Computer Sciences Department

A New Time-Space Lower Bound for Nondeterministic Algorithms Solving Tautologies

Scott Diehl
Dieter van Melkebeek
Ryan Williams

Technical Report #1601

October 2007

UNIVERSITY OF WISCONSIN MADISON

A New Time-Space Lower Bound for Nondeterministic Algorithms Solving Tautologies

Scott Diehl *
University of Wisconsin-Madison
sfdiehl@cs.wisc.edu

Dieter van Melkebeek †
University of Wisconsin-Madison
dieter@cs.wisc.edu

Ryan Williams
Carnegie Mellon University
ryanw@cs.cmu.edu

October 6, 2007

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

1 Introduction

Tautologies is the set of Boolean formulas that are satisfied by every assignment to their variables. Lower bounds for nondeterministic algorithms deciding tautologies relate to the fundamental question of NP versus coNP. Although we expect that NP $\neq$ coNP, the best lower bound we have to date only shows that conondeterministic time t is not contained in nondeterministic time $o(t)$, and we cannot rule out even a linear-time nondeterministic algorithm for tautologies.

One way to approach this shortcoming is to show lower bounds within specific families of propositional proof systems, such as resolution. This corresponds to establishing time lower bounds for the nondeterministic algorithms that solve tautologies within those proof systems. Although the lower bounds achieved in this manner are often quantitatively very strong (see [1] for a survey), they only apply to these specific algorithms.

A different direction that has been successful at proving lower bounds focuses on nondeterministic algorithms that are only restricted in the amount of space they use. For example, one result along these lines gives a time lower bound of $n^{\sqrt{2}-o(1)} \approx n^{1.414}$ for nondeterministic algorithms that solve tautologies while using a subpolynomial amount of space, i.e., space $n^{o(1)}$ [4]. In this report, we present a new time-space lower bound for nondeterministic algorithms solving tautologies, boosting the time lower bound for subpolynomial space algorithms to $n^{\sqrt[3]{4}-o(1)} \approx n^{1.587}$. We state the result in the less restrictive setting of small-polynomial space bounds:

*Supported by NSF Career award CCR-0133693 and Cisco Systems Distinguished Graduate Fellowship.
†Partially supported by NSF Career award CCR-0133693.

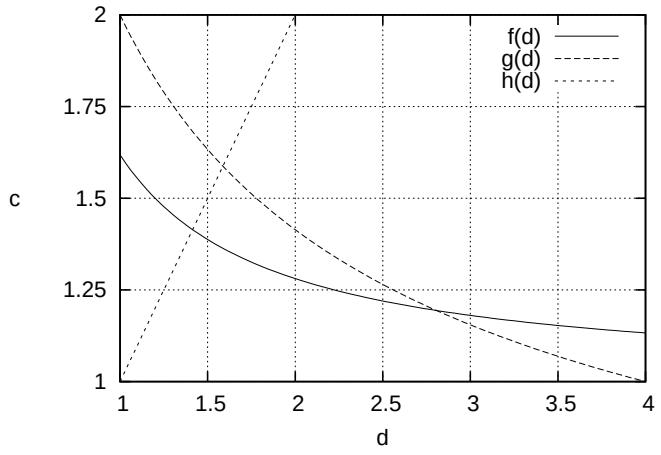

Figure 1: $f(d)$ solves the bound due to [4], $(c^2 - 1)d = c$, for c, $g(d)$ solves the bound due to Theorem 2, $c^2 d = 4$, for c, and $h(d)$ is the identity

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.*

In fact, the result of [4] 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 values c, d, and e. One particularly interesting setting for which the argument works, actually dating back to [3], is that tautologies cannot have both a nondeterministic quasilinear-time algorithm and a nondeterministic logarithmic-space algorithm. The precise relationship between the parameters given by [4] is that for every c and d such that $(c^2 - 1)d < c$, there is a positive e such that the statement holds. The main result of this report adds another condition on the running times of the algorithms in question.

