An Improved Time-Space Lower Bound for Tautologies

Scott Diehl[1]*, Dieter van Melkebeek[2] **, and Ryan Williams[3]***

[1] Computer Science Department, Siena College, Loudonville, NY 12211
sfdiehl@siena.edu
[2] Computer Sciences Department, University of Wisconsin, Madison, WI 53706
dieter@cs.wisc.edu
[3] School of Mathematics, Institute for Advanced Study, Princeton, NJ 08540
ryanw@ias.edu

Abstract. We show that for all reals c and d such that $c^2 d < 4$ there exists a real $e > 0$ such that tautologies of length n cannot be decided by both a nondeterministic algorithm that runs in time n^c, and a nondeterministic algorithm that runs in time n^d and space n^e. In particular, for all $d < \sqrt[3]{4}$ there exists an $e > 0$ such that tautologies cannot be decided by a nondeterministic algorithm that runs in time n^d and space n^e.

1 Introduction

Proof complexity studies the NP versus coNP problem — whether tautologies can be recognized efficiently by nondeterministic machines. Typical results in proof complexity deal with specific types of nondeterministic machines that implement well-known proof systems, such as resolution. They establish strong (superpolynomial or even exponential) lower bounds for the size of any proof of certain families of tautologies within that system, and thus for the running time of the corresponding nondeterministic machine deciding tautologies.

Another, more generic, approach to the NP versus coNP problem follows along the lines of the recent time-space lower bounds for satisfiability on deterministic machines [4]. Similar arguments yield lower bounds for satisfiability on conondeterministic machines, or equivalently, for tautologies on nondeterministic machines. Those results show that no nondeterministic algorithm can decide tautologies in time n^d and space n^e for interesting combinations of d and e. The lower bounds obtained are very robust with respect to the model of computation and apply to any proof system. However, the arguments only work in the polynomial time range (constant d) and sublinear space range ($e < 1$). For example, Fortnow [1] proved that we must have $d > 1$ whenever $e < 1$, and Fortnow and

* Research partially supported by NSF award CCR-0728809 and a Cisco Systems Distinguished Graduate Research Fellowship while at the University of Wisconsin.

** Research partially supported by NSF award CCR-0728809 and partially performed while visiting the Weizmann Institute of Science.

*** Research partially supported by NSF award CCF-0832797.

Van Melkebeek [3] (see also [2]) showed a time lower bound of n^d for any $d < \sqrt{2}$ in the case of subpolynomial space bounds ($e = o(1)$).

In this paper we build on these generic techniques and boost the exponent in the time lower bound for subpolynomial-space nondeterministic algorithms recognizing tautologies from $\sqrt{2} \approx 1.414$ to $\sqrt[3]{4} \approx 1.587$.

Theorem 1. *For every real $d < \sqrt[3]{4}$ there exists a positive real e such that tautologies cannot be decided by nondeterministic algorithms running in time n^d and space n^e.*

The earlier result of Fortnow and Van Melkebeek [3] can be refined to rule out either nondeterministic algorithms solving tautologies in time n^c (regardless of space) *or* nondeterministic algorithms solving tautologies in simultaneous time n^d and space n^e for certain combinations of c, d, and e. More precisely, for every c and d such that $(c^2 - 1)d < c$, there is an $e > 0$ satisfying the lower bound. For example, tautologies cannot have both a nondeterministic algorithm using $n^{1+o(1)}$ time and a nondeterministic algorithm using logarithmic space [1]. Correspondingly, our argument yields the following refinement.

Theorem 2. *For all reals c and d such that $c^2 d < 4$, there exists a positive real e such that tautologies of length n cannot be solved by both*

(i) a nondeterministic algorithm that runs in time n^c and
(ii) a nondeterministic algorithm that runs in time n^d and space n^e.

The interesting range of parameters in Theorem 2 is $d \geq c \geq 1$, since an algorithm of type (ii) is a special case of an algorithm of type (i) for $d \leq c$, and a sublinear-time algorithm can be ruled out unconditionally by simple diagonalization. The condition due to this paper, $c^2 d < 4$, is less restrictive for values of d that are close to c. In particular, for $c = d$, our condition requires $d < \sqrt[3]{4} \approx 1.587$, whereas that of [3] requires $d < \sqrt{2} \approx 1.414$; this setting is the improvement stated in Theorem 1.

