Time-Space Lower Bounds for the Polynomial-Time Hierarchy on Randomized Machines

Scott Diehl[⋆] and Dieter van Melkebeek[⋆⋆]

University of Wisconsin-Madison, Madison WI 53706, USA
{sfdiehl, dieter}@cs.wisc.edu

Abstract. We establish the first polynomial-strength time-space lower bounds for problems in the linear-time hierarchy on randomized machines with bounded two-sided error. We show that for any integer $\ell > 1$ and constant $c < \ell$, there exists a positive constant d such that QSAT_ℓ cannot be computed by such machines in time n^c and space n^d, where QSAT_ℓ denotes the problem of deciding the validity of a Boolean first-order formula with at most $\ell - 1$ quantifier alternations. Corresponding to $\ell = 1$, we prove that for any constant $c < \phi \approx 1.618$, there exists a positive constant d such that the set of Boolean tautologies cannot be decided by a randomized machine with one-sided error in time n^c and space n^d.

1 Introduction

Proving lower bounds remains one of the most challenging tasks in computational complexity. Satisfiability, the seminal NP-complete problem, is particularly unyielding in this respect. While we believe that any algorithm for satisfiability takes time linear exponential in the number of variables in the formula, we have been unable to prove super-linear time lower bounds on random access machines despite several decades of effort. Additionally, problems complete for higher levels of the polynomial-time hierarchy, while not receiving as much attention, have also resisted nontrivial time lower bounds.

A few years ago, Fortnow [5] realized that if we restrict the work space that a machine can use to solve satisfiability, then we can establish nontrivial lower bounds. Fortnow's technique has its roots in earlier work by Kannan [6] and has been further developed in recent years [8, 9]. For example, Fortnow and Van Melkebeek derived the following time-space lower bound for nondeterministic linear time, which gives the same lower bound for satisfiability due to the tight connection between satisfiability and nondeterministic linear time.

Theorem 1 (Fortnow-Van Melkebeek [4]). *Let $\phi \doteq (\sqrt{5} + 1)/2 \approx 1.618$ denote the golden ratio. For any constant $c < \phi$ there exists a positive constant*

[⋆] Supported by NSF Career award CCR-0133693.

[⋆⋆] Partially supported by NSF Career award CCR-0133693.

L. Caires et al. (Eds.): ICALP 2005, LNCS 3580, pp. 982–993, 2005.

d such that nondeterministic linear time cannot be simulated by deterministic random-access machines running in time n^c and space n^d.

Fortnow and Van Melkebeek also considered higher levels of the linear-time hierarchy and managed to prove the following time-space lower bound.

Theorem 2 (Fortnow-Van Melkebeek [4]). *For any integer $\ell \geq 2$ and constant $c < \ell$, there exists a positive constant d such that $\Sigma_\ell\mathrm{TIME}[n]$ cannot be simulated by deterministic random-access machines running in time n^c and space n^d.*

The same tight relationship between nondeterministic linear time and satisfiability also exists between $\Sigma_\ell\mathrm{TIME}[n]$ and QSAT_ℓ, the problem of deciding the validity of a given Boolean first-order formula with at most $\ell - 1$ quantifier alternations. Thus, the time-space lower bound for $\Sigma_\ell\mathrm{TIME}[n]$ of Theorem 2 holds for QSAT_ℓ as well.

In this paper, we establish time-space lower bounds for the same problems on *randomized* machines with two-sided error bounded away from 1/2.

Theorem 3 (Main Theorem). *For any integer $\ell \geq 2$ and constant $c < \ell$, there exists a positive constant d such that $\Sigma_\ell\mathrm{TIME}[n]$ cannot be simulated by randomized random-access machines with two-sided error bounded away from 1/2 running in time n^c and space n^d.*

Observe that our bounds essentially match those for the deterministic simulations given in Theorem 2, the only difference being the exact dependence of d on c. As in the deterministic case, Theorem 3 implies the same time-space lower bounds for QSAT_ℓ.

