy to see that the simulation is correct. The input constraints on the quantifier blocks are satisfied, since after the universal guess, the input is only x, C_i, and C_{i+1}, which is of size $n + 2n^e \leq n^{\max\{1,e\}+o(1)}$. $\square$

The Speedup Lemma dates back to work of Nepomnjascii [Nep70] and Kannan [Kan84]. Note that in the above alternating simulation, the input to the final DTISP computation is linear in $n + n^e$, *regardless of the choice of* x. This is a surprisingly subtle property that is exploited heavily in alternation-trading proofs. The Slowdown Lemma is the following folklore result:

Lemma A.2 (Slowdown Lemma) *Let $a \geq 1$, $e \geq 0$, $a' \geq 0$, and $b \geq 1$. If $\mathsf{NTIME}[n] \subseteq \mathsf{DTISP}[n^c, n^e]$, then for both $Q \in \{\exists, \forall\}$,*

$$(Q\ n^{a'})^b\mathsf{DTIME}[n^a] \subseteq \mathsf{DTISP}[n^{c\cdot\max\{a,a',b\}}, n^{e\cdot\max\{a,a',b\}}].$$

In particular, if $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$, then

$$(Q\ n^{a'})^b\mathsf{DTIME}[n^a] \subseteq \mathsf{DTS}[n^{c\cdot\max\{a,a',b\}}].$$

Proof. Let L be a problem in $(Q\ n^{a'})^b\mathsf{DTIME}[n^a]$, and let A be an algorithm satisfying $L(A) = L$. On an input x of length n, A guesses a string y of length $n^{a'+o(1)}$, then feeds an $n^{b+o(1)}$ bit string z to $A'(z)$, where A' is a deterministic algorithm that runs in n^a time. Since $\mathsf{NTIME}[n] \subseteq \mathsf{DTISP}[n^c, n^e]$ and DTISP is closed under complement, by padding we have $\mathsf{NTIME}[p(n)] \cup \mathsf{coNTIME}[p(n)] \subseteq \mathsf{DTISP}[p(n)^c, p(n)^e]$ for polynomials $p(n) \geq n$. Therefore A can be simulated with a deterministic algorithm B. Since the total runtime of A is $n^{a'+o(1)} + n^{b+o(1)} + n^a$, the runtime of B is $n^{c\cdot\max\{a,a',b\}+o(1)}$ and the space usage is similar. $\square$

The final component of an alternation-trading proof is a time hierarchy theorem, the most general of which is the following, provable by a simple diagonalization.

Theorem A.2 (Alternating Time Hierarchy) *For $k \geq 0$, for all $Q_i \in \{\exists, \forall\}$, $a'_i > a_i \geq 1$, and $b'_i \geq b_i \geq 1$,*

$$(Q_1\ n^{a_1})^{b_2} \cdots ^{b_k} (Q_k\ n^{a_k})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}] \not\subseteq (R_1\ n^{a'_1})^{b'_2} \cdots ^{b'_k} (R_k\ n^{a'_k})^{b'_{k+1}}\mathsf{DTS}[n^{a'_{k+1}}],$$

where $R_i \in \{\exists, \forall\}$ and $R_i \neq Q_i$.

Remark 1 *Are alternation-trading proofs relativizing? The Slowdown Lemma relativizes, but the Speedup Lemma does not relativize in most oracle models, for the simple reason that the original machine runs longer than the (sped-up) host machine, and can therefore ask longer queries. This is typically the case. For example, the proof that $\mathsf{NTIME}[n] \neq \mathsf{DTIME}[n]$ is non-relativizing, since a powerful enough oracle makes the two classes equal. Therefore, we consider alternation-trading proofs to be in that rare class of non-relativizing and non-naturalizing lower bounds (but acknowledge that our belief is not unanimously held).*

Two Instructive Examples. In order to understand alternation-trading proofs, it is necessary to consider some examples. The art behind their construction consists of finding the proper sequence of rules to apply, and the right settings of the parameter x in the Speedup Lemma.

1. In FOCS'99, Lipton and Viglas proved that SAT cannot be solved by algorithms running in $n^{\sqrt{2}-\varepsilon}$ time and $n^{o(1)}$ space, for all $\varepsilon > 0$. Their proof can be summarized as follows: by Theorem A.1, the assumption that there is such a SAT algorithm implies that $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ with $c^2 < 2$. Then

$$\begin{aligned}(\exists\, n^{2/c^2})(\forall\, n^{2/c^2})\mathsf{DTS}[n^{2/c^2}] &\subseteq (\exists\, n^{2/c^2})\mathsf{DTS}[n^{2/c}] \quad \text{(Slowdown Lemma)}\\ &\subseteq \mathsf{DTS}[n^2] \quad \text{(Slowdown Lemma)}\\ &\subseteq (\forall\, n)(\exists\, \log n)\mathsf{DTS}[n] \quad \text{(Speedup Lemma, with } x = 1\text{)}.\end{aligned}$$

   But $(\exists\, n^{2/c^2})(\forall\, n^{2/c^2})\mathsf{DTS}[n^{2/c^2}] \subseteq (\forall\, n)(\exists\, \log n)\mathsf{DTS}[n]$ contradicts Theorem A.2. In fact, one can show that if $c^2 = 2$, we still have a contradiction with $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$, so the ε can be removed from the previous statement and state that SAT cannot be solved in $n^{\sqrt{2}}$ time and $n^{o(1)}$ exactly.[7]

2. Improving on the previous example, one can show SAT $\notin \mathsf{DTS}[n^{1.6004}]$. If $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ and $\sqrt{2} \leq c < 2$, then by applying the Speedup and Slowdown Lemmas appropriately, one can derive:

$$\begin{aligned}\mathsf{DTS}[n^{c^2/2+2}] &\subseteq (\exists\, n^{c^2/2})(\forall\, \log n)\mathsf{DTS}[n^2]\\ &\subseteq (\exists\, n^{c^2/2})(\forall\, \log n)(\forall\, n)(\exists\, \log n)\mathsf{DTS}[n]\\ &= (\exists\, n^{c^2/2})(\forall\, n)(\exists\, \log n)\mathsf{DTS}[n]\\ &\subseteq (\exists\, n^{c^2/2})(\forall\, n)\mathsf{DTS}[n^c]\\ &\subseteq (\exists\, n^{c^2/2})\mathsf{DTS}[n^{c^2}]\\ &\subseteq (\exists\, n^{c^2/2})(\exists\, n^{c^2/2})(\forall\, \log n)\mathsf{DTS}[n^{c^2/2}]\\ &= (\exists\, n^{c^2/2})(\forall\, \log n)\mathsf{DTS}[n^{c^2/2}]\\ &\subseteq (\exists\, n^{c^2/2})\mathsf{DTS}[n^{c^3/2}]\\ &\subseteq \mathsf{DTS}[n^{c^4/2}]\end{aligned}$$

   When $c^2/2 + 2 > c^4/2$ (which happens if $c < 1.6004$), we have $\mathsf{DTS}[n^a] \subseteq \mathsf{DTS}[n^{a'}]$ for some $a > a'$. Notice that we do not know if $\mathsf{DTS}[n^a] \not\subseteq \mathsf{DTS}[n^{a'}]$ when $a' > a$, as the space bounds on both sides of the inequality are the same. However one can still show by a translation argument (similar to the footnote) that either $\mathsf{DTS}[n^a] \not\subseteq \mathsf{DTS}[n^{a'}]$ or $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$, concluding the proof.

While the second example is more clever in structure, a more interesting fact is that the proof was found by a computer program. By "found", we mean that the program applied the Speedup and Slowdown Lemmas in precisely the same order and the same parameter settings, having only minimum knowledge of these Lemmas along with a way to check the validity of the parameters. Moreover, the program verified that the above is the *best possible* alternation-trading proof that applies the Speedup and Slowdown Lemmas for at most 7 times. A precise definition of "alternation-trading proof" is given in Section 3.

[7]Suppose $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ and $\Sigma_2\mathsf{TIME}[n] \subseteq \Pi_2\mathsf{TIME}[n^{1+o(1)}]$. The first assumption, along with the Speedup and Slowdown Lemmas, implies that for every k there's a K satisfying $\Sigma_2\mathsf{TIME}[n^k] \subseteq \mathsf{NTIME}[n^{kc}] \subseteq \Sigma_K\mathsf{TIME}[n]$. But the second assumption implies that $\Sigma_K\mathsf{TIME}[n] = \Sigma_2\mathsf{TIME}[n^{1+o(1)}]$. Hence $\Sigma_2\mathsf{TIME}[n^k] \subseteq \Sigma_2\mathsf{TIME}[n^{1+o(1)}]$, which contradicts the time hierarchy for $\Sigma_2\mathsf{TIME}$.

B New Lower Bound on the Time-Space Product for SAT[8]

Using Lemma A.1 in its full generality, it is possible to adapt our linear programming framework to prove time lower bounds for SAT for any fixed space bound n^{δ}, where $\delta \in (0,1)$. Trying a range of values for δ, we found that for each of them, the optimal annotations for the $n^{o(1)}$ space setting also appeared to be optimal for *every* space bound n^{δ}. The following table gives time-space pairs for which our theorem prover has shown that no SAT algorithm can satisfy both time and space requirements simultaneously.

Time	Space
$n^{1.06}$	$n^{.9}$
$n^{1.17}$	$n^{.75}$
$n^{1.24}$	$n^{.666}$
$n^{1.36}$	$n^{.5}$
$n^{1.51}$	$n^{.333}$
$n^{1.58}$	$n^{.25}$
$n^{1.7}$	$n^{.1}$
$n^{1.75}$	$n^{.05}$

Based on this table, it is natural to conjecture that the time-space product for any algorithm solving SAT is at least $\Omega(n^{2\cos(\pi/7)}) \geq \Omega(n^{1.801})$, and the product is minimized when the space is as small as possible. In the below, we establish the conjecture. To the best of our knowledge, the previously best known bound on the time-space product was only $\Omega(n^{1.573})$ [FLvMV05]. While the proof annotations in the below are analogous to the $2\cos(\pi/7)$ bound (as suggested by the experiments), the parameter settings in the proof rely a great deal on our study of the theorem prover's output.

Theorem B.1 *Let $t(n)$ and $s(n)$ be bounded above by polynomials. Any algorithm solving SAT in time t and space s requires $t \cdot s = \Omega(n^{2\cos(\pi/7)-\varepsilon})$ for all $\varepsilon > 0$.*

Proof. Suppose SAT is solved in time $t = n^c$ and space $s = n^d$, with $c + d \leq 2\cos(\pi/7)$. Of course we must have $c \geq 1$, and so $d \leq 1 - 2\cos(\pi/7) < 1$. By Theorem A.1, it follows that $\mathsf{NTIME}[n] \subseteq \mathsf{DTISP}[n^{c+o(1)}, n^{d+o(1)}]$.

Define the sequences $c_1 := 2 - d$, $c_{k+1} := \frac{c+c_k}{c+d}$, and $d_1 := d$, $d_{k+1} := \frac{d \cdot c_{k+1}}{c}$. It is easy to see that $c_k \geq c$ for all k, and that the sequences $\{c_k\}$ and $\{d_k\}$ are monotone nondecreasing for $c + d \leq 2$ and $d < 1$.

First we prove that for all k,

$$\mathsf{DTISP}[n^{c_k}, n^{d_k}] \subseteq (\exists\, n^{1+o(1)})(\forall\, \log n)\mathsf{DTISP}[n^{1+o(1)}, n^{d+o(1)}]. \quad (2)$$

By the Speedup Lemma,

$$\mathsf{DTISP}[n^{c_1}, n^{d_1+o(1)}] = \mathsf{DTISP}[n^{2-d}, n^{d+o(1)}] \subseteq (\exists\, n^{1+o(1)})(\forall\, \log n)\mathsf{DTISP}[n, n^{d+o(1)}].$$

For the inductive step, we have the following (subtle) series of inclusions:

$$\begin{aligned}
\mathsf{DTISP}[n^{c_{k+1}}, n^{d_{k+1}+o(1)}] &\subseteq (\exists\, n^{1+o(1)})(\forall\, \log n)\mathsf{DTISP}[n^{c_{k+1}-(1-d_{k+1})}, n^{d_{k+1}+o(1)}] \quad \text{(Speedup)}\\
&= (\exists\, n^{1+o(1)})(\forall\, \log n)\mathsf{DTISP}[n^{c_k/c}, n^{d_{k+1}+o(1)}] \quad \text{(def. of } c_{k+1}, \text{ \& } c_k \geq c)\\
&\subseteq (\exists\, n^{1+o(1)})\mathsf{DTISP}[n^{c_k+o(1)}, n^{d_k+o(1)}] \quad \text{(Slowdown \& def. of } d_k)\\
&\subseteq (\exists\, n^{1+o(1)})(\forall\, \log n)\mathsf{DTISP}[n^{1+o(1)}, n^{d+o(1)}] \quad \text{(by induction).}
\end{aligned}$$

[8]A early version of the work in this section was reported in the author's PhD thesis in 2007.

The sequence $\{c_k\}$ converges to $c_\infty = c/(c-1+d)$, hence $\{d_k\}$ converges to $d_\infty = d/(c-1+d)$. (Note we have $c \geq c-1+d$, since $d \leq 1$.) Therefore for all $c' < c_\infty$, $d' < d_\infty$,

$$\mathsf{DTISP}[n^{c'}, n^{d'}] \subseteq (\exists\, n^{1+o(1)})(\forall\, \log n)\mathsf{DTISP}[n^{1+o(1)}, n^{d+o(1)}]. \tag{3}$$

Note if $c^2 \leq c_k$ for any k, we already obtain a contradiction: for sufficiently large t, we have

$$\mathsf{NTIME}[t^{c_k/c}] \subseteq \mathsf{DTISP}[t^{c_k}, t^{d_k}] \subseteq (\exists\, t^{1+o(1)})(\forall\, \log n)\mathsf{DTISP}[t^{1+o(1)}, t^{d+o(1)}] \subseteq \mathsf{NTIME}[t^c],$$

where the first inclusion follows from Slowdown, the second from eq.(2), and the third from Slowdown. From these containments, one can derive a time hierarchy style contradiction, along the lines of Lemma 3.1. Therefore we may assume that $c^2 > c_k$ for all k.

Equation (3) can be combined with another inductive argument to produce a contradiction. In particular, for all k and ℓ,

$$\mathsf{DTISP}[n^{c_\ell - kd_\ell + \sum_{i=1}^{k}(c^2/c_\ell)^i}, n^{d_\ell}] \subseteq (\exists\, n^{(c^2/c_\ell)^k})(\forall\, \log n)\mathsf{DTISP}[n^{(c^2/c_\ell)^k+o(1)}, n^{d(c^2/c_\ell)^k+o(1)}]. \tag{4}$$

The proof of equation (4) is very similar in structure to Lemma 6.8 in [Wil08]. For completeness, we give the sequence of inclusions to derive it. When $k=1$, for arbitrary ℓ we have

$$\begin{aligned}
\mathsf{DTISP}[n^{c_\ell+(c^2/c_\ell)-d_\ell}, n^{d_\ell}] &\subseteq (\exists\, n^{c^2/c_\ell})(\forall\, \log n)^1\mathsf{DTISP}[n^{c_\ell}, n^{d_\ell}] \quad \text{(Speedup)}\\
&\subseteq (\exists\, n^{c^2/c_\ell})(\forall\, \log n)^1(\forall\, n^{1+o(1)})(\exists\, \log n)\mathsf{DTISP}[n^{1+o(1)}, n^{d+o(1)}] \quad \text{(Eq. (2))}\\
&\subseteq (\exists\, n^{c^2/c_\ell})(\forall\, n^{1+o(1)})\mathsf{DTISP}[n^{c+o(1)}, n^{d+o(1)}] \quad \text{(Slowdown)}\\
&\subseteq (\exists\, n^{c^2/c_\ell})\mathsf{DTISP}[n^{c^2+o(1)}, n^{dc+o(1)}] \quad \text{(Slowdown; note } c_\ell \geq c \text{ so } c \geq c^2/c_\ell)\\
&\subseteq (\exists\, n^{c^2/c_\ell})(\exists\, n^{c^2/c_\ell})(\forall\, \log n)\mathsf{DTISP}[n^{c^2/c_\ell+o(1)}, n^{dc^2/c_\ell+o(1)}] \quad \text{(Eq. (2))}\\
&= (\exists\, n^{c^2/c_\ell})(\forall\, \log n)\mathsf{DTISP}[n^{c^2/c_\ell+o(1)}, n^{dc^2/c_\ell+o(1)}].
\end{aligned}$$

For the inductive step, we have

$$\begin{aligned}
&\mathsf{DTISP}[n^{c_\ell - kd_\ell + \sum_{i=1}^{k}(c^2/c_\ell)^i}, n^{d_\ell}]\\
&\subseteq (\exists\, n^{(c^2/c_\ell)^k})(\forall\, \log n)\mathsf{DTISP}[n^{c_\ell-(k-1)d_\ell+\sum_{i=1}^{k-1}(c^2/c_\ell)^i+o(1)}, n^{d_\ell}] \quad \text{(Speedup)}\\
&\subseteq (\exists\, n^{(c^2/c_\ell)^k})(\forall\, \log n)(\forall\, n^{(c^2/c_\ell)^{k-1}})(\exists\, \log n)\mathsf{DTISP}[n^{(c^2/c_\ell)^{k-1}+o(1)}, n^{d(c^2/c_\ell)^{k-1}+o(1)}] \quad \text{(Induction)}\\
&\subseteq (\exists\, n^{(c^2/c_\ell)^k})(\forall\, \log n)(\forall\, n^{(c^2/c_\ell)^{k-1}})\mathsf{DTISP}[n^{c(c^2/c_\ell)^{k-1}+o(1)}, n^{d(c^2/c_\ell)^{k-1}+o(1)}] \quad \text{(Slowdown)}\\
&\subseteq (\exists\, n^{(c^2/c_\ell)^k})\mathsf{DTISP}[n^{c^2(c^2/c_\ell)^{k-1}+o(1)}, n^{dc(c^2/c_\ell)^{k-1}+o(1)}] \quad \text{(Slowdown)}\\
&\subseteq (\exists\, n^{(c^2/c_\ell)^k})(\exists\, n^{(c^2/c_\ell)^k})(\forall\, \log n)\mathsf{DTISP}[n^{(c^2/c_\ell)^k+o(1)}, n^{d(c^2/c_\ell)^k+o(1)}] \quad \text{(Eq. (2))}\\
&= (\exists\, n^{(c^2/c_\ell)^k})(\forall\, \log n)\mathsf{DTISP}[n^{(c^2/c_\ell)^k+o(1)}, n^{d(c^2/c_\ell)^k+o(1)}]
\end{aligned}$$

Now, for sufficiently large $t(n) \geq n$ we have

$$\begin{aligned}
&\mathsf{NTIME}[t^{(c_\ell-kd_\ell+\sum_{i=1}^{k}(c^2/c_\ell)^i)/c}]\\
&\subseteq \mathsf{DTISP}[t^{c_\ell-kd_\ell+\sum_{i=1}^{k}(c^2/c_\ell)^i+o(1)}, n^{d_\ell}] \quad \text{(Slowdown)}\\
&\subseteq (\exists\, t^{(c^2/c_\ell)^k})(\forall\, \log n)\mathsf{DTISP}[t^{(c^2/c_\ell)^k+o(1)}, n^{d(c^2/c_\ell)^k+o(1)}] \quad \text{(Eq. (4))}\\
&\subseteq (\exists\, t^{(c^2/c_\ell)^k})\mathsf{DTISP}[t^{c(c^2/c_\ell)^k+o(1)}, n^{d(c^2/c_\ell)^k+o(1)}] \quad \text{(Slowdown)}\\
&\subseteq \mathsf{NTIME}[t^{c(c^2/c_\ell)^k+o(1)}].
\end{aligned}$$

A contradiction with the nondeterministic time hierarchy follows, when

$$c_\ell - kd_\ell + \sum_{i=1}^{k}(c^2/c_\ell)^i < c^2(c^2/c_\ell)^k. \tag{5}$$

Finally, we claim that when $c + d < 2\cos(\pi/7)$, inequality (5) holds for sufficiently large k and ℓ. The theorem follows immediately from this claim. To see why the claim is true, we analyze the case where k and ℓ grow unboundedly and use the fact that the underlying sequences are monotone nondecreasing. Rewrite the inequality into the form

$$\frac{c_\ell - kd_\ell}{(c^2/c_\ell)^k} + \sum_{i=1}^{k}(c^2/c_\ell)^{i-k} < c^2. \tag{6}$$

As $k \to \infty$, the first term on the LHS vanishes since $c^2 > c_\ell$ for all ℓ. The second term converges to $\frac{1}{1-c_\ell/c^2}$. Now as $\ell \to \infty$, $c_\ell \to c_\infty = c/(c-1+d)$. Hence in the limit, inequality (5) becomes

$$0 + \frac{1}{1-\frac{1}{c(c-1+d)}} < c^2 \iff 1 < c^2 - \frac{c^2}{c^2-c+dc} \iff c^2 - c + dc < c^2(c^2-c+dc) - c^2$$

$$\iff c - 1 + d < c(c^2 - c + dc) - c \iff 2c - 1 + d < c^3 - c^2 + dc^2 \iff 0 < c^3 - c^2 + dc^2 - 2c + 1 - d.$$

It remains to show that the bivariate polynomial $p(x, y) = x^3 - x^2 - 2x + 1 + y(x^2 - 1)$ is greater than 0 over all points (x, y) where $x + y > 2\cos(\pi/7)$. When $y = 0$, $p(x) = 0$ over the range $x \in [1, \infty]$ precisely when $x = 2\cos(\pi/7)$, and $p(x) > 0$ for all $x < 2\cos(\pi/7)$. But for all $y > 0$ and $x > 1$, the resulting polynomial strictly dominates $p(x, 0)$ and we also have $p(x, y) > 0$ for any $x + y < 2\cos(\pi/7)$.