Our main technical contribution is another level of sophistication in the indirect diagonalization paradigm, corresponding to the transition from linear to nonlinear dynamics. We start from the hypothesis that tautologies have machines of types (i) and (ii), and aim to derive a contradiction. Fortnow and Van Melkebeek [3] use (ii) to obtain a nondeterministic time-space efficient simulation of conondeterministic computations. Next, they speed up the space-bounded nondeterministic computation à la Savitch [5] by introducing alternations, and subsequently eliminate those alternations efficiently using (i). When $(c^2-1)d < c$, the net effect is a speedup of generic conondeterministic computations on nondeterministic machines, implying the sought-after contradiction.

The above argument exploits (ii) in a rather limited way, namely only in the very first step. One could use (ii) instead of (i) to eliminate alternations. Since $d \geq c$ this costs at least as much time as using (i), but the space bound induced by (ii) allows us to run another layer of alternation-based speedups and alternation eliminations. Due to the additional layer, the recurrence relation for the net speedup becomes of degree two (rather than one as before) and has nonconstant coefficients, but we can still handle it analytically. We point out that this is the first application of nonlinear dynamics in analyzing time-space lower bounds for satisfiability and related problems.

2 Preliminaries

2.1 Notation

For functions t and s we denote by NTIME(t) the class of languages recognized by nondeterministic machines that run in time $O(t)$, and by NTISP(t, s) those recognized by nondeterministic machines that run in simultaneous time $O(t)$ and space $O(s)$. We use the prefix "co" to represent the complementary classes. We often use the same notation to refer to classes of machines rather than languages.

Our results are robust with respect to the choice of machine model underlying our complexity classes; for concreteness, we use the random-access machine model as described in [4]. Note that all instances of t and s in this paper are polynomials in n, so they are easily constructible.

Recall that a space-bounded nondeterministic machine does not have two-way access to its guess bits unless it explicitly writes them down on its worktape at the expense of space. It is often important for us to take a finer-grained view of such computations to separate out the resources required to write down a nondeterministic guess string from those required to verify that the guess is correct. To this end, we adopt the following notation.

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 $\forall^f \mathcal{C}$.

2.2 Tautologies versus Conondeterministic Linear Time

All known time-space lower bounds for satisfiability or tautologies hinge on the tight connection between the tautologies problem and the class of languages recognized by conondeterministic linear-time machines, coNTIME(n). Strong versions of the Cook-Levin Theorem have been formulated, showing that the tautologies problem captures the simultaneous time *and* space complexity of conondeterministic linear time on nondeterministic machines, up to polylogarithmic factors. As a consequence, time-space lower bounds for coNTIME(n) on nondeterministic machines transfer to tautologies with little loss in parameters. In particular we use the following result; see [4] for an elementary proof.

Lemma 1. *For positive reals d and e, if*

$$\text{coNTIME}(n) \not\subseteq \text{NTISP}(n^d, n^e),$$

then for any reals $d' < d$ and $e' < e$,

$$\textit{Tautologies} \notin \mathrm{NTISP}(n^{d'}, n^{e'}).$$

Since a lower bound for coNTIME(n) yields essentially the same lower bound for tautologies, we shift our focus to proving lower bounds for the former.

2.3 Indirect Diagonalization

Our proofs follow the paradigm of indirect diagonalization. The paradigm works by contradiction. In the case of Theorem 2 we assume that

$$\mathrm{coNTIME}(n) \subseteq \mathrm{NTIME}(n^c) \cap \mathrm{NTISP}(n^d, n^e). \tag{1}$$

This unlikely assumption is used to derive more and more unlikely inclusions of complexity classes, until some inclusion contradicts a known diagonalization result. The main two tools we use to derive inclusions go in opposite directions:

(a) Speed up nondeterministic space-bounded computations by adding alternations, and
(b) Eliminate these alternations via assumption (1), at a moderate increase in running time.

