Noname manuscript No.
(will be inserted by the editor)

An Improved Time-Space Lower Bound for Tautologies

Scott Diehl · Dieter van Melkebeek · Ryan Williams

Abstract We show that 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 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 every $d < \sqrt[3]{4}$ there exists a positive e such that tautologies cannot be decided by a nondeterministic algorithm that runs in time n^d and space n^e.

Keywords Computational Complexity, Time-Space Lower Bounds, Tautologies, Satisfiability

Scott Diehl
Department of Computer Science, Siena College
515 Loudon Road, Loudonville, NY 12211
Phone: (518)783-4198, Fax: (518)783-2986
E-mail: `sfdiehl@siena.edu`

Dieter van Melkebeek
Department of Computer Sciences, University of Wisconsin-Madison
1210 W. Dayton St., Madison, WI 53706
Phone: (608)262-4196, Fax: (608)262-9777
E-mail: `dieter@cs.wisc.edu`

Ryan Williams
School of Mathematics, Institute for Advanced Study
1 Einstein Drive, Princeton, NJ 08540
Phone: (609)734-8100, Fax: (609)951-4459
E-mail: `ryanw@ias.edu`

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. We refer to (Beame and Pitassi, 2001) for a survey of such results.

A 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 (Van Melkebeek, 2007). Similar arguments as in the deterministic setting yield somewhat weaker 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 certain nontrivial 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 (Fortnow, 2000) established a slightly superlinear time lower bound in the case of constant $e < 1$, and Fortnow and Van Melkebeek (Fortnow and Van Melkebeek, 2000) (see also (Fortnow et al., 2005)) 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 $\sqrt[3]{4} \approx 1.587$. More precisely, we obtain the following result.

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.*

Although raising the exponent from $\sqrt{2}$ to $\sqrt[3]{4}$ may seem like a step that could soon be further improved upon, there is some experimental evidence that $\sqrt[3]{4}$ may be the best exponent that can be obtained within the framework of the recent time-space lower bounds. All of the previous lower bound arguments follow the paradigm of indirect diagonalization — they start from the hypothesis that the lower bound fails and use a few simple rules (in increasingly complex patterns) to obtain a contradiction with a direct diagonalization result. This format provides structure amenable to computerized exhaustive search for better proofs. In fact, it was the outcome of such a search that spurred our current investigations. The search showed that a tiny improvement over $\sqrt{2}$ was possible, and convinced us to revisit the conondeterministic setting. This led us to a new intuitive idea, which we further developed analytically in order to determine how far beyond $\sqrt{2}$ it could take us. The limit of this technique pushes the initial tiny improvement to $\sqrt[3]{4}$, as stated in Theorem 1. Following up on our analytical work, a subsequent large automated search did not discover any proofs that lead to an exponent better than $\sqrt[3]{4}$. We refer to Williams (2009) for details of the automated searches.

The earlier result of Fortnow and Van Melkebeek (Fortnow and Van Melkebeek, 2000) 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. The precise relationship between the parameters is that for every c and d such that $(c^2-1)d < c$, there is a positive e such that the statement holds. In particular, tautologies cannot have both a nondeterministic algorithm that runs in time $n^{1+o(1)}$ and a nondeterministic algorithm that runs in logarithmic space (Fortnow, 2000). Correspondingly, our argument yields the following refinement.

Theorem 2 *For all reals c and d such that $c^2d < 4$, 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.

In order to compare the condition $c^2d < 4$ in Theorem 2 with the corresponding condition $(c^2-1)d < c$ in (Fortnow and Van Melkebeek, 2000), we include a plot of those bounds in Figure 1. Note that the interesting range of parameters satisfies $d \geq c \geq 1$. This is because an algorithm of type (ii) is a special case of an algorithm of type (i) for $d \leq c$, and that an algorithm of type (i) with $c < 1$ can be ruled out unconditionally by simple diagonalization. The condition due to this paper, $c^2d < 4$, is less restrictive for values of d not too much larger than c. Thus, Theorem 2 gives a better lower bound in this range. In particular, for $c = d$, our condition requires $d < \sqrt[3]{4} \approx 1.587$, whereas that of (Fortnow and Van Melkebeek, 2000) requires $d < \sqrt{2} \approx 1.414$; this setting yields 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 are easy in the sense that they have machines of types (i) and (ii), and aim to derive a contradiction. Fortnow and Van Melkebeek (Fortnow and Van Melkebeek, 2000) 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 (Savitch, 1970) 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

