Curiouser and Curiouser: The Link between Incompressibility and Complexity

Eric Allender

Department of Computer Science, Rutgers University, Piscataway, NJ 08855,
allender@cs.rutgers.edu

Abstract. This talk centers around some audacious conjectures that attempt to forge firm links between computational complexity classes and the study of Kolmogorov complexity.
More specifically, let R denote the set of Kolmogorov-random strings. Let BPP denote the class of problems that can be solved with negligible error by probabilistic polynomial-time computations, and let NEXP denote the class of problems solvable in nondeterministic exponential time.
Conjecture 1: NEXP $=$ NP^R.
Conjecture 2: BPP is the class of problems non-adaptively polynomial-time reducible to R.
These conjectures are not only audacious; they are obviously false! R is not a decidable set, and thus it is absurd to suggest that the class of problems reducible to it constitutes a complexity class.
The absurdity fades if, for example, we interpret "NP^R" to be "the class of problems that are NP-Turing reducible to R, no matter which universal machine we use in defining Kolmogorov complexity". The lecture will survey the body of work (some of it quite recent) that suggests that, when interpreted properly, the conjectures may actually be true.

1 Introduction

This is a story about mathematical notions that refuse to stay put in their proper domain. Complexity theory is supposed to deal with decidable sets (and preferably with sets that are *very* decidable – primitive recursive at least, and ideally much lower in the complexity hierarchy than that). Undecidable sets inhabit a very different realm, and they look suspiciously out-of-place popping up in a discussion of computational complexity classes.

And yet, the (undecidable) set of Kolmogorov-random strings persists in intruding into complexity-theoretic investigations. It has become much harder to deny that there is a connection, although the precise nature of the relationship remains unclear.

1.1 Cast of Characters

The primary focus of our attention will be a familiar list of deterministic and nondeterministic time- and space-bounded complexity classes: P, NP, PSPACE, NEXP, EXPSPACE, along with the class BPP of languages accepted in polynomial time by

probabilistic machines with negligible error, and the class P/poly of problems with polynomial-size circuit complexity. Detailed definitions can be found in a standard text such as [5].

Much of the action in our story revolves around the set of Kolmogorov-random strings. Before this set can be introduced properly, some definitions are required.

Given a Turing machine M, the (plain) Kolmogorov complexity function $C_M(x)$ is defined to be the minimum element of the set $\{|d| : M(d) = x\}$ (and is undefined if this set is empty). A machine U is said to be *universal* for this measure, if

$$\forall M \exists c \forall x C_U(x) \leq C_M(x) + c.$$

As usual in the study of Kolmogorov complexity (see, e.g., [9, 8]), we pick one such universal Turing machine and define $C(x)$ to be $C_U(x)$. (There is something rather arbitrary in the selection of U; we will come back to this later.)

For some applications, a better-behaved Kolmogorov complexity measure is the prefix-free measure $K_M(x)$ that has an identical definition, but where the Turing machine M is restricted to be prefix-free (meaning that if $M(x)$ halts, then M does not halt on input xy for any non-empty string y). It turns out that a universal prefix-free machine U exists such that, for all prefix-free machines M there is a constant c such that for all x $K_U(x) \leq K_M(x) + c$, and we select one such U and define $K(x)$ to be $K_U(x)$. Again, consult [9, 8] for details.

A string x is *random* (or *incompressible*) if there is no "description" d with $|d| < |x|$ such that $U(d) = x$. Depending on which notion of Kolmogorov complexity we are using, this gives us two sets of random strings:

- $R_C = \{x : C(x) \geq |x|\}$.
- $R_K = \{x : K(x) \geq |x|\}$.

When it does not make any difference which of these two sets is meant, we will use the simplified notation "R". (Similarly, if it is necessary to make explicit mention of a universal machine U, we will refer to R_{K_U} and R_{C_U}.)

2 Some Odd Inclusions

It has long been known [10] that R is Turing-equivalent to the halting problem. However, it is much less clear what can be *efficiently* reduced to R. To date, the only known proof that the halting problem can be Turing-reduced to R via polynomial-size *circuits* relies on the arsenal of derandomization techniques that were developed in the 1990s [2]. For efficient "uniform" reductions (i.e., reductions computed by polynomial-time *machines*), it is not easy to see how to make use of R as an oracle. (To illustrate this, we encourage the reader to spend a minute trying to see how to reduce their favorite NP-complete problem to R.) Thus the following theorem is of some interest.

Theorem 2.1. *The following inclusions hold:*

- BPP $\subseteq \mathrm{P}^R_{tt}$ *[6].*
- PSPACE $\subseteq \mathrm{P}^R$ *[2].*

– NEXP $\subseteq$ NP^R *[1].*

Here, the notation P_{tt}^A denotes the class of problems that are reducible to A via polynomial-time *truth-table* reductions (also known as "non-adaptive" reductions). These are reductions computed by an oracle Turing machine M that, on input x, computes a list of queries $y_1, \ldots, y_m$, and then asks the oracle about each of these m queries, and then uses the oracle answers to decide whether to accept or reject. (In a more general Turing reduction, the list of queries can depend on the answers that the oracle gives.)