□

C Speedup Rule 2 is Redundant

To prove this we first need a lemma relating the a_i's and b_i's of an alternating class.

Definition C.1 *A class* $(Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2}) \cdots ^{b_k} (Q_k\ n^{a_k})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}]$ *is* orderly *if it is either (a) a* $\mathsf{DTS}[n^a]$ *class, or (b) for all* $i = 1, \ldots, k$, $a_i \leq b_{i+1}$.

Lemma C.1 *Suppose* A_1 *is orderly. Then every alternation-trading proof beginning with* A_1 *consists of only orderly classes.*

Proof. Induction on the number of lines. The base case is trivial. The induction hypothesis is that the ℓth line,

$$(Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2}) \cdots ^{b_k} (Q_k\ n^{a_k})^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}}],$$

is orderly. For the $(\ell + 1)$th line, we consider the rules in turn:

- (Speedup Rule 0) Clearly $(Q_k\ n^x)^{\max\{x,1\}}(Q_{k+1}\ n^0)^1\mathsf{DTS}[n^{a_{k+1}-x}]$ is orderly.
- (Speedup Rule 1) Suppose the line is

  $$(Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2}) \cdots ^{b_k} (Q_k\ n^{\max\{a_k,x\}})^{\max\{x,b_{k+1}\}}(Q_{k+1}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}].$$

  Then $a_k \leq b_{k+1}$ by the induction hypothesis, so $\max\{a_k, x\} \leq \max\{x, b_{k+1}\}$, $1 \leq b_{k+1}$, thus the class is orderly.

- (Speedup Rule 2) This case is clear, as the line is:
$$(Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^{x})^{\max\{x,b_{k+1}\}}(Q_{k+2}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}].$$
- (Slowdown Rule) Obvious, given the hypothesis.

This concludes all the cases. □

Lemma C.2 *Let A_1 be orderly. For every alternation-trading proof that* $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c] \implies A_1 \subseteq A_2$*, there is another alternation-trading proof of the same implication that does not use Speedup Rule 2.*

Proof. Consider a proof P that applies Speedup Rule 2 at some line. The line has the form
$$A = (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^{x})^{\max\{x,b_{k+1}\}}(Q_{k+2}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}].$$
We consider two cases:

1. If $x \le a_k$, then $x \le b_{k+1}$ by Lemma C.1. By applying Speedup Rule 2, one obtains
$$\begin{aligned} A &= (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^{x})^{\max\{x,b_{k+1}\}}(Q_{k+2}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}] \\ &= (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^{x})^{b_{k+1}}(Q_{k+2}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}]. \end{aligned}$$
   If we instead apply Speedup Rule 1 with $x' = x$, the class is
$$\begin{aligned} B &= (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{\max\{a_k,x'\}})^{\max\{x',b_{k+1}\}}(Q_{k+1}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x'}] \\ &= (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}]. \end{aligned}$$
   Then by applying Speedup Rule 1 with $x' = 0$, the above class is in
$$(Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^0)^{b_{k+1}}(Q_{k+2}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}].$$
   It is clear that $B \subseteq A$: every parameter in B is at most the corresponding parameter in A. Thus any inclusion derived with Rule 2 could only be made stronger by applying Rule 1 twice instead.

2. If $x \ge a_k$, then Speedup Rule 2 gives
$$A = (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^{x})^{\max\{x,b_{k+1}\}}(Q_{k+2}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}].$$
   Speedup Rule 1 with $x' = a_k$ gives
$$\begin{aligned} B &= (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{\max\{a_k,x'\}})^{\max\{x',b_{k+1}\}}(Q_{k+1}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x'}]. \\ &= (Q_1\ n^{a_1})^{b_2}(Q_2\ n^{a_2})\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-a_k}]. \end{aligned}$$
   where we used the fact that $x' = a_k \le b_{k+1}$ (Lemma C.1). Applying Speedup Rule 1 again with $x' = x - a_k$, B is contained in
$$(Q_1\ n^{a_1})^{b_2}\cdots^{b_k}(Q_k\ n^{a_k})^{b_{k+1}}(Q_{k+1}\ n^{\max\{x-a_k,1\}})^{\max\{x-a_k,b_{k+1}\}}(Q_{k+2}\ n^0)^{b_{k+1}}\mathsf{DTS}[n^{a_{k+1}-x}].$$
   Again, observe $B \subseteq A$ in this case, and every parameter in B is at most the corresponding parameter in A.

This completes the proof. □

As a consequence, Speedup Rule 2 is not necessary for normal form proofs.

Theorem C.1 *For every alternation-trading proof of* $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c] \implies A_1 \subseteq A_2$ *in normal form, there is another alternation-trading proof of the same that does not use Speedup Rule 2.*

Proof. By Lemma C.1, every normal form alternation-trading proof is orderly. So by Lemma C.2, there is an equivalent alternation-trading proof that does not use Speedup Rule 2. □

D Proofs of Normal Form Properties

Here we establish Lemma 3.1 and Theorem 3.1, which show that it suffices to consider alternation-trading proofs written in normal form:

Lemma 3.1 *Let $c \geq 1$. If there is an alternation-trading proof for c in normal form having at least two lines, then* $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$.

Proof. Let P be an alternation-trading proof for c in normal form. We consider two cases.

- Suppose $a > a'$. In this case, $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ implies $\mathsf{DTS}[n^a] \subseteq \mathsf{DTS}[n^{a-\delta}]$ for some $\delta > 0$. By translation, $\mathsf{DTS}[n^a] \subseteq \mathsf{DTS}[n^{a-\delta}]$ implies

  $$\mathsf{DTS}[n^{a^2/(a-\delta)}] \subseteq \mathsf{DTS}[n^a] \subseteq \mathsf{DTS}[n^{a-\delta}],$$

  and $\mathsf{DTS}[n^{a\cdot(a/(a-\delta))^i}] \subseteq \mathsf{DTS}[n^{a-\delta}]$ for all $i \geq 0$. Since $\delta > 0$, this implies $\mathsf{DTS}[n^L] \subseteq \mathsf{DTS}[n^{a-\delta}]$ for all $L \geq a - \delta$. Therefore, if $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ then for all $L \geq a$,

  $$\mathsf{NTIME}[n^L] \subseteq \mathsf{DTS}[n^{Lc}] \subseteq \mathsf{DTS}[n^{a-\delta}] \subseteq \mathsf{coNTIME}[n^{a-\delta}],$$

  a contradiction to the time hierarchy (Theorem A.2).

- Suppose $a = a'$. Let A be a line in P with a positive number of alternations. (Such a line must exist since P has at least two lines.) The proof P proves that $\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c]$ implies $\mathsf{DTS}[n^a] \subseteq A \subseteq \mathsf{DTS}[n^{a'}]$, so $A = \mathsf{DTS}[n^a]$.

  Since $\mathsf{DTS}[n^a]$ is closed under complement,

  $$A = A', \tag{7}$$

  where A' is the complement of A. Without loss of generality, assume $A = (\exists\, n^\delta)B$ and $A' = (\forall\, n^\delta)B'$ for some $\delta > 0$ and complementary classes B and B'. Clearly,

  $$A' = (\forall\, n^\delta)A' \text{ and } A = (\exists\, n^\delta)A. \tag{8}$$

  Now consider the class $\mathsf{DTS}[n^{\delta\lceil \frac{k}{\delta}\rceil}] \supseteq \mathsf{DTS}[n^k]$, for arbitrary $k \geq 1$. By the Speedup Lemma (Lemma A.1) and the fact that $\mathsf{DTS}[n^\varepsilon] \subseteq A'$ for some $\varepsilon > 0$,

  $$\mathsf{DTS}[n^k] \subseteq \mathsf{DTS}[n^{\delta\lceil \frac{k}{\delta}\rceil}] \subseteq \underbrace{(\exists\, n^\delta)(\forall\, n^\delta)\cdots(\exists\, n^\delta)(\forall\, n^\delta)}_{\lceil k/\delta\rceil} A'.$$

  Applying equations (7) and (8), we have

  $$\begin{aligned}
  &(\exists\, n^\delta)(\forall\, n^\delta)\cdots(\exists\, n^\delta)(\forall\, n^\delta)A' \\
  &= (\exists\, n^\delta)(\forall\, n^\delta)\cdots(\exists\, n^\delta)A' \\
  &= (\exists\, n^\delta)(\forall\, n^\delta)\cdots(\exists\, n^\delta)A \\
  &= (\exists\, n^\delta)(\forall\, n^\delta)\cdots A \\
  &= \cdots = (\exists\, n^\delta)(\forall\, n^\delta)A' = (\exists\, n^\delta)A' = (\exists\, n^\delta)A = A.
  \end{aligned}$$

Therefore $\mathsf{DTS}[n^k] \subseteq A$, for *every* $k \geq 1$. Hence $\mathsf{NP} \subseteq \mathsf{DTS}[n^{O(1)}, n^{o(1)}] \subseteq A$. But by applying a slowdown step for a finite number of times to A, there is an alternation-trading proof that $A \subseteq \mathsf{DTS}[n^K]$ for a constant K. It follows that $\mathsf{NP} \subseteq A \subseteq \mathsf{DTS}[n^K] \subseteq \mathsf{coNTIME}[n^K]$, contradicting the time hierarchy (Theorem A.2). So $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$ in this case as well.

□

Theorem 3.1 *Let A_1 and A_2 be complementary. If there is an alternation-trading proof P for c that shows* $(\mathsf{NTIME}[n] \subseteq \mathsf{DTS}[n^c] \Longrightarrow A_1 \subseteq A_2)$*, then there is a normal form proof for c, of length at most that of P.*

Proof. Consider an alternation-trading proof P for c, written as

$$P = A_1, C_1, \ldots, C_k, A_2.$$

Define the *dual proof P'* by

$$P' = A_2, \neg C_1, \ldots, \neg C_k, A_1,$$

where the notation $\neg C$ denotes the unique complementary simple class for C, *i.e.* every '$\forall$' in C is replaced with '$\exists$', and vice-versa. Note that P' is an alternation-trading proof if and only if P is one.

Since the quantifiers of the first and last line of P are different, note there there must be a line $C_i = \mathsf{DTS}[n^a]$ for some a.