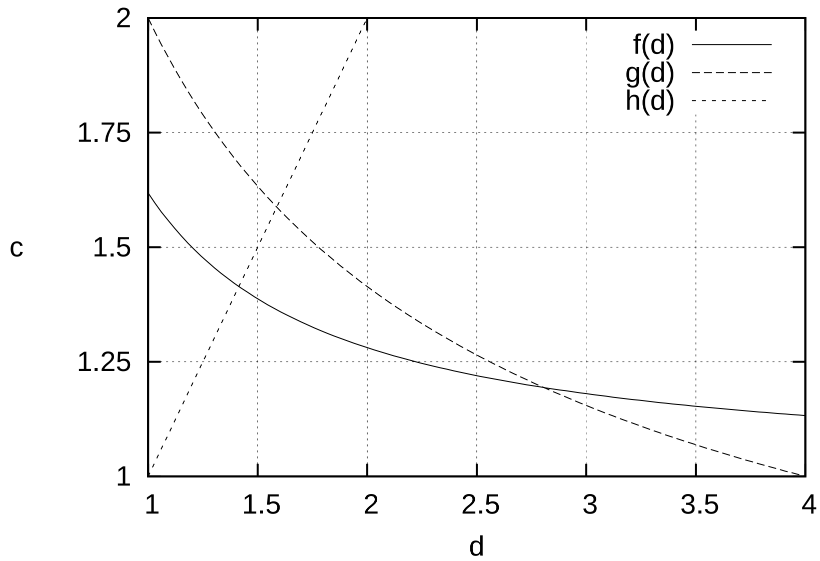

Fig. 1 Tradeoff Curves for Tautologies: Tautologies cannot have both (i) a nondeterministic algorithm that runs in time n^c and (ii) a nondeterministic algorithm that runs in time n^d and space $n^{o(1)}$. The function $f(d)$ solves for the best bound on c due to the prior results of (Fortnow and Van Melkebeek, 2000), and $g(d)$ does the same for Theorem 2; the function $h(d)$ marks the case $c = d$, illustrating the bound in Theorem 1.

induced by (ii) allows us to run another layer of alternation-based speedups and alternation eliminations. Earlier approaches to take advantage of the latter possibility failed. We show how to do it and compensate for any extra cost in running time, as long as d isn't much larger than c. Due to the additional layer involved in the argument, 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.

Our main technical contribution is the introduction of 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 are easy in the sense that they have machines of both types (i) and (ii), and we aim to derive a contradiction. Fortnow and Van Melkebeek (Fortnow and Van Melkebeek, 2000) use (ii) to obtain a time-space efficient simulation of conondeterministic computations by nondeterministic machines. Next, they speed up the space-bounded nondeterministic computation à la Savitch (Savitch, 1970) by introducing additional alternations and subsequently eliminate those extra alternations using (i) at a moderate cost in running time. When $(c^2 - 1)d < c$, the net effect is a speedup of generic conondeterministic computations on nondeterministic machines, which yields the sought-after contradiction. Note that this argument uses (ii) in a rather limited way, namely only in the very first step. Approaches to exploit (ii) in the rest of the argument have failed so far. Note that eliminating alternations using (ii) costs at least as much time as using (i) in the interesting case where $d \geq c$. On the other hand, the space bound induced by (ii) allows us to run another layer of alternation-based speedups and alternation eliminations. We show how to take advantage of the latter possibility and compensate for any extra cost in running time, provided d isn't too much larger than c. 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 the area of time-space lower bounds for satisfiability and related problems.

In the remainder of this paper, we focus on proving the condition $c^2 d < 4$ given in Theorem 2. Section 2 discusses our machine conventions and describes some key techniques. We prove the main result in Section 3. Finally, Section 4 concludes with a brief discussion of future work.

2 Preliminaries

In this section we describe some definitions, conventions, and basic techniques that we use throughout the paper.

2.1 Notation

For functions t and s we denote by $\mathrm{NTIME}(t)$ the class of languages recognized by nondeterministic machines that run in time $O(t)$, and by $\mathrm{NTISP}(t, s)$ 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 classes of 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 (Van Melkebeek, 2007). We omit constructibility considerations for the bounds t and s in this paper as our final results apply to polynomial bounds which satisfy all the constructibility properties needed.

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 work-tape 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 to date hinge on the tight connection between the tautologies problem and the class of languages recognized by conondeterministic linear-time machines, coNTIME(n). The Cook-Levin Theorem, the seminal result showing that satisfiability is NP-complete, can be interpreted as saying that satisfiability captures the time complexity of all of NP up to polynomial factors; the complement of this statement applies to the tautologies problem and coNP. Stronger versions have been formulated for various machine models, showing that tautologies 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; we refer to (Van Melkebeek, 2007) for an elementary and model independent proof.

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

$$\text{coNTIME}(n) \nsubseteq \text{NTISP}(n^d, n^e),$$

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

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

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, i.e., we begin by assuming that the desired lower bound does not hold. In the case of Theorem 2 we assume that

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

We then use this unlikely assumption to derive a series of more and more unlikely inclusions of complexity classes. The argument concludes when we derive an inclusion so unlikely that it contradicts a known diagonalization result.

Most of the challenge in formulating an indirect diagonalization argument lies in deriving new inclusions from the assumption (1). The main two tools we use towards this end 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 the assumption (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. Item (b) eliminates the alternations 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 the function 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 the aforementioned result with which we ultimately derive a contradiction.

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

$$\mathrm{coNTIME}(n^b) \not\subseteq \mathrm{NTIME}(n^a).$$

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 (Savitch, 1970). Briefly, the idea is to divide the 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, $1 \leq i \leq b$, 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: First existentially guess $b-1$ configurations of M, universally guess a block number i, and conclude by deciding 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 $b = O(\sqrt{t/s})$ optimizes this running time to $O(\sqrt{ts})$. However, minimizing the overall running time of (3) produces suboptimal results in our arguments. Instead, we apply (3) for an unspecified b and choose the optimal value after all of our derivations.

We 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 $O(n+s)$ rather than the quantity $O(n+bs)$ suggested by the general complexity-class inclusion of (3). This fact has a subtle but key impact on our analysis in Section 3.

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 the indirect diagonalization assumption (1). The assumption that $\mathrm{coNTIME}(n) \subseteq \mathrm{NTIME}(n^c)$ allows us to eliminate an alternation at the cost of raising the running time to the power of c. Alternatively, the assumption that $\mathrm{coNTIME}(n) \subseteq \mathrm{NTISP}(n^d, n^e)$ 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 techniques in our analysis.

However, it is important to point out an issue that arises in this context due to the necessity of treating the guess bits of previous alternating stages as input to the final stage: The running-time of the final stage must be linear in the original input *and* the guess bits of the previous alternating stages in order to apply the indirect diagonalization assumption. An example of accounting for this effect is as follows:

Proposition 1 *Suppose that*

$$\mathrm{coNTIME}(n) \subseteq \mathrm{NTIME}(n^c)$$

for some real $c \geq 1$. Then for any time functions t and t',

$$\exists^{t'}\mathrm{coNTIME}(t) \subseteq \mathrm{NTIME}((t+t'+n)^c).$$

Proof Consider a machine M recognizing a language in $\exists^{t'}\mathrm{coNTIME}(t)$. Its acceptance condition on input x can be written as

$$\exists y \in \{0,1\}^{O(t')} P(x,y),$$

where $P(\cdot,\cdot)$ is a predicate recognized by a conondeterministic machine running in time $O(t)$ on input $\langle x, y\rangle$. Since P takes input of size $O(n+t')$, the hypothesis allows P to be recognized by a nondeterministic machine running in time $O((t+t'+n)^c)$ by a padding argument. In this way, we can characterize the acceptance of M by two consecutive existential guesses. Thus, M can be simulated by a nondeterministic machine that requires time $O(t')$ for its guess of y and $O((t+t'+n)^c)$ for the part recognizing P, for a total of $O((t+t'+n)^c)$ since $c \geq 1$. $\square$

In a typical setting of $t = t' = n^{1+\Omega(1)}$, Proposition 1 allows us to go from the second level of the polynomial-time hierarchy to the first at the cost of increasing the running-time to the power of c, as described above. The finer point to make is that although the argument only applies the hypothesis to the final conondeterministic phase, Proposition 1 indicates that, in general, the t' guess bits of the initial phase factor into the cost of eliminating the alternation as much as the running time of the final phase does, even when the latter is much smaller. This point is where the special property of the speedup (3) becomes important, since the input to the final stage is only a small portion of the bits guessed in the initial stage, dramatically reducing the effect just described.

We now have all the tools we need to carry out our indirect diagonalization argument to prove Theorem 2.

3 Proof of the Lower Bound

We begin with a brief discussion of the techniques required to prove Fortnow and Van Melkebeek's condition of $(c^2 - 1)d < c$ (Fortnow and Van Melkebeek, 2000). We then show how to build on these techniques to arrive at our new condition $c^2 d < 4$.

The relevant technical lemma from (Fortnow and Van Melkebeek, 2000) 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, (Fortnow and Van Melkebeek, 2000) 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 the space-for-time statement represented by (5), let us sketch the $n^{\sqrt{2}-o(1)}$ lower bound of (Fortnow and Van Melkebeek, 2000; Fortnow et al., 2005) for subpolynomial-space nondeterministic algorithms solving tautologies. We assume, by way of contradiction, that

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

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

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

This is a contradiction with 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 $g = g_\ell$, $\ell = 1, 2, \ldots$. The idea can be summarized as follows: We start with a space-bounded nondeterministic machine and apply the speedup (3):

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

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

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

We conclude the inductive argument by using the assumption (6) to eliminate the two alternations in this simulation, ending up with another statement of the form

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

Notice that the above inductive argument does not rely on the space bound in (6); the weaker assumption that $\mathrm{coNTIME}(n) \subseteq \mathrm{NTIME}(n^c)$ 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 new argument exploit the space bound while others do not. In the analysis we allow for different parameters in those two assumptions, say we assume that

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

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 not too large compared 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. That our approach works better in this range of c and d makes plausible the behavior illustrated in Figure 1.

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 only $O(n + s)$. 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), as exemplified by Proposition 1.

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 $s \leq t$,

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

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 simulation with the goal of achieving a speedup 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 $(11a)$ as described above. Recall that 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

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

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

$$\mathrm{coNTISP}\left(((\frac{t}{b}s^{\ell})^{\gamma_\ell} + (n+s)^{a_\ell})^d, ((\frac{t}{b}s^{\ell})^{\gamma_\ell} + (n+s)^{a_\ell})^e\right), \tag{13}$$

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

$$\begin{aligned}\text{coNTIME}\left(((\tfrac{t}{b}s^\ell)^{\gamma_\ell} + (n+s)^{a_\ell})^{\gamma_\ell(d+e\ell)} + (n+s+((\tfrac{t}{b}s^\ell)^{\gamma_\ell}+(n+s)^{a_\ell})^e)^{a_\ell}\right) \\ \subseteq \text{coNTIME}\left((\tfrac{t}{b}s^\ell)^{\gamma_\ell\mu_\ell} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell}\right).\end{aligned} \tag{14}$$

Replacing (11a) in (11) by (14) eliminates an alternation, lowering the simulation of M to the second level of the polynomial-time hierarchy:

$$\exists^{bs}\underbrace{\forall^{\log t}\text{coNTIME}\left((\frac{t}{b}s^\ell)^{\gamma_\ell\mu_\ell} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell}\right)}_{\text{(15a)}} \tag{15}$$

