Electronic Colloquium on Computational Complexity, Report No. 28 (2000)

Time-Space Tradeoffs for Nondeterministic Computation

Lance Fortnow
NEC Research Institute
4 Independence Way
Princeton, NJ 08540, USA
fortnow@research.nj.nec.com

Dieter van Melkebeek
DIMACS
96 Frelinghuysen Road
Piscataway, NJ 08854-8018, USA
dieter@dimacs.rutgers.edu.

April 16, 2000

Abstract

We show new tradeoffs for satisfiability and nondeterministic linear time. Satisfiability cannot be solved on general purpose random-access Turing machines in time $n^{1.618}$ and space $n^{o(1)}$. This improves recent results of Fortnow and of Lipton and Viglas.

In general, for any constant a less than the golden ratio, we prove that satisfiability cannot be solved in time n^a and space n^b for some positive constant b. Our techniques allow us to establish this result for $b < \frac{1}{2}(\frac{a+2}{a^2} - a)$. We can do better for a close to the golden ratio, for example, satisfiability cannot be solved by a random-access Turing machine using $n^{1.46}$ time and $n^{.11}$ space. We also show the first nontrivial lower bounds for nondeterministic linear time machines using sublinear space. For example, there exists a language computable in nondeterministic linear time and $n^{.619}$ space that cannot be computed in deterministic $n^{1.618}$ time and $n^{o(1)}$ space.

Higher up the polynomial-time hierarchy we can get better bounds. We show that linear-time Σ_ℓ-computations require essentially n^ℓ time if we only allow $n^{o(1)}$ space. We also show new lower bounds on conondeterministic versus nondeterministic computation.

ISSN 1433-8092

1 Introduction

Proving lower bounds remains the most difficult of tasks in computational complexity theory. While we expect problems like satisfiability to take time $2^{\Omega(n)}$, we do not know how to prove that no linear-time algorithms exist on random-access Turing machines. In recent years we have seen a new approach to show that problems like satisfiability require a nonminimal amount of time or space. This technique assumes that satisfiability can be solved in a small amount of both time and space and uses careful simulations to allow a contradiction by diagonalization.

Fortnow [4] used this approach to give the first nontrivial time-space tradeoffs for satisfiability on random-access Turing machines. He showed that there is no algorithm solving satisfiability using $n^{1+o(1)}$ time and $n^{1-\epsilon}$ space for any fixed $\epsilon > 0$ on general purpose random-access machines.

Lipton and Viglas [10] showed how to use the nondeterministic time-hierarchy to show that satisfiability has no algorithm using n^a time and $n^{o(1)}$ space for $a < \sqrt{2}$. Their paper also claimed to improve to result to $a < 2$ by using a recursive variation of their proof but the authors have since withdrawn this claim [9]. We show that no proof of that form can improve upon their $\sqrt{2}$ result.

We use a slightly different technique to indeed improve the Lipton-Viglas result and show that satisfiability has no algorithm using n^a time and $n^{o(1)}$ space for a less than the golden ratio, about 1.618. Our trick consists of reversing the order of alternation in the recursive Lipton-Viglas argument. An easier diagonalization argument than the nondeterministic time hierarchy theorem suffices for the arguments, namely the fact that nondeterministic time t cannot be solved in time $o(t)$ on conondeterministic machines.

We also obtain direct tradeoffs between time and space. We show that satisfiability cannot be solved in time n^a and space n^b if $b < \frac{1}{2}(\frac{a+2}{a^2} - a)$. For certain values of a up to the golden ratio, we obtain stronger results. For example, satisfiability cannot be solved by any random-access Turing machine using $n^{1.46}$ time and $n^{.11}$ space. In general, for any constant a less than the golden ratio, a machine solving satisfiability in time n^a requires space n^b for some positive constant b depending on a. These tradeoffs generalize both the work of Fortnow [4] and Lipton-Viglas [10].

Satisfiability is complete for nondeterministic quasilinear time under reductions that use quasilinear time and logarithmic space [3]. Up to polylogarithmic factors we can consider results about satisfiability as results about nondeterministic linear time and vice-versa. We can get lower bounds for nondeterministic linear time even if we restrict the amount of space we use. We show that for every a less than the golden ratio there exists a language computable in linear time and $n^{1/a}$ space that is not computable in n^a time and $n^{o(1)}$ space. We can also get tradeoffs between the various time and space parameters. These are the first nontrivial time-space tradeoffs for nondeterministic linear-time machines with sublinear space.

We also show lower bounds for conondeterministic time, though the bounds are slightly weaker. We prove that conondeterministic linear time cannot be simulated on a nondeterministic machine using n^a time and $n^{o(1)}$ space for $a < \sqrt{2}$.

We establish new bounds for languages higher in the polynomial-time hierarchy as well. We show that $\Sigma_\ell\mathcal{TIME}[n]$ cannot be solved in deterministic time n^a and space $n^{o(1)}$ for $a < \ell$.

1.1 Technical Summary

We now give an informal description of the techniques that we apply in this paper. First we describe the basic idea of the recent work of Lipton and Viglas [10]. We use the weaker diagonalization argument mentioned above rather than the nondeterministic time hierarchy theorem used by Lipton and Viglas. The latter does not yield stronger results.

Consider the tableau of a Turing machine, i.e., a representation of the entire computation of the Turing machine which consists of a description of the configuration of the machine at each time step. A configuration consists of the work tapes, state and head positions of the machine. We exclude the read-only input tape and the write-only output tape. Each configuration has size $O(s(n))$ and we have $O(t(n))$ configurations for a machine that uses time $t(n)$ and space $s(n)$.

Note that this machine accepts if there are configurations $C_0, \ldots, C_{\sqrt{t(n)}}$ such that C_0 is the initial configuration, $C_{\sqrt{t(n)}}$ is an accepting configuration and for each i, C_{i-1} can get to C_i in $\sqrt{t(n)}$ steps. This observation goes back to Nepomnjaščiĭ [11] and played an important role in works of Kannan [8] and Fortnow [4].

Suppose that we have that nondeterministic linear time is computable deterministically in $O(n^a)$ time and say $O(\log n)$ space. By padding and complementation we have conondeterministic quadratic time is in $O(n^{2a})$ time and $O(\log n)$ space deterministically. Suppose we are given a potential list of configurations $C_0, \ldots, C_{\sqrt{n^{2a}}}$ of the deterministic machine as described above. We can verify that list is incorrect in nondeterministic time $O(n^a)$ by guessing an i such that C_{i-1} does not generate C_i in $\sqrt{n^{2a}}$ steps.

By assumption this verification can be done deterministically in time $O((n \log n)^{a^2})$. The whole simulation can now be done nondeterministically in time $O((n \log n)^{a^2})$ by guessing the C_i's and verifying that this list is not incorrect. If $a < \sqrt{2}$ we have simulated conondeterministic quadratic time in nondeterministic subquadratic time, a contradiction.

Lipton and Viglas try to increase a by recursively splitting the computation but their calculations are flawed. We show that simply using recursion will not help increase a. We also show that some recursion with changing the quantifier order can help to increase a and we improve the result to where $a(a-1) < 1$, i.e., a is less than the golden ratio, about 1.618.

The Lipton-Viglas argument requires small space for the deterministic machine because one needs to guess the C_i's and if these are large then it takes a lot of time to guess them. We show one can have a nontrivial tradeoff here.

An analysis of our proofs shows that they do not need the full power of nondeterministic linear time. We argue that the lower bounds apply to problems nondeterministically computable in linear time with only n^{a-1} space for a the golden ratio. We show various tradeoffs with this parameter as well.

Fortnow's paper [4] exhibits lower bounds for nondeterministic versus conondeterministic computation. Lipton-Viglas appears to work only for nondeterministic versus deterministic computation. We show how to extend their approach and obtain nondeterministic versus conondeterministic bounds, though weaker than the deterministic ones. Here recursion does help.