- Suppose there is only one deterministic class in P; call it C_i. Then

$$P'' = C_i, C_{i+1}, \ldots C_k, A_2, \neg C_1, \ldots, \neg C_i$$

  is also an alternation-trading proof, obtained by piecing together the appropriate lines from P and P'. However, $C_i = \neg C_i$, since $\mathsf{DTS}[n^a]$ is closed under complement. Hence P'' is in normal form: its first and last lines are DTS classes, and no intermediate class is a DTS class.

- Suppose there are $k \geq 2$ different DTS classes in P. Write P as

$$P = A_1, \ldots, \mathsf{DTS}[n^{a_1}], \ldots, \mathsf{DTS}[n^{a_2}], \ldots, \ldots, \mathsf{DTS}[n^{a_k}], \ldots, A_2.$$

  There are two cases:

  - If there is an $i \in [k]$ satisfying $a_i \geq a_{i+1}$, we are done: let P'' to be the sequence of lines from $\mathsf{DTS}[n^{a_i}]$ and $\mathsf{DTS}[n^{a_{i+1}}]$, and this is in normal form.
  - If $a_i < a_{i+1}$ for every i, then set $P'' = \mathsf{DTS}[n^{a_k}], \ldots, A_2, \ldots, \mathsf{DTS}[n^{a_1}]$, where the classes in the first "…" in P'' are taken directly from P, and the classes in the second "…" in P'' are obtained by taking the lines $A_2, \ldots, \mathsf{DTS}[n^{a_1}]$ in P'. P'' is in normal form since $a_k > a_1$.

□

E Experimental Results: Time-Space Lower Bounds for SAT

We wrote a program that given a proof annotation generates the relevant linear programming instance, solves it, then prints the proof in human-readable form. For proof annotations exceeding 100 lines, we used the `lp_solve` package to solve the corresponding linear program.[9] We also wrote heuristic search routines that try to derive new proofs from old ones. One program starts with a queue of annotations, pulls the head of the queue, and then tries all possible ways to add at most four bits to the annotation. If the resulting lower bound from the new annotation increases, the new annotation is added to the queue. Interestingly, this simple strategy generated all the optimal lower bounds that were found by exhaustive search, and more.

First we verified all the previously known lower bounds, such as the $n^{2\cos(\pi/7)}$ bound. In some cases, we found better settings of the parameters than had been found in the past, but no proof better than $n^{2\cos(\pi/7)}$. Then we searched the space of proof annotations, looking for interesting patterns. For all even $k = 2, \ldots, 26$, we exhaustively searched over all valid proof annotations with k lines. The best proof annotations for each k are given in the below table. For $k > 26$ we have not exhaustively searched all proofs, but instead used a heuristic search as described above; these rows of the table are marked with an asterisk. For rows with multiple annotations, we checked the annotations to two more decimal places to further verify that the obtained lower bounds are likely the same. The Δ of a row is the difference between the exponent of that row and the exponent of the previous row.

#Lines	Best Proof Annotation(s)	L.B.	Δ
4	$[1,0,0]$	1.4142	0
6	$[1,0,1,0,0]$	1.5213	0.1071
	$[1,1,0,0,0]$		
8	$[1,1,0,0,1,0,0]$	1.6004	0.0791
10	$[1,1,0,0,1,0,1,0,0]$	1.633315	0.032915
	$[1,1,0,1,0,0,1,0,0]$		
	$[1,1,1,0,0,0,1,0,0]$		
12	$[1,1,1,0,0,1,0,0,1,0,0]$	1.6635	0.0302
14	$[1,1,1,0,0,1,0,0,1,0,1,0,0]$	1.6871	0.0236
16	$[1,1,1,0,0,1,0,1,0,0,1,0,1,0,0]$	1.699676	0.012576
	$[1,1,1,0,1,0,0,1,0,0,1,0,1,0,0]$		
	$[1,1,1,1,0,0,0,1,0,0,1,0,1,0,0]$		
18	$[1,1,1,1,0,0,1,0,0,1,0,0,1,0,1,0,0]$	1.7121	0.0125
20	$[1,1,1,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,0]$	1.7232	0.0111
22	$[1,1,1,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0]$	1.7322	0.0090
24	$[1,1,1,1,0,0,1,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0]$	1.737851	0.005651
	$[1,1,1,1,0,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0]$		
	$[1,1,1,1,1,0,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0]$		
26	$[1,1,1,1,1,0,0,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0]$	1.7437	0.005849
28*	$[1,1,1,1,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0]$	1.7491	0.0054
30*	$[1,1,1,1,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0,1,0,1,0,1,0,0]$	1.7537	0.0046
32*	$[1,1,1,1,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,0,0]$	1.7577	0.0040
34*	$[1,1,1,1,1,0,0,1,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,0,0]$	1.760632	0.002932
	$[1,1,1,1,1,0,1,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,0,0]$		
	$[1,1,1,1,1,1,0,0,0,1,0,0,1,0,1,0,0,1,0,1,0,1,0,0,1,0,1,0,1,0,1,0,0]$		

We observe that the proofs produced by the annotations in the table have strong similarities to those in the $2\cos(\pi/7)$ lower bound. For example, the best 14-line proof (proving an $\Omega(n^{1.6871})$ lower bound) looks like:

```
0, DTS[n^5.275587925]
1, (E n^1.853485593)(A n^1.)DTS[n^3.422102331]
2, (E n^1.853485593)(A n^1.422102331)(E n^1.)DTS[n^2.000000001]
```

[9]The `lp_solve` package is an open source simplex-based linear programming solver. It is maintained by a community on Yahoo Groups: `http://groups.yahoo.com/group/lp_solve`.

```
3, (E n^1.853485593)(A n^1.422102331)(E n^1.000000001)(A n^1.000000000)DTS[n^1.]
4, (E n^1.853485593)(A n^1.422102331)(E n^1.000000001)DTS[n^1.687100000]
5, (E n^1.853485593)(A n^1.422102331)DTS[n^2.846306408]
6, (E n^1.853485593)(A n^1.423153204)(E n^1.000000000)DTS[n^1.423153204]
7, (E n^1.853485593)(A n^1.423153204)DTS[n^2.401001771]
8, (E n^1.853485593)DTS[n^4.050730087]
9, (E n^1.853485593)(A n^1.000000000)DTS[n^2.197244494]
10, (E n^1.853485593)DTS[n^3.706971186]
11, (E n^1.853485593)(A n^1.000000000)DTS[n^1.853485593]
12, (E n^1.853485593)DTS[n^3.127015544]
13, DTS[n^5.275587925]
```

Looking closely at the table, there is a strong correlation between later rows of the table and earlier ones. For example, there is a tie for best annotation at 10, 16, 24, and 34 lines, among three annotations that differ only in three of their bits. To develop a greater understanding of what is happening, let us introduce some abbreviations in the annotation. Where an annotation contains the string $(1\ 0)^k\ 0$, we put the symbol **k**, for $k \geq 1$. Where an annotation contains the string 11000, we just put **0**. The following table emerges:

#Lines	Best Proof Annotation(s)	L.B.	Δ
4	**1**	1.4142	0
6	**2**	1.5213	0.1071
	0		
8	1 **2**	1.6004	0.0791
10	1 **1 2**	1.633315	0.032915
	1 **2 1**		
	1 **0 1**		
12	1 1 **1 1 1**	1.6635	0.0302
14	1 1 **1 1 2**	1.6871	0.0236
16	1 1 **1 2 2**	1.699676	0.012576
	1 1 **2 1 2**		
	1 1 **0 1 2**		
18	1 1 1 **1 1 1 2**	1.7121	0.0125
20	1 1 1 **1 1 2 2**	1.7232	0.0111
22	1 1 1 **1 1 2 3**	1.7322	0.0090
24	1 1 1 **1 2 2 3**	1.737851	0.005651
	1 1 1 **2 1 2 3**		
	1 1 1 **0 1 2 3**		
26	1 1 1 1 **1 1 1 2 3**	1.7437	0.005849
28*	1 1 1 1 **1 1 2 2 3**	1.7491	0.0054
30*	1 1 1 1 **1 1 2 3 3**	1.7537	0.0046
32*	1 1 1 1 **1 1 2 3 4**	1.7577	0.0040
34*	1 1 1 1 **1 2 2 3 4**	1.760632	0.002932
	1 1 1 1 **2 1 2 3 4**		
	1 1 1 1 **0 1 2 3 4**		

For an optimal annotation that ends with a non-zero **k**, a longer optimal annotation can be obtained by adding either a **k** or **k+1** to the end, and a 1 at the beginning. (There are of course some restrictions– there are no more than three consecutive **1**'s, no more than two consecutive **2**'s, *etc.*) While we do not yet know how to *prove* that all of the best proofs must have this behavior, it seems extraordinarily unlikely that this pattern deviates at some later point.

The table suggests that we examine proof annotations of the form $1 \cdots 1\,\mathbf{0\ 1\ 2\ 3\ 4}\cdots$. Unfortunately these annotations do not lead to an improvement. To illustrate, for the 424 line proof annotation denoted by

$$1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ \mathbf{0\ 1\ 2\ 3\ 4}\ \cdots\ \mathbf{17\ 18\ 19},$$

experiments with `lp_solve` revealed that the optimal exponent is only in the interval $[1.80175, 1.8018)$. These results (along with the fact that any annotation in the above form *provably* can yield no better than a $2\cos(\pi/7)$ exponent, cf. Theorem 3.3) point strongly to the conjecture that there is no alternation-trading proof that $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^c]$, for any $c > 2\cos(\pi/7) \approx 1.8019$.

F Proof of Theorem 3.4: No Quadratic Lower Bound

In their paper showing that SAT cannot be solved in $O(n^{\phi})$ time and $n^{o(1)}$ space, Fortnow *et al.* [FLvMV05] write that "some complexity theorists feel that improving the golden ratio exponent beyond 2 would require a breakthrough." Here we give a formal proof of this sentiment. Although the proof is simple, we believe it is important as a formalization of a folklore conjecture.

Theorem 3.4 *There is no alternation-trading proof of* $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTS}[n^2]$.

Proof. (Sketch) Suppose there is such a proof, and let A be a minimum length annotation in normal form for it. We claim that A can be made shorter, yet the resulting LP is still feasible if the original LP was feasible.

First we observe that every normal form annotation contains a sequence $1, 0$. Normal form annotations can be put in 1-1 correspondence with strings of balanced parentheses of the form `(`x`)`, where x is an non-empty balanced parentheses string. The first speedup in a proof corresponds to `((`, as it introduces two quantifiers, all other speedup applications correspond to a `(`, and a slowdown correponds to a `)`. For example, `(())` corresponds to $[1, 0, 0]$. Since there is always an adjacent parentheses pair `()` in any string of balanced parentheses, there must also be some occurrence of $1, 0$ in a valid proof annotation. If this $1, 0$ can be removed from A without changing the feasibility of the underlying linear program, the claim is proved.