Theorem 2. *For all reals c and d such that $c^2 d < 4$, there exists a positive real e such that tautologies 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 appreciate the range of parameters for which the condition given by Theorem 2 represents a stronger lower bound than that from the condition $(c^2 - 1)d < c$, we briefly explore the parameter space of c and d. A simple diagonalization argument rules out values of c or d that are smaller than 1— see Section 2, Lemma 4. Also, an algorithm of type (ii) is a special case of an algorithm of type (i) for $d \leq c$. Therefore, the interesting range of parameters has $d \geq c \geq 1$. We refer the reader to Figure 1 for a plot of the bounds involved. Notice that the condition due to this report, $c^2 d < 4$, 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 [4] requires $d < \sqrt{2} \approx 1.414$; this setting yields the improvement stated in Theorem 1.

In the remainder of the report, we focus on proving the condition $c^2d < 4$ given in Theorem 2. To do so, we first present some essential techniques and lemmas in Section 2 before proving the result in Section 3.

2 Preliminaries

In this section, we describe some conventions and definitions that we follow throughout the report, as well as describe some techniques and results that are crucial to our proofs.

2.1 Machine Model

All of our results are robust with respect to choice of machine model. For concreteness, we use the random-access machine model throughout the report. We refer to [6] for a description of this model. Our arguments skim over the details regarding the constructibility of functions involved—in the end, the functions we are interested in are polynomials, which are sufficiently constructible for our purposes.

We take this opportunity to point out an important limitation that space restrictions impose on a nondeterministic machine: although a nondeterministic machine can guess one bit per unit of running time, it does not have two-way access to these bits unless it explicitly writes them down on its worktapes. Such a machine can only remember as many of its guess bits at one time as its space bound allows. Thus, a space-bounded nondeterministic machine cannot necessarily follow the typical behavior of writing down its guess in full (say, an assignment to Boolean variables) and then verifying if this guess has a certain property (say, satisfies a given formula) while enjoying two-way access to the guess.

2.2 Notation

Much of our notation is standard, although we adopt some nonstandard abbreviations to be concise. For functions t and s, we denote $\mathrm{NT}(t)$ to be the class of languages recognized by nondeterministic machines running in time $O(t)$ and $\mathrm{NTS}(t, s)$ to be those recognized by nondeterministic machines running in simultaneous time $O(t)$ and space $O(s)$. We prefix "co" to either to represent their complementary class. We often use the same notation to refer to classes of machines rather than classes of languages; where the interpretation isn't clear from context, either should work.

In our arguments, it is often necessary to describe an alternating computation where the number of guess bits in a given phase is explicit. To this end, we adopt the following notation (see [2] for more discussion).

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

2.3 Conondeterministic Linear Time Versus Tautologies

One result that we use (as with every known time-space lower bound for satisfiability or tautologies) is a tight connection between the tautologies problem and the class of languages recognized by conondeterministic linear-time machines, coNT(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. Fortnow et al. [4] formulated a sufficiently strong version of this statement which shows that tautologies tightly captures the time *and* space complexity of *conondeterministic linear time* in a way that allows lower bounds for the latter to transfer with little loss in parameters to the former (and vice-versa).

Lemma 3 ([4]). *For positive reals d and e, if*

$$\text{coNT}(n) \not\subseteq \text{NTS}(n^d, n^e),$$

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

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

Since a lower bound for coNT(n) yields essentially the same lower bound for tautologies by Lemma 3, we focus on proving lower bounds of the former type for the remainder of the report.

2.4 Indirect Diagonalization

Our proofs follow the now-standard technique used to prove time-space lower bounds for satisfiability or tautologies known as indirect diagonalization. This is a proof by contradiction wherein we begin by assuming that the desired lower bound does not hold, which in the case of Theorem 2 is that

$$\text{coNT}(n) \subseteq \text{NT}(n^c) \cap \text{NTS}(n^d, n^e). \tag{1}$$

We then use this unlikely assumption to derive a series of more and more unlikely inclusions. 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 is in the step of deriving new inclusions from the hypothesis (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 at a moderate increase in running time via the hypothesis (1).