To envision the utility of these items, notice that (1) allows the simulation of a conondeterministic machine by a space-bounded nondeterministic machine. Item (a) allows us to simulate the latter machine by an alternating machine that runs in less time. Using item (b), the alternations can be eliminated from this simulation, increasing the running time modestly. In this way, we end up back at a nondeterministic computation, so that overall we have derived a simulation of a conondeterministic machine by a nondeterministic one. The complexity class inclusion that this simulation yields is a complementation of the form

$$\mathrm{coNTIME}(t) \subseteq \mathrm{NTIME}(f(t)), \tag{2}$$

where we seek to make f as small as possible by carefully compounding applications of (a) and (b). In fact, we know how to rule out inclusions of the type (2) for small functions f, say $f(t) = t^{1-\epsilon}$, by a folklore diagonalization argument. This supplies us with a means for deriving a contradiction.

Lemma 2. *Let a and b be positive reals such that $a < b$, then*

$$\mathrm{coNTIME}(n^b) \nsubseteq \mathrm{NTIME}(n^a).$$

Let us discuss how to achieve items (a) and (b). Item (a) is filled in by the divide-and-conquer strategy that underlies Savitch's Theorem [5]. Briefly, the idea is to divide the rows in a computation tableau of a space-bounded nondeterministic machine M into b time blocks. Observe that M accepts x in time t if and only if there are $b - 1$ configurations $C_1, C_2, \ldots, C_{b-1}$ at the boundaries of these blocks such that for every block i, $1 \le i \le b$, the configuration at the beginning of that block, C_{i-1}, can reach the configuration at the end of that block, C_i, in t/b steps, where C_0 is the initial configuration and C_b is the accepting configuration. This condition is implemented on an alternating machine to realize a speedup of M as follows. First existentially guess $b-1$ configurations of M, universally guess a block number i, and decide if C_{i-1} reaches C_i via a simulation of M for t/b steps. Thus, we can derive that

$$\mathrm{NTISP}(t,s) \subseteq \exists^{bs}\forall^{\log b}\mathrm{NTISP}(t/b,s). \tag{3}$$

The above simulation runs in overall time $O(bs+t/b)$. Choosing $b = O(\sqrt{t/s})$ minimizes this running time, to $O(\sqrt{ts})$. However, this minimization produces suboptimal results in our arguments. Instead, we apply (3) for an unspecified b and choose the optimal value after all of our derivations.

Let us point out one important fact about the simulation underlying (3). The final phase of this simulation, that of simulating M for t/b steps, does not need access to all of the configurations guessed during the initial existential phase — it only reads the description of two configurations, C_{i-1} and C_i, in addition to the original input x. Thus, the input size of the final stage is $O(n+s)$, as opposed to $O(n+bs)$ as the complexity-class inclusion of (3) suggests in general. This fact has a subtle but key impact in our lower bound proof.

We now turn to item (b), that of eliminating the alternations introduced by (3). In general, eliminating alternations comes at an exponential cost. However, in our case we are armed with assumption (1). The assumption $\mathrm{coNTIME}(n) \subseteq \mathrm{NTIME}(n^c)$ allows us to eliminate an alternation at the cost of raising the running time to the power of c. Alternately, assuming $\mathrm{coNTIME}(n) \subseteq \mathrm{NTISP}(n^d, n^e)$ allows us to eliminate an alternation at the cost of raising the running time to the power of d while at the same time maintaining the space restriction of $O(n^e)$ on the final stage. We use both of these ideas in our argument.

3 Proof of the Lower Bound

We begin with a brief discussion of the strategy used to prove the condition $(c^2-1)d < c$ of [3]. The relevant technical lemma from [3] can be thought of as trading space for time within NP under the indirect diagonalization assumption (1). More precisely, it tries to establish

$$\mathrm{NTISP}(t,s) \subseteq \mathrm{NTIME}(g(t,s)) \tag{4}$$

for the smallest possible functions g, with the hope that $g(t,s) \ll t$. In particular, for subpolynomial space bounds ($s = t^{o(1)}$) and sufficiently large polynomial t, [3] achieves $g = t^{c-1/c+o(1)}$,

$$\mathrm{NTISP}(t,t^{o(1)}) \subseteq \mathrm{NTIME}(t^{c-1/c+o(1)}), \tag{5}$$

which is smaller than t when $c < \phi \approx 1.618$.