One can also view the instantiation of our Main Result for $\ell = 2$ as an analog of Theorem 1. Note that Theorem 1 relates to the question of P versus NP. We know the trivial inclusion that $\mathrm{P} \subseteq \mathrm{NP}$ and do not believe the converse but fail to prove that conjecture. Thus, research evolved towards time-space lower bounds to achieve partial negative results for the converse, as in [5, 8, 4, 9]. Similarly for BPP, we know that $\mathrm{BPP} \subseteq \Sigma_2^p$ but we do not believe the converse. In this work, we turn to time-space lower bounds to achieve nontrivial negative results about the converse.

We also further strengthen Theorem 3 by showing similar lower bounds for problems that can be decided by Σ_ℓ machines in linear time and space n^a for some constant $a < 1$.

We note that Theorem 3 establishes the first polynomial-strength time-space lower bounds for two-sided error randomized simulations of the polynomial-time hierarchy. By time-space lower bounds of "polynomial strength" we mean time lower bounds of the form $\Omega(n^c)$ for some constant $c > 1$ under nontrivial space upper bounds. Previous works establish randomized time-space lower bounds but they either consider problems believed not to be in the polynomial-time hierarchy, or the time lower bounds involved are only slightly super-linear. Allender et al.'s [1] time-space lower bounds for problems in the counting hierarchy on probabilistic machines with unbounded error fall within the first category. Beame et al. [2] give a (nonuniform) time-space lower bound for a problem in P based on a binary quadratic form, which falls in the second category.

At first glance, it might seem that the known results about space-bounded derandomization let us derive time-space lower bounds on randomized machines as immediate corollaries to the known time-space lower bounds on deterministic machines. In particular, assuming that we can solve satisfiability on a randomized machine in logarithmic space and time n^c, Nisan's deterministic simulation [11] yields a deterministic algorithm for satisfiability that runs in polylogarithmic space and polynomial time. However, even for $c = 1$, the degree of the latter polynomial is far too large for this simulation to yield a contradiction with Theorem 1 as we would like. Thus, we need a more delicate approach for the randomized setting.

Our proofs follow the paradigm of indirect diagonalization. The technique establishes a desired separation by contradiction – assuming the separation does not hold, we derive a sequence of progressively unlikely inclusions of complexity classes until we reach one that contradicts a known diagonalization result. Kannan [6] used the paradigm avant la lettre to investigate the relationship between deterministic linear time and nondeterministic linear time. All of the recent work on time-space lower bounds for satisfiability and problems higher up in the polynomial-time hierarchy [5, 8, 4, 9] follow it as well. Allender et al. [1] employed the technique to establish time-space lower bounds for problems in the counting hierarchy.

The critical ingredient that allows us to apply the paradigm to two-sided error randomized algorithms for problems in the polynomial-time hierarchy is a time and space efficient simulation of randomized computations in the second level of the polynomial-time hierarchy with very few guess bits. The latter follows from a careful combination of Nisan's partial space-bounded derandomization [10] and a version of Lautemann's proof that BPP $\subseteq \Pi_2^p$ [7].

We point out that earlier work implies lower bounds for (the complements of) the above problems on randomized machines with *one-sided error*. This follows because the lower bound arguments for conondeterministic linear time on *deterministic* machines typically also work on *nondeterministic* machines, of which randomized machines with one-sided error are special cases. However, the bounds become weaker. For example, the bound on c in Theorem 1 reduces from the golden ratio to $\sqrt{2}$.

Theorem 4 (Fortnow-Van Melkebeek [4]). *For any constant $c < \sqrt{2}$ there exists a positive constant d such that conondeterministic linear time cannot be simulated by randomized random-access machines with one-sided error running in time n^c and space n^d.*

Using ideas from the proof of our main result, we manage to strengthen Theorem 4 so that the bound matches the one in Theorem 1.

Theorem 5. *For any constant $c < \phi$, there exists a positive constant d such that conondeterministic linear time cannot be simulated by randomized random-access machines with one-sided error running in time n^c and space n^d.*

A similar strengthening holds for the analog of Theorem 2.

2 Preliminaries

Most of the notation we use is standard [3, 12]. For a detailed description of the machine model we use, we refer the reader to [9]. We adopt the convention that time and space functions refer to constructible functions from natural numbers to natural numbers. Our results ultimately apply to computations with polynomial time and space bounds, which certainly meet these conditions.

