Nondeterministic Separations

Lance Fortnow(✉)

Georgia Institute of Technology, Atlanta, USA
fortnow@cc.gatech.edu

Abstract. We survey recent research on the power of nondeterministic computation and how to use nondeterminism to get new separations of complexity classes. Results include separating NEXP from NP with limited advice, a new proof of the nondeterministic time hierarchy and a surprising relativized world where NP is as powerful as NEXP infinitely often.

1 Results

In this talk we focus on new results by the speaker about the power of nondeterminism which sits at the heart of the famous P versus NP problem. The results in this paper first appeared in works by Buhrman, Fortnow and Santhanam [1–3]

Theorem 1. *For any constant c,* $\mathsf{NEXP} \not\subseteq \mathsf{NP}/n^c$.

Eric Allender asked whether even Theorem 1 ($\mathsf{NEXP} \not\subseteq \mathsf{NP}/n^c$) can be strengthened to a lower bound that works on almost all input lengths, rather than on infinitely many. Direct diagonalizations tend to work on almost all input lengths– our separation is indirect, and technique does not give this stronger property. We give a new relativized world showing that relativizing techniques cannot get the stronger separation even without the advice.

Theorem 2. *There exists a relativized world such that* $\mathsf{NEXP} \subseteq \mathsf{i.o.NP}$.

Cook [4] first showed a nondeterministic time hierarchy, given in its strongest form by Seiferas, Fischer and Meyer [5] and simplified by Žàk [6]. We give yet a new proof that gives a far more compact diagonalization.

Theorem 3. *If* t_1 *and* t_2 *are time-constructable functions such that*

- $t_1(n) = o(t_2(n))$, *and*
- $n \leqslant t_1(n) \leqslant n^c$ *for some constant* c

then $\mathsf{NTIME}(t_2(n)) \not\subseteq \mathsf{NTIME}(t_1(n))$.

Corollary 1. *For any reals* $1 \leqslant r < s$, $\mathsf{NTIME}(n^s) \not\subseteq \mathsf{NTIME}(n^r)$.

We can use the techniques of this new proof to get a time hierarchy with advice.

R. Jain et al. (Eds.): TAMC 2015, LNCS 9076, pp. 10–17, 2015.
DOI: 10.1007/978-3-319-17142-5_2

Theorem 4. *Let $d \geqslant 1$ be any constant, and let t be a time-constructible time bound such that $t = o(n^d)$. Then* $\mathsf{NTIME}(n^d) \not\subseteq \mathsf{NTIME}(t)/n^{1/d}$.

Theorem 4 improves on known results handling advice in two respects. First, the amount of advice in the lower bound can be as high as $n^{\Omega(1)}$, in contrast to earlier results in which it was limited to be $O(\log(n))$. Second, the hierarchy is provably tight in terms of the time bounds, while earlier results handling advice could only separate $\mathsf{NTIME}(n^d)$ from $\mathsf{NTIME}(\mathrm{n}^c)$ with advice, where $c < d$.

We are able to use Theorem 4 to derive a new circuit lower bound for NP, improving a 30-year old result of Kannan [7].

Corollary 2. *Let $k > 1$ be any constant.* NP *does not have* NP*-uniform nondeterministic circuits of size $O(n^k)$.*

2 Proof of Theorem 1

We first need the following lemma, a slightly stronger version of a result in Homer and Mocas [8] about lower bounds for deterministic exponential time against advice. The proof we give is folklore.

Lemma 1. *For any constant d,* $\mathsf{EXP} \not\subseteq \mathsf{i.o.DTIME}(2^{n^d})/n^d$.

Proof. The proof is by diagonalization. We define a diagonalizing language L which is not in $\mathsf{i.o.DTIME}(2^{n^d})/n^d$ by defining a machine M which runs in exponential time and decides L.