We establish our results for general random-access machines. If one only cares about multitape Turing machines, in which each tape head can move over only one tape cell during each computation step, one can obtain stronger time-space tradeoffs for satisfiability using simpler proofs [15]. These results follow immediately from the known time-space tradeoffs for languages like the set of palindromes. They do not rely on the inherent difficulty of nondeterministic computation but rather exploit an artefact of the multitape Turing machine model — that the machine may have to waste a lot of time in moving its tape head between both ends of the input tape.

1.2 Organization

Section 2 contains some preliminaries, including a description of the machine model and the statements of the diagonalization theorems we will be using.

In Section 3, we analyze the Lipton-Viglas approach for lower bounds for nondeterministic linear time (Sections 3.1–3.3), obtain time-space tradeoffs and show the limitations of their approach (Section 3.3). We extend their technique in various ways in Section 3.4. We establish time-space tradeoffs for languages in simultaneous nondeterministic linear time and sublinear space (Section 3.4.2) and for conondeterministic versus nondeterministic computation (Sections 3.4.1 and 3.4.3).

Section 4 contains our improvements to these results based on a new idea. We describe our trick (Section 4.1), analyze it (Section 4.2) and derive the improved time-space tradeoffs (Section 4.3).

We describe our results for linear-time Σ_ℓ-computations in Section 5.

We do not explicitly state the corollaries about time-space tradeoffs for satisfiability. They easily follow from the stated results on nondeterministic linear time using Theorem 2.4 or Corollary 2.5. See Section 2.4.

2 Preliminaries

In this section, we only describe some specific preliminaries for this paper. We refer the reader to the textbooks by Balcázar, Díaz and Gabarró [1], and by Papadimitriou [12] for general background on computational complexity and for notation.

2.1 Machine model

We use random-access Turing machines with any number of tapes. There are two types of tapes: non-index tapes and index tapes. Every non-index tape T except the output tape has an associated index tape I. The machine can move the head on T in one step to the position indexed by the contents of I.

The input tape is read-only and has an associated index tape; the output tape is write-only, has no index-tape, and is one-directional. The input and output tapes do not count towards the space usage of the machine. Non-index tapes contribute the largest position ever read (indexed) to the space usage.

2.2 Diagonalization results

Fix a positive integer ℓ. Paul, Prauß, and Reischuk [13] showed that every Σ_ℓ-machine running in time t with an arbitrary number of tapes can be simulated in time $O(t)$ by a Σ_ℓ-machine with a fixed number of tapes. The next result follows by straightforward diagonalization.

Theorem 2.1 *Let ℓ be a positive integer and $t(n)$ a time constructible function. Then*

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

Except in Section 5, we will apply Theorem 2.1 with $\ell = 1$, i.e., we will use the fact that $\mathcal{NTIME}[t] \not\subseteq \mathrm{co}\mathcal{NTIME}[o(t)]$.

$\mathcal{DTISP}[t,s]$ denotes the class of languages that can be accepted by a deterministic machine running in time $O(t)$ and space $O(s)$. $\mathcal{NTISP}[t,s]$ symbols the corresponding nondeterministic class. Fortnow and Lund [5] argued that the result of Paul, Prauß, and Reischuk also holds in the time-space bounded setting. Theorem 2.1 also carries over. We will only use the instance for $\ell = 1$:

Theorem 2.2 *Let $t(n)$ and $s(n)$ be functions that are (t,s) time-space constructible. Then*

$$\mathcal{NTISP}[t,s] \not\subseteq \mathrm{co}\mathcal{NTISP}[o(t),o(s)].$$

2.3 Constructibility Issues

Let $f(n)$, $t(n)$, and $s(n)$ be functions. We say that f is (t, s) time-space constructible if there exists a machine that outputs $1^{f(|x|)}$ on input x and runs in time $O(t)$ and space $O(s)$.

Constructibility conditions are needed when we do time and/or space bounded simulations, as in the diagonalization results above or in padding results like the following.

Theorem 2.3 *Let $f(n)$, $t(n)$, $s(n)$, $s'(n)$, and $\sigma(n)$ be functions such that $f(n) \geqslant n + 1$ and is (f, s') time-space constructible and $s' \in O(\sigma(f))$. Then*

$$\mathcal{NTISP}[n, \sigma] \subseteq \mathrm{co}\mathcal{NTISP}[t, s]$$

implies

$$\mathcal{NTISP}[f, \sigma(f)] \subseteq \mathrm{co}\mathcal{NTISP}[t(f) + f, s(f) + s'].$$

We will not explicitly state the constructibility conditions in the rest of this paper. We will tacitly assume that the bounds in the general formulations of our results satisfy these requirements. We will also assume that they are at least logarithmic, nondecreasing and do not grow too fast in the sense that $f(O(n)) \subseteq O(f(n))$. The bounds we use in our concrete results will be polynomials, which are sufficiently smooth to meet all these conditions.

2.4 Satisfiability

Cook [3], building on work of Pippenger and Fischer [14] and Hennie and Stearns [7], showed that satisfiability is complete for nondeterministic quasilinear time on multitape Turing machines under reductions that use deterministic quasilinear time and logarithmic space. Gurevich and Shelah [6], using a result of Schnorr [16], constructed efficient simulations of nondeterministic random-access Turing machines on nondeterministic multitape Turing machines. Combining the ideas of these papers, we can show the following result. We defer the proof to the Appendix.

Theorem 2.4 *There exists a constant c such that the following holds: If*

$$SAT \in \mathcal{DTISP}[t, s],$$

then

$$\mathcal{NTIME}[n] \subseteq \mathcal{DTISP}[t(n \log^c n) \cdot \log^c n, s(n \log^c n) + \log n].$$

Theorem 2.4 allows us to translate our lower bounds on nondeterministic linear time into essentially the same lower bounds for satisfiability. In particular, for polynomial bounds we obtain:

Corollary 2.5 *Let a and b be constants such that*

$$\mathcal{NTIME}[n] \not\subseteq \mathcal{DTISP}[n^a, n^b].$$

Then for any constants $a' < a$ and $b' < b$,

$$SAT \notin \mathcal{DTISP}[n^{a'}, n^{b'}].$$

The same results hold if we replace $\mathcal{DTISP}$ by $\mathcal{NTISP}$ or by $\mathrm{co}\mathcal{NTISP}$ in both the hypothesis and the conclusion.

3 Analysis of Lipton-Viglas

In this section we analyze the Lipton-Viglas approach for proving lower bounds for nondeterministic linear time. We will derive time-space tradeoffs, show the limitations of this approach, and extend the technique in various ways.

3.1 Outline

The main construction yields results of the form

$$\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTIME}[n^a] \cap \mathcal{DTISP}[t, s]$$

for certain values of the constant $a \geqslant 1$ and interesting functions $t(n)$ and $s(n)$.

The proof is by contradiction and consists of two parts. The first part uses the hypothesis that

$$\mathcal{NTIME}[n] \subseteq \mathrm{co}\mathcal{NTIME}[n^a] \tag{1}$$

and goes as follows. Consider a deterministic Turing machine M that runs in time $T(n)$ and space $S(n)$ on inputs x of length n. The following property of configurations holds for any positive integer b and time bound τ that is a multiple of b.

$$C \vdash^{\tau}_{M,x} C' \Leftrightarrow (\exists\, C_1, C_2, \ldots, C_{b-1})(\forall\, 1 \leqslant i \leqslant b)\, [C_{i-1} \vdash^{\tau/b}_{M,x} C_i], \tag{2}$$

where $C_0 = C$ and $C_b = C'$.

Using (2) recursively k times for $b = b_1(n), b_2(n), \ldots, b_k(n)$, we can construct a Σ_{2k}-machine M' that is equivalent to M and runs in time $O(\sum_{\ell=1}^{k}(b_\ell \cdot S + \log b_\ell) + (T/(b_1 \cdot b_2 \cdot \ldots \cdot b_k)))$. Under the hypothesis (1), we can transform M' into an equivalent nondeterministic machine M''. For certain settings of $a \geqslant 1$, T and S, M'' will run considerably faster than M.

The second part uses the hypothesis

$$\mathrm{co}\mathcal{NTIME}[n] \subseteq \mathcal{DTISP}[t, s] \tag{3}$$

and combines it with the first part to derive a contradiction with Theorem 2.1.

3.2 Analysis of the first part

Fix a positive integer k. For $\ell = k, k-1, \ldots, 1$, we will inductively define b_ℓ as a function of $b_1, b_2, \ldots, b_{\ell-1}$. We will pick b_ℓ so as to minimize the running time of the nondeterministic machine of the form M'' accepting

$$A_{\ell-1} \doteq \left\{ \langle x, C, C' \rangle \mid C \vdash^{T/(b_1 \cdot b_2 \cdot \ldots \cdot b_{\ell-1})}_{M,x} C' \right\}.$$

Say that we can decide A_ℓ for $0 \leqslant \ell \leqslant k$ on a nondeterministic Turing machine in time O of

$$T_l \doteq \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_\ell} \right)^{c_\ell} \cdot S^{d_\ell}.$$

We will determine the constants c_ℓ and d_ℓ inductively for $\ell = k, k-1, \ldots, 0$. At the same time, we will determine the functions b_ℓ.

For the base case, we can take $c_k = 1$ and $d_k = 0$. We will also treat $\ell = k - 1$ as a special case, but we will first deal with the inductive step $\ell \to \ell - 1$ for $1 \leqslant \ell \leqslant k - 1$.

By (2), we know that

$$\langle x, C, C'\rangle \in A_{\ell-1} \Leftrightarrow \underbrace{(\exists\, C_1, C_2, \ldots, C_{b_\ell-1})\, \underbrace{(\forall\, 1 \leqslant i \leqslant b_\ell)\, \underbrace{[\langle x, C_{i-1}, C_i\rangle \in A_\ell]}_{(\alpha)}}_{(\beta)}}_{(\gamma)}$$

(α) is a nondeterministic computation with the following parameters:

$$\begin{array}{rl} \text{input size:} & n + 2S \\ \text{running time:} & O(T_\ell). \end{array}$$

Provided

$$T_\ell \in \Omega(n + S), \tag{4}$$

(α) can be transformed using (1) and padding into a conondeterministic computation taking time $O(T_\ell^a)$.

(β) then becomes a conondeterministic computation with the following parameters:

$$\begin{array}{rl} \text{input size:} & n + b_\ell \cdot S \\ \text{running time:} & O(\log b_\ell + S + T_\ell^a). \end{array}$$

Provided that

$$T_\ell^a \in \Omega(n + b_\ell \cdot S), \tag{5}$$

(β) can be transformed using (1) (and padding) into a nondeterministic computation taking time $O(T_\ell^{a^2})$. This turns (γ) into a nondeterministic computation taking time $O(b_\ell \cdot S + T_\ell^{a^2}) = O(T_\ell^{a^2})$.

We will now compute a value for b_ℓ that minimizes this time up to a constant factor. Note $T_\ell^{a^2}$ decreases when b_ℓ increases. Condition (5) implies, up to constant factors, that

$$b_\ell \cdot S \leqslant T_\ell^a = \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_\ell}\right)^{ac_\ell} \cdot S^{ad_\ell}, \tag{6}$$

which yields an upper bound for b_ℓ. Setting b_ℓ equal to this upper bound, i.e.,

$$b_\ell = \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_{\ell-1}}\right)^{\frac{ac_\ell}{1+ac_\ell}} \cdot S^{\frac{ad_\ell - 1}{1+ac_\ell}},$$

is optimal provided conditions (4) and (5) are met. Because of the equality in (6), the latter two conditions reduce to merely condition (4).

The time needed for the nondeterministic evaluation of (γ) now becomes $O(T_{\ell-1})$ where

$$T_{\ell-1} = T_\ell^{a^2} = (b_\ell \cdot S)^a = \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_{\ell-1}}\right)^{\frac{a^2 c_\ell}{1+ac_\ell}} \cdot S^{\frac{a^2(c_\ell + d_\ell)}{1+ac_\ell}}.$$

Thus, we can set

$$\begin{cases} c_{\ell-1} &= \frac{a^2 c_\ell}{1+ac_\ell} \\ d_{\ell-1} &= \frac{a^2(c_\ell + d_\ell)}{1+ac_\ell}. \end{cases} \tag{7}$$

This finishes the inductive step.

Since $T_{\ell-1} = T_\ell^{a^2}$, $1 \leqslant \ell \leqslant k-1$, it follows that

$$T_\ell = (T_0)^{1/a^{2\ell}} = (T^{c_0} \cdot S^{d_0})^{1/a^{2\ell}}, 0 \leqslant \ell \leqslant k-1.$$

Therefore, since $b_\ell \cdot S = T_\ell^a$,

$$b_\ell = T^{\frac{c_0}{a^{2\ell-1}}} \cdot S^{\frac{d_0}{a^{2\ell-1}}-1}, 1 \leqslant \ell \leqslant k-1. \tag{8}$$

All of the conditions (4) for $1 \leqslant \ell \leqslant k-1$ are met provided

$$T^{c_0} \cdot S^{d_0} \in \Omega\left((n+S)^{a^{2(k-1)}}\right).$$

As for the remaining case $\ell \to \ell-1$ for $\ell = k$, we can skip in the above inductive step the transformation of (α) into a conondeterministic computation, since (α) already is a conondeterministic computation because it is in fact deterministic. This leads to the optimal values $c_{k-1} = d_{k-1} = a/2$ with

$$T_k = b_k \cdot S = T_{k-1}^{1/a} = (T^{c_0} \cdot S^{d_0})^{1/a^{2k-1}} \tag{9}$$

and b_k as given by (8) for $\ell = k$, under the condition that $T_k \in \Omega(n)$.

We can solve the recurrences (7). First we observe that $d_\ell = (k-\ell) \cdot c_\ell$ for $0 \leqslant \ell \leqslant k$. An explicit expression for c_ℓ for $0 \leqslant \ell \leqslant k-1$ is

$$c_\ell = \begin{cases} (a - \frac{1}{a}) \cdot \frac{1}{1+\frac{a^2-2}{a^{2(k-\ell)}}} & \text{if } a > 1 \\ 1/(k-\ell+1) & \text{if } a = 1. \end{cases} \tag{10}$$

Because of the fact that the b_ℓ's have to be positive integers, the above analysis becomes inaccurate for small b_ℓ's. Therefore, we require that $b_\ell \in \omega(1)$ for $1 \leqslant \ell \leqslant k$, which is equivalent to the condition $T_k \in \omega(S)$.

Summarizing, we have established the following lemma.

Lemma 3.1 *Suppose that $\mathcal{NTIME}[n] \subseteq \text{co}\mathcal{NTIME}[n^a]$ for some constant $a \geqslant 1$. Then for any integer $k \geqslant 1$ and functions $T(n)$ and $S(n)$,*

$$\mathcal{DTISP}[T, S] \subseteq \mathcal{NTIME}[(T \cdot S^k)^{e_k}]$$

provided that

$$(T \cdot S^k)^{e_k} \in \omega(S^{a^{2k-1}}) \cap \Omega(n^{a^{2k-1}}), \tag{11}$$

where

$$e_k = \begin{cases} (a - \frac{1}{a}) \cdot \frac{1}{1-\frac{2-a^2}{a^{2k}}} & \textit{if } a > 1 \\ 1/(k+1) & \textit{if } a = 1. \end{cases} \tag{12}$$

Note the sequence e_k in (12) starts with $e_1 = a/2$, $e_2 = a^3/(2+a^2)$, ..., and converges to $a - 1/a$.

3.3 Analysis of the second part and results

Combining Lemma 3.1 with hypothesis (3) and Theorem 2.1 yields the following result.

Theorem 3.2 *For any constant $a \geqslant 1$ and functions $t(n)$ and $s(n)$,*

$$\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTIME}[n^a] \cap \mathcal{DTISP}[t, s] \tag{13}$$

if for some integer $k \geqslant 1$

$$(t \cdot s^k)^{e_k} \in \omega(s^{a^{2k-1}}) \cap o(n), \tag{14}$$

where e_k is given by (12).

Proof Since (13) trivially holds for $t \notin \Omega(n)$, we can assume for the rest of the proof that $t \in \Omega(n)$.

Suppose that (13) fails, i.e., assume that both (1) and (3) hold. Fix an integer k for which (14) holds. Let $\tau(n) = n^{(a^{2k-1}/e_k)}$, $T(n) = t(\tau(n))$, and $S(n) = s(\tau(n))$. By hypothesis (3) and padding,

$$\mathrm{co}\mathcal{NTIME}[\tau] \subseteq \mathcal{DTISP}[T, S]. \tag{15}$$

By the choice of parameters and the fact that $t \in \Omega(n)$, $(T \cdot S^k)^{e_k} \geqslant n^{a^{2k-1}}$. Combined with the leftmost part of (14), this shows that condition (11) is met. Therefore, by Lemma 3.1,

$$\mathcal{DTISP}[T, S] \subseteq \mathcal{NTIME}[(T \cdot S^k)^{e_k}]. \tag{16}$$

Combining (15) and (16) with the rightmost part of (14), we obtain a contradiction with Theorem 2.1, namely that $\mathrm{co}\mathcal{NTIME}[\tau] \subseteq \mathcal{NTIME}[o(\tau)]$. □

In the case of only one application of (2), we obtain:

Corollary 3.3 *Let a be any constant. If $t \in \omega(s)$ and $ts \in o(n^{2/a})$, then (13) holds.*

Proof Under the given hypothesis, condition (14) is satisfied for $k = 1$. □

Corollary 3.3 yields the following lower bound on the time-space product for deterministic simulations of nondeterministic linear time.

Corollary 3.4 *Suppose that $\mathcal{NTIME}[n] \subseteq \mathcal{DTISP}[t, s]$. Then $ts \notin o(n^{\sqrt{2}})$.*

Proof Let a be any positive constant. We consider three cases:

- If $s \notin o(t)$ then $ts \notin o(n^2)$ since $t(n) \geqslant n$.
- If $t \notin O(n^a)$ then $ts \notin o(n^a)$.
- If $t \in O(n^a)$ and $t \in \omega(s)$ then by Corollary 3.3 $ts \notin o(n^{2/a})$.

The minimum of the three lower bounds is maximized for $a = \sqrt{2}$. Thus we obtain unconditionally that $ts \notin o(n^{\sqrt{2}})$. □

Taking $t(n) = n^a$ in Theorem 3.2 and Corollary 3.3 gives:

Theorem 3.5 *For any constant* $a \geqslant 1$,

$$\mathcal{NTIME}[n] \not\subseteq \mathcal{DTISP}[n^a, o(n^b)],$$

where

$$b = \max\left(\bigcup_{k \geqslant 1} \{c \mid a^{2k-1}c \leqslant (a+kc)e_k \leqslant 1\}\right) \tag{17}$$

and e_k *is defined by (12).*

In the special case of one application of (2) we get:

Corollary 3.6 *For any constant* a,

$$\mathcal{NTIME}[n] \not\subseteq \mathcal{DTISP}[n^a, o(n^{\frac{2}{a}-a})].$$

Note Corollary 3.6 is only interesting in case $1 \leqslant a < \sqrt{2}$. In particular, Corollary 3.6 implies:

Corollary 3.7 *For any constant* $a < \sqrt{2}$,

$$\mathcal{NTIME}[n] \not\subseteq \mathcal{DTISP}[n^a, n^{o(1)}]. \tag{18}$$

One might hope that multiple recursive applications of (2) would allow us to improve Corollary 3.7 and establish (18) for constants $a \geqslant \sqrt{2}$. However, this is not the case because of the following reason. Theorem 3.5 says that k recursive applications allow us to establish (18) provided that $a \cdot e_k < 1$. The sequence e_k turns out to be monotonically decreasing for $a < \sqrt{2}$ and monotonically increasing for $a > \sqrt{2}$. Therefore, increasing k cannot help us for $a \geqslant \sqrt{2}$, and we can only establish (18) this way for $a < \sqrt{2}$. In fact, a careful analysis shows that the maximum on the right-hand side of (17) is achieved for $k = 1$. Therefore, Corollary 3.6 is as powerful as Theorem 3.5, and the exponent $\frac{2}{a} - a$ in the time-space tradeoff of Corollary 3.6 cannot be improved by multiple applications of (2) for any constant a.

Note the sequence $a \cdot e_k$ converges to $(a^2 - 1)$, which is less than 1 iff $a < \sqrt{2}$.

3.4 Extensions

3.4.1 Conondeterministic Time-Space Tradeoffs

Equation (2) holds for nondeterministic Turing machines M as well. This allows us to extend Lemma 3.1 to nondeterministic computations.

Lemma 3.8 *Suppose that* $\mathcal{NTIME}[n] \subseteq \mathrm{co}\mathcal{NTIME}[n^a]$ *for some constant* $a \geqslant 1$. *Then for any integer* $k \geqslant 1$, *and functions* $T(n)$ *and* $S(n)$,

$$\mathcal{NTISP}[T, S] \subseteq \mathcal{NTIME}[(T \cdot S^k)^{f_k}]$$

provided that

$$(T \cdot S^k)^{f_k} \in \Omega((n+S)^{a^{2k}}) \cap \omega(S),$$

where

$$f_k = \begin{cases} (a - \frac{1}{a}) \cdot \frac{1}{1 - \frac{1+a-a^2}{a^{2k+1}}} & \textit{if } a > 1 \\ 1/(k+1) & \textit{if } a = 1. \end{cases} \tag{19}$$

Proof Consider the proof of Lemma 3.1 in the case that M is nondeterministic. Everything carries through except that the induction step $\ell \to \ell - 1$ for $\ell = k$ now follows the general pattern and no longer needs a special treatment. As a result, c_{k-1} and d_{k-1} are now given by the recurrence (7), i.e., $c_{k-1} = d_{k-1} = a^2/(1+a)$, and equation (9) reads

$$T_k = b_k \cdot S = T_{k-1}^{1/a^2} = (T^{c_0} \cdot S^{d_0})^{1/a^{2k}}.$$

Thus, the condition $T_k \in \Omega(n+S)$ becomes equivalent to $T^{c_0} \cdot S^{d_0} \in \Omega((n+S)^{a^{2k}})$. Since $b_k S = T_k^a$, this guarantees that $b_k \in \omega(1)$ unless $a = 1$. In that case, the condition $T_0 \in \omega(S)$ does the job. $\square$

Note the sequence f_k defined by (19) satisfies the recurrence $f_{k+1} = a^2 f_k/(1 + af_k)$. It starts out with $f_1 = a^2/(1+a)$ and converges to $a - 1/a$.

Using the nondeterministic time hierarchy theorem [17, 18] instead of Theorem 2.1, Theorem 3.2 and Corollary 3.3 translate as follows.

Theorem 3.9 *For any constant $a \geqslant 1$ and functions $t(n)$ and $s(n)$,*

$$\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTIME}[n^a] \cap \mathcal{NTISP}[t,s] \tag{20}$$

if for some integer $k \geqslant 1$

$$(t \cdot s^k)^{f_k} \in \Omega(s^{a^{2k}}) \cap \omega(s) \cap o(n),$$

where f_k is given by (19).

Corollary 3.10 *Let a be any constant. If $t \in \Omega(s^a) \cap \omega(s)$ and $ts \in o(n^{(1+a)/a^2})$, then (20) holds.*

Theorem 3.5 and its corollaries do not carry over in the same way, i.e., we do not obtain time-space tradeoffs for nondeterministic linear time on nondeterministic machines. However, we can get time-space tradeoffs for nondeterministic linear time on conondeterministic machines using the following application of Lemma 3.8 and Theorem 2.1.

Theorem 3.11 *For any constant $a \geqslant 1$ and functions $t(n)$ and $s(n)$,*

$$\mathrm{co}\mathcal{NTIME}[n] \not\subseteq \mathcal{NTIME}[n^a] \cap \mathcal{NTISP}[t,s] \tag{21}$$

if for some integer $k \geqslant 1$

$$(t \cdot s^k)^{f_k} \in \Omega(s^{a^{2k}}) \cap \omega(s) \cap o(n),$$

where f_k is given by (19).

Proof Suppose that (21) fails. Let $\tau(n) = n^{(a^{2k}/f_k)}$. By the proof of Theorem 3.2, using Lemma 3.8 instead of Lemma 3.1, we have that $\mathrm{co}\mathcal{NTIME}[\tau] \subseteq \mathcal{NTIME}[o(\tau)]$, a contradiction with Theorem 2.1. $\square$

In the case of one application of (2), Theorem 3.11 reads:

Corollary 3.12 *Let a be any constant. If $t \in \Omega(s^a) \cap \omega(s)$ and $ts \in o(n^{(1+a)/a^2})$, then (21) holds.*

Corollary 3.12 gives the following lower bound for the time-space product of nondeterministic linear time on conondeterministic machines:

Corollary 3.13 *Suppose that $\mathcal{NTIME}[n] \subseteq \mathrm{co}\mathcal{NTISP}[t, s]$. Then $ts \notin o(n^a)$ where a is the positive solution of $a(a^2 - 1) = 1$, about 1.324.*

Proof Let a be an arbitrary positive constant. We consider three cases:

- If $s \notin o(t^{1/a})$ then $ts \notin o(n^{1+1/a})$. This follows from the fact that $t(n) \geqslant n$.
- If $t \notin O(n^a)$ then $ts \notin o(n^a)$.
- If $t \in O(n^a)$ and $t \in \omega(s^a)$ then by Corollary 3.12 $ts \notin o(n^{(1+a)/a^2})$.

The minimum of the lower bounds for the three cases is maximized when $\frac{1+a}{a^2} = a$, i.e., when $a(a^2 - 1) = 1$, which yields the result. $\square$

Multiple applications of (2) allow us to establish larger lower bounds on ts^k where k denotes the number of applications.

By taking $t(n) = n^a$ in Theorem 3.11 we obtain the following conondeterministic time-space tradeoffs for nondeterministic linear time analogous to the deterministic ones of Section 3.3.

Theorem 3.14 *For any constant $a \geqslant 1$,*

$$\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTISP}[n^a, o(n^b)],$$

where

$$b = \max\left(\bigcup_{k \geqslant 1} \{c \mid a^{2k} c \leqslant (a + kc) f_k \leqslant 1\}\right)$$

and f_k is defined by (19).

In case of one application of (2) we get:

Corollary 3.15 *For any constant a,*

$$\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTISP}[n^a, o(n^{\frac{1+a}{a^2} - a})].$$

Note Corollary 3.15 is only interesting in case $a \geqslant 1$ and $\frac{1+a}{a^2} - a > 0$. The latter condition is equivalent to $a(a^2 - 1) < 1$ and implies an upper bound for a of about 1.324. This is the limit value of a for which one application of (2) allows us to establish that $\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTISP}[n^a, n^{o(1)}]$. As opposed to the deterministic case of Section 3.3, multiple recursive applications of (2) do yield better results. They let us relax the condition $a(a^2 - 1) < 1$ to $a^2 - 1 < 1$, allowing values of a up to $\sqrt{2}$.

Corollary 3.16 *For any constant $a < \sqrt{2}$,*

$$\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTISP}[n^a, n^{o(1)}]. \tag{22}$$

Proof According to Theorem 3.14, k recursive applications of (2) allow us to establish (22) for any constant a satisfying $af_k < 1$, where f_k is given by (19). The sequence af_k converges to $a^2 - 1$, which is less than 1 iff $a < \sqrt{2}$. $\square$

Note that the sequence in the proof of Corollary 3.16 is monotonically decreasing[1]. Therefore, the more applications of (2), the stronger a result of the type (22) we get.

[1] In fact, the sequence is monotonically decreasing as long as $a(a - 1) < 1$.

3.4.2 Deterministic Time-Space Tradeoffs for $\mathcal{NTISP}$

By analyzing the space needed for the transformation of the Σ_{2k}-machine into a nondeterministic machine in Lemma 3.1, we obtain the following strengthening of Lemma 3.1.

Lemma 3.17 *Suppose that $\mathcal{NTISP}[n, \sigma] \subseteq \mathrm{co}\mathcal{NTISP}[n^a, r]$ for some constant $a \geqslant 1$ and functions $\sigma(n)$ and $r(n)$. Then for any integer $k \geqslant 1$ and functions $T(n)$ and $S(n)$,*

$$\mathcal{DTISP}[T, S] \subseteq \mathcal{NTISP}[R, R^{1/a} + r(R^{1/a})],$$

where $R \doteq (T \cdot S^k)^{e_k}$ and e_k is defined by (12), provided that

$$\begin{array}{rcl} R & \in & \omega(S^{a^{2k-1}}) \cap \Omega(n^{a^{2k-1}}) \\ \sigma(R^{1/a^{2k-1}}) & \in & \Omega(S) \\ \sigma(n) & \in & \Omega(n^{1/a} + r(n^{1/a})). \end{array}$$

The last condition is not required in case $k = 1$.

In Section 3.3, we combined Lemma 3.1 with Theorem 2.1 to obtain separations of the form (13). Similarly, by combining Lemma 3.17 with Theorem 2.2 we can derive separations of the form

$$\mathcal{NTISP}[n, \sigma] \not\subseteq \mathrm{co}\mathcal{NTISP}[n^a, r] \cap \mathcal{DTISP}[t, s] \tag{23}$$

for some constants $a \geqslant 1$ and interesting functions $\sigma(n)$, $r(n)$, $s(n)$, and $t(n)$. This gives a strengthening of Theorem 3.2. Analogous strengthenings of Corollaries 3.3 and 3.4, Theorem 3.5, and Corollaries 3.6 and 3.7 also follow. Here, we only spell out the time-space tradeoffs that strengthen Corollaries 3.6 and 3.7.

Theorem 3.18 *For any constants $a \geqslant 1$ and c such that $a(a + c) < 2$,*

$$\mathcal{NTISP}[n, \omega(n^{\frac{a+c}{2}})] \not\subseteq \mathcal{DTISP}[n^a, n^c]. \tag{24}$$

Note in Theorem 3.18 the interesting values of a are in the range $1 \leqslant a < \sqrt{2}$ and of c in $0 < c < 1$. From Theorem 3.18 we get:

Corollary 3.19 *For any constant $a < \sqrt{2}$ and constant $d > a/2$,*

$$\mathcal{NTISP}[n, n^d] \not\subseteq \mathcal{DTISP}[n^a, n^{o(1)}].$$

Note Theorem 3.18 and Corollary 3.19 use only one application of (2). In Section 3.3 we mentioned that multiple applications of (2) do not allow larger values of c in (24) for a given value of a. Similarly, given a and c, multiple applications of (2) do not yield stronger statements than (24).

3.4.3 Conondeterministic Time-Space Tradeoffs for $\mathcal{NTISP}$

A space analysis of Lemma 3.8 shows the following stronger result.