There is indeed something odd about Theorem 2.1. Is it interesting to study *efficient* reductions to an undecidable set? Since R is Turing-equivalent to the halting problem, one ought to wonder whether *every* computably-enumerable set is in P_{tt}^R (in which case, Theorem 2.1 would not be very interesting).

In truth, it *is* still an open question whether the halting problem (and hence every c.e. set) is in $\mathrm{P}_{tt}^{R_C}$. In contrast, for the prefix-free measure K, the situation is intriguing, as the next section will relate.

3 An Upper Bound on Complexity

The proofs of the inclusions in Theorem 2.1, such as NEXP $\subseteq$ NP^R, make use of no special properties of the universal Turing machine that defines Kolmogorov complexity. Thus it follows that we actually have the inclusion NEXP $\subseteq \bigcap_U$ NP$^{R_{K_U}}$, where the intersection is taken over all universal prefix-free Turing machines. This might seem to be a trivial observation, but it is actually essential, if we want to obtain an upper bound on the complexity of classes such as NP^R. This is because there exist universal prefix-free Turing machines U such that even $\mathrm{P}_{tt}^{R_{K_U}}$ contains arbitrarily complex decidable sets.

However, a paper presented at the 2011 ICALP conference shows that, if we consider only those problems that are reducible to R_K *regardless* of which universal Turing machine is used in defining K-complexity, then we do indeed obtain something that looks very much like a complexity class:

Theorem 3.1. *[4]*

– BPP $\subseteq \Delta_1^0 \cap \bigcap_U \mathrm{P}_{tt}^{R_{K_U}} \subseteq$ PSPACE
– PSPACE $\subseteq \Delta_1^0 \cap \bigcap_U \mathrm{P}^{R_{K_U}}$
– NEXP $\subseteq \Delta_1^0 \cap \bigcap_U$ NP$^{R_{K_U}} \subseteq$ EXPSPACE.

Here, as usual, Δ_1^0 denotes the class of decidable sets.

Theorem 3.1 is stated in terms of the prefix-free measure K. It seems reasonable to conjecture that it holds also for the plain measure C, but there does not seem to be an easy way to modify the proof of Theorem 3.1 to deal with the C measure.

The proof of the inclusion $\Delta_1^0 \cap \bigcap_U$ NP$^{R_{K_U}} \subseteq$ EXPSPACE proceeds by first observing that an NP-Turing reduction can be simulated by an exponential-time truth-table reduction, and then noting that the PSPACE upper bound on $\Delta_1^0 \cap \bigcap_U \mathrm{P}_{tt}^{R_{K_U}}$ translates into an EXPSPACE upper bound on the class $\Delta_1^0 \cap \bigcap_U \mathrm{EXP}_{tt}^{R_{K_U}}$. Since NP is widely conjectured to be a small subclass of exponential time, it seems likely we are throwing

away too much information in the initial step in this argument (replacing an NP-Turing reduction by an exponential-time truth-table reduction). That is, we suspect that the inclusion $\Delta_1^0 \cap \bigcap_U \mathrm{EXP}_{tt}^{R_{K_U}} \subseteq \mathrm{EXPSPACE}$ is not optimal. In fact, we suspect that the inclusion $\mathrm{NEXP} \subseteq \mathrm{NP}^R$ is tight, in the following sense:

Conjecture 3.2. $\mathrm{NEXP} = \Delta_1^0 \cap \bigcap_U \mathrm{NP}^{R_{K_U}}$.

Such a characterization of NEXP in terms of reducibility to R_K would certainly be unusual. Perhaps it would also be useful.

4 Towards a Characterization of BPP

There is more to report, regarding the inclusion $\Delta_1^0 \cap \bigcap_U \mathrm{P}_{tt}^{R_{K_U}} \subseteq \mathrm{PSPACE}$ of Theorem 3.1.

In a still-unpublished paper [3], it is argued that it is likely that the PSPACE upper bound can be improved to PSPACE $\cap$ P/poly. If it true, then this would imply that $\mathrm{BPP} \subseteq \Delta_1^0 \cap \bigcap_U \mathrm{P}_{tt}^{R_{K_U}} \subseteq \mathrm{PSPACE} \cap \mathrm{P/poly}$. Since there is a dearth of interesting complexity classes between BPP and PSPACE $\cap$ P/poly, this motivates the following:

Conjecture 4.1. $\mathrm{BPP} = \Delta_1^0 \cap \bigcap_U \mathrm{P}_{tt}^{R_{K_U}}$.