To envision the utility of these items, notice that the hypothesis 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

$$\text{coNT}(t) \subseteq \text{NT}(f(t)),$$

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 above type for small functions f, say, $f(t) = t^{1-\epsilon}$ by a simple diagonalization argument. This supplies us with the aforementioned result with which we ultimately derive a contradiction:

Lemma 4 (Folklore). *Let a and b be positive reals such that $a < b$, then*

$$\mathrm{coNT}(n^b) \not\subseteq \mathrm{NT}(n^a).$$

Item (a) is filled in by the divide-and-conquer strategy that underlies Savitch's Theorem [5]. Consider the definition of acceptance for a nondeterministic machine: a nondeterministic machine M running in time t and space s accepts an input x if and only if there is a computation tableau of this machine that begins with the initial configuration C_0 and ends with the accepting configuration C_A. Our simulation unfolds by stating this condition in a different way by dividing the computation tableau of this machine into b blocks: M accepts x if an 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, for $1 \leq i \leq b$, the configuration at the beginning of that block, C_{i-1}, reaches the configuration at the end of that block, C_i, in t/b steps (where we define $C_b \doteq C_A$). The latter task of verifying that one configuration of M reaches another in t/b steps now requires simulating the space-bounded nondeterministic machine M for fewer steps. We formulate an alternating machine that captures the above characterization to realize a speedup of M: First existentially guess $b-1$ configurations of M, each of which can be described by $O(s)$ bits for a total of $O(bs)$ bits, then universally guess a block number i in time $O(\log b)$, and then conclude by deciding if C_{i-1} reaches C_i via a simulation of M for t/b steps. Therefore, we have

$$\mathrm{NTS}(t,s) \subseteq \exists^{bs}\forall^{\log b}\mathrm{NTS}(t/b,s). \tag{2}$$

The above simulation runs in overall time $O(bs + t/b)$. Choosing $b = O(\sqrt{t/s})$ optimizes this running time to $O(\sqrt{ts})$, which realizes almost a square-root speedup of M when s is small, at the cost of adding two alternations. However, minimizing the overall running time of (2) actually produces suboptimal results when it is used in our arguments. This makes it necessary for our arguments to apply (2) for an unspecified b and then choose the value of b that is optimal for the particular derivation in question.

We point out one important fact about the simulation underlying (2): The final phase of this simulation, that of simulating M for t/b steps, reads the description of two configurations, C_{i-1} and C_i, in addition to the original input x (as well as the description of the block number i, whose size is negligible). Thus, the input size of the final stage is $O(n+s)$ as opposed to $O(n+bs)$ as the complexity-class inclusion of (2) interpreted in its full generality would suggest. 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 (2). In general, eliminating alternations comes at an exponential cost. However, in our case we are armed with the hypothesis (1). The assumption that $\mathrm{coNT}(n) \subseteq \mathrm{NT}(n^c)$ allows us to eliminate an alternation at the cost of raising the running-time to the power of c. Alternatively the assumption that $\mathrm{coNT}(n) \subseteq \mathrm{NTS}(n^d, n^e)$ 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 hypothesis. An example of accounting for this effect is as follows:

Proposition 5. *Suppose that*

$$\mathrm{coNT}(n) \subseteq \mathrm{NT}(n^c)$$

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

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

Proof. Consider a machine M recognizing a language in $\exists^{t'}\mathrm{coNT}(t)$. 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 $c \geq 1$. □

In a typical setting of $t = t' = n^{1+\Omega(1)}$, Proposition 5 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 5 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 (2) 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 proof Theorem 2.

3 Proof of the Lower Bound

We prove Theorem 2 in this section, beginning with a brief discussion of the techniques required to prove the $(c^2 - 1)d < c$ condition of [4]. We then show how to build on these techniques to arrive at the $c^2 d < 4$ condition.

The relevant technical lemma from [4] can be thought of as trading space for time within NP under the hypothesis (1). More precisely, it tries to establish

$$\mathrm{NTS}(t, s) \subseteq \mathrm{NT}(f(t, s)) \tag{3}$$

for the smallest possible functions f, with the hope that $f(t, s) \ll t$. In particular, for subpolynomial space bounds, $s = t^{o(1)}$, [4] achieves $f = t^{c-1/c+o(1)}$, which is smaller than t when $c < \phi \approx 1.618$.

As an example of the utility of the space-for-time statement, let us sketch the $n^{\sqrt{2}-o(1)}$ lower bound for subpolynomial-space nondeterministic algorithms solving tautologies mentioned in the

introduction. We assume, by way of contradiction, that $\mathrm{coNT}(n) \subseteq \mathrm{NTS}(n^c, n^{o(1)})$. Then we have that:

$$\begin{array}{rcll} \mathrm{coNT}(n) & \subseteq & \mathrm{NTS}(n^c, n^{o(1)}) & \text{[by hypothesis of indirect diagonalization]} \\ & \subseteq & \mathrm{NT}(n^{c^2-1+o(1)}) & \text{[by trading space for time using (3).]} \end{array}$$

This is a contradiction with Lemma 4 when $c < \sqrt{2}$, yielding the desired lower bound.

The space-for-time statement is shown by an inductive argument that derives statements of the type (3) for a sequence of smaller and smaller functions f_ℓ. The idea can be summarized as follows: We start with a space-bounded nondeterministic machine and apply the speedup (2).

$$\mathrm{NTS}(t,s) \subseteq \underbrace{\exists^{bs} \forall^{\log b} \underbrace{\mathrm{NTS}(t/b, s)}_{(*)}}_{(**)}.$$

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

$$\mathrm{NTS}(t,s) \subseteq \exists^{bs}\forall^{\log b}\mathrm{NT}(f_{\ell-1}(t/b, s)).$$

We conclude the inductive argument by using the assumption that $\mathrm{coNT}(n) \subseteq \mathrm{NT}(n^c)$ to eliminate the two alternations in this simulation, ending up with another statement of the form

$$\mathrm{NTS}(t,s) \subseteq \mathrm{NT}(f_\ell(t,s)).$$

Our main modification to this argument comes by noticing that it uses only the time-bounded half of the hypothesis of the indirect diagonalization argument (1). By replacing $(*)$ in the above using the time *and* space bounded half of the hypothesis, that $\mathrm{coNT}(n) \subseteq \mathrm{NTS}(n^d, n^e)$, we eliminate an alternation at the same time as re-introducing a space-bound. This allows us to apply the inductive hypothesis for a *second time* in the argument to trade this space bound for a speedup in time. Provided that the space-bounded half of the hypothesis is not too expensive, i.e., that d is not too much larger than c, the net effect is to eliminate the alternation in $(**)$ even more efficiently, yielding a smaller f_ℓ after completing the argument. That this approach works better than the previous when d is close to c makes plausible the behavior as 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 $(**)$ is only over $\log b$ bits and (ii) the fact mentioned in Section 2 that $(*)$ has input size only $O(n+s)$. Because of (i), the running time of $(**)$ is dominated by that of $(*)$, allowing us to reduce the cost of simulating $(**)$ without an alternation only by reducing the cost of simulating $(*)$ in coNT. Item (ii) is important for the latter task, as per the discussion of Proposition 5, because the input size of $(*)$ is much smaller than the $O(n+bs)$ bits taken by $(**)$; in particular, it does not increase with b.

Now that we have sketched the important ideas, we present the details.

Lemma 6. *If*

$$\mathrm{coNT}(n) \subseteq \mathrm{NT}(n^c) \cap \mathrm{NTS}(n^d, n^e)$$

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

$$\mathrm{NTS}(t,s) \subseteq \mathrm{NT}\left((ts^\ell)^{\alpha_\ell} + (n+s)^{a_\ell}\right),$$

where $\alpha_0 = 1$, $a_0 = 1$, *and* α_ℓ *and* a_ℓ *are defined recursively for* $\ell > 0$ *as follows: Let*

$$\mu_\ell = \max(\alpha_\ell(d + e\ell), ea_\ell), \tag{4}$$

then

$$\alpha_{\ell+1} \quad = \quad c\alpha_\ell\mu_\ell/(1 + \alpha_\ell\mu_\ell), \tag{5}$$

and

$$a_{\ell+1} \quad = \quad ca_\ell \cdot \max(1, \mu_\ell). \tag{6}$$

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 (2) using $b > 0$ blocks (to be determined later); this simulation is in