M operates as follows on input x of length n. It enumerates advice taking machines $M_1, M_2 \ldots M_{\log(n)}$ each running in time at most 2^{n^d} and taking advice of length n^d. It then enumerates all $\log(n)2^{n^d}$ truth tables computed by these machines when every possible string of length n^d is given as advice. It then computes a truth table of an n-bit function f which is distinct from all the truth tables enumerated so far–this can be done in exponential time by a simple pruning strategy. Finally it outputs $f(x)$.

Now we are ready to prove our lower bound for NEXP.

Proof. We will show that either $\mathsf{NEXP} \not\subseteq \mathsf{NP}/poly$ or $\mathsf{NEXP} \not\subseteq \mathsf{NE}/\mathrm{n}^c$. From this, the result follows.

Assume, to the contrary, that both these inclusions hold, i.e., $\mathsf{NEXP} \subseteq \mathsf{NP}/poly$ and $\mathsf{NEXP} \subseteq \mathsf{NE}/\mathrm{n}^c$. We will derive a contradiction. Let L be a complete language for NE with respect to linear-time reductions. Since $\mathsf{NEXP} \subseteq \mathsf{NP}/poly$, we get that $L \in \mathsf{NTIME}(n^k)/n^k$ for some constant k. Since L is complete for NE with respect to linear-time reductions, we get that $\mathsf{NE} \subseteq \mathsf{NTIME}(n^k)/O(n^k)$.

By translation, we get that $\mathsf{NE}/\mathrm{n}^c \subseteq \mathsf{NTIME}(n^{kc})/O(n^{kc})$. To see this, let L' be a language in NE/n^c, and let M' be an advice-taking NE machine accepting L' with advice length n^c. Define a language $L'' \in \mathsf{NE}$ as follows: a string $< x, a >$ is in L'' iff M' accepts x with advice a. Since M' is an NE machine, it follows that

$L'' \in \mathsf{NE}$. Thus, by assumption $L'' \in \mathsf{NTIME}(m^k)/O(m^k)$, where m is the input length for L''. Let M'' be an advice-taking machine solving L'' using resources as stated. Now we can solve L' in $\mathsf{NTIME}(n^{kc})/O(n^{kc})$ as follows. The advice-taking machine M we construct for solving L' interprets its advice as consisting of two parts: the first part is an advice string a of length n^c, where n is the input size, and the second part is an advice string b of length $O((n+n^c)^k) = O(n^{kc})$. M simulates M'' on input $< x, a >$ with advice string b, where x is the input for L'. M accepts iff M'' accepts. M operates within time $O(n^{kc})$ (since it simulates an $O(n^k)$ time machine on an input of length $O(n^c)$), uses advice of length $O(n^{kc})$, and decides L' correctly, by definition of L'' and the assumption on M''.

Thus, we have $\mathsf{NEXP} \subseteq \mathsf{NE/n^c}$ and $\mathsf{NE/n^c} \subseteq \mathsf{NTIME}(n^{kc})/O(n^{kc})$, which together imply $\mathsf{NEXP} \subseteq \mathsf{NTIME}(n^{kc})/O(n^{kc})$. But since $\mathsf{EXP} \subseteq \mathsf{NEXP}$ and $\mathsf{NTIME}(n^{kc})/O(n^{kc}) \subseteq \mathsf{DTIME}(2^{n^{kc}})/O(n^{kc})$ we get $\mathsf{EXP} \subseteq \mathsf{DTIME}(2^{n^{kc}})/O(n^{kc})$, which is a contradiction to Lemma 1.

3 Proof of Theorem 2

We show the surprising relativized world where NEXP is infinitely often contained in NP.

Proof. Let M_i be a standard enumeration of non-deterministic relativized Turing machines that runs in time at most 2^{n^i}. Since these machines are paddable, for any A and any $L \in \mathsf{NEXP}^A$ there will some i such that $L = L(M_i^A)$. We will create A such that for every i there are an infinite number of n such that for all x of length n,

$$x \in L(M_i^A) \Leftrightarrow \text{there exists a } y \text{ with } |y| = 2|x|^i \text{ and } (i, x, y) \in A$$

which immediately implies Theorem 2.