We now complement the conondeterministic computation represented by (15a) via the lemma's assumption that $\text{NTIME}(n) \subseteq \text{coNTIME}(n^c)$, eliminating one more alternation in the simulation of M along the lines of Proposition 1. Specifically, since (15a) takes input of size $O(n+bs)$, this places the simulation in

$$\begin{aligned}&\exists^{bs}\text{NTIME}\left(\left((\tfrac{t}{b}s^\ell)^{\gamma_\ell\mu_\ell} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell} + (bs+n)\right)^c\right) \\ &\subseteq \text{NTIME}\left(\left(\underbrace{\left(\frac{t}{b}s^\ell\right)^{\gamma_\ell\mu_\ell}}_{\text{(16a)}} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell} + \underbrace{bs}_{\text{(16b)}}\right)^c\right),\end{aligned} \tag{16}$$

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 $c \geq 1$).

Therefore, we have arrived at a simulation that gives rise to an inclusion of $\text{NTISP}(t,s)$ in $\text{NTIME}(\cdot)$; all that remains is to choose b to optimize the running time. Notice that the running time of the simulation in (16) has one term, (16b), that increases with b and one term, (16a), that decreases with b. The running time is optimized up to a constant factor by choosing b to equate the two terms, resulting in a choice of

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

When this value is at least 1, the running time of the nondeterministic simulation (16) 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 (16a) + (16b) $= O(s)$, which is dominated by the $(n+s)^{a_{\ell+1}}$ term. $\square$

Under the assumption of Lemma 3, we can further deduce that for a sufficiently large polynomial τ,

$$\begin{aligned}\mathrm{coNTIME}(\tau) \subseteq \mathrm{NTISP}(\tau^d, \tau^e) &\subseteq \mathrm{NTIME}(\tau^{(d+e\ell)\gamma_\ell} + \tau^{ea_\ell}) \\ &= \mathrm{NTIME}(\tau^{\mu_\ell}),\end{aligned} \tag{17}$$

which is a contradiction with Lemma 2 when $\mu_\ell < 1$. Therefore, the key question is for what values of c, d, and e does μ_ℓ take on a value less than 1. Our analysis focuses on small values of e and shows how such a setting allows us to exhibit the desired behavior in μ_ℓ.

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

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

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

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

for a value of e to be determined later. As noted above, the theorem's assumption in conjunction with Lemma 3 yields the complementation (17) 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

$$\mu'_\ell = \max(\gamma'_\ell(d+\beta), ea'_\ell), \tag{18}$$

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

$$\gamma'_{\ell+1} = c\gamma'_\ell \mu'_\ell / (1 + \gamma'_\ell \mu'_\ell), \tag{19}$$

and

$$a'_{\ell+1} = ca'_\ell \cdot \max(1, \mu'_\ell). \tag{20}$$

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 (18) 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:

$$\begin{aligned}\nu_0 &= d + \beta \\ \nu_{\ell+1} &= \nu_\ell^2 c(d+\beta)/((d+\beta) + \nu_\ell^2).\end{aligned} \tag{21}$$

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

$$\eta \to \eta^2 c(d+\beta)/((d+\beta)+\eta^2) \tag{22}$$

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

$$e\ell^* \leq \beta. \tag{23}$$

Item (ii) requires that the first term in the definition (18) of μ'_ℓ dominates the second up to this point, namely,

$$\gamma'_\ell(d+\beta) \geq e a'_\ell \text{ for all } \ell \leq \ell^*. \tag{24}$$

When all of these conditions are satisfied, we have that

$$\mu_{\ell^*} \leq \mu'_{\ell^*} = \nu_{\ell^*} < 1,$$

and the running time of the nondeterministic simulation represented by (17) for $\ell = \ell^*$ runs in time

$$O(\tau^{\mu_{\ell^*}}) = O(\tau^{\mu'_{\ell^*}}) = O(\tau^{\nu_{\ell^*}}). \tag{25}$$

Therefore, by choosing a small enough positive e to satisfy the finite number of constraints in (23) and (24), we arrive at our goal of exhibiting an exponent cost in the complementation of (17) that is smaller than 1. This is a contradiction, which proves the desired lower bound. $\square$

The proof of Theorem 3 establishes $c^2 d < 4$ as a sufficient condition for our approach to work but it isn't clear that the condition is also necessary. For completeness we include an argument showing that our analysis in the proof is tight — our approach does not work for any setting with $c^2 d \geq 4$.

