WHAT IS...

a Natural Proof?

Timothy Y. Chow

Have you ever wondered whether the reason there is (apparently) no simple proof that P $\neq$ NP is that P $\neq$ NP? Or to turn it around, that an easy proof that P $\neq$ NP would somehow solve a problem that is hard not only in the Millennium Prize sense but also in the computational-complexity sense?

Stated this naïvely, the above idea does not quite make sense, but in a paper that won them the 2007 Gödel Prize, Alexander Razborov and Steven Rudich [3] proved a result that showed that there is something to this intuition after all. Informally, their argument is as follows. Let T be an NP-hard function; for example, let T take as input a list of cities and distances and as output an optimal traveling-salesman tour through the cities. Suppose that your strategy for proving that P $\neq$ NP is to identify some property $\mathscr{P}$ that T has, but that no polynomial-time computable function has. Suppose further that $\mathscr{P}$ has the following *natural* characteristics: $\mathscr{P}$ is *efficiently computable* (i.e., it is computationally easy to determine whether any given function possesses property $\mathscr{P}$) and moreover *many functions* possess $\mathscr{P}$—enough that a *random function* would possess $\mathscr{P}$ with some non-negligible probability. By exhibiting $\mathscr{P}$, you would indeed separate P from NP, but at the same time you would do something else equally spectacular: you would *break some of the strongest known cryptosystems!* Cryptographers consider a cryptosystem broken if the outputs (a.k.a. *ciphertexts*) of the encryption function E can be efficiently distinguished from random strings—or equivalently, if E, when presented as a lookup table, can be efficiently distinguished from a random function. But this can be done simply by checking whether E possesses $\mathscr{P}$; any practical encryption function will be computable in polynomial time and therefore will *not* possess property $\mathscr{P}$—as opposed to a random function, which will possess property $\mathscr{P}$ with reasonably high probability.

Turning this around, if you believe that there do exist secure cryptosystems, then your strategy for proving that P $\neq$ NP cannot be too simple-minded; your proposed property $\mathscr{P}$ must either be difficult to compute or it must focus on some very special features of NP-hard functions that are not shared by random functions.

For those who like precision, we now give a formal statement of the Razborov–Rudich theorem. (The reader not interested in technicalities may skip ahead to just past the theorem statement without loss of continuity.) By a *Boolean function on n variables* we mean a $\{0,1\}$-valued function on n $\{0,1\}$-valued variables. (We use the term "Boolean" because we think of 0 as FALSE and 1 as TRUE.) We would like to measure the computational complexity of Boolean functions; to do so we must consider not just a single Boolean function but a *sequence* (f_n), where each f_n is a Boolean function of n variables. If the minimum number of ANDs, ORs, and NOTs needed to express f_n is bounded by a polynomial function of n, then we write $(f_n) \in \mathrm{P/poly}$. It can be shown that if there is a polynomial-time Turing machine that outputs $f_{|x|}(x)$ for any input binary string x (where $|x|$ denotes the length of x), then $(f_n) \in \mathrm{P/poly}$. That is, $\mathrm{P} \subseteq \mathrm{P/poly}$. Therefore, one strategy for showing that P $\neq$ NP is to show the stronger statement that NP $\not\subseteq$ P/poly. It is this strategy that Razborov and Rudich's result addresses.

Timothy Y. Chow is a member of the research staff at the IDA Center for Communications Research in Princeton. His email address is tchow@alum.mit.edu.

More specifically, they consider *properties* of Boolean functions that are not possessed by any function in P/poly. A property is just a sequence (C_n) where each C_n is a set of Boolean functions on n variables, and to say that no function in P/poly possesses the property means that if $f_n \in C_n$ for infinitely many n, then $(f_n) \notin$ P/poly. Notice that C_n can itself be thought of as a Boolean function on 2^n variables, sending the truth table of f_n to 1 if and only if $f_n \in C_n$. We say that (C_n) is *efficiently computable* if it is in P/poly when thought of as a sequence of Boolean functions in this way. Finally, we say that *many functions possess the property* (C_n) if C_n contains at least $2^{-O(n)}$ of all Boolean functions on n variables. If a property is simultaneously efficiently computable and possessed by many functions, then we say that it is *natural*. Then Razborov and Rudich's main theorem is:

Theorem. *If there exists a natural property* (C_n) *that is not possessed by any function in* P/poly, *then there do not exist any* 2^{n^ϵ}*-hard pseudorandom number generators.*

The definition of a 2^{n^ϵ}-hard pseudorandom number generator is somewhat technical, and we will omit it here; suffice it to say that if factoring integers or computing discrete logarithms is sufficiently hard, or if any of various popular candidates for symmetric-key cryptography is indeed secure, then 2^{n^ϵ}-hard pseudorandom number generators do exist.

Note that the Razborov–Rudich theorem does not mention NP specifically and therefore applies equally to attempts along the same lines to prove, for example, that P $\neq$ PