New Surprises from Self-Reducibility

Eric Allender

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

Abstract. Self-reducibility continues to give us new angles on attacking some of the fundamental questions about computation and complexity.

1 Introduction

Perhaps the most surprising thing about *Self-Reducibility* is its longevity. Who would have suspected that this simple idea would continue to play a key role in important developments in our evolving understanding of computation and complexity over a span of four decades? Yet recent developments demonstrate that self-reducibility is still able to lead us to new insights, both in computability theory and in complexity theory.

Trakhtenbrot introduced the notion of autoreducibility in a paper published forty years ago [28]. Briefly, a set A is *autoreducible* if A is accepted by an oracle Turing machine M that has A as an oracle, with the restriction that M, on input x, does not ask its oracle about x. Already in his 1970 paper, Trakhtenbrot studied autoreducibility in the context of resource-bounded computation, although in early years the notion was studied primarily in the context of computability [23, 20]. Polynomial-time autoreducibility *per se* seems to have been studied first by Ambos-Spies [5], although some types of polynomial-time *self-reducibility* (corresponding to restricted versions of autoreducibility) had been studied earlier (most notably in the work of Karp and Lipton [21]). Balcázar was among the first to give a systematic study of different types of polynomial-time self-reducibility [6]; a set A is said to be *length-decreasing self-reducible* (or "downward self-reducible") if it is polynomial-time autoreducible via a reduction that, on input x, asks only questions of length less than $|x|$.

This lecture will not attempt to survey 40 years of work on this topic. Fortunately, there are a number of excellent surveys to which I am happy to refer the reader. The Masters Thesis of Selke gives a comprehensive and systematic overview of most of this work [27], including a summary of the results of Balcázar, Mayordomo, Merkle and others, clarifying the relationship between different notions of genericity and randomness, and variants of autoreducibility [7, 24, 12].

A particularly exciting line of research on autoreducibility was introduced by Buhrman, Fortnow, van Melkebeek, and Torenvliet [8]. They showed that for "*large*" complexity classes containing DSPACE($2^{2^{n^{O(1)}}}$), *not* all $\leq^{\mathrm{p}}_T$-complete sets are polynomial-time autoreducible, while also giving a non-relativizing proof that all $\leq^{\mathrm{p}}_T$-complete sets for EXP *are* polynomial-time autoreducible. Even more intriguingly, they showed that for "intermediate" classes (such as EXPSPACE and DTIME($2^{2^{n^{O(1)}}}$)), any resolution of the question about autoreducibility of complete sets would result in solving some long-standing open questions in complexity theory; if they are autoreducible, then NL $\neq$ NP; if not, then EXP is not equal to the polynomial hierarchy. There has been quite a bit of additional progress regarding the self-reducibility properties of complete sets for various complexity classes; for example, see these excellent surveys: [10, 9, 17, 16].

The following two sections describe some aspects of self-reducibility that are not discussed by the aforementioned surveys, but that may be of interest to participants in this conference.

2 Circuit Size Lower Bounds

A recent development presents a way in which self-reducibility might point to a path around a daunting obstacle to proving *circuit size* lower bounds. TC^0 is a well-studied circuit complexity class, consisting of those problems that can be solved by polynomial-size *threshold circuits* of constant-depth. (That is, there is some constant d such that, for every input length, there is a depth-d circuit of (negated and non-negated) MAJORITY gates solving the problem.) Although it is widely believed that many problems in P lie outside of TC^0, it remains unknown whether NEXP is contained in TC^0! Razborov and Rudich, in their work on "Natural Proofs," explained our current inability to prove lower bounds against TC^0 by showing that any lower bound argument that adheres to a certain "natural" approach is doomed to failure, if there are pseudorandom function generators computable in TC^0 [26]. (If popular conjectures regarding the cryptographic complexity of factoring are true, then there *are* cryptographically secure pseudorandom function generators computable in TC^0 [25].)