The two lines in the proof corresponding to the sequence $1, 0$ (including the previous line) have the form:

$$\cdots^{b_{k-1}} (Q_{k-1}\ n^{a_{k-1}})^{b_k} (Q_k\ n^{a_k})^{b_{k+1}} \mathsf{DTS}[n^{a_{k+1}}] \tag{9}$$

$$\cdots^{b_{k-1}} (Q_{k-1}\ n^{a_{k-1}})^{b_k} (Q_k\ n^{\max\{x,a_k\}})^{\max\{x,b_{k+1}\}} (Q_{k+1}\ \log n^0)^{b_{k+1}} \mathsf{DTS}[n^{a_{k+1}-x}] \tag{10}$$

$$\cdots^{b_{k-1}} (Q_{k-1}\ n^{a_{k-1}})^{b_k} (Q_k\ n^{\max\{x,a_k\}})^{\max\{x,b_{k+1}\}} \mathsf{DTS}[n^{\max\{c(a_{k+1}-x), cx, cb_{k+1}\}}] \tag{11}$$

Every parameter in the class (11) is at least the corresponding parameter in the class (9), except for possibly the runtime of the DTS computation. Hence if $a_{k+1} \le c(a_{k+1} - x)$, or $a_{k+1} \le cx$, or $a_{k+1} \le cb_{k+1}$, then $1, 0$ could be removed without changing the feasibility of the LP. However, if both $a_{k+1} > c(a_{k+1} - x)$ and $a_{k+1} > cx$, then $2a_{k+1} > c(a_{k+1} - x) + cx$, a contradiction when $c \ge 2$. □

G Experimental Results: Lower Bounds for Nondeterministic Algorithms Solving Tautologies

Below is a table of results found by exhaustive search over valid annotations.

#Lines	Best Proof Annotation(s)	L.B.
5	[1, 0, 0, 0]	1.323
8	[1, 1, 0, 0, 0, 0, 0]	1.380
	[1, 0, 1, 0, 0, 0, 0]	
11	[1, 1, 0, 0, 0, 1, 0, 0, 0, 0]	1.419
14	[1, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]	1.433
	[1, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0]	
	[1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0]	
	[1, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0]	
17	[1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0]	1.445
	[1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0]	
20	[1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0]	1.455
	[1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0]	
	[1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0]	
	[1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0]	
23	[1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0]	1.465

As one can see, the structure of good lower bound proofs for the "TAUTOLOGY versus NTISP" problem turns out to be different from those for the "SAT versus DTISP" problem. Taken from the program, the best 11-line proof reads:

```
0, NTS[n^4.788956584]
1, (E n^2.369956583)(A n^1.)NTS[n^2.419]
2, (E n^2.369956583)(A n^1.)(E n^1.419)(A n^1.)NTS[n^1.]
3, (E n^2.369956583)(A n^1.)(E n^1.419)(A n^1.)coNTS[n^1.419]
4, (E n^2.369956583)(A n^1.)(E n^1.419)NTS[n^2.013561000]
5, (E n^2.369956583)(A n^1.)coNTS[n^2.857243059]
6, (E n^2.369956583)(A n^2.378351877)(E n^1.)coNTS[n^1.181167036]
7, (E n^2.369956583)(A n^2.378351877)(E n^1.)NTS[n^1.676076023]
8, (E n^2.369956583)(A n^2.378351877)coNTS[n^2.378351877]
9, (E n^2.369956583)NTS[n^3.374881314]
10, coNTS[n^4.788956584]
```

Note how larger annotations are composed of smaller ones: for example, $[1,1,1,0,0,0,0,0,1,0,0,0,0]$ is $[1, A_7, A_4, 0]$, where A_4 and A_7 are optimal annotations for four and seven lines. In particular, observe that three optimal annotations from the table have a distinctive pattern, namely

$$[1,0,0,0], [1,1,0,0,0,1,0,0,0,0], \text{ and } [1,1,1,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,0,0].$$

The pattern suggests that we look for an inductive proof where an induction hypothesis is applied twice in the inductive step. The next annotation in the pattern would be

$$\begin{gathered}[1,1,1,1,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,0,0,\\ 1,1,1,0,0,0,1,0,0,0,0,1,1,0,0,0,1,0,0,0,0,0,0],\end{gathered}$$

a 47-line annotation that gives a 1.49 exponent. A heuristic search found the above 47-line annotation, and no other annotations found (with at most 47 lines) attained a lower bound of that quality. For many line numbers ℓ, heuristic search found a large number of ℓ-line proof annotations that achieve the same lower bound. For example, there are eight such annotations of 26 lines. Each optimal annotation found could be written as a concatenation of smaller optimal annotations along with an additional 1 and 0.

H Proof of Theorem 4.2: The Nondeterministic Time-Space Lower Bound

To prove Theorem 4.2, we use an inductive lemma. For a given constant $c \geq 1$, define the sequence $x_1 := 1$, $x_2 := c$, $x_k := c^3(x_{k-1})^2/(\sum_{i=1}^{k-1} x_i)$.

Lemma H.1 *If* $\mathsf{coNTIME}[n] \subseteq \mathsf{NTS}[n^c]$ *and* $c^2(x_k)^2 \geq \sum_{i=1}^{k} x_i$, *then for all* $k \geq 2$, $\mathsf{NTS}[n^{\sum_{i=1}^{k} x_i}] \subseteq (\exists\, n^{x_k})(\forall\, n^{x_k})\mathsf{coNTS}[n^{x_k}]$.

In the following, we do not specify the inputs to quantifier blocks, except where absolutely necessary for the argument.

Proof. For $k = 2$ we have $\mathsf{NTS}[n^{c+1}] \subseteq (\exists\, n^c)(\forall\, \log n)\mathsf{NTS}[n] \subseteq (\exists\, n^c)(\forall\, \log n)\mathsf{coNTS}[n^c]$. Note that $c^2(x_k)^2 \geq \sum_{i=1}^{k} x_i$ implies $x_{k+1} \geq 1$. By induction we have that

$$\begin{aligned}
\mathsf{NTS}[n^{\sum_{i=1}^{k+1} x_i}] &\subseteq (\exists\, n^{x_{k+1}})(\forall\, \log n)^1\, \mathsf{NTS}[n^{\sum_{i=1}^{k} x_i}] \text{ (Speedup)}\\
&\subseteq (\exists\, n^{x_{k+1}})(\forall\, \log n)^1(\exists\, n^{x_k})(\forall\, n^{x_k})\mathsf{coNTS}[n^{x_k}] \text{ (by Induction Hypothesis)}\\
&\subseteq (\exists\, n^{x_{k+1}})(\forall\, \log n)^1(\exists\, n^{x_k})\mathsf{NTS}[n^{cx_k}] \text{ (Slowdown)}\\
&\subseteq (\exists\, n^{x_{k+1}})(\forall\, \log n)^1\mathsf{coNTS}[n^{c^2 x_k}] \text{ (Slowdown)}\\
&\subseteq (\exists\, n^{x_{k+1}})(\forall\, \log n)^1(\forall\, n^{c^2(x_k)^2/(\sum_i x_i)})(\exists\, n^{c^2(x_k)^2/(\sum_i x_i)})\mathsf{NTS}[n^{c^2(x_k)^2/(\sum_i x_i)}]\\
&\qquad\qquad \text{(Induction Hypothesis, and Assumption)}\\
&\subseteq (\exists\, n^{x_{k+1}})(\forall\, \log n)^1(\forall\, n^{c^2(x_k)^2/(\sum_i x_i)})\mathsf{coNTS}[n^{c^3(x_k)^2/(\sum_i x_i)}] \text{ (Slowdown)}\\
&\subseteq (\exists\, n^{x_{k+1}})(\forall\, n^{x_{k+1}})\mathsf{coNTS}[n^{x_{k+1}}].
\end{aligned}$$

□

Note the lemma indeed applies its induction hypothesis twice, as suggested by experiments. The lower bound proof for tautologies can now be derived.

Proof of Theorem 4.2. Assume $\mathsf{coNTIME}[n] \subseteq \mathsf{NTS}[n^c]$. Suppose ℓ is the smallest integer satisfying $c^2(x_\ell)^2 < \sum_{i=1}^{\ell} x_i$. Note that $c^2(x_2)^2 = c^4 \geq c + 1 = x_1 + x_2$ for $c > 1.2207$, which we know holds due to Fortnow and Van Melkebeek. Therefore $\ell \geq 3$. By Lemma H.1 and the Slowdown Lemma, for every $k < \ell$ we have

$$\mathsf{NTS}[n^{\sum_{i=1}^{k} x_i}] \subseteq (\exists\, n^{x_k})(\forall\, n^{x_k})\mathsf{coNTS}[n^{x_k}] \subseteq \mathsf{coNTS}[n^{c^2 x_k}]. \tag{12}$$

Define the sequence $s_k := (\sum_{i=1}^{k} x_i)/x_k = 1 + \sum_{i=1}^{k-1} x_i/x_k$. By induction one can show that $s_k = 1 + (s_{k-1})^2/c^3$, and that this sequence is increasing.

The inclusion (12) says that a contradiction is obtained with Lemma 4.3 when $c^2 \leq s_k$. Hence if $c^2 \leq s_{\ell-1}$, we have a contradiction with Lemma 4.3 (the NTS versus coNTS hierarchy). However, we know that $c^2 x_\ell < \sum_{i=1}^{\ell} x_i/x_\ell$ and $c^2 x_{\ell-1} \geq \sum_{i=1}^{\ell-1} x_i/x_{\ell-1}$, by our choice of ℓ. Taken together, the two inequalities are equivalent to the condition $x_{\ell+1} < c \leq x_\ell$. With algebra manipulation and the fact that $c^3 > c + 1$ (which holds for $c > 1.33$), one can show that this condition implies $c^2 \leq s_{\ell-1}$. Hence no such ℓ exists.

Now suppose instead that $c^2(x_k)^2 \geq \sum_{i=1}^{k} x_i$, for all k. Then inclusion (12) holds for all k. For which c can we obtain $c^2 \leq s_k$, for sufficiently large k? Either the sequence $\{s_k\}$ is unbounded (in which case we are done, as the inequality holds for all c) or it has a limit point. In the latter case, we have $s_\infty = 1 + s_\infty^2/c^3$. The polynomial $p(x) = 1 + x^2/c^3 - x$ has roots $x = c \cdot (c^2/2 \pm \sqrt{(c^4 - 4c)/2})$. When $c = 4^{1/3}$, this root is imaginary, therefore s_∞ would be imaginary, a contradiction. It follows that $c < 4^{1/3}$. □

I No Golden Ratio Lower Bound for Solving Tautologies With Nondeterminism

Here we prove that there is no alternation trading proof that $\mathsf{coNTIME}[n] \not\subseteq \mathsf{NTS}[n^c]$, for any $c \geq \phi \approx 1.618$, the golden ratio.