Referring to the notation used in the proof of Theorem 3, our approach works if we can find a value of $\mu_\ell < 1$ so that (17) contradicts Lemma 3. Without loss of generality, we can assume that the space-efficient simulation only uses logarithmic space, in which case the sequence μ_ℓ simplifies to ν_ℓ given by (21), where the term β is negligible. Since $\nu_1 \geq 1$, ν_ℓ has to decrease in order to obtain a contradiction; as this happens when $(c-1)d < 1$, we can rule out success in settings with $c^2 d \geq 4$ and $d = 1$, or $c \geq 2$. For $c^2 d = 4$ and $d > 1$, the underlying transformation (22) has a unique positive fixed point to which the sequence ν_ℓ converges, namely $cd/2 = \sqrt{d}$, which is less than d. If we let c grow, the double fixed point separates into two positive fixed points that gradually diverge from but remain centered around $cd/2$. As long as the largest of the two fixed points remains less than d, the sequence converges to it in a monotone decreasing way. At the verge where that fixed point reaches d, the sequence remains constant, which means that $(c-1)d = 1$. Beyond that the sequence monotonically increases to the larger fixed point. This argument implies that at all times $\nu_\ell \geq cd/2$. In order to have $\nu_\ell < 1$, we would need $cd < 2$, which is incompatible with our assumption that $c^2 d \geq 4$ and $d > 1$.

4 Future Work

The most compelling avenue for future work is to improve the quantitative strength of the lower bounds for tautologies. There is no a priori reason to believe that the exponent of $\sqrt[3]{4}$ could not be improved upon in the near future by making modifications to current arguments. For example, one might think that applying the inductive hypothesis more than twice per step and switching to qubic or higher-order dynamics would lead to further progress, but this seems to fail[1]. In fact, an automated search exploiting the regularity within current indirect diagonalization practices revealed no evidence of *any* proof doing better than $\sqrt[3]{4}$ Williams (2009). Therefore, we believe that an improvement to our lower bound for tautologies must involve a new approach to proving lower bounds for NP-complete and related hard problems.

Acknowledgements The first author would like to thank Bess Berg for a helpful discussion about the behavior of the sequence μ_ℓ. The first author was supported by National Science Foundation award CCF-0728809 and Cisco Systems Distinguished Graduate Fellowship while at the University of Wisconsin.

The second author was partially supported by National Science Foundation award CCF-0728809. His research was partially performed while visiting the Weizmann Institute of Science.

The third author was partially supported by National Science Foundation award CCF-0832797.

References

Beame, P. and Pitassi, T. (2001). Propositional proof complexity: Past, present, and future. In Paun, G., Rozenberg, G., and Salomaa, A., editors, *Current Trends in Theoretical Computer Science: Entering the 21st Century*, pages 42–70. World Scientific.

Fortnow, L. (2000). Time-space tradeoffs for satisfiability. *Journal of Computer and System Sciences*, 60:337–353.

Fortnow, L., Lipton, R., Van Melkebeek, D., and Viglas, A. (2005). Time-space lower bounds for satisfiability. *Journal of the ACM*, 52:835–865.

[1] The reason can be traced back to the fact that the size of the input to (7b) in (7) is the full $O(n + bs)$ rather than the smaller $O(n + s)$ taken by (7a).

Fortnow, L. and Van Melkebeek, D. (2000). Time-space tradeoffs for nondeterministic computation. In *Proceedings of the 15th IEEE Conference on Computational Complexity*, pages 2–13. IEEE.

Van Melkebeek, D. (2007). A survey of lower bounds for satisfiability and related problems. *Foundations and Trends in Theoretical Computer Science*, 2:197–303.

Savitch, W. (1970). Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences*, 4:177–192.

Williams, R. (2009). Alternation-trading proofs, linear programming, and lower bounds. Manuscript available at `http://www.cs.cmu.edu/~ryanw/automated-lbs.pdf`.