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

We focus on simulating the computation of $(*)$ as described above. Notice that the input to $(*)$ 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 $(*)$ in

$$\mathrm{NT}((\frac{t}{b}s^\ell)^{\alpha_\ell} + (n + s)^{a_\ell}). \tag{8}$$

In turn, this simulation can be complemented while simultaneously introducing a space bound via the hypothesis of the lemma; namely, (8) is in

$$\mathrm{coNTS}\left((\frac{t}{b}s^\ell)^{\alpha_\ell} + (n + s)^{a_\ell})^d, ((\frac{t}{b}s^\ell)^{\alpha_\ell} + (n + s)^{a_\ell})^e\right), \tag{9}$$

where here the $(n + s)^{a_\ell}$ term subsumes the $(n + s)$ 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 $(*)$ in

$$\begin{array}{c}\mathrm{coNT}\left(((\frac{t}{b}s^\ell)^{\alpha_\ell} + (n + s)^{a_\ell})^{\alpha_\ell(d+e\ell)} + (n + s + ((\frac{t}{b}s^\ell)^{\alpha_\ell} + (n + s)^{a_\ell})^e)^{a_\ell}\right) \\ \subseteq \mathrm{coNT}\left((\frac{t}{b}s^\ell)^{\alpha_\ell\mu_\ell} + (n + s)^{a_\ell\mu_\ell} + (n + s)^{a_\ell}\right).\end{array} \tag{10}$$

Replacing $(*)$ in (7) by (10) eliminates an alternation, lowering the simulation of M to the second level:

$$\underbrace{\exists^{bs}\,\forall^{\log t}\mathrm{coNT}\left((\frac{t}{b}s^\ell)^{\alpha_\ell\mu_\ell} + (n + s)^{a_\ell\mu_\ell} + (n + s)^{a_\ell}\right)}_{(**)} \tag{11}$$

We now complement the conondeterministic computation represented by $(**)$ via the non-space-bounded half of the hypothesis, eliminating one more alternation in the simulation of M along the

lines of Proposition 5. Specifically, since $(**)$ takes input of size $O(n+bs)$, this places the simulation in

$$\begin{aligned} & \exists^{bs}\mathrm{NT}\left(((\tfrac{t}{b}s^{\ell})^{\alpha_\ell\mu_\ell} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell} + (bs+n))^c\right) \\ \subseteq\ & \mathrm{NT}((\underbrace{(\frac{t}{b}s^{\ell})^{\alpha_\ell\mu_\ell}}_{(\searrow)} + (n+s)^{a_\ell\mu_\ell} + (n+s)^{a_\ell} + \underbrace{bs}_{(\nearrow)})^c), \end{aligned} \tag{12}$$

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 $\mathrm{NTS}(t,s)$ in $\mathrm{NT}(\cdot)$; all that remains is to choose b to optimize the running-time. Notice that the running time of the simulation in (12) has one term, $(\nearrow)$, that increases with b and one term, $(\searrow)$, 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

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

When this value is at least 1, the running-time of the nondeterministic simulation (12) is

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

resulting in the recurrences (5) and (6). If $b^* < 1$, then $b = 1$ is the best we can do; the desired bound still holds since in this case $(\nearrow)+(\searrow) = O(s)$, which is dominated by the $(n+s)^{a_{\ell+1}}$ term.□

Under the hypothesis of Lemma 6, we can further deduce that for a sufficiently large polynomial τ,

$$\mathrm{coNT}(\tau) \subseteq \mathrm{NTS}(\tau^d, \tau^e) \subseteq \mathrm{NT}(\tau^{(d+e\ell)\alpha_\ell} + \tau^{ea_\ell}) = \mathrm{NT}(\tau^{\mu_{\ell+1}}), \tag{13}$$

which is a contradiction with Lemma 4 when $\mu_{\ell+1} < 1$. Therefore, the key question is for what values of c, d, and e does $\mu_{\ell+1}$ 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 7. *For all reals c and d such that $c^2d < 4$ there exists a positive real e such that*