We introduce some additional terminology to reason about randomized computation. In particular, we use the notation BPTISP$[t, s]$ to refer to the class of languages recognized by randomized machines using time t and space s with error bounded by $\frac{1}{3}$ on both sides.

Our arguments involve Σ_k^p and Π_k^p computations in which the numbers of bits guessed at each stage are bounded by explicitly given small functions. To this end, we use the following notation to describe such computations:

Definition 1. *Given a complexity class $\mathcal{C}$ and a function f, we define the class $\exists^f \mathcal{C}$ to be the set of languages that can be described as*

$$\{x | \exists y \in \{0,1\}^{O(f(|x|))} P(x, y)\},$$

where P is a predicate accepting a language in the class $\mathcal{C}$ when its complexity is measured in terms of $|x|$ (not $|x| + |y|$). We analogously define $\forall^f \mathcal{C}$.

For example, $\exists^f$DTIME$[n]$ and $\forall^f$DTIME$[n]$ are subsets of NP and coNP for $f(n) = n^{O(1)}$. The requirement that the complexity of P be measured in terms of $|x|$ allows us to express the running times simply in terms of the original input length, which is a more natural notion for our arguments.

We also make use of the standard divide-and-conquer approach for speeding up space bounded computation by introducing alternations. Namely, by splitting up the computation tableau of a DTISP$[T, S]$ computation into $b > 0$ equal size blocks, we obtain

$$\mathrm{DTISP}[T, S] \subseteq \exists^{bS} \forall^{\log b} \mathrm{DTISP}[T/b, S] \subseteq \Sigma_2 \mathrm{TIME}[bS + T/b]. \tag{1}$$

If we choose b to optimize the running time of the resulting Σ_2 computation, the result is

$$\mathrm{DTISP}[T, S] \subseteq \exists^{\sqrt{TS}} \exists^{\log T} \subseteq \Sigma_2 \mathrm{TIME}[\sqrt{TS}]. \tag{2}$$

When used within the framework of time-space lower bounds, it is not always desirable to choose the block size to optimize the running time in this way, as exhibited by [4]. Therefore, most of our arguments make use of (1) for some unspecified b, and then set b later to yield the strongest results.

Finally, we need a standard diagonalization result from which we can derive contradictions. The following lemma states that for a fixed number of alternations, if we switch from universal to existential initial states and allow for a little more time, we can compute something we couldn't compute before.

Lemma 1 (Folklore). *Let ℓ be a positive integer and t a time function. Then*

$$\Sigma_\ell \mathrm{TIME}[t] \not\subseteq \Pi_\ell \mathrm{TIME}[o(t)].$$

3 Framework of Earlier Deterministic Results

In this section, we provide an overview of the arguments used to establish Theorem 1 since our approach uses the same general framework. Specifically, both arguments follow the paradigm of indirect diagonalization, which can be divided into the following general steps:

1. Assume the inclusion that we wish to show does not hold. For example, $\Sigma_2\mathrm{TIME}[n] \subseteq \mathrm{BPTISP}[t, s]$.
2. Using the hypothesis, derive inclusions of complexity classes which are increasingly unlikely.
3. Eventually one of these inclusions contradicts a known diagonalization result, proving the desired result.

Let us step through a weaker instantiation of Theorem 1 as an example. Namely, we prove the result of [8] that $\mathrm{NTIME}[n] \nsubseteq \mathrm{DTISP}[n^c, n^{o(1)}]$ for $c < \sqrt{2}$ following the outline described above. Therefore, the first step is to assume that

$$\mathrm{NTIME}[n] \subseteq \mathrm{DTISP}[n^c, n^{o(1)}]. \tag{3}$$

Consider the class $\mathrm{DTISP}[T, T^{o(1)}]$ for some polynomial T, say $T(n) = n^2$. Using (2), we can speed up this computation by introducing alternations, resulting in a $\Sigma_2\mathrm{TIME}[T^{1/2+o(1)}]$ simulation. Now observe that we can use the hypothesis (3) to collapse Σ_2 to NP, eliminating one alternation at the small cost of raising the running time of the simulation to the power of c. Since DTISP is closed under complement, this process gives the inclusion

$$\mathrm{DTISP}[T, T^{o(1)}] \subseteq \mathrm{coNTIME}[T^{c/2+o(1)}, T^{o(1)}]. \tag{4}$$

If the cost of removing an alternation by this technique is less than the speedup we gained by its introduction, then (4) is a more unlikely inclusion than the hypothesis, which could lead to a contradiction with Lemma 1.