Proof. (Sketch) Suppose there is a proof that $\mathsf{coNTIME}[n] \not\subseteq \mathsf{NTS}[n^c]$ with $c \geq \phi$, and let A be a normal form annotation for it, of minimum length. First, observe that every valid annotation contains the sequence $1, 0, 0$ in it, for if every occurrence of $1, 0$ was followed by a 1, the proof could not possibly be in normal form. (In particular, when a speedup rule and slowdown rule are applied to a simple class A, the resulting class A' has *more* quantifiers than A, in this setting.) Therefore $1, 0, 0$ must occur somewhere in the proof.

Next, we show that any subsequence $1, 0, 0$ can be removed from A, and the resulting LP will still be feasible for the constant c. This implies a contradiction.

Consider the four lines in a prospective proof corresponding to the sequence $1, 0, 0$, where we include the line before the three rules are applied. The first line is one of four possibilities:

$$\cdots^{b'} (\exists n^{a'})^b \mathsf{NTS}[n^a], \quad \cdots^{b'} (\forall n^{a'})^b \mathsf{coNTS}[n^a], \quad \cdots^{b'} (\exists n^{a'})^b \mathsf{coNTS}[n^a], \text{ or } \cdots^{b'} (\forall n^{a'})^b \mathsf{NTS}[n^a].$$

The first two cases are symmetric to each other, as are the last two cases, so it suffices for us to consider $\cdots^{b'} (\exists n^{a'})^b \mathsf{NTS}[n^a]$ and $\cdots^{b'} (\exists n^{a'})^b \mathsf{coNTS}[n^a]$.

In the first case, the four lines have the form:

$$\cdots^{b'} (\exists n^{a'})^b \mathsf{NTS}[n^a] \tag{13}$$
$$\cdots^{b'} (\exists\, n^{\max\{a',x\}})^{\max\{b,x\}} (\forall\, \log n)^b \mathsf{NTS}[n^{a-x}] \tag{14}$$
$$\cdots^{b'} (\exists\, n^{\max\{a',x\}})^{\max\{b,x\}} (\forall\, \log n)^b \mathsf{coNTS}[n^{\max\{c(a-x),cb\}}] \tag{15}$$
$$\cdots^{b'} (\exists\, n^{\max\{a',x\}})^{\max\{b,x\}} \mathsf{NTS}[n^{\max\{c^2(a-x),c^2 b,cx\}}] \tag{16}$$

Observe that each parameter in class (16) is at least the corresponding parameter in class (13), except for possibly the runtime of the NTS computation. However, if any one of $a \leq c^2(a-x)$, $a \leq c^2 b$, or $a \leq cx$ hold, then the above lines can be removed from the proof, and the optimal assignment to the parameters would only be larger. So suppose $a > c^2(a-x)$, $a > c^b$, and $a > cx$. Then $c^2 a < a + c^2 x < a + ca$, implying that $c^2 < (1+c)$, or $c(c-1) < 1$. For $c \geq \phi$, this is a contradiction.

One can argue similarly for the second case. There the four lines have the form:

$$\cdots^{b'} (\exists n^{a'})^b \mathsf{coNTS}[n^a] \tag{17}$$
$$\cdots^{b'} (\exists\, n^{a'})^b (\forall\, n^x)^{\max\{x,b\}} (\exists\, \log n)^b \mathsf{coNTS}[n^{a-x}] \tag{18}$$
$$\cdots^{b'} (\exists\, n^{a'})^b (\forall\, n^x)^{\max\{x,b\}} (\exists\, \log n)^b \mathsf{NTS}[n^{\max\{c(a-x),cb\}}] \tag{19}$$
$$(\exists\, n^{a'})^b (\forall\, n^x)^{\max\{x,b\}} \mathsf{coNTS}[n^{\max\{c^2(a-x),c^2 b,cx\}}] \tag{20}$$

Using an argument similar to the above, class (20) contains class (17) when $c \geq \phi$, so removing the above lines can only improve the optimum setting of the parameters. $\square$

J Experimental Results: Lower Bounds for Multidimensional TMs Solving SAT

For 1-dimensional machines, a summary of lower bounds found by the LP-based theorem prover is given in the below table. Unlike the previous two cases, the optimal bounds attained by optimal proofs have non-monotonic behavior (with respect to length) at first. Perhaps surprisingly, the table looks the same for the 2-dimensional and 3-dimensional cases, albeit with smaller lower bound exponents.

#Lines	Best Proof Annotation(s)	L.B.
5	[1, 1, 0, 0]	1.224
6	[1, 1, 0, 1, 0]	
7	[1, 1, 1, 0, 0, 0]	1.201
8,9	[1, 1, 0, 1, 1, 0, 0], [1, 1, 0, 1, 1, 0, 1, 0]	1.262
10	[1, 1, 1, 0, 0, 1, 1, 0, 0]	1.261
11, 12	[1, 1, 0, 1, 1, 0, 1, 1, 0, 0], [1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0]	1.274
13	[1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0]	1.277
14, 15	[1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0],[1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0]	1.278
16, 17	[1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0], [1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 0]	1.287
19	[1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0]	1.292
25	[1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0]	1.297
28	[1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0]	1.298

A subset of the optimal annotations have the form

$$A = [1\ (1\ 1\ 0)^k\ 0\ (1\ 1\ 0)^\ell\ 0],$$

for integers k, ℓ. (In fact those that do not can be written in this way.) In other words, $0\ 0$ occurs exactly twice. Might it be that for longer proofs there are optimal annotations with three occurrences of $0\ 0$? As before, we used a heuristic search to investigate. The search uncovered more interesting annotations, but all of the best had the form of A above. For instance, the best 25 line proof was:

```
0, DTIME1[n^1.751958454]
1, (E n^1.)DTISP[n^1.751958454,n^.7519608720]
2, (E n^1.040108911)(A n^1.)DTISP[n^1.463810415,n^.7519608720]
3, (E n^1.040108911)(A n^1.)(E n^1.)DTISP[n^1.215771287,n^.7519608720]
4, (E n^1.040108911)(A n^1.)DTIME1[n^1.577881470]
5, (E n^1.040108911)(A n^1.)DTISP[n^1.577881470,n^.5778814720]
6, (E n^1.040108911)(A n^1.)(E n^1.)DTISP[n^1.155762944,n^.5778814720]
7, (E n^1.040108911)(A n^1.)DTIME1[n^1.5]
8, (E n^1.040108911)(A n^1.)DTISP[n^1.5,n^.5]
9, (E n^1.040108911)(A n^1.)(E n^1.)DTISP[n^1.,n^.5]
10, (E n^1.040108911)(A n^1.)DTIME1[n^1.297844000]
11, (E n^1.040108911)DTIME1[n^1.684399048]
12, (E n^1.040108909)DTISP[n^1.684399048,n^.6442901394]
13, (E n^1.040108909)(A n^1.)DTISP[n^1.288580278,n^.6442901394]
14, (E n^1.040108909)DTIME1[n^1.672376183]
15, (E n^1.040108909)DTISP[n^1.672376183,n^.6322672739]
16, (E n^1.040108909)(A n^1.)DTISP[n^1.264534548,n^.6322672739]
```

```
17, (E n^1.040108909)DTIME1[n^1.641168576]
18, (E n^1.040108909)DTISP[n^1.641168576,n^.6010596669]
19, (E n^1.040108911)(A n^1.)DTISP[n^1.202119332,n^.6010596669]
20, (E n^1.040108911)DTIME1[n^1.560163362]
21, (E n^1.040108911)DTISP[n^1.560163362,n^.5200544533]
22, (E n^1.040108908)(A n^1.)DTISP[n^1.040108908,n^.5200544533]
23, (E n^1.040108908)DTIME1[n^1.349899105]
24, DTIME1[n^1.751958454]
```

K Proof of Theorem 5.1: The d-Dimensional TM Lower Bound

Here we prove that SAT cannot be solved by a Turing machine with random access to its input and sequential access to a d-dimensional tape, in $O(n^{r_d})$ time, where $r_d \geq 1$ is a root of the polynomial

$$p_d(x) = (2d+1)(d+1)^2x^5 - 2(d+1)(2d+1)x^4 - d^2x^3 + 2(d+1)x^2 - ((2d+1)(d+1)^2+1)x + d(d+1).$$

As corollaries, SAT $\notin \mathsf{DTIME}_1[n^{1.3009}]$, SAT $\notin \mathsf{DTIME}_2[n^{1.1887}]$, and SAT $\notin \mathsf{DTIME}_3[n^{1.1372}]$.

Before we prove Theorem 5.1, we first give an inductive lemma. Let $c \geq 1$, and define the sequence $e_1 := (d+2)/(d+1)$, $e_{k+1} := 1 + \frac{e_k}{c(d+1)}$.

Lemma K.1 *Suppose* $\mathsf{NTIME}[n] \subseteq \mathsf{DTIME}_d[n^c]$. *Then for all* $k \geq 1$,

$$\mathsf{DTIME}_d[n^{e_k}] \subseteq (\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n, n^{d/(d+1)}].$$

Proof. When $k = 1$,

$$\mathsf{DTIME}_d[n^{e_k}] \subseteq (\exists\, n)\mathsf{DTISP}[n^{(d+2)/(d+1)}, n^{.5}] \subseteq (\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n, n^{d/(d+1)}],$$

by Lemma 5.1 (DTIME_d to DTISP) and Lemma A.1 (DTISP Speedup), respectively. For the inductive step,

$$\begin{aligned}
\mathsf{DTIME}_d[n^{1+\frac{e_k}{c(d+1)}}] &\subseteq (\exists\, n)\mathsf{DTISP}[n^{1+\frac{e_k}{c(d+1)}}, n^{\frac{e_k}{c(d+1)}}] \quad (\mathsf{DTIME}_d \text{ to } \mathsf{DTISP})\\
&\subseteq (\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n^{e_k/c}, n^{\frac{e_k}{c(d+1)}}] \quad (\text{Speedup})\\
&\subseteq (\exists\, n)\mathsf{DTIME}_d[n^{e_k}] \quad (\mathsf{DTIME}_d \text{ Slowdown})\\
&\subseteq (\exists\, n)(\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n, n^{d/(d+1)}] = (\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n, n^{d/(d+1)}],
\end{aligned}$$

where the last containment holds by induction. □

Note the proof annotations for the derivations in the above lemma have the form $(1\ 1\ 0)^{k-1}\ 1\ 1$.