$$\mathrm{coNT}(n) \not\subseteq \mathrm{NT}(n^c) \cap \mathrm{NTS}(n^d, n^e).$$

Proof. The case where either $c < 1$ or $d < 1$ is ruled out by Lemma 4. For $c \geq 1$ and $d \geq 1$, assume (by way of contradiction) that

$$\mathrm{coNT}(n) \subseteq \mathrm{NT}(n^c) \cap \mathrm{NTS}(n^d, n^e)$$

for a value of e to be determined later. As noted above, the hypothesis in conjunction with Lemma 6 yields the complementation (13) 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

$$\mu'_\ell = \max(\alpha'_\ell(d+\beta), ea'_\ell), \tag{14}$$

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

$$\alpha'_{\ell+1} = c\alpha'_\ell\mu'_\ell/(1+\alpha'_\ell\mu'_\ell), \tag{15}$$

and

$$a'_{\ell+1} = ca'_\ell \cdot \max(1, \mu'_\ell). \tag{16}$$

As long as β behaves as intended, i.e., that $e\ell \le \beta$, we can show by induction that $\alpha_\ell \le \alpha'_\ell$, $a_\ell \le a'_\ell$, and $\mu_\ell \le \mu'_\ell$. 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 (14) of μ'_ℓ is larger than the second when e is very small. Provided that this is the case, μ'_ℓ follows the sequence ν_ℓ defined as follows:

$$\begin{array}{rcl} \nu_0 & = & d+\beta \\ \nu_{\ell+1} & = & \nu_\ell^2 c(d+\beta)/((d+\beta)+\nu_\ell^2). \end{array} \tag{17}$$

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 reals, the sequence ν_ℓ is monotone. 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 $\nu_{\ell+1} < 1$. Then to satisfy item (i), we require that

$$e(\ell^*+1) \le \beta. \tag{18}$$

For item (ii), we require that the first term in the definition (14) of μ'_ℓ dominates the second up to this point, namely,

$$\alpha'_\ell(d+\beta) \ge er'_\ell \text{ for all } \ell \le \ell^*+1. \tag{19}$$

When all of these conditions are satisfied, we have that

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

and the running-time of the conondeterministic simulation represented by (13) for $\ell = \ell^*$ runs in time

$$O(\tau^{\mu_{\ell^*+1}}) = O(\tau^{\mu'_{\ell^*+1}}) = O(\tau^{\nu_{\ell^*+1}}). \tag{20}$$

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 exhibiting an exponent cost in the complementation of (13) that is smaller than 1. This is a contradiction, which proves the desired lower bound. □

The proof of Theorem 7 shows that $c^2 d < 4$ is a sufficient condition for our approach to work. It turns out that our strategy does not work for $c^2 d \geq 4$. The analysis becomes more involved but shows that our argument does not yield a contradiction anywhere in this range of parameters. We include a brief explanation that applies even for logarithmic space bounds. The logarithmic-space setting has the benefit of simplifying the sequences involved in the analysis: in this case, we have that $\mu_\ell = \nu_\ell$ for $\beta = o(1)$; since its effect is negligible, we can assume $\beta = 0$ in the following discussion. We start out by observing that if μ_ℓ is not decreasing, we cannot hope to use Lemma 6 to obtain a contradiction. Settings with $c^2 d \geq 4$ and $d = 1$, or $c \geq 2$ are incompatible with the condition $(c-1)d < 1$ required for μ_ℓ to be decreasing. For $c^2 d = 4$ and $d > 1$, the underlying transformation has a unique positive fixed point to which the sequence 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 $\mu_\ell \geq cd/2$. In order to have $\mu_\ell < 1$ (which we require for contradiction in (13)), we would need $cd < 2$, which is incompatible with our assumption that $c^2 d \geq 4$ and $d > 1$.

Acknowledgements

The first author would like to thank Bess Berg for a helpful discussion about the behavior of the sequence μ_ℓ.

References

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

[2] S. Diehl and D. van Melkebeek. Time-space lower bounds for the polynomial-time hierarchy on randomized machines. *SIAM Journal on Computing*, 36:563–594, 2006.

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

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

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

[6] 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.