As an example of the utility of inclusion (5), let us sketch the $n^{\sqrt{2}-o(1)}$ lower bound of [3] for subpolynomial-space nondeterministic algorithms solving tautologies. We assume, by way of contradiction, that

$$\text{coNTIME}(n) \subseteq \text{NTISP}(n^c, n^{o(1)}). \tag{6}$$

Then, for sufficiently large polynomials t, we have that:

$$\begin{aligned} \text{coNTIME}(t) &\subseteq \text{NTISP}(t^c, t^{o(1)}) && \text{[by assumption (6)]} \\ &\subseteq \text{NTIME}(t^{c^2-1+o(1)}) && \text{[by trading space for time using (5)].} \end{aligned}$$

This contradicts Lemma 2 when $c < \sqrt{2}$, yielding the desired lower bound.

The space-for-time inclusion (5) is shown by an inductive argument that derives statements of the type (4) for a sequence of smaller and smaller running times $\{g_\ell\}$. The idea can be summarized as follows. We start with a space-bounded nondeterministic machine and apply the speedup (3), yielding

$$\text{NTISP}(t, s) \subseteq \underbrace{\exists^{bs}\, \forall^{\log b}\, \underbrace{\text{NTISP}(t/b, s)}_{\text{(7a)}}}_{\text{(7b)}}. \tag{7}$$

The inductive hypothesis is then applied to trade the space bound of the final stage (7a) of this Σ_3-simulation for time:

$$\text{NTISP}(t, s) \subseteq \exists^{bs}\forall^{\log b}\text{NTIME}(g_{\ell-1}(t/b, s)).$$

Finally, we use assumption (6) to eliminate the two alternations in this simulation, ending up with another statement of the form

$$\text{NTISP}(t, s) \subseteq \text{NTIME}(g_\ell(t, s)).$$

Notice that the above argument does not rely on the space bound in (6); the weaker assumption that $\text{coNTIME}(n) \subseteq \text{NTIME}(n^c)$ is enough to eliminate the alternations introduced by the speedup. Our new argument does exploit the fact that when we transform (7a) using the assumption (6), we eliminate an alternation *and* re-introduce a space-bound. This allows us to apply the inductive hypothesis for a *second time* and trade the space bound for a speedup in time once more. This way, we hope to eliminate the alternation in (7b) more efficiently than before, yielding a smaller g_ℓ after completing the argument.

Some steps of our argument exploit the space bound while others do not. We allow for different parameters in those two types of steps; we assume

$$\text{coNTIME}(n) \subseteq \text{NTISP}(n^c) \cap \text{NTISP}(n^d, n^{o(1)}),$$

where $d \geq c \geq 1$. The success of our approach to eliminate the alternation in (7b) now depends on how large d is compared with c. If d is close to c, then the increased cost of complementing via the space-bounded assumption is counteracted by the benefit of trading this space bound for time.

Two key ingredients that allow the above idea to yield a quantitative improvement for certain values of c and d are (i) that the conondeterministic guess at the beginning of stage (7b) is only over $\log b$ bits and (ii) the fact mentioned in Section 2 that (7a) has input size $O(n+s)$. Because of (i), the running time of (7b) is dominated by that of (7a), allowing us to reduce the cost of simulating (7b) without an alternation by reducing the cost of complementing (7a) into coNP. Item (ii) is important for the latter task because the effective input size for the computation (7a) is much smaller than the $O(n+bs)$ bits taken by (7b); in particular, it does not increase with b. This allows the use of larger block numbers b to achieve greater speedups while maintaining that the final stage runs in time at least linear in its input. The latter behavior is crucial in allowing alternation removal at the expected cost — raising the running time to the power of c or d — because we can pad the indirect diagonalization assumption (1) up (to superlinear time) but not down (to sublinear time).