Lemma 3.20 *Suppose that*

$$\mathcal{NTISP}[n, \sigma] \subseteq \mathrm{co}\mathcal{NTISP}[n^a, r] \tag{25}$$

for some constant $a \geqslant 1$ and functions $\sigma(n)$ and $r(n)$. Then for any integer $k \geqslant 1$ and functions $T(n)$ and $S(n)$,

$$\mathcal{NTISP}[T, S] \subseteq \mathcal{NTISP}[R, R^{1/a} + r(R^{1/a})], \tag{26}$$

where $R \doteq (T \cdot S^k)^{f_k}$ and f_k is defined by (19), provided that

$$\begin{aligned} R &\in \Omega((n+S)^{a^{2k}}) \cap \omega(S) \\ \sigma(R^{1/a^{2k}}) &\in \Omega(S) \\ \sigma(n) &\in \Omega(r(n^{1/a})) \\ \sigma(n) &\in \Omega(n^{1/a}). \end{aligned}$$

The last condition is not required in case $k = 1$.

The $\mathcal{NTISP}$ hierarchy theorem [2] implies that $\mathcal{NTISP}[t, s] \not\subseteq \mathcal{NTISP}[o(t), o(s)]$ provided $t(n)$ and $s(n)$ are sufficiently nice functions that don't grow too fast. Note (26) by itself does not contradict the $\mathcal{NTISP}$ hierarchy theorem for $a \geqslant 1$. For some settings of the parameters in Lemma 3.20, the time on the right-hand side of (26) is o of the time on the left-hand side, but the space on the right-hand side always exceeds the space on the left-hand side. Under the hypothesis (25), Lemma 3.20 lets us simulate nondeterministic time T and space S in less time but somewhat more space. The hypothesis (25) itself allows us to do the opposite – increase time but reduce space – at the cost of switching from nondeterministic to conondeterministic computations. By combining the two we get a contradiction with Theorem 2.2 and thereby refute hypothesis (25).

This is what happens in the strengthenings of Theorem 3.11, Corollaries 3.12 and 3.13, Theorem 3.14, and Corollaries 3.15 and 3.16, based on the use of Lemma 3.20 instead of Lemma 3.8. Here, we only state the time-space tradeoffs that strengthen Theorem 3.14 and Corollaries 3.15 and 3.16.

Theorem 3.21 *For any integer $k \geqslant 1$ and constants $a \geqslant 1$ and c such that $a^{2k}c \leqslant (a + kc)f_k < 1$ (where the first inequality has to be strict in case $a = 1$),*

$$\mathcal{NTISP}[n, n^d] \not\subseteq \text{co}\mathcal{NTISP}[n^a, n^c],$$

where $d = \max(\frac{1}{a}, \frac{a^{2k}c}{(a+kc)f_k})$ and f_k is given by (19).

Corollary 3.22 *For any constants a and c such that $a^2(a + c) < 1 + a$,*

$$\mathcal{NTISP}[n, n^d] \not\subseteq \text{co}\mathcal{NTISP}[n^a, n^c],$$

where $d = \max(\frac{1}{a}, \frac{(1+a)c}{a+c})$.

Corollary 3.23 *For any constant $a < \sqrt{2}$,*

$$\mathcal{NTISP}[n, n^{1/a}] \not\subseteq \text{co}\mathcal{NTISP}[n^a, n^{o(1)}].$$

4 Improvements

In this section, we describe our improvements to the results of Section 3. We first give the idea and then analyze it. Finally, we state the improved time-space tradeoffs we get from it.

4.1 Idea

In case of a deterministic Turing machine M, the following property of configurations holds for any positive integer b and time bound τ that is a multiple of b.

$$C \vdash^{\tau}_{M,x} C' \Leftrightarrow (\forall\, C_1, C_2, \ldots, C_{b-1}, C_b \neq C')(\exists\, 1 \leqslant i \leqslant b)\, [C_{i-1} \nvdash^{\tau/b}_{M,x} C_i], \tag{27}$$

where $C_0 = C$. Equation (27) says that C' is reachable from C in τ steps iff any tableau of size τ that starts with C and ends in a configuration C_b different from C' contains a mistake in at least one of the consecutive subtableaus of size τ/b. Note this only holds for deterministic machines M. In case M uses little space, (27) gives us another tool besides (2) to speed up space bounded computations using nondeterminism. By applying (27) k times recursively, we obtain an alternating machine N' equivalent to M, similar to the machine M' of Section 3.1. The running time of N' is the same as of M' but, for $k > 1$, N' has fewer alternations than M': N' is a Π_{k+1}-machine, whereas M' is a Σ_{2k}-machine. The number of alternations determines how often the running time of M' is raised to the power a when transforming M' into an equivalent nondeterministic machine M'' using the hypothesis that $\mathcal{NTIME}[n] \subseteq \mathrm{co}\mathcal{NTIME}[n^a]$. Therefore, when we similarly transform N' into an equivalent conondeterministic machine N'', the running time of N'' will be smaller than of M''. This will allow us to obtain stronger separation results.

4.2 Analysis

We first determine how much faster we can make N'' run than M.

Lemma 4.1 *Suppose that*

$$\mathcal{NTIME}[n] \subseteq \mathrm{co}\mathcal{NTIME}[n^a] \tag{28}$$

for some constant $a \geqslant 1$. Then for any integer $k \geqslant 1$ and functions $T(n)$ and $S(n)$,

$$\mathcal{DTISP}[T, S] \subseteq \mathcal{NTIME}[(T \cdot S^k)^{g_k}] \tag{29}$$

provided that

$$(T \cdot S^k)^{g_k} \in \omega(S^{a^k}) \cap \Omega(n^{a^{2k}}),$$

where

$$g_k = \begin{cases} (a-1) \cdot \frac{1}{1 - \frac{2-a}{a^k}} & \textit{if } a > 1 \\ 1/(k+1) & \textit{if } a = 1. \end{cases} \tag{30}$$

Proof Let M be a deterministic Turing machine running in time $T(n)$ and space $S(n)$ on inputs x of length n. Fix a positive integer k. For $\ell = k, k-1, \ldots, 1$, we will inductively define a function $b_\ell(n)$ in terms of $b_1, b_2, \ldots, b_{\ell-1}$. We will pick b_ℓ so as to minimize the running time of the conondeterministic machine of the form N'' accepting

$$A_{\ell-1} \doteq \left\{ \langle x, C, C' \rangle \mid C \vdash^{T/(b_1 \cdot b_2 \cdot \ldots \cdot b_{\ell-1})}_{M,x} C' \right\}.$$

Say that we can decide A_ℓ for $0 \leqslant \ell \leqslant k$ on a conondeterministic Turing machine in time O of

$$T_l \doteq \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_\ell} \right)^{c_\ell} \cdot S^{d_\ell}.$$

We will determine the constants c_ℓ and d_ℓ inductively for $\ell = k, k-1, \ldots, 0$. At the same time, we will determine the functions b_ℓ.

For the base case, we take $c_k = 1$ and $d_k = 0$. We now discuss the inductive step $\ell \to \ell - 1$ for $1 \leqslant \ell \leqslant k$.

By (27), we know that

$$\langle x, C, C' \rangle \in A_{\ell-1} \Leftrightarrow \underbrace{(\forall C_1, C_2, \ldots, C_{b_\ell - 1}, C_b \neq C')\, \underbrace{(\exists\, 1 \leqslant i \leqslant b_\ell)\, \underbrace{[\langle x, C_{i-1}, C_i \rangle \notin A_\ell]}_{(\alpha)}}_{(\beta)}}_{(\gamma)}$$

By the induction hypothesis, (α) is a nondeterministic computation taking time $O(T_\ell)$. Therefore, (β) is a nondeterministic computation with the following parameters:

$$\begin{aligned} \text{input size:} \quad & n + b_\ell \cdot S \\ \text{running time:} \quad & O(\log b_\ell + S + T_\ell). \end{aligned}$$

Provided that

$$T_\ell \in \Omega(n + b_\ell \cdot S) \tag{31}$$

(β) can be transformed using (28) (and padding) into a conondeterministic computation taking time $O(T_\ell^a)$. This turns (γ) into a conondeterministic computation taking time $O(b_\ell \cdot S + T_\ell^a) = O(T_\ell^a)$.