To find the values of c which yield a contradiction, consider the hypothesis (3) padded to time T. Combining this with (4), we can conclude

$$\mathrm{NTIME}[T] \subseteq \mathrm{DTISP}[T^c, T^{o(1)}] \subseteq \mathrm{coNTIME}[T^{c^2/2+o(1)}],$$

which contradicts Lemma 1 so long as $c^2/2 < 1$. This proves the desired result.

One can obtain stronger results by applying these arguments recursively. Specifically, once the speedup of (1) is applied, the final stage involved in the resulting Σ_2 simulation is itself a space bounded computation taking less time than what we started with. Therefore, we can obtain a contradiction for larger c if we recursively apply the same arguments to further speed up this computation. Doing so in an economical way (with respect to alternations) and choosing the block numbers optimally at each recursive step yields Theorem 1.

4 Lautemann's Proof and Derandomization

For this paper, we wish to derive a contradiction from the assumption

$$\Sigma_2\mathrm{TIME}[n] \subseteq \mathrm{BPTISP}[t, s]. \tag{5}$$

Taking a cue from the deterministic results, we would like to figure out a way to transfer the hypothesis (5) into a statement giving a strong collapse of the polynomial-time hierarchy, such as $\Sigma_2\mathrm{TIME}[n] \subseteq \Pi_2\mathrm{TIME}[f(n)]$ for some small function $f(n)$. We can then use such a collapse to eliminate alternations introduced by applying the speedup of (1). The focus of this section is the derivation of such a statement.

Lautemann's proof that BPP $\subseteq \Sigma_2^p \cap \Pi_2^p$ is the first tool which helps us accomplish this task. The proof relies on the large gap in sizes of witness sets for a BPP algorithm which uses R bits to accept a language L with sufficiently small error. When $x \in L$, the witness set is large enough so that for most sets of R shifts, the union of the R shifted witness sets covers the entire universe of possible witnesses; when $x \notin L$, the witness set is so small that no set of R shifts covers the universe. These complementary conditions can be expressed by a Σ_2^p predicate. Since BPP is closed under complement, this shows that BPP $\subseteq \Sigma_2^p \cap \Pi_2^p$. Specifically, we are interested in the Π_2^p side of the inclusion for our results.

Theorem 6 (Lautemann [7]). *Let L be a language recognized by a randomized machine M with error bounded on both sides by $1/R$ that runs in time T, space S (when provided two-way access to the random bits), and uses R random bits. Then*

$$L \in \forall^{R^2}\exists^{R}\mathrm{DTISP}[RT, S + \log R]. \tag{6}$$

Theorem 6 is a natural candidate to derive the desired strong collapse of the polynomial-time hierarchy from (5). However, the key question is if the inclusion given by Theorem 6 is efficient enough to allow for a sufficiently strong collapse. To answer this, first note the requirement that the error be $1/R$ does not pose a problem, since this can be achieved by taking the majority vote of $O(\log R) = O(\log T)$ repetitions of a standard BPP algorithm. Thus, we can transform an arbitrary BPP computation into an equivalent one satisfying the conditions of Theorem 6 with only a logarithmic time blowup. This is good news since we would be in trouble if the theorem required error exponentially small in R – in general, the running time of the amplified algorithm would be at least quadratic.

On the other hand, the Π_2 simulation resulting in an application of Theorem 6 must run the randomized machine on R different shifts of a witness so its running time is a factor of R greater than that of the randomized machine. Since, in general, R can be as large as T, this slow-down is too much for our arguments to compensate. Therefore, an additional ingredient is needed.

That ingredient exploits the fact that the hypothesis (5) gives simulations by *space-bounded* BPP computations. In that setting, we know of techniques to reduce the number of random bits used, which in turn reduces the overhead in the Π_2 simulation given by Theorem 6. The means by which we achieve the needed reduction in randomness is the space-bounded derandomization of Nisan [10].

Theorem 7 (Nisan [10]). *Any randomized algorithm running in time T and space S with error ϵ can be simulated by one running in time $O(T)$ and space $O(S \log T)$, which uses only $O(S \log T)$ random bits and has error $\epsilon + 2^{-S}$. If two-way access to the random bits is allowed, the space requirement is reduced to $O(S)$.*

Note that we do not apply Theorem 7 to deterministically simulate the randomized algorithm. Instead, we use it to reduce the randomness required by a BPTISP$[T, S]$ algorithm to $O(S \log T)$ with no blowup in time. If we subsequently apply Theorem 6, we only incur a blowup of $O(S \log T)$, which is acceptable for polynomial T and small S. Therefore, we can conclude:

Theorem 8.

$$\mathrm{BPTISP}[T, S] \subseteq \forall^{(S \log T)^2} \exists^{S \log T} \mathrm{DTISP}[TS \log^2 T, S]. \tag{7}$$

Proof. Let A be the randomized time T, space S procedure for recognizing $L \in$ BPTISP$[T, S]$. We first take the majority of $O(\log T)$ independent repetitions of A to reduce the error to $1/T^2$, increasing the running time to $O(T \log T)$. We then apply the derandomization provided by Theorem 7 to obtain a procedure A' taking time $O(T \log T)$, using $O(S \log T)$ random bits, and using space $O(S)$ when allowed two-way access to the random bits. Additionally, A' has error at most $1/T^2 + 2^{-S}$. Since we can assume without loss of generality that $S \leq T \leq 2^S$ and $S = \omega(1)$, we have that $1/T^2 + 2^{-S} = o(1/(S \log T))$. We now apply Theorem 6 to derive (7). □

This result gives exactly the efficient inclusion of space bounded randomized classes in the polynomial time hierarchy that we need. Combining it with the hypothesis (5), we derive the inclusion

$$\Sigma_2\mathrm{TIME}[n] \subseteq \forall^{(s \log t)^2} \exists^{s \log t} \mathrm{DTISP}[ts \log^2 t, s], \tag{8}$$

which gives the desired strong collapse for small enough t and s.

5 Main Result

We now use the techniques discussed in the previous sections to complete our indirect diagonalization argument for the proof of Theorem 3. Using (8) as a starting point, we derive a series of inclusions giving stronger and stronger collapses of the polynomial-time hierarchy, towards the end of contradicting Lemma 1. Specifically, we derive inclusions of the form $\Sigma_2\mathrm{TIME}[n] \subseteq \Pi_2\mathrm{TIME}[f(n)]$ for smaller and smaller $f(n)$, eventually hoping to find such an inclusion for $f(n) = o(n)$. Given that the hypothesis allows a simulation of $\Sigma_2\mathrm{TIME}[n]$ by

BPTISP$[t, s]$, we rely on finding a sequence of inclusions for the latter to give these collapses. The following process derives the first such inclusion giving an improvement over (8):

1. First, we use Theorem 8 to derive a Π_2 simulation of a BPTISP$[t, s]$ computation.
2. Since the quantifiers of the Π_2 simulation derived in the previous step are over a small number of bits, the dominant term in the running time comes from the final space bounded deterministic computation. Therefore, we can apply the speedup of (1) to the final deterministic computation to achieve a simulation taking less time, whose quantifiers look like $\forall\exists\exists\forall$, i.e., a Π_3 computation.
3. We now eliminate an alternation from this Π_3 computation in two steps. We first apply the hypothesis to simulate the computation represented by the last two quantifiers (a Σ_2 computation) by a BPTISP computation.
4. Finally, we complete the collapse by applying Theorem 8 to simulate this BPTISP computation by a Π_2 computation. Merging the two initial universal quantifiers gives us a Π_2TIME$[f(n)]$ simulation of BPTISP$[t, s]$ for small $f(n)$, depending on the choice of t and s.

For small enough t and s, we have derived a stronger collapse of Σ_2 to Π_2 than that given by (8). We can achieve even stronger collapses by viewing the above procedure as a process that takes as input a BPTISP computation and returns a Π_2 simulation which possibly takes less time. If we recursively apply this procedure to the BPTISP computation in step 4 instead of Theorem 8, we can complete the collapse to Π_2 while possibly further speeding up the computation, providing a stronger collapse. Running this recursive argument more and more times yields the sequence of collapses we require.

The following lemma formalizes the above idea. Specifically, we consider the hypothesis (5) for polynomial t and s, namely $t = n^c, s = n^d$, and derive the running time of the resulting Π_2 simulation in terms c, d, and k, the number of times the argument is recursively applied.

Lemma 2. *Suppose that*

$$\Sigma_2\text{TIME}[n] \subseteq \text{BPTISP}[n^c, n^d] \tag{9}$$

for some constants $c \geq 1$ and $d > 0$ where $c + 2d \leq 2$. Then for any functions T and S and positive integer k such that $2d \leq f_k$,

$$\text{BPTISP}[T, S] \subseteq \Pi_2\text{TIME}\left[\left((TS^2)^{f_k} + (n + S^2)^{c+2d}\right) \text{polylog}(T + n)\right],$$

where

$$f_k = \left(\tfrac{c+2d}{2}\right)^k. \tag{10}$$

Lemma 2 gives the sequence of collapses which, for certain values of c and d, leads to a contradiction with Lemma 1. We now derive the resulting time-space lower bound for Σ_2TIME$[n]$.

Theorem 9. *For any constant $c < 2$, there exists a positive constant d such that $\Sigma_2\mathrm{TIME}[n]$ cannot be simulated by randomized random-access machines with error bounded away from $1/2$ running in time n^c and space n^d.*

Proof. For $c < 1$, the Theorem holds for any d by standard techniques.

We prove the case for $c \geq 1$ via indirect diagonalization. Suppose, by way of contradiction, that

$$\Sigma_2\mathrm{TIME}[n] \subseteq \mathrm{BPTISP}[n^c, n^d], \tag{11}$$

for some constant $d > 0$ to be determined later. Then for any time function $\tau(n) \geq n$, the hypothesis and Lemma 2 give us the inclusions

$$\begin{aligned}\Sigma_2\mathrm{TIME}[\tau] &\subseteq \mathrm{BPTISP}[\tau^c, \tau^d] \\ &\subseteq \Pi_2\mathrm{TIME}\left[\left((\tau^{c+2d})^{f_k} + (\tau^{2d} + n)^{c+2d}\right) \mathrm{polylog}(\tau)\right]\end{aligned}$$

when $c + 2d \leq 2$ and $2d \leq f_k$. In the case that $2d \leq f_k$, the dominant power of τ will be $2f_{k+1}$, allowing us to simplify the running time of the Π_2 machine to big-O of

$$(\tau^{2f_{k+1}} + n^{c+2d})\mathrm{polylog}(\tau).$$

Choosing a sufficiently large polynomial τ, we can further simplify this to

$$\tau^{2f_{k+1}}\mathrm{polylog}(\tau).$$

Therefore, we have shown

$$\Sigma_2\mathrm{TIME}[\tau] \subseteq \Pi_2\mathrm{TIME}[\tau^{2f_{k+1}}\mathrm{polylog}(\tau)]. \tag{12}$$

The inclusion (12) gives a contradiction with Lemma 1 for any k with $f_{k+1} < 1/2$. Note that $f_k \to 0$ as $k \to \infty$ if $c + 2d < 2$. Therefore, all that remains is to show that the latter condition is compatible with the other ones, i.e., that we can pick a constant $d > 0$ and an integer $k > 0$ such that

$$\begin{aligned} c + 2d &< 2, \quad (13) \\ 2d &\leq f_k, \text{ and} \quad (14) \\ f_{k+1} &< 1/2. \quad (15) \end{aligned}$$

It remains to show that d and k can be chosen to satisfy these constraints. For any c and d satisfying (13), consider choosing k to be the smallest integer such that (15) is satisfied. For such a choice, we can see that $f_k \geq 1/2$, so $d \leq 1/4$ satisfies (14). Therefore, choosing d such that $d \leq \min(1/4, \frac{2-c}{2})$ and then calculating k as described above yields a d and k satisfying all of the constraints. □

We point out that the dependence of d on c in Theorem 9 differs from the deterministic setting. The proofs of Theorem 1 and Theorem 2 show that as c approaches 1 from above, d approaches 1 from below. In the proof of Theorem 9, however, the strategy described for choosing d yields a value approaching 1/4 from below as c approaches 1 from above. Although we have not optimized our arguments to obtain the largest value of d possible, a smaller value than in the deterministic case seems inherent to our approach.