Now that we have sketched the intuition and key ingredients, we proceed with the actual argument. The following lemma formalizes the inductive process of speeding up nondeterministic space-bounded computations on space-unbounded nondeterministic machines.

Lemma 3. *If*

$$\mathrm{coNTIME}(n) \subseteq \mathrm{NTIME}(n^c) \cap \mathrm{NTISP}(n^d, n^e)$$

for some reals c, d, and e then for every nonnegative integer ℓ, time function t, and space function $s \le t$,

$$\mathrm{NTISP}(t,s) \subseteq \mathrm{NTIME}\left((ts^{\ell})^{\gamma_\ell} + (n+s)^{a_\ell}\right),$$

where $\gamma_0 = 1$, $a_0 = 1$, and γ_ℓ and a_ℓ are defined recursively for $\ell > 0$ as follows: Let

$$\mu_\ell = \max(\gamma_\ell(d+e\ell), ea_\ell), \tag{8}$$

then

$$\gamma_{\ell+1} = c\gamma_\ell\mu_\ell/(1+\gamma_\ell\mu_\ell), \tag{9}$$

and

$$a_{\ell+1} = ca_\ell \cdot \max(1, \mu_\ell). \tag{10}$$

Proof. The proof is by induction on ℓ. The base case $\ell = 0$ is trivial. To argue the inductive step, $\ell \to \ell+1$, we consider a nondeterministic machine M running in time t and space s and construct a faster simulation at the cost of sacrificing the space bound. We begin by simulating M in the third level of the polynomial-time hierarchy via the speedup (3) using $b > 0$ blocks (to be determined later); this simulation is in

$$\exists^{bs}\forall^{\log t}\underbrace{\mathrm{NTISP}(t/b, s)}_{\text{(11a)}}. \tag{11}$$

We focus on simulating the computation of (11a). Recall the input to (11a) consists of the original input x of M as well as two configuration descriptions of size $O(s)$, for a total input size of $O(n+s)$. The inductive hypothesis allows the simulation of (11a) in

$$\mathrm{NTIME}\left(\left(\frac{t}{b}s^{\ell}\right)^{\gamma_\ell} + (n+s)^{a_\ell}\right). \tag{12}$$

In turn, this simulation can be complemented while simultaneously introducing a space bound via the assumption of the lemma; namely, (12) is in

$$\mathrm{coNTISP}\left(\left(\left(\frac{t}{b}s^{\ell}\right)^{\gamma_\ell} + (n+s)^{a_\ell}\right)^{d}, \left(\left(\frac{t}{b}s^{\ell}\right)^{\gamma_\ell} + (n+s)^{a_\ell}\right)^{e}\right),$$

where here the $(n+s)^{a_\ell}$ term subsumes the $O(n+s)$ term from the input size because $a_\ell \geq 1$. The space bound allows for a simulation via the inductive hypothesis once more, yielding a simulation of (11a) in

$$\begin{gathered}\mathrm{coNTIME}\left(\left(\left(\tfrac{t}{b}s^{\ell}\right)^{\gamma_\ell} + (n+s)^{a_\ell}\right)^{\gamma_\ell(d+e\ell)} + \left(n+s+\left(\left(\tfrac{t}{b}s^{\ell}\right)^{\gamma_\ell} + (n+s)^{a_\ell}\right)^{e}\right)^{a_\ell}\right) \\ \subseteq \mathrm{coNTIME}\left(\left(\tfrac{t}{b}s^{\ell}\right)^{\gamma_\ell\mu_\ell} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell}\right).\end{gathered} \tag{13}$$

Replacing (11a) in (11) by (13) eliminates an alternation, lowering the simulation of M to the second level of the polynomial hierarchy:

$$\exists^{bs}\underbrace{\forall^{\log t}\mathrm{coNTIME}\left(\left(\frac{t}{b}s^{\ell}\right)^{\gamma_\ell\mu_\ell} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell}\right)}_{\text{(14a)}} \tag{14}$$

We now complement the conondeterministic computation of (14a) via the assumption that $\mathrm{NTIME}(n) \subseteq \mathrm{coNTIME}(n^c)$, eliminating one more alternation. Since (14a) takes input of size $O(n+bs)$, this places (14) in

$$\begin{aligned}&\exists^{bs}\mathrm{NTIME}\left(\left(\left(\tfrac{t}{b}s^{\ell}\right)^{\gamma_\ell\mu_\ell} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell} + (bs+n)\right)^{c}\right) \\ &\subseteq \mathrm{NTIME}\left(\left(\underbrace{\left(\frac{t}{b}s^{\ell}\right)^{\gamma_\ell\mu_\ell}}_{\text{(15a)}} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell} + \underbrace{bs}_{\text{(15b)}}\right)^{c}\right),\end{aligned} \tag{15}$$

where the inclusion holds by collapsing the adjacent existential phases (and the time required to guess the $O(bs)$ configuration bits is accounted for by the observation that $c \geq 1$).

We have now given a simulation of $\mathrm{NTISP}(t,s)$ in $\mathrm{NTIME}(\cdot)$; all that remains is to choose the parameter b. Notice that the running time of (15) has one term, (15b), that increases with b and one term, (15a), that decreases with b. The running time is minimized up to a constant factor by choosing b to equate the two terms, resulting in

$$b^* = \left(\frac{(ts^{\ell})^{\gamma_\ell\mu_\ell}}{s}\right)^{1/(1+\gamma_\ell\mu_\ell)}.$$

When this value is at least 1, the running time of the simulation (15) is

$$O\left((ts^{\ell+1})^{c\gamma_\ell\mu_\ell/(1+\gamma_\ell\mu_\ell)} + (n+s)^{ca_\ell\mu_\ell} + (n+s)^{ca_\ell}\right),$$

resulting in the recurrences (9) and (10). If $b^* < 1$, then $b = 1$ is the best we can do; the desired bound still holds since in this case (15a) + (15b) $= O(s)$, which is dominated by $(n+s)^{a_{\ell+1}}$. □

Applying Lemma 3, we deduce that for large enough polynomial τ,

$$\text{coNTIME}(\tau) \subseteq \text{NTISP}(\tau^d, \tau^e) \subseteq \text{NTIME}(\tau^{(d+e\ell)\gamma_\ell} + \tau^{ea_\ell}) = \text{NTIME}(\tau^{\mu_\ell}), \tag{16}$$

which is a contradiction with Lemma 2 when $\mu_\ell < 1$. We now determine values of c, d, and e that imply this contradiction, focusing on small values of e.

Theorem 3. *For all reals c and d such that $c^2 d < 4$ there exists a positive real e such that*

$$\text{coNTIME}(n) \nsubseteq \text{NTIME}(n^c) \cap \text{NTISP}(n^d, n^e).$$

Proof. The case where either $c < 1$ or $d < 1$ is ruled out by Lemma 2. For $c \geq 1$ and $d \geq 1$, assume (by way of contradiction) that

$$\text{coNTIME}(n) \subseteq \text{NTIME}(n^c) \cap \text{NTISP}(n^d, n^e)$$

for a value of e to be determined later. As noted above, the theorem's assumption in conjunction with Lemma 3 yields the complementation (16) for any integer $\ell \geq 0$ and sufficiently large polynomial bound τ.

Our goal is now to characterize the behavior of μ_ℓ in terms of c, d, and e. This task is facilitated by focusing on values of e that are small enough to smooth out the complex behavior of μ_ℓ caused by (i) the appearance of the nonconstant term $e\ell$ in the recurrence and (ii) its definition via the maximum of two functions.