We will now compute a value for b_ℓ that minimizes this time up to a constant factor. Note T_ℓ^a decreases when b_ℓ increases. Condition (31) implies, up to constant factors, that

$$b_\ell \cdot S \leqslant T_\ell = \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_\ell} \right)^{c_\ell} \cdot S^{d_\ell}, \tag{32}$$

which yields an upper bound for b_ℓ. Setting b_ℓ equal to this upper bound, i.e.,

$$b_\ell = \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_{\ell-1}} \right)^{\frac{c_\ell}{1+c_\ell}} \cdot S^{\frac{d_\ell - 1}{1+c_\ell}},$$

is optimal provided condition (31) is met. This will be the case if

$$T_\ell \in \Omega(n). \tag{33}$$

The time needed for the conondeterministic evaluation of (γ) now becomes $O(T_{\ell-1})$ where

$$T_{\ell-1} = T_\ell^a = (b_\ell \cdot S)^a = \left(\frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_{\ell-1}} \right)^{\frac{ac_\ell}{1+c_\ell}} \cdot S^{\frac{a(c_\ell + d_\ell)}{1+c_\ell}}.$$

Thus, we can set

$$\left\{ \begin{array}{lcl} c_{\ell-1} & = & \frac{ac_\ell}{1+c_\ell} \\ d_{\ell-1} & = & \frac{a(c_\ell + d_\ell)}{1+c_\ell}. \end{array} \right. \tag{34}$$

This finishes the inductive step.

Since $T_{\ell-1} = T_\ell^a$, $1 \leqslant \ell \leqslant k$, it follows that

$$T_\ell = (T_0)^{1/a^\ell} = (T^{c_0} \cdot S^{d_0})^{1/a^\ell}, 0 \leqslant \ell \leqslant k.$$

Therefore, since $b_\ell \cdot S = T_\ell$,

$$b_\ell = T^{\frac{c_0}{a^{2\ell-1}}} \cdot S^{\frac{d_0}{a^{2\ell-1}} - 1}, 1 \leqslant \ell \leqslant k.$$

All of the conditions (33) for $1 \leqslant \ell \leqslant k$ are met provided

$$T^{c_0} \cdot S^{d_0} \in \Omega(n^{a^k}).$$

As for the recurrences (34), we observe that $d_\ell = (k-\ell)\cdot c_\ell$ for $0 \leqslant \ell \leqslant k$. An explicit expression for c_ℓ for $0 \leqslant \ell \leqslant k$ is

$$c_\ell = \begin{cases} (a-1)\cdot \frac{1}{1-\frac{2-a}{a^{k-\ell}}} & \text{if } a > 1 \\ 1/(k-\ell+1) & \text{if } a = 1. \end{cases}$$

In order to guarantee that the b_ℓ's are close enough to integers and that the above analysis is accurate, we require that $b_k \in \omega(1)$, i.e., that $T_k \in \omega(S)$.

The lemma follows by setting $g_k = c_0$. □

Note the sequence g_k defined by (30) starts with $g_1 = a/2$, $g_2 = a^2/(2+a)$, ..., satisfies the recurrence $g_{k+1} = ag_k/(1+g_k)$, and converges to $a-1$.

Analyzing the space need in Lemma 4.1 as we did for Lemma 3.1 in Section 3.4.2, we obtain the following strengthening.

Lemma 4.2 *Suppose that*

$$\mathcal{NTISP}[n, \sigma] \subseteq \mathrm{co}\mathcal{NTISP}[n^a, r]$$

for some constant $a \geqslant 1$ and functions $\sigma(n)$ and $r(n)$. Then for any integer $k \geqslant 1$ and functions $T(n)$ and $S(n)$,

$$\mathcal{DTISP}[T, S] \subseteq \mathcal{NTISP}[R, R^{1/a} + r(R^{1/a})],$$

where $R \doteq (T \cdot S^k)^{g_k}$ and g_k is defined by (30), provided that

$$\begin{aligned} R &\in \omega(S^{a^k}) \cap \Omega(n^{a^{2k}}) \\ \sigma(R^{1/a^{2k}}) &\in \Omega(S) \\ \sigma(n) &\in \Omega(n^{1/a} + r(n^{1/a})). \end{aligned}$$

The last condition is not required in case $k = 1$.

4.3 Results

We only state the consequences of Lemma 4.1 and Lemma 4.2 that improve upon the results of Section 3. In particular, we will not spell out any result for one level of recursion, since for $k = 1$ Lemma 4.1 and Lemma 4.2 are equivalent to Lemma 3.1 and Lemma 3.17 respectively.

We could state all results in the stronger $\mathcal{NTISP}$ version, using Lemma 4.2 instead of Lemma 4.1. However, as in Section 3.4.2, we will refrain from describing general $\mathcal{NTISP}$ results of the form (23) because they involve too many parameters. We will state the weaker $\mathcal{NTIME}$ versions instead.

Theorem 4.3 *For any constant $a \geqslant 1$ and functions $t(n)$ and $s(n)$,*

$$\mathcal{NTIME}[n] \not\subseteq \mathrm{co}\mathcal{NTIME}[n^a] \cap \mathcal{DTISP}[t, s]$$

if for some integer $k \geqslant 1$

$$(t \cdot s^k)^{g_k} \in \omega(s^{a^k}) \cap o(n),$$

where g_k is given by (30).

Theorem 4.4 *For any constant $a \geqslant 1$,*

$$\mathcal{NTIME}[n] \not\subseteq \mathcal{DTISP}[n^a, o(n^b)],$$

where

$$b = \max\left(\bigcup_{k\geqslant 1}\{c \mid a^k c \leqslant (a+kc)g_k \leqslant 1\}\right) \tag{35}$$

and g_k is defined by (30).

Recall that taking $k = 1$ in Theorem 4.4 gives us Corollary 3.6, or equivalently, Theorem 3.5. We observe that $k = 2$ always allows larger values of c on the right-hand side of (35) than $k = 1$. Thus, the following corollary of Theorem 4.4 improves Corollary 3.6 and Theorem 3.5.

Corollary 4.5 *For any constant $a \geqslant 1$,*

$$\mathcal{NTIME}[n] \not\subseteq \mathcal{DTISP}[n^a, o(n^{\frac{1}{2}(\frac{a+2}{a^2}-a)})].$$

Theorem 4.3 allows us to obtain lower bounds on ts^k for $\mathcal{DTISP}[t,s]$-simulations of nondeterministic linear time. For $k \geqslant 2$ these bounds are stronger than the analogs of Corollary 3.4 that follow from Theorem 3.2. We state the result for $k = 2$.

Corollary 4.6 *Suppose that $\mathcal{NTIME}[n] \subseteq \mathcal{DTISP}[t,s]$. Then $ts^2 \notin o(n^a)$, where a is the positive solution of $a(a^2-1) = 2$, about 1.521.*

Note Corollary 4.5 yields nontrivial statements as long as $\frac{a+2}{a^2} - a > 0$, i.e., $a(a^2-1) < 2$, or approximately, $a \leqslant 1.521$. Theorem 4.4 gives interesting results for a up to the golden ratio, as we will see next.

We get the following improvements over the $\mathcal{NTISP}$ time-space tradeoffs of Section 3.4.2.

Theorem 4.7 *For any integer $k \geqslant 2$ and constants $a \geqslant 1$ and c such that $a^k c < (a+kc)g_k < 1$,*

$$\mathcal{NTISP}[n, n^d] \not\subseteq \mathcal{DTISP}[n^a, n^c],$$

where $d = \max(\frac{1}{a}, \frac{a^k c}{(a+kc)g_k})$ and g_k is given by (30).