Start with $A = \varnothing$. We construct A in stages (i, j) chosen in any order that cover all possible (i, j).

Stage (i, j): Pick n such that n is larger than any frozen string as well as the n chosen in any previous stage.

Set all strings x of length n to be unmarked.

Repeat the following as long as there is an unmarked x of length n such that $M_i^A(x)$ accepts: Fix an accepting path of $M_i^A(x)$ and freeze every string queried along that path. Mark x. Pick a y, $|y| = 2|x|^i$ such that (i, x, y) is not frozen and let $A = A \cup \{(i, x, y)\}$.

We can always find such a y since we have 2^{2n^i} possible (i, x, y) and at this point since we have frozen at most 2^{n^i} strings for at most 2^n possible x's for a total of $2^{n^i}2^n < 2^{2n^i}$ frozen strings.

By adding every (i, x, y) that is non frozen in the proof above one can get an even stronger oracle.

Corollary 3. *There exists a relativized world such that* $\mathsf{NEXP} \subseteq \mathsf{i.o.RP}$.

4 New Proof of Nondeterministic Time Hierarchy

Here we give an alternate proof of Theorem 3.

Proof (Proof of Theorem 3). Let $M_1, M_2, \ldots$ be an enumeration of multitape nondeterministic machines that run in time $t_1(n)$.

Define a nondeterministic Turing machine M that on input 1^i01^m0w does as follows:

- If $|w| < t_1(i+m+2)$ accept if both $M_i(1^i01^m0w0)$ and $M_i(1^i01^m0w1)$ accept.
- If $|w| \geqslant t_1(i+m+2)$ accept if $M_i(1^i01^m0)$ rejects on the path specified by the bits of w.

Since we can universally simulate $t(n)$-time nondeterministic multitape Turing machines on an $O(t(n))$-time 2-tape nondeterministic Turing machine, $L(M) \in \mathsf{NTIME}(O(t_1(n+1))) \subseteq \mathsf{NTIME}(t_2(n))$. Note $(n+1)^c = O(n^c)$ for any c.

Suppose $\mathsf{NTIME}(t_2(n)) \subseteq \mathsf{NTIME}(t_1(n))$. Pick a c such that $t_1(n) \ll n^c$. By assumption there is a language $L \in \mathsf{NTIME}(t_1(n))$ such that $L(M) = L$. Fix i such that $L = L(M_i)$. Then $z \in L(M_i) \Leftrightarrow z \in L(M)$ for all $z = 1^i01^{n_0}0w$ for $w \leqslant t_1(i+n_0+2)$.

By induction we have $M_i(1^i01^{n_0}0)$ accepts if $M_i(1^i01^{n_0}0w)$ accepts for all $w \leqslant t_1(i+n_0+2)$. So $M_i(1^i01^{n_0}0)$ accepts if and only $M_i(1^i01^{n_0}0)$ rejects on every computation path, contradicting the definition of nondeterministic time.

5 Proof of Theorem 4

Theorem 4 follows immediately from the following result.

Theorem 5. *Fix any constant $d > 1$. Let t_1 and t_2 be time-constructible functions such that $t_2 = O(n^d)$ and $t_1(n+1) = o(t_2(n))$. Then there is a language in $\mathsf{NTIME}(t_2)$ which is not in $\mathsf{NTIME}(t_1)/t_2^{-1}(n)$.*

We need a new notion of "cumulative advice", defined as follows. Given a time function $t : \mathbb{N} \to \mathbb{N}$ and an advice function $a : \mathbb{N} \to \mathbb{N}$, a language L is said to be in $\mathsf{NTIME}(t)/_c a$ if there is an advice-taking non-deterministic machine M such that, for each n, there is a string b_n of length at most $a(n)$ for which M, given $< n, b_n >$ on its advice tape, halts in time $t(n)$ and accepts an input x of length at most n iff $x \in L$.