The proof of Theorem 9 generalizes to higher levels of the linear-time hierarchy, as stated in our Main Theorem. In this setting, we can use the hypothesis that $\Sigma_\ell\mathrm{TIME}[n] \subseteq \mathrm{BPTISP}[n^c, n^d]$ along with Theorem 8 to eliminate more than one alternation at the same cost of removing one alternation in the setting of Theorem 9. This lets us eliminate the alternations introduced by many recursive applications of (1), achieving a greater speedup and allowing contradictions for values of c less than ℓ.

Through a tight connection to nondeterministic linear time, the results of Theorem 1 extend to satisfiability and many other NP-complete problems. This connection also exists between $\Sigma_\ell\mathrm{TIME}[n]$ and Σ_ℓ-complete problems such as QSAT_ℓ, allowing us to extend our time-space lower bounds to randomized computations of such problems.

Corollary 1. *For any integer $\ell \geq 2$ and constant $c < \ell$, there exists a positive constant d such that QSAT_ℓ cannot be solved by randomized random-access machines with error bounded away from 1/2 running in time n^c and space n^d.*

Paying close attention to the space used by the simulations in our proofs, we actually obtain time-space lower bounds for $\Sigma_2\mathrm{TIME}[n, n^a]$ for certain values of $a < 1$.

Theorem 10. *For any integer $\ell \geq 2$ and any positive constants c and a with $c < 1 + (\ell - 1)a$, there exists a constant $d > 0$ such that $\Sigma_\ell\mathrm{TISP}[n, n^a] \nsubseteq \mathrm{BPTISP}[n^c, n^d]$.*

6 Other Results

In this section, we show how to extend the golden ratio result of Fortnow and Van Melkebeek for deterministic machines (Theorem 1) to randomized machines with one-sided error, yielding Theorem 5. As we will argue, Theorem 5 follows from the next extension of Theorem 1 to a a slightly stronger class of machines.

Theorem 11. *For any constant $c < \phi$ there exist positive constants d and b such that conondeterministic linear time cannot be simulated by nondeterministic random-access machines which run in time n^c, space n^d, and nondeterministically guess only n^b bits.*

Nisan's space-bounded derandomization [10] given in Theorem 7 allows us to reduce the number of random bits used by a randomized machine with one-sided error running in time T and space S to $O(S \log T)$ without significantly increasing the time or space used. Since a randomized machine with one-sided error is also a special type of nondeterministic machine, we can view such a derandomized machine as a space bounded nondeterministic machine which guesses only

$O(S \log T)$ bits. Thus, the time-space lower bounds of Theorem 5 for conondeterministic linear time on randomized machines with one-sided error follow as a corollary to Theorem 11.

In order to generalize Theorem 1 to nondeterministic machines that guess a bounded number of bits, we derive an analog to Lemma 2 which gives an unlikely inclusion of $\exists^B \mathrm{DTISP}[T, S]$ into NTIME. Recall that in Section 5, we observed that if the number of bits guessed in a $\Pi_2\mathrm{TISP}[T, S]$ computation is much less than the running time T, then the entire computation can be sped up by applying (1) to the final deterministic stage of the computation. The same observation allows us to speed up a $\exists^B \mathrm{DTISP}[T, S]$ computation when T dominates B. We accomplish the latter through an extension of Lemma 3.1 from [4], which gives a speedup of $\mathrm{DTISP}[T, S]$ on nondeterministic machines under the hypothesis that $\mathrm{coNTIME}[n] \subseteq \mathrm{NTIME}[n^c]$.