Note the sequence $a \cdot g_k$ converges to $a(a-1)$, which is less than 1 as long as a is less than the golden ratio (approximately 1.618). Thus, we obtain:

Corollary 4.8 *For any constant a such that $a(a-1) < 1$,*

$$\mathcal{NTISP}[n, n^{1/a}] \not\subseteq \mathcal{DTISP}[n^a, n^{o(1)}].$$

5 Results for Other Classes

This section describes our result for linear-time Σ_ℓ-computations.

Theorem 5.1 *For any integer $\ell \geqslant 2$ and functions $t(n)$ and $s(n)$,*

$$\Sigma_\ell\mathcal{TIME}[n] \not\subseteq \mathcal{DTISP}[t,s],$$

provided $ts^{\ell-1} \in o(n^\ell)$ and $t \in \omega(s)$.

Proof Suppose t and s satisfy the conditions of the lemma but

$$\Sigma_\ell\mathcal{TIME}[n] \subseteq \mathcal{DTISP}[t, s]. \tag{36}$$

Let $L \in \Sigma_\ell\mathcal{TIME}[n]$. By (36), there exists a deterministic Turing machine running in time $T = t$ and space $S = s$ that accepts L. Consider the machine N' described in Section 4.1 for $k = \ell - 1$. The machine N' accepts L, is of type $\Pi_{k+1} = \Pi_\ell$, and runs in time

$$O\left(\sum_{i=1}^{k}(b_i \cdot S + \log b_i) + \frac{T}{b_1 \cdot b_2 \cdot \ldots \cdot b_k}\right). \tag{37}$$

Up to a constant factor, (37) is minimized by picking each b_i equal to b where b is such that $b \cdot S = T/b^k$, i.e., $b = (T/S)^{1/(k+1)}$. Note $b \in \omega(1)$ since $T \in \omega(S)$.

For this choice of parameters, N' runs in time $O((T \cdot S^k)^{1/(k+1)})$. Thus, we get that

$$\Sigma_\ell\mathcal{TIME}[n] \subseteq \Pi_\ell\mathcal{TIME}[(TS^{\ell-1})^{1/\ell}] \subseteq \Pi_\ell\mathcal{TIME}[o(n)],$$

a contradiction with Theorem 2.1. $\square$

Corollary 5.2 *For any integer $\ell \geqslant 2$ and constants $\epsilon > 0$ and $\delta < \epsilon/(\ell - 1)$,*

$$\Sigma_\ell\mathcal{TIME}[n] \not\subseteq \mathcal{DTISP}[n^{\ell-\epsilon}, n^\delta].$$

Acknowledgments

The second author would like to thank Rahul Santhanam and Iannis Tourlakis for discussions about the topic of this paper and for comments on an earlier draft. He is grateful to Steve Cook for bringing him into contact with Iannis Tourlakis.

References

[1] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity I*, volume 11 of *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1995.

[2] A. Bruss and A. Meyer. On time-space classes and their relation to the theory of real addition. *Theoretical Computer Science*, 11:59–69, 1980.

[3] S. Cook. Short propositional formulas represent nondeterministic computations. *Information Processing Letters*, 26:269–270, 1988.

[4] L. Fortnow. Nondeterministic polynomial time versus nondeterministic logarithmic space: Time-space tradeoffs for satisfiability. In *Proceedings of the 12th IEEE Conference on Computational Complexity*, pages 52–60. IEEE, 1997.

[5] L. Fortnow and C. Lund. Interactive proof systems and alternating time-space complexity. *Theoretical Computer Science*, 113:55–73, 1993.

[6] Y. Gurevich and S. Shelah. Nearly-linear time. In *Proceedings, Logic at Botik '89*, volume 363 of *Lecture Notes in Computer Science*, pages 108–118. Springer-Verlag, 1989.

[7] F. Hennie and R. Stearns. Two-tape simulation of multitape Turing machines. *Journal of the ACM*, 13:533–546, 1966.

[8] R. Kannan. Towards separating nondeterminism from determinism. *Mathematical Systems Theory*, 17:29–45, 1986.

[9] R. Lipton and A. Viglas, November 1999. Personal communication.

[10] R. Lipton and A. Viglas. On the complexity of SAT. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 459–464. IEEE, 1999.

[11] V. Nepomnjaščiĭ. Rudimentary predicates and Turing calculations. *Soviet Mathematics–Doklady*, 11:1462–1465, 1970.

[12] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[13] W. Paul, E. Prauß, and R. Reischuk. On alternation. *Acta Informatica*, 14:243–255, 1980.

[14] N. Pippenger and M. Fischer. Relations among complexity measures. *Journal of the ACM*, 26:361–381, 1979.

[15] R. Santhanam, October 1999. Personal communication.

[16] C. Schnorr. Satisfiability is quasilinear complete in NQL. *Journal of the ACM*, 25:136–145, 1978.

[17] J. Seiferas, M. Fischer, and A. Meyer. Separating nondeterministic time complexity classes. *Journal of the ACM*, 25:146–167, 1978.

[18] S. Žàk. A Turing machine time hierarchy. *Theoretical Computer Science*, 26:327–333, 1983.

A Appendix

We now prove Theorem 2.4.

Gurevich and Shelah [6] showed how to simulate nondeterministic random-access machines by multitape Turing machines. Their proof built on Schnorr's result [16] that one can sort in quasilinear time on a nondeterministic multitape Turing machine.

Theorem A.1 (Gurevich-Shelah) *There exists a constant c such that every language that is accepted by a nondeterministic random-access Turing machine using time t is also accepted by a nondeterministic multitape Turing machine using time $O(t \log^c t)$.*

Because of Theorem A.1 the following theorem finishes the proof of Theorem 2.4.

Theorem A.2 *There exists a constant c such that the following holds: If*

$$SAT \in \mathcal{DTISP}[t, s],$$

then every language accepted by a linear-time nondeterministic multitape Turing machine belongs to

$$\mathcal{DTISP}[t(n \log^2 n) \cdot \log^c n, s(n \log^2 n) + \log n].$$

Proof Let M be a deterministic random-access Turing machine deciding satisfiability in time t and space s.

Consider an arbitrary language L accepted by a linear-time nondeterministic multitape Turing machine. Hennie and Stearns [7] showed that there exists an oblivious 2-tape nondeterministic Turing machine that accepts L in time $O(n \log n)$. Cook [3], building on work by Pippenger and Fischer [14], used this result to construct for a given input x of length n, a Boolean formula ϕ such that $\phi \in SAT$ iff $x \in L$. The formula ϕ has size $m = dn \log n$ for some constant d depending on L, only depends on n, and uses the bits x_i of the input x as well as some some additional Boolean variables y. More precisely, ϕ is of the form $(\wedge_{i=1}^{n} x_i = y_i) \wedge \psi$ where ψ only uses the variables y. Given a pointer to a bit of ψ, we can compute that bit in simultaneous time $O(\log^{c_1} m) = O(\log^{c_1} n)$ and space $O(\log m) = O(\log n)$ for some constant c_1 independent of L.

The following algorithm decides L in time $O(t(n \log n) \cdot \log^c n)$ and space $O(s(n \log n) + \log n)$ on a deterministic random-access Turing machine for some constant c independent of L. We will simulate running M on input ϕ without storing ϕ in memory and without recomputing all of ϕ each time we have to access one of its bits. When given ϕ, running M on ϕ takes time $t(dn \log n)$ and space $s(dn \log n)$. Whenever M needs a bit from ψ, we compute that individual bit from scratch in time $O(\log^{c_1} n)$ and space $O(\log n)$, without moving the input tape head above x. During the periods when M is accessing the part of ϕ that depends on the input x, the easy structure of that part allows us to compute the bit of ϕ we need in time $O(\log^{c_2} n)$ and space $O(\log n)$ for some constant c_2 independent of L. This operation may require moving the input tape head above x. All together, we can simulate M on ϕ with a multiplicative time overhead of $O(\log^c n)$ and an additive space overhead of $O(\log n)$, where $c = \max(c_1, c_2)$. $\square$

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'