The notion of cumulative advice is defined here for non-deterministic time but it extends naturally to any complexity measure.

Informally, an advice string given as cumulative advice helps to decide all inputs of length *at most* a given length, while the traditional notion of advice only applies to inputs which are all of the same length. If a language L is in $\mathsf{NTIME}(t)/a$ and a is a non-decreasing function, then it is obvious that L is $\mathsf{NTIME}(t)/_c na$, since cumulative advice for length n can be formed simply by concatenating all advice strings of length at most n. However, it is far from clear

whether advice of length a can be simulated with cumulative advice $o(na)$, when a is polynomially bounded.

We will first prove a hierarchy theorem for non-deterministic polynomial time against sub-linear cumulative advice, and then show how to strengthen this to a hierarchy theorem for non-deterministic polynomial time against sub-linear advice. Note that though the notion of cumulative advice plays an important role in our proof, it does not appear in our main theorem - the main theorem holds for the traditional notion of advice.

Lemma 2. *Fix any constant $d > 1$. Let t_1 and t_2 be time-constructible functions such that $t_2 = O(n^d)$, $t_1(n+1) = o(t_2(n))$. Then there is a language $L \in \mathsf{NTIME}(t_2)$ which is not in $\mathsf{NTIME}(t_1)/_c t_2^{-1}(n)$.*

Note that the statement of Lemma 2 is identical to that of Theorem 5, except that the lower bound is against cumulative advice.

Proof. First fix a function $f : \mathbb{N} \to \mathbb{N}$ such that $f(n)$ is computable in time $O(n)$, and for each constant k, there are only finitely many triples (n_1, n_2, n_3) of integers such that $n_1 \leqslant n_2 \leqslant n_3 \leqslant n_1^k$ such that $f(n_1), f(n_2), f(n_3)$ are all distinct, and also such that each positive integer has infinitely many pre-images under f. We will use the function $f(n) = i$ if $2^{2^{2^m}} \leqslant n < 2^{2^{2^{m+1}}}$, where i is the unique number such that $bin(m)$ is of the form $1^k 0 bin(i)$ for some $k \geqslant 0$. Here $bin(j)$ denotes the binary representation of the number j.

Intuitively, f selects which cumulative advice-taking non-deterministic Turing machine we attempt to diagonalize against at a given input length n. The properties of f ensure that the same machine is being diagonalized against for a long enough stretch of inputs, and that it is easy to compute for any given input length which machine we're diagonalizing against. Let $M_1, M_2, M_3 \ldots$ be an efficiently computable enumeration of all cumulative advice-taking 2-tape non-deterministic Turing machines. We define a non-deterministic machine M without advice which operates as follows.

On input x, M first computes $n = |x|$, $i = f(n)$ and the number $t_2(n)$, the last of which it uses as a clock for its computation. It then computes the largest m such that $2^{2^{2^m}} \leqslant n < 2^{2^{2^{m+1}}}$. Set $A = 2^{2^{2^m}}$. If $n > t_2(A)$, M simply rejects. Otherwise M decomposes x as yz, where $|y| = A$. If $n < t_2(A)$, M simulates M_i on input $x0$ with advice $< t_2(A), y >$ on the advice tape[1]. If M_i halts within the allotted time, M next simulates M_i on input $x1$ with advice $< t_2(A), y >$ on the advice tape. If this simulation halts as well within the allotted time, M accepts iff both simulations (i.e., of M_i on $x0$ and M_i on $x1$) accept. In every other case, M rejects.

If $n = t_2(A)$, M simulates M_i on y with guess sequence z (i.e., z is treated as an encoding of all the non-deterministic choices of M_i), and with advice $< n, y >$ on the advice tape. It accepts iff the simulation halts and rejects. Note that the simulation on such an input length n is completely deterministic.

[1] We assume that if M_i needs only $r < |y|$ bits of advice, then only the first r bits of y are used.

By definition of M, $L(M) \in \mathsf{NTIME}(t_2)$. We claim $L(M) \notin \mathsf{NTIME}(t_1)/{}_c t_2^{-1}(n)$. The proof of this claim is by contradiction. Suppose, to the contrary, that there is a cumulative advice-taking non-deterministic Turing machine deciding $L(M)$ in time $O(t_1)$ with $t_2^{-1}(n)$ bits of advice. By the tape reduction theorem for non-deterministic time, there is a 2-tape advice-taking non-deterministic machine M_i which decides $L(M)$ in time $O(t_1)$ with $t_2^{-1}(n)$ bits of advice.

Let $g : \mathbb{N} \to \mathbb{N}$ be a function such that the simulation of t steps of a machine M_i is performed within $g(i)t$ steps of M. Choose A a power of a power of a power of 2 large enough so that $f(A) = i$ and $2g(i)t_1(n'+1) + 100n' < t_2(n')$ for all $n' \geqslant A$. By choice of f and since $t_1(n+1) = o(t_2(n))$, such an A exists. Now, for all n such that $A \leqslant n < t_2(A)$, the simulations of M_i by M halt within the allotted time, since all the extra computations (of $n, i, t_2(n)$ and the decomposition) can be performed in time $< 100n$. Note also that the simulations at length $n = t_2(A)$ complete succesfully since $t_2(n) - n \geqslant t_1(n)$.

By assumption, there is a sequence of advice strings $\{b_m\}$ such that for each m, for each x of length at most m, M_i accepts x with advice $< m, b_m >$ iff $x \in L(M)$, and $|b_m| \leqslant t_2^{-1}(m)$. Let y be any string of length A such that $b_{t_2(A)}$ is a prefix of y. By the assumption on size of advice strings, such a string y exists.

Now we have that M accepts on y iff M_i accepts on both $y0$ and $y1$ with $< t_2(A), y >$ on the advice tape. Continuing inductively, we have that M accepts y iff M_i accepts on all strings of the form yz, $|z| \leqslant t_2(A) - A$ with $< t_2(A), y >$ on the advice tape. Now we take advantage of the behavior of M on strings of length $t_2(A)$. M accepts on a string yz, $|z| = t_2(A) - A$ iff z is not a sequence of non-deterministic choices leading to acceptance of M_i on y with $< t_2(A), y >$ on the advice tape. Hence, if M_i with $< t_2(A), y >$ on the advice tape agrees with M on all strings of the form $yz, |yz| = t_2(A)$, we have that M accepts y iff M_i rejects y with $< t_2(A), y >$ on the advice tape, which contradicts the assumption that M on y agrees with M_i on y with $< t_2(A), y >$ on the advice tape.

Lemma 3. *Let L be any language, and let $L' = \{0^k 1x | x \in L, k \geqslant 0\}$. For any non-decreasing advice function $a : \mathbb{N} \to \mathbb{N}$, and for any non-decreasing time function $t : \mathbb{N} \to \mathbb{N}$ which is $\Omega(n)$, we have that $L \in \mathsf{NTIME}(t(n+1))/{}_c a(n+1)$ iff $L' \in \mathsf{NTIME}(t(n))/a(n)$.*

Proof. We define $L' = \{0^k 1x | x \in L, k \geqslant 0\}$. We first show the forward implication, and then the reverse one.

Suppose $L \in \mathsf{NTIME}(t(n+1))/{}_c a(n+1)$, for some time function t and cumulative advice function a. Let M be an advice-taking non-deterministic Turing machine which always halts in time $t(n+1)$ on inputs of length n and decides L correctly with $a(n+1)$ bits of cumulative advice. For each input length m, let b_m be a correct advice string of length at most $a(m+1)$ for M at length m, i.e., for all x of length at most m, M accepts x given advice $< m, b_m >$ iff $x \in L$. We define an advice-taking non-deterministic Turing machine M' which always halts in time $t(n)$ on inputs of length n and decides L correctly with at most $a(n)$ bits of advice.