Corollary K.1 *For all* $\varepsilon > 0$ *and* $c \geq 1$, *if* $\mathsf{NTIME}[n] \subseteq \mathsf{DTIME}_d[n^c]$ *then* $\mathsf{DTIME}_d[n^{1+\frac{1}{c(d+1)-1}-\varepsilon}] \subseteq (\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n, n^{d/(d+1)}]$.

Proof. For $e < 1 + \frac{1}{c(d+1)-1}$, we have $e < 1 + \frac{e}{c(d+1)-1}$. The sequence $s_k = 1 + \frac{s_{k-1}}{c(d+1)-1}$ converges to $\eta = 1 + \frac{1}{c(d+1)-1}$ for all $c \geq 1$. (Note $e = 1 + e/(c(d+1))$ implies $e = 1 + \frac{1}{c(d+1)-1}$.) So for any $e^* < \eta$, by setting $e = (d+1)/(d+2)$ and observing $\mathsf{DTIME}_d[n^{(d+1)/(d+2)}] \subseteq (\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n, n^{d/(d+1)}]$, one can apply Lemma K.1 a constant number of times to get that the same containment holds for $\mathsf{DTIME}_d[n^{e^*}]$. □

Intuitively, the corollary says that as we make stronger assumptions about how quickly SAT can be solved on a d-dimensional one-tape TM, then we can place more of $\mathsf{DTIME}_d[n^{O(1)}]$ in $(\exists\, n)(\forall\, \log n)\mathsf{DTISP}[n, n^{d/(d+1)}]$, when $c < (d+2)/(d+1)$. We can now prove the lower bound.

Proof of Theorem 5.1. Let $a \geq 1$ be a parameter. Then

$$\begin{aligned} \mathsf{DTIME}_d[n^a] &\subseteq (\exists\, n)\mathsf{DTISP}[n^a, n^{d(a-1)}] \quad (\mathsf{DTIME}_d \text{ to } \mathsf{DTISP}) \\ &\subseteq (\exists\, n^{x+d(a-1)})^{x+d(a-1)}(\forall\, \log n)^1\mathsf{DTISP}[n^{a-x}, n^{d(a-1)}] \quad \text{(Speedup)} \end{aligned}$$

where x is a parameter satisfying $c \geq x + d(a-1) \geq 1$. By Speedup, the above class is in

$$\begin{aligned} & (\exists\, n^{x+d(a-1)})^{x+d(a-1)}(\forall\, n^{(1-d(a-1))+d(a-1)})^1(\exists\, \log n)^1\mathsf{DTISP}[n^{a-x-(1-d(a-1))}, n^{d(a-1)}] \\ = \;& (\exists\, n^{x+d(a-1)})^{x+d(a-1)}(\forall\, n^1)^1(\exists\, \log n)^1\mathsf{DTISP}[n^{a-x-(1-d(a-1))}, n^{d(a-1)}] \\ \subseteq \;& (\exists\, n^{x+d(a-1)})^{x+d(a-1)}(\forall\, n^1)^1\mathsf{DTIME}_d[n^{c(a-x-(1-d(a-1)))}] \quad \text{(Slowdown)}, \end{aligned}$$

assuming (for the moment) that $1 - d(a-1) \geq 0$. Suppose that a and x satisfy $c(a - x - (1 - d(a-1))) = c((d+1)(a-1) - x) \geq 1 + \frac{1}{c(d+1)-1} - \varepsilon$, for some $\varepsilon > 0$. Applying Corollary K.1, the above is contained in

$$\begin{aligned} (\exists\, n^{x+d(a-1)})(\forall\, n)(\exists\, \log n)\mathsf{DTISP}[n^1, n^{d/(d+1)}] &\subseteq (\exists\, n^{x+d(a-1)})(\forall\, n)\mathsf{DTISP}[n^c, n^{d/(d+1)}] \quad \text{(Slowdown)} \\ &\subseteq (\exists\, n^{x+d(a-1)})^{x+d(a-1)}\mathsf{DTIME}_d[n^{c^2}], \quad \text{(Slowdown)} \end{aligned}$$

since $c \geq x + d(a-1)$. Now suppose a and c satisfy $c^2/(x + d(a-1)) = 1 + 1/(c(d+1) - 1) - \varepsilon$. Then Corollary K.1 can be applied again, obtaining the class

$$\begin{aligned} (\exists\, n^{x+d(a-1)})(\forall\, n)\mathsf{DTISP}[n^{(x+d(a-1))}, n^{(x+d(a-1))\cdot\frac{d}{d+1}}] &\subseteq (\exists\, n^{x+d(a-1)})\mathsf{DTIME}_d[n^{c(x+d(a-1))}] \quad \text{(Slowdown)} \\ &\subseteq \mathsf{DTIME}_d[n^{c^2(x+d(a-1))}]. \quad \text{(Slowdown).} \end{aligned}$$

Setting $a = c^2(x + d(a-1))$ yields a contradiction with Lemma 5.3. Observe that a proof annotation for the above has the form $[1\ (1\ 1\ 0)^k\ 0\ (1\ 1\ 0)^\ell\ 0]$. The analysis introduced three parameters (c, a, x) along with three equations to satisfy:

$$a = c^2(x + d(a-1)),\; c(a - x - d(a-1) + 1) = 1 + \frac{1}{c(d+1)-1},\; \frac{c^2}{x + d(a-1)} = 1 + \frac{1}{c(d+1)-1}.$$

Adding the constraint that $c \geq 1$, a solution to this system also satisfies the constraints $c \geq x + d(a-1) \geq 1$ (and therefore $1 - d(a-1) \geq 0$) that arose in the analysis. With substantial algebraic manipulation, one can show that $c \geq 1$ satisfying the equations is the unique root (greater than 1) of the quintic

$$p_d(x) = (2d+1)(d+1)^2x^5 - 2(d+1)(2d+1)x^4 - d^2x^3 + 2(d+1)x^2 - ((2d+1)(d+1)^2+1)x + d(d+1).$$

For any $r < c$, we can find a, x, and $\varepsilon > 0$ satisfying $r(d(a-1) - x) = 1 + 1/(r(d+1) - 1) - \varepsilon$, $a = r^2(x + d(a-1))$, and $r^2/(x + d(a-1)) = 1 + 1/(r(d+1) - 1) - \varepsilon$. This completes the proof. □

L No $n^{1+1/(d+1)}$ Lower Bound for Solving SAT With d-Dimensional Machines

Proof. (Sketch) Proof by minimal counterexample. Suppose there is an alternation-trading proof that $\mathsf{NTIME}[n] \not\subseteq \mathsf{DTIME}_d[n^d]$ with $c \geq 1 + 1/(d+1)$, and let A be a normal form annotation. We may assume that $|A| > 4$, otherwise the only annotation is $[1, 1, 0, 0]$ which we know does not yield the result. First, we prove that in this setting, every sequence $0, 1, 0$ in a normal form annotation can be replaced with just $0, 0$. After a slowdown, the deterministic portion of a class is DTIME_d, therefore $0, 1, 0$ produces the lines:

$$\cdots^{b'} (\exists n^{a'})^b \mathsf{DTIME}_d[n^a] \tag{21}$$
$$\cdots^{b'} (\exists\, n^{a',a-s})^{\max\{b,a-s\}} \mathsf{DTISP}[n^a, n^{ds}] \tag{22}$$
$$\cdots^{b'} \mathsf{DTIME}_d[n^{\max\{cb,ca-cs,ca',ca,cds\}}], \tag{23}$$

but this gives no improvement over applying the sequence $0, 0$ since $ca - cs \leq ca$.

So without loss of generality, every 1 in an annotation can be assumed to occur adjacent to another 1. Once we have removed $0, 1, 0$, the proof annotations can be placed in 1-1 correspondence with strings of balanced parentheses of the form $\mathtt{(}x\mathtt{)}$, as in Theorem 3.4. Every run of k 1's corresponds to $k - 1$ open parentheses, and every 0 corresponds to a closed parenthesis. Hence there is a sequence $1, 1, 0, 0$ in a proof annotation, as this corresponds to the substring $\mathtt{())}$, which must occur in a string of the form $\mathtt{(}x\mathtt{)}$ where x is not the empty string.

Finally, we claim that if $c \geq (d + 2)/(d + 1)$, then $1, 1, 0, 0$ can be replaced with just 0, corresponding to $\mathtt{)}$. To prove this, one examines the outcome of four lines where two speedups and two slowdowns are applied, then argues that when $c \geq (d + 2)/(d + 1)$, the resulting constants are no better than the case where one slowdown is applied. There are two cases to analyze: one where the first line has a DTISP class and one where the first line has a DTIME_d class. The reasoning follows the style of Theorems 3.4 and I but is much more technical in nature, so we omit its derivation here. □

M Lower Bounds for QBF_k [10]

As first observed by Fortnow and Van Melkebeek [FvM00], the alternation-trading scheme for lower bounds against nondeterminism extends naturally to lower bounds against alternating computations. Since $\mathsf{AP} = \mathsf{PSPACE}$ [CKS81], it follows that $\mathsf{ATIME}[n] \not\subseteq \mathsf{DTISP}[n^k, n^{o(1)}]$ for *every* $k \geq 1$.[11] So we already have a polynomial time lower bound for the quantified Boolean formula problem (QBF) in the small space setting.

How large can lower bounds for quantified Boolean formulas be, when the number of quantifier blocks is a fixed constant? Define QBF_k to be the problem of solving a QBF with k quantifier blocks (*i.e.* deciding the truth of Σ_k and Π_k sentences in first-order Boolean logic). Building on Fortnow and Van Melkebeek [FvM00] who proved that QBF_k requires $\Omega(n^{k-\varepsilon})$ time on $n^{o(1)}$-space machines, we prove time lower bounds for QBF_k of the form $\Omega(n^{k+1-\varepsilon_k})$ on the same model, where $\{\varepsilon_k\}$ is a decreasing sequence such that $\lim_{k\to\infty} \varepsilon_k = 0$. We use the fact that QBF_k is "robustly complete" in the appropriate sense, then show $\Sigma_k\mathsf{TIME}[n] \not\subseteq \mathsf{DTS}[n^c]$ for certain $c > k$ by proving a series of class containments. Let us recall the completeness result of Fortnow *et al.*:

Theorem M.1 (Fortnow-Lipton-Van Melkebeek-Viglas [FLvMV05]) *For all $k \geq 1$, QBF_k is robustly complete for $\Sigma_k\mathsf{QL} \cup \Pi_k\mathsf{QL}$. In particular, there is a quasi-linear reduction from an arbitrary language in the class to QBF_k, where an arbitrary bit of the reduction can be computed in polylogarithmic time.*

We can modify the LP framework for SAT lower bounds to obtain a similar LP framework for QBF_k lower bounds: only the Slowdown Rule differs, as its application removes two quantifiers instead of just one. Doing

[10] A preliminary version of the work in this section was reported in the author's PhD thesis in 2007. Also cf. [vM07], Section 4.2, for an overview of the result.

