Electronic Colloquium on Computational Complexity, Report No. 93 (2004)

From logarithmic advice to single-bit advice

Oded Goldreich* Madhu Sudan† Luca Trevisan‡

November 9, 2004

Abstract

Building on Barak's work (*Random'02*), Fortnow and Santhanam (*FOCS'04*) proved a time hierarchy for probabilistic machines with one bit of advice. Their argument is based on an implicit translation technique, which allow to translate separation results for short (say logarithmic) advice (as shown by Barak) into separations for a single-bit advice. In this note, we make this technique explicit, by introducing an adequate translation lemma.

Keywords: Time hierarchies, probabilistic polynomial-time, non-uniformity, short advice.

*Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, ISRAEL. Email: oded.goldreich@weizmann.ac.il. Work was done while the author was a fellow at the Radcliffe Institute for Advanced Study of Harvard University.

†Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139. Email: madhu@mit.edu. Work was done while the author was a fellow at the Radcliffe Institute for Advanced Study of Harvard University.

‡EECS – Computer Science Div., UC-Berkeley, CA 94720. Email: luca@eecs.berkeley.edu

ISSN 1433-8092

1 Introduction and High Level Description

Trying to address the open problem of providing a probabilistic time hierarchy, Barak [1] presented a time hierarchy for slightly non-uniform probabilistic machines. Specifically, he showed that, in presence of double-logarithmic advice, there exists a hierarchy of probabilistic polynomial-time. Recently, Fortnow and Santhanam [2] showed that a similar hierarchy holds in the presence of a single-bit advice. Their argument is based on an implicit translation technique, which allow to translate separation results for short (say logarithmic) advice into separations for a single-bit advice. In this note, we make this technique explicit, by introducing an adequate translation lemma and showing that applying it to Barak's result [1] yields the aforementioned result of [2].

Interestingly (as in [2]), we rely on the fact that Barak [1] actually shows a time separation that holds even when the more time-restricted machine is given a somewhat longer advice. In contrast, arguably, the more natural statement of such results refers to machines that use the same advice length.[1]

The basic idea underlying the proof in [2] is that short advice can be incorporated in the (length of the) instance of a padded language, while using a single bit of advice to indicate whether or not the resulting instance length encodes a valid advice. For this to work, the length of the resulting instance should indicate a unique length of the original instance as well as a value of a corresponding advice (for this instance length).

Suppose we wish to treat a language L that is decidable (within some time) using *eight* bits of advice. Viewing the possible values of the advice as integers in $\{0, 1, ..., 255\}$, we define a (padded) language L' as follows: the pair $(x, 0^{255|x|+i})$ is in L' if and only if $x \in L$ and i is an adequate advice for instances of length $|x|$. Note that L' can be decided using a single bit of advice that indicates whether the instance length encodes a valid advice for L. Specifically, the advice bit for length m instances (of L') is 1 if and only if $m \bmod 256$ is a valid advice for instances of length $\lfloor m/256 \rfloor$ (of L). Thus, on input $y = (x, 0^{255|x|+i})$, where $i \in \{0, ..., 255\}$, we accept if and only if the advice bit is 1 and the original machine accepts x when given advice i.

Note that we should also show that if L is undecidable using less time (and, say, *nine* bits of advice) then L' is correspondingly hard (even using a single bit of advice). This is shown by using a machine for deciding L' as a subroutine for deciding L, while using part of the advice (given for deciding L) for determining an adequate instance for L'. In other words, we present a non-uniform reduction of L to L', where the non-uniformity is accounted by the longer advice allowed in deciding L.

2 Detailed Technical Presentation

We assume that the machine model supports some trivial computations with little overhead. Specifically, we refer to computing the square root of the length of the input in linear time.

[1] That is, in order to show, say, that $\mathrm{BPtime}(n^3)/1$ is not contained in $\mathrm{BPtime}(n^2)/1$, we use the fact that Barak showed that $\mathrm{BPtime}(n^6)/\log n$ is not contained in $\mathrm{BPtime}(n^4)/2\log n$ (rather than $\mathrm{BPtime}(n^6)/\log n$ is not contained in $\mathrm{BPtime}(n^4)/\log n$).

We state our main (translation) lemma for probabilistic machines. An analogous lemma holds for deterministic (and non-deterministic) machines.

Lemma 1 (Translation Lemma): *Let $f(m)$ be a fixed function growing roughly as $\sqrt{m}$, and suppose that $1^m \mapsto f(m)$ can be computed in linear time. Suppose that L is a language that is decided by some advice-taking probabilistic machine M, with $A_M(n) \le \log n$ bits of advice in time $T_M(n)$. Suppose further that L is not decided by any $a(n)$-advice probabilistic machine in time $t(n)$, where $a(n) \ge A_M(n)$. Then there exists a language $L' = L'_M$ that is decided by some probabilistic machine M' with 1 bit of advice in time $m + T_M(f(m))$, on inputs of length m, but is not decidable by any $(a(f(m)) - A_M(f(m)))$-advice probabilistic machine in time $t(f(m)) - m$.*

Before proving the Translation Lemma let us spell-out its implication. Below, we denote by $\mathrm{BPtime}(T)/A$ the class of languages decidable by advice-taking probabilistic machines of time complexity T and advice complexity A.