Lemma 3. *Suppose that*

$$\mathrm{coNTIME}[n] \subseteq \mathrm{NTIME}[n^c]$$

for some constant $c \geq 1$. Then for any functions T, S, and B and any integer $k \geq 0$

$$\exists^B \mathrm{DTISP}[T, S] \subseteq \mathrm{NTIME}[(T \cdot S^k)^{f_k} + (n + B + S)^{c^k}],$$

where f_k is given by,

$$\begin{aligned} f_0 &= 1 \\ f_{k+1} &= c \cdot f_k / (1 + f_k). \end{aligned} \tag{16}$$

We now describe how to use Lemma 3 to prove Theorem 11. First assume that $\mathrm{coNTIME}[n] \subseteq \exists^{n^b} \mathrm{DTISP}[n^c, n^d]$ for b and d to be determined later. Padding this assumption and applying Lemma 3 yields the inclusion

$$\mathrm{coNTIME}[\tau] \subseteq \mathrm{NTIME}[\tau^{(c+kd)f_k} + (n + \tau^b + \tau^d)^{c^k}] \tag{17}$$

for any time function $\tau(n) \geq n$. Letting τ be a large enough polynomial and choosing small enough values for b and d, (17) forms a contradiction to Lemma 1 as long as there is a k such that $cf_k < 1$. Since the sequence $(f_k)_k$ defined by (16) decreases monotonically to $c - 1$ for $c < 2$, this is the case if and only if $c(c-1) < 1$. Since $c(c-1) = 1$ defines the golden ratio ϕ, this establishes Theorem 11.

We point out that a similar strengthening as Theorem 5 holds for the analog of Theorem 2.

7 Further Research

The techniques discussed in this work allow us to establish time-space lower bounds for two-sided error randomized simulations of the polynomial-time hierarchy at the second level and higher. They do not seem to extend to the first level in a straightforward way. This is mainly due to the fact that the assumption NTIME$[n] \subseteq$ BPTISP$[t, s]$ doesn't seem to allow the collapsing of alternations in an efficient manner. Thus, establishing time-space lower bounds for satisfiability on randomized machines with two-sided error remains open.

Improving the quantitative results of this paper is another direction needing further work. Very recently, Williams [13] used a bootstrapping argument to improve the lower bounds for NTIME$[n]$ on deterministic machines. He was able to boost the bound on the exponent c in Theorem 1 from the golden ratio $\phi \approx 1.618$ to 1.732, and we have been able to boost it further to 1.759. Using the same technique, we can improve our lower bound for coNTIME$[n]$ on randomized machines with one-sided error, boosting the bound on the exponent c in Theorem 5 from the golden ratio to 1.759. However, we have been unable to apply the technique to improve our main result, the lower bounds for Σ_ℓTIME$[n]$ on randomized machines with two-sided error, as given in Theorem 3.

Acknowledgements

We would like to thank Bess Berg and the anonymous referees for helpful comments.

References

1. E. Allender, M. Koucky, D. Ronneburger, S. Roy, and V. Vinay. Time-space tradeoffs in the counting hierarchy. In *CCC*, pages 295-302. IEEE, 2001.
2. P. Beame, M. Saks, X. Sun, and E. Vee. Time-space tradeoff lower bounds for randomized computation of decision problems. *Journal of the ACM,* 50(2):154-195, 2003.
3. J. Balcázar, J. Díaz, J. Gabarró. *Structural Complexity I,* volume 11 of *EATCS Monographs on Theoretical Computer Science.* Springer-Verlag, 1995.
4. L. Fortnow and D. van Melkebeek. Time-space tradeoffs for nondeterministic computation. In *CCC*, pages 2-13. IEEE, 2000.
5. L. Fortnow. Time-space tradeoffs for satisfiability. *Journal of Computer and System Sciences*, 60:337-353, 2000.
6. R. Kannan. Towards separating nondeterminism from determinism. *Mathematical Systems Theory,* 17:29-45, 1984.
7. C. Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17(4):215-217, 1983.
8. R. Lipton and A. Viglas. On the complexity of SAT. In *FOCS*, pages 459-464. IEEE, 1999.
9. D. van Melkebeek. Time-Space Lower Bounds for NP-Complete Problems. In G. Paun, G. Rozenberg, and A. Salomaa, editors, *Current Trends in Theoretical Computer Science*, pages 265-291. World Scientific, 2004.
10. N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449-461, 1992.
11. N. Nisan. RL $\subseteq$ SC. *Computational Complexity,* 4:1-11, 1994.
12. C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.
13. R. Williams. Better Time-Space Lower Bounds for SAT and Related Problems. To appear in *CCC.* IEEE, 2005.