[11] Otherwise, $\mathsf{SPACE}[n] \subseteq \mathsf{ATIME}[n^2] \subseteq \mathsf{SPACE}[n^{o(1)}]$, contradicting the space hierarchy theorem.

so, we wrote a program for proving QBF_k time lower bounds, which produced proofs that closely resembled the below argument.

Theorem M.2 *For all $k \geq 1$, QBF_k requires $\Omega(n^c)$ time on $n^{o(1)}$ space RAMs, where $c^3/k - c^2 - 2c + k < 0$.*

Note this result generalizes the $2\cos(\pi/7)$ lower bound for SAT. The remainder of this section proves Theorem M.2, which was partly inspired by some short proofs generated by our theorem prover.The main tool we use is the following.

Theorem M.3 (Conditional Speedup for the Polynomial Hierarchy) *If $\Sigma_k\mathsf{TIME}[n] \subseteq \mathsf{DTS}[n^c]$ for some $c > k$, then for all d satisfying $c \leq d < \frac{c}{c-k}$, $\mathsf{DTS}[n^d] \subseteq \Sigma_{k+1}\mathsf{TIME}[n^{1+o(1)}] \cap \Pi_{k+1}\mathsf{TIME}[n^{1+o(1)}]$.*

Proof. Similar to the proof of the Conditional Speedup Theorem in [Wil08]. We show that $\mathsf{DTS}[n^d] \subseteq \Sigma_{k+1}\mathsf{TIME}[n^{1+o(1)}] \cap \Pi_{k+1}\mathsf{TIME}[n^{1+o(1)}]$ implies $\mathsf{DTS}[n^{1+dk/c}] \subseteq \Sigma_{k+1}\mathsf{TIME}[n^{1+o(1)}] \cap \Pi_{k+1}\mathsf{TIME}[n^{1+o(1)}]$. This process converges when $d = 1 + dk/c$, or $d = c/(c-k)$.

The Speedup Lemma (Lemma A.1) implies that

$$\mathsf{DTS}[n^{1+dk/c}] \subseteq (\exists\ n)(\forall\ \log n)\mathsf{DTS}[n^{dk/c}, n^{o(1)}].$$

Applying speedup for k more times,

$$\mathsf{DTS}[n^{1+dk/c}] \subseteq (\exists\ n)(\forall\ \log n)\underbrace{(\forall\ n^{d/c})\cdots(Q\ n^{d/c})}_{k-1}(\neg Q\ \log n)\mathsf{DTS}[n^{d/c}]$$

for some $Q \in \{\exists, \forall\}$, where $\neg Q$ is opposite to Q. Since $\Sigma_k\mathsf{TIME}[n] \subseteq \mathsf{DTS}[n^c]$,

$$(\exists\ n)\underbrace{(\forall\ n^{d/c})\cdots(Q\ n^{d/c})(\neg Q\ \log n)}_{k}\mathsf{DTS}[n^{d/c}] \subseteq (\exists\ n)\mathsf{DTS}[n^d].$$

Finally, since $\mathsf{DTS}[n^d] \subseteq \Sigma_{k+1}\mathsf{TIME}[n^{1+o(1)}] \cap \Pi_{k+1}\mathsf{TIME}[n^{1+o(1)}]$,

$$(\exists\ n)\mathsf{DTS}[n^d] \subseteq \Sigma_{k+1}\mathsf{TIME}[n^{1+o(1)}].$$

An analogous argument implies $\mathsf{DTS}[n^{1+dk/c}] \subseteq \Pi_{k+1}\mathsf{TIME}[n^{1+o(1)}]$. $\square$

Theorem M.4 *If $\Sigma_k\mathsf{TIME}[n] \subseteq \mathsf{DTS}[n^c]$, then for all $\ell \geq 1$ and d satisfying $c \leq d < c/(c-k)$,*

$$\mathsf{DTS}[n^{d+\sum_{i=1}^{\ell}\left(\frac{c^2}{dk}\right)^i}] \subseteq \Sigma_{k+1}\mathsf{TIME}[n^{\left(\frac{c^2}{dk}\right)^\ell + o(1)}] \cap \Pi_{k+1}\mathsf{TIME}[n^{\left(\frac{c^2}{dk}\right)^\ell + o(1)}].$$

Proof. Induction on ℓ. The case $\ell = 0$ is immediate, by the previous theorem. For the inductive step, suppose $\mathsf{DTS}[n^{d+\sum_{i=1}^{\ell}\left(\frac{c^2}{dk}\right)^i}] \subseteq \Sigma_{k+1}\mathsf{TIME}[n^{\left(\frac{c^2}{dk}\right)^\ell + o(1)}]$. First, the Speedup Lemma implies

$$\mathsf{DTS}[n^{d+\sum_{i=1}^{\ell+1}\left(\frac{c^2}{dk}\right)^i}] \subseteq (\exists\ n^{\left(\frac{c^2}{dk}\right)^{\ell+1}})(\forall\ \log n)\mathsf{DTS}[n^{d+\sum_{i=1}^{\ell}\left(\frac{c^2}{dk}\right)^i}],$$

where the input to the DTS part has length $n + 2n^{o(1)}$. By the induction hypothesis, the above is contained in

$$(\exists\ n^{\left(\frac{c^2}{dk}\right)^{\ell+1}})(\forall\ \log n)\Pi_{k+1}\mathsf{TIME}[n^{\left(\frac{c^2}{dk}\right)^\ell + o(1)}].$$

Applying $\Sigma_k\mathsf{TIME}[n] \subseteq \mathsf{DTS}[n^c]$ to the Σ_k part of the $\Pi_{k+1}\mathsf{TIME}$ class, the above lies in

$$(\exists\, n^{\left(\frac{c^2}{dk}\right)^{\ell+1}})(\forall\ \log n)(\forall\, n^{\left(\frac{c^2}{dk}\right)^{\ell}})\mathsf{DTS}[n^{c\left(\frac{c^2}{dk}\right)^{\ell}}].$$

If $c < k$, we already have a contradiction, because $\Sigma_k\mathsf{TIME}[n^k] \subseteq \mathsf{DTS}[n^{kc}] \subseteq \Pi_k\mathsf{TIME}[n^c]$ (this follows from applying the Speedup Lemma, k times).

If $c \geq k$, the above class is contained in

$$(\exists\, n^{\left(\frac{c^2}{dk}\right)^{\ell+1}})(\forall\ \log n)(\forall\, n^{\left(\frac{c^2}{dk}\right)^{\ell}})\Pi_k\mathsf{TIME}[n^{\frac{c}{k}\cdot\left(\frac{c^2}{dk}\right)^{\ell}+o(1)}] = (\exists\, n^{\left(\frac{c^2}{dk}\right)^{\ell+1}})\Pi_k\mathsf{TIME}[n^{\frac{c}{k}\cdot\left(\frac{c^2}{dk}\right)^{\ell}+o(1)}].$$

Note $(\frac{c^2}{dk})^{\ell+1} \leq \frac{c}{k}\cdot(\frac{c^2}{dk})^{\ell}$, because $d \geq c$. Invoking the assumption $\Sigma_k\mathsf{TIME}[n] \subseteq \mathsf{DTS}[n^c]$ again results in the class

$$(\exists\, n^{\left(\frac{c^2}{dk}\right)^{\ell+1}})\mathsf{DTS}[n^{\frac{c^2}{k}\cdot\left(\frac{c^2}{dk}\right)^{\ell}}].$$

Finally, since $d(\frac{c^2}{dk})^{\ell+1} = \frac{c^2}{k}\cdot(\frac{c^2}{dk})^{\ell}$, Theorem M.3 applies, and the above class is in

$$(\exists\, n^{\left(\frac{c^2}{dk}\right)^{\ell+1}})\Sigma_{k+1}\mathsf{TIME}[n^{\frac{c^2}{dk}\cdot\left(\frac{c^2}{dk}\right)^{\ell}+o(1)}] = \Sigma_{k+1}\mathsf{TIME}[n^{\left(\frac{c^2}{dk}\right)^{\ell+1}+o(1)}].$$

An analogous argument proves the containment for $\Pi_{k+1}\mathsf{TIME}[n^{\left(\frac{c^2}{dk}\right)^{\ell+1}+o(1)}]$. □

Let $K_\ell = d + \sum_{i=1}^{\ell}\left(\frac{c^2}{dk}\right)^i$, for $\ell \geq 1$. We claim (the proof is not hard) that

$$\left(\frac{c^2}{dk}\right)^{\ell} \leq K_\ell\left(1 - \frac{dk}{c^2} - \varepsilon_\ell\right),$$

for a small constant $\varepsilon_\ell > 0$ satisfying $\lim_{\ell\to\infty}\varepsilon_\ell = 0$. We deduce the chain:

$$\begin{aligned}\Sigma_{k+1}\mathsf{TIME}[n^{K_\ell}] &\subseteq (\exists\, n^{K_\ell})\mathsf{DTS}[n^{cK_\ell}]\\ &\subseteq (\exists\, n^{K_\ell})\Sigma_k\mathsf{TIME}[n^{(c/k)K_\ell}] \subseteq \mathsf{DTS}[n^{(c^2/k)K_\ell}] \subseteq \Pi_{k+1}\mathsf{TIME}[n^{(c^2/k)K_\ell(1-\frac{dk}{c^2}-\varepsilon_\ell)}].\end{aligned}$$

For sufficiently large K_ℓ, a contradiction is reached when $c^2/k(1-(dk)/(c^2)) < 1$. Recalling that $d < c/(c-k)$, the condition simplifies to $p_k(c) = c^3/k - c^2 - 2c + k < 0$.

For concrete bounds, cf. Table 1. As the evidence suggests, at least one root of the polynomial p_k gradually approaches $k+1$ as k increases unboundedly; hence the lower bound exponent for QBF$_k$ approaches $k+1$.

Proposition 2 $\lim_{k\to\infty} p_k(k+1) = 0$. *In particular, for all* k, $p_k(k+1-1/k) < 0$ *and* $p_k(k+1) > 0$.

Proof. Algebraic manipulation gives $p_k(k+1) = 1/k > 0$ and $p_k(k+1-1/k) = 3/k^3 - 1 - 1/k - 1/k^2 - 1/k^4 < -1/k^4 < 0$, for all $k \geq 1$. □

Problem	**Time Lower Bound Exponent**
SAT	$n^{1.801}$
QBF$_2$	$n^{2.903}$
QBF$_3$	$n^{3.942}$
QBF$_4$	$n^{4.962}$
QBF$_{10}$	$n^{10.991}$
QBF$_{100}$	$n^{100.999902}$

Table 1: **Time lower bounds for** QBF$_k$ **on small space RAMs.**