The connection to self-reducibility involves a type of "*strong*" downward self-reducibility that was originally defined by Goldwasser *et al.* [18]. A set A is strongly downward self-reducible if, for all input lengths n, there is a constant-depth *oracle circuit* of polynomial-size for A, where the oracle is A, and all queries are *very short* (say, of size $\sqrt{n}$). (A related notion, defined in terms of polynomial-time computation instead of constant-depth circuits, has also been considered [13, Theorem 3.3].) It turns out that several well-studied problems (such as the problem of evaluating a Boolean formula) are strongly downward self-reducible via *linear-size* reductions [4], and furthermore, if any such problem lies in TC^0, then it has TC^0 circuits of *nearly linear* size. The significance of this is that, in order to prove that such a problem lies outside of TC^0, it suffices to give a "natural" proof of a modest size lower bound (such as size $n^{1.0001}$, and then this would yield a "non-natural" lower bound, showing that P does not lie in TC^0 [4]. (For a very different line of attack, showing how modest lower bounds would yield "non-natural" lower bounds for non-linear logarithmic-depth circuits, see [11].)

3 Random Self-Reducibility and Kolmogorov Complexity

A set A is *random self-reducible* if there is a probabilistic oracle machine accepting A, using A as an oracle, where queries to the oracle are made "at random". (For a more satisfactory definition, see, e.g., [27].) Random self-reducible sets have found wide application in complexity theory. For instance, Fortnow and Santhanam [15] were able to give an improved time hierarchy theorem for probabilistic computation, showing that BPTime(n^k)/1 $\neq$ BPTime($n^{k'}$)/1 if $k < k'$, by making crucial use of the fact that there is a problem that is complete for PSPACE that is both downward self-reducible and random self-reducible [29]. (Random self-reducibility is usually considered to be a property of quite complex sets such as PSPACE-complete sets; it is unlikely that there are NP-complete sets that are random self-reducible [14]. Note however that there are some *regular* sets that are both random self-reducible and downward self-reducible, and this has been used to show that if Boolean formula evaluation requires TC^0 circuits of size $n^{1.0001}$, then probabilistic TC^0 circuits can be simulated in subexponential time [1].)

The promised connection to Kolmogorov complexity is indirect, and is tied up with the following inclusion [3]:

$$\text{PSPACE} \subseteq \text{P}^{R_{\text{C}}}$$

where R_{C} is the set of Kolmogorov-random strings: $R_{\text{C}} = \{x : C(x) \geq |x|\}$. What is special about PSPACE? Is *every* decidable set efficiently reducible to R_{C}? Is the halting problem efficiently reducible to R_{C}? Kummer does show that the halting problem is reducible to R_{C} in some computable time bound [22], but for the type of reduction that he gives (a disjunctive truth-table reduction) it is known that at least exponential time is required [2].

The main reason why PSPACE is the largest class known to be efficiently reducible to R_{C} is this: No larger class can have a complete problem that is downward self-reducible. The reduction showing that PSPACE can be reduced to R_{C} exploits the properties of a pseudorandom generator G_f that Impagliazzo and Wigderson show how to construct from any function f that is both downward and random self-reducible [19] (and, as we have mentioned above, such problems exist that are complete for PSPACE). The output of this generator can be distinguished from truly random strings, using R_{C} as an oracle. Impagliazzo and Wigderson show that this allows one to use R_{C} to efficiently compute f. For details, see [3].

Although this proof relies heavily on downward self-reducibility, it would be good to know if this is essential. One intriguing possibility is that it could be possible to *characterize* certain complexity classes in terms of efficient reductions to the non-computable set R_{C}; some preliminary steps in this direction have already been taken [2]. It is tempting (albeit premature) to speculate about what the implications would be, of adding such an unlikely avenue to apply the techniques of computability theory to questions of complexity.

Acknowledgments

The research of the author is supported in part by NSF Grants DMS-0652582, CCF-0830133, and CCF-0832787.

References

1. E. Allender, V. Arvind, and F. Wang. Uniform derandomization from pathetic lower bounds. Technical Report TR10-069, Electronic Colloquium on Computational Complexity (ECCC), 2010.