Given an input x', M' operates as follows. M' first computes the unique string x such that $0^k1x = x'$, for some $k \geqslant 0$. This computation can be done easily in linear time. M' then interprets its advice string c_n as the cumulative advice b_{n-1} for M at length $n-1$, and simulates M on x with advice $< n-1, c_n >$. It accepts iff M accepts. M' always halts in time $O(t(n))$ since the string x' is of length at most $n-1$ and since M always halts in time $t(m+1)$ on inputs of length m. The correctness of M' follows from the fact that M is a correct advice-taking machine deciding L with cumulative advice.

For the reverse implication, suppose $L' \in \mathsf{NTIME}(t(n))/a(n)$. Let M' be an advice-taking non-deterministic machine which always halts in time $t(\cdot)$ and accepts L' with at most $a(n)$ bits of advice. We define an advice-taking machine M halting in time $t(n+1)$ and accepting L with at most $a(n+1)$ bits of cumulative advice as follows.

Say M is given a string x on its input tape, and $< m, b_m >$ on its advice tape, with $m \geqslant |x|$. Note that we can assume wlog that $m \geqslant |x|$, since otherwise M is allowed to behave arbitrarily. M forms the string $x' = 0^{m-|x|}1x$ and then simulates M' on input x' with advice b_m. Namely, it interprets its advice string as advice for M' at length $m+1$. The time taken for the simulation is $O(t(m+1))$ since t is at least linear, and the advice is of length at most $a(m+1)$. The correctness of M follows from the correctness of M'.

Proof of Theorem 5. Applying Lemma 2 to the time functions $t_1(n+1), t_2(n+1)$ and the cumulative advice function $t_2^{-1}(n+1)$, we have that there is a language L which is in $\mathsf{NTIME}(t_2(n+1))$ but not in $\mathsf{NTIME}(t_1(n+1))/t_2^{-1}(n+1)$. Using Lemma 3 with $t = t_2$ and $a = 0$, we have that $L' \in \mathsf{NTIME}(t_2)$. Using Lemma 3 with $t = t_1$ and $a = t_2^{-1}$, we have that $L' \notin \mathsf{NTIME}(t_1)/t_2^{-1}(n)$. Thus L' satisfies the required conditions.

We note that the polynomial upper bound on t_2 in Theorem 5 is in fact redundant. It helps to simplify the choice of f in the proof, but in fact for any time-constructible t_2 an appropriate f can be chosen to make the proof go through.

References

1. Buhrman, H., Fortnow, L., Santhanam, R.: Unconditional lower bounds against advice. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part I. LNCS, vol. 5555, pp. 195–209. Springer, Heidelberg (2009)
2. Fortnow, L., Santhanam, R.: Robust simulations and significant separations. In: Aceto, L., Henzinger, M., Sgall, J. (eds.) ICALP 2011, Part I. LNCS, vol. 6755, pp. 569–580. Springer, Heidelberg (2011)
3. Fortnow, L., Santhanam, R.: Hierarchies against sublinear advice. Technical report TR14-171, Electronic Colloquium on Computational Complexity (2014)
4. Cook, S.: A hierarchy for nondeterministic time complexity. J. Comput. Syst. Sci. **7**(4), 343–353 (1973)
5. Seiferas, J., Fischer, M., Meyer, A.: Separating nondeterministic time complexity classes. J. ACM **25**(1), 146–167 (1978)

6. Žàk, S.: A turing machine time hierarchy. Theor. Comput. Sci. **26**(3), 327–333 (1983)
7. Kannan, R.: Circuit-size lower bounds and non-reducibility to sparse sets. Inf. Control **55**, 40–56 (1982)
8. Homer, S., Mocas, S.: Nonuniform lower bounds for exponential time classes. In: Hájek, Petr, Wiedermann, Jiří (eds.) MFCS 1995. LNCS, vol. 969. Springer, Heidelberg (1995)