We first handle item (i) by introducing a related, nicer sequence by substituting a real β (to be determined) as an upper bound for $e\ell$. Let

$$\mu'_\ell = \max(\gamma'_\ell(d+\beta), ea'_\ell), \tag{17}$$

where $\gamma'_0 = 1$, $a'_0 = 1$ and

$$\begin{aligned}\gamma'_{\ell+1} &= c\gamma'_\ell\mu'_\ell/(1+\gamma'_\ell\mu'_\ell), \text{ and}\\ a'_{\ell+1} &= ca'_\ell \cdot \max(1, \mu'_\ell).\end{aligned}$$

As long as β behaves as intended, i.e., $e\ell \leq \beta$, we can show by induction that $\gamma_\ell \leq \gamma'_\ell$, $a_\ell \leq a'_\ell$, and $\mu_\ell \leq \mu'_\ell$. Therefore, μ'_ℓ upper bounds μ_ℓ up to a value of ℓ that depends on e, and this ℓ-value becomes large when e is very small. This allows us to use μ'_ℓ as a proxy for μ_ℓ in our analysis.

To smooth out the behavior caused by issue (ii), we point out that the first term in the definition (17) of μ'_ℓ is larger than the second when e is very small. Provided that this is the case, μ'_ℓ equals the sequence ν_ℓ defined as follows:

$$\begin{aligned}\nu_0 &= d+\beta\\ \nu_{\ell+1} &= \nu_\ell^2 c(d+\beta)/((d+\beta)+\nu_\ell^2).\end{aligned}$$

This delivers a simpler sequence to analyze. Notice that because the underlying transformation

$$\eta \to \eta^2 c(d+\beta)/((d+\beta)+\eta^2)$$

is increasing over the positive reals, the sequence ν_ℓ is monotone in this range. It is decreasing if and only if $\nu_1 < \nu_0$, which is equivalent to $(c-1)(d+\beta) < 1$. Furthermore, when $c^2(d+\beta) < 4$, the transformation has a unique real fixed point at 0. Since the underlying transformation is also bounded and starts positively, the sequence ν_ℓ must decrease monotonically to 0 in this case.

Therefore, when $c^2 d < 4$ we can choose a positive β such that ν_ℓ becomes as small as we want for large ℓ. Provided that β, e, and ℓ satisfy the assumptions required to smooth out items (i) and (ii), this also gives us that μ_ℓ is small. More formally, let ℓ^* be the first value of ℓ such that $\nu_\ell < 1$. Item (i) requires that

$$e\ell^* \leq \beta. \tag{18}$$

Item (ii) requires that the first term of μ'_ℓ in (17) dominates the second up to this point, namely,

$$\gamma'_\ell(d+\beta) \geq ea'_\ell \text{ for all } \ell \leq \ell^*. \tag{19}$$

When all of these conditions are satisfied, we have that $\mu_{\ell^*} \leq \mu'_{\ell^*} = \nu_{\ell^*} < 1$, and the running time of the NTIME computation in (16) for $\ell = \ell^*$ is $O(\tau^{\mu_{\ell^*}}) = O(\tau^{\mu'_{\ell^*}}) = O(\tau^{\nu_{\ell^*}})$.

Therefore, by choosing a small enough positive e to satisfy the finite number of constraints in (18) and (19), we arrive at our goal of proving that $\mu_\ell < 1$ in (16). This is a contradiction, which proves the desired lower bound. □

We remark that our above analysis is tight, in the sense the above proof does not work for $c^2 d \geq 4$. The details will appear in the full version of the paper.

References

1. L. Fortnow. Time-space tradeoffs for satisfiability. *Journal of Computer and System Sciences*, 60:337–353, 2000.
2. L. Fortnow, R. Lipton, D. van Melkebeek, and A. Viglas. Time-space lower bounds for satisfiability. *Journal of the ACM*, 52:835–865, 2005.
3. L. Fortnow and D. van Melkebeek. Time-space tradeoffs for nondeterministic computation. In *Proceedings of the 15th IEEE Conference on Computational Complexity*, pages 2–13. IEEE, 2000.
4. D. van Melkebeek. A survey of lower bounds for satisfiability and related problems. *Foundations and Trends in Theoretical Computer Science*, 2:197–303, 2007.
5. W. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4:177–192, 1970.
6. R. Williams. Time-space tradeoffs for counting NP solutions modulo integers. *Computational Complexity*, 17:179–219, 2008.
7. R. Williams. Alternation-trading proofs, linear programming, and lower bounds. Manuscript, 2009.