2. 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.
3. E. Allender, H. Buhrman, M. Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM Journal on Computing*, 35:1467–1493, 2006.
4. E. Allender and M. Koucký. Amplifying lower bounds by means of self-reducibility. *Journal of the ACM*, 57:14:1 – 14:36, 2010.
5. K. Ambos-Spies. P-mitotic sets. In *Logic and Machines*, volume 171 of *Lecture Notes in Computer Science*, pages 1–23, 1983.
6. J. Balcázar. Self-reducibility. *J. Comput. Syst. Sci.*, 41(3):367–388, 1990.
7. J. Balcázar and E. Mayordomo. A note on genericity and bi-immunity. In *Structure in Complexity Theory Conference*, pages 193–196, 1995.
8. H. Buhrman, L. Fortnow, D. van Melkebeek, and L. Torenvliet. Separating complexity classes using autoreducibility. *SIAM J. Comput.*, 29(5):1497–1520, 2000.
9. H. Buhrman and L. Torenvliet. Complete sets and structure in subrecursive classes. In *In Proc. Logic Colloquium '96*, volume 12 of *Lecture Notes in Logic*, pages 45–78, 1998.
10. H. Buhrman and L. Torenvliet. A Post's program for complexity theory. *Bulletin of the EATCS*, 85:41–51, 2005.
11. Z. Dvir. On matrix rigidity and locally self-correctable codes. In *IEEE Conference on Computational Complexity (CCC)*, 2010. to appear.
12. T. Ebert, W. Merkle, and H. Vollmer. On the autoreducibility of random sequences. *SIAM Journal on Computing*, 32(6):1542–1569, 2003.
13. P. Faliszewski and M. Ogihara. On the autoreducibility of functions. *Theory Comput. Syst.*, 46(2):222–245, 2010.
14. J. Feigenbaum and L. Fortnow. On the random-self-reducibility of complete sets. *SIAM Journal on Computing*, 22:994–1005, 1993.
15. L. Fortnow and R. Santhanam. Hierarchy theorems for probabilistic polynomial time. In *Proc. IEEE Symp. on Found. of Comp. Sci. (FOCS)*, pages 316–324, 2004.
16. C. Glaßer, M. Ogihara, A. Pavan, A. L. Selman, and L. Zhang. Autoreducibility and mitoticity. *SIGACT News*, 73(5):735–754, 2007.
17. C. Glaßer, A. Pavan, A. L. Selman, and L. Zhang. Mitosis in computational complexity. In *Theory and Applications of Models of Computation (TAMC)*, volume 3959 of *Lecture Notes in Computer Science*, pages 61–67, 2006.
18. S. Goldwasser, D. Gutfreund, A. Healy, T. Kaufman, and G. Rothblum. Verifying and decoding in constant depth. In *Proc. ACM Symp. on Theory of Computing (STOC)*, pages 440–449, 2007.
19. R. Impagliazzo and A. Wigderson. Randomness vs. time: de-randomization under a uniform assumption. *Journal of Computer and System Sciences*, 63:672–688, 2001.
20. C. Jockusch and M. Paterson. Completely autoreducible degrees. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 22:571–575, 1977.
21. R. M. Karp and R. J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proc. ACM Symp. on Theory of Computing (STOC)*, pages 302–309, 1980.
22. M. Kummer. On the complexity of random strings. In *Proc. of Symp. on Theo. Aspects of Comp. Sci. (STACS)*, volume 1046 of *Lecture Notes in Computer Science*, pages 25–36, 1996.
23. R. E. Ladner. Mitotic recursively enumerable sets. *J. Symb. Log.*, 38(2):199–211, 1973.
24. W. Merkle and N. Mihailovic. On the construction of effectively random sets. *J. Symb. Log.*, 69(3):862–878, 2004.
25. M. Naor and O. Reingold. Number-theoretic constructions of efficient pseudo-random functions. *J. ACM*, 51(2):231–262, 2004.
26. A. A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:24–35, 1997.

27. J. Selke. Autoreducibility and friends: About measuring redundancy in sets. Master's thesis, Universität Hanover, 2006.
28. B. A. Trakhtenbrot. On autoreducibility. *Dokl. Akad. Nauk SSSR*, 11:814–817, 1970.
29. Luca Trevisan and Salil P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.