Corollary 2 *Let $T, A, t, a : \mathsf{N} \to \mathsf{N}$ such that $a(n) \ge A(n)$.*

If $\mathrm{BPtime}(T)/A$ *contains sets not in* $\mathrm{BPtime}(t)/a$

then $\mathrm{BPtime}(T')/1$ *contains sets not in* $\mathrm{BPtime}(t')/a'$*, where* $T'(m) \stackrel{\mathrm{def}}{=} T(\sqrt{m}) + m$*,* $t'(m) \stackrel{\mathrm{def}}{=} t(\sqrt{m}) - m$ *and* $a'(m) \stackrel{\mathrm{def}}{=} a(\sqrt{m}) - A(\sqrt{m})$*.*

For example, we can apply Corollary 2 to Barak's result [1] that asserts the existence of a language L in, say, $\mathrm{BPtime}(n^6)/\log\log n \setminus \mathrm{BPtime}(n^4)/\log n$. Doing so, we conclude that there exists a language in $\mathrm{BPtime}(m^3)/1 \setminus \mathrm{BPtime}(m^2)/(0.5\log m - \log\log m)$. (Thus, we establish the aforementioned result of [2].)

Note that in order to obtain an interesting consequence out of Corollary 2, we need $a(n) \ge A(n) + 1$. The reason is that only this setting yields that the separation is due to the time complexity (rather than being due to (higher) non-uniformity, as could be the case when $a(n) = A(n)$, which implies $a'(m) = 0 < 1$).

Proof of the Translation Lemma: Define L' to consists of pairs $(x, 0^{n^2 + \mathtt{adv}_M(n)})$ such that $x \in L$, $|x| = n$, and $\mathtt{adv}_M(n)$ is a correct advice for M on inputs of length n, written as an integer in $\{0, \ldots, 2^{A_M(n)} - 1\} \subseteq \{0, ..., n-1\}$.

Define $f(m) \stackrel{\mathrm{def}}{=} \lfloor \sqrt{m} \rfloor$. That is, for every $m \in \{n + n^2 + 0, ..., n + n^2 + n - 1\}$ (which in turn is contained in $\{n^2, ..., (n+1)^2 - 1\}$), it holds that $f(m) = n$. Below, n (resp., m) will always denote the length of instances to L (resp., L').

We first show that L' is decidable by a probabilistic machine M' taking one bit of advice and running in time $m + T_M(f(m))$. Machine M' parses its input $y \in \{0,1\}^m$ into the form $(x, 0^{n^2+i})$, where $|x| = n = f(m)$. Given the advice bit σ, machine M' always rejects if $\sigma = 0$ and invokes M on input x and advice i (viewed as an $A_M(n)$-bit long string) otherwise. Thus, M accepts $y = (x, 0^{n^2+i})$ using advice σ if and only if $\sigma = 1$ and M accepts x using advice i. The (bit) advice regarding m-bit inputs is determined in correspondence to the aforementioned parsing: the advice bit is 1 if and only if $m = f(m) + f(m)^2 + \mathtt{adv}_M(f(m))$.

Indeed, this setting of the advice guarantees that M' accepts $y = (x, 0^{|x|^2+i})$ if and only if $x \in L$ and $i = \mathtt{adv}_M(|x|)$. Thus, using the adequate advice, M' decides L', and, indeed, the running time of M' is as stated.

We next show that L' is not decidable by any probabilistic machine in time $t(f(m)) - m$ that takes a $(a(f(m)) - A_M(f(m)))$-bit long advice. Actually, we will show that if L' is decidable by some probabilistic machine in time $t'(m)$ with $a'(m)$ bits of advice, then L is decidable by a probabilistic machine in time $t'(n + n^2 + \mathtt{adv}_M(n)) + O(n^2)$ with $A_M(n) + a'(n + n^2 + \mathtt{adv}_M(n))$ bits of advice. Suppose that M' is a machine deciding L' as in the hypothesis, and let $\mathtt{adv}_{M'}(m)$ be the advice it uses for m-bit inputs. Then consider the following machine M'' (designed to decide L) whose advice on inputs of length n is the pair $(\mathtt{adv}_M(n), \mathtt{adv}_{M'}(n + n^2 + \mathtt{adv}_M(n)))$. On input x and advice (i, j), machine M'' invokes M' on input $(x, 0^{n^2+i})$ with advice j. Thus, M'' accepts x when given the adequate advice if and only if M' accepts $(x, 0^{|x|^2+\mathtt{adv}_M(|x|)})$ when given the adequate advice. It follows that M'' decides L, and does so within the stated complexities. ∎

Remark: Note that once L' is defined, the proof proceeds in two steps:

1. Assuming that $L \in \text{BPtime}(T)/A$, we establish that $L' \in \text{BPtime}(T')/1$, where $T'(m) = m + T(f(m))$.

2. Assuming that $L' \in \text{BPtime}(t')/a'$, we establish that $L \in \text{BPtime}(t)/a$, where $t(n) = t'(n^2 + O(n)) + O(n^2))$ and $a(n) = A(n) + a'(n^2 + O(n))$.

References

[1] B. Barak. A Probabilistic-Time Hierarchy Theorem for "Slightly Non-uniform" Algorithms. In *Random'02*, LNCS 2483, pages 194–208, 2002.

[2] L. Fortnow and R. Santhanam. Hierarchy theorems for probabilistic polynomial time. In *45th FOCS*, pages 316–324, 2004.

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'