The evidence presented in [3] in support of the P/poly upper bound can be summarized in this way: The authors present a true statement of the form $\forall n \forall j \Psi(n, j)$ (provable in ZF), with the property that if, for each fixed (**n,j**) there is a proof in Peano Arithmetic of the statement ψ(**n,j**), then the P/poly upper bound holds. (In fact, under this assumption, for each length n, it suffices to restrict attention to truth-table reductions that make queries only of length $O(\log n)$ and have as oracle a subset of R (possibly a different subset for each input length – which can be encoded as a circuit for inputs of length n).

Motivated largely by the results of [3], Buhrman and Loff [7] have proved a very recent result that can also be seen as supporting the P/poly upper bound. For a polynomial-time reduction from a decidable set A to the undecidable set R, it seems reasonable to hypothesize that the reduction would also work if one used a very high time-complexity approximation to R, such as $R_K^{t(n)}$ for some very rapidly-growing time bound $t(n)$. Buhrman and Loff have shown that, for each decidable set A and polynomial-time truth-table reduction M, it is the case that for *every* large-enough time bound t, if M reduces A to $R_K^{t(n)}$, then $A \in \mathrm{P/poly}$.

Interestingly, the techniques used by Buhrman and Loff also allowed them to show that the sentences ψ(**n,j**) considered in [3] are, in fact, independent of Peano Arithmetic. Worse, they present a polynomial-time reduction with the property that it can *not* be directly replaced by a reduction that makes queries only of length $O(\log n)$, having as oracle a subset of R. Thus the general approach discussed in [3] will need to be revised substantially, if it is to be used to obtain a P/poly upper bound on this class.

5 Speculations

Since, to date, no interesting characterizations of complexity classes in terms of efficient reductions to R have been obtained, it may be premature to speculate about the usefulness of such a characterization. Nonetheless, it is fun to engage in such speculation. Could it be possible that linking the study of Kolmogorov-random strings to the study of computational complexity classes could enable the application of tools from one domain, to problems where these tools had seemed inapplicable? It is an exciting prospect to contemplate.

The techniques of computability theory usually relativize, which might seem like an impediment to the realization of this program. However, the inclusions PSPACE $\subseteq \mathrm{P}^R$ and NEXP $\subseteq \mathrm{NP}^R$ each utilize techniques that do not relativize. Perhaps there is room to explore new combinations of tools and techniques from these fields.

The inclusion BPP $\subseteq \mathrm{P}^R_{tt}$ does relativize, in the following sense. The argument of [6] shows that, for every decidable set A, every set in BPP^A is P^A-truth-table reducible to R. Thus it is conceivable that a stronger version of Conjecture 4.1 holds, characterizing BPP^A as the class of decidable sets that are P^A-truth-table reducible to R. Thus, any attempt to prove P $=$ BPP (as many suspect) by proving that $\mathrm{P} = \Delta_1^0 \cap \bigcap_U \mathrm{P}^{R_{K_U}}_{tt}$ will require some new non-relativizing proof techniques. (Note however that analogous equalities have been proved for some limited classes of truth-table reductions [1].)

At the very least, results such as Theorem 3.1 provide motivation for some questions in computability theory that have not received much attention. For instance, is "$\Delta_1^0\cap$" redundant in each line of Theorem 3.1? That is, if a set is in $\mathrm{NP}^{R_{K_U}}$ for each universal machine U, is it decidable?

Acknowledgments

The research of the author is supported in part by NSF Grants CCF-0830133, CCF-0832787, and CCF-1064785.

References

1. E. Allender, H. Buhrman, and M. Koucký. What can be efficiently reduced to the Kolmogorov-random strings? *Annals of Pure and Applied Logic*, 138:2–19, 2006.
2. E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.
3. E. Allender, G. Davie, L. Friedman, S. B. Hopkins, and I. Tzameret. Kolmogorov complexity, circuits, and the strength of formal theories of arithmetic. Submitted for publication, 2012.
4. E. Allender, L. Friedman, and W. Gasarch. Limits on the computational power of random strings. In *Proc. of International Conference on Automata, Languages, and Programming (ICALP)*, volume 6755 of *Lecture Notes in Computer Science*, pages 293–304. Springer, 2011. To appear in special issue of Information and Computation for ICALP 2011.
5. Sanjeev Arora and Boaz Barak. *Computational Complexity, a modern approach*. Cambridge University Press, 2009.

6. H. Buhrman, L. Fortnow, M. Koucký, and B. Loff. Derandomizing from random strings. In *25th IEEE Conference on Computational Complexity (CCC)*, pages 58–63. IEEE Computer Society Press, 2010.
7. H. Buhrman and B. Loff. Personal Communication, 2012.
8. R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.
9. M. Li and P. Vitanyi. *Introduction to Kolmogorov Complexity and its Applications*. Springer, third edition, 2008.
10. D.A. Martin. Completeness, the recursion theorem and effectively simple sets. *Proceedings of the American Mathematical Society*, 17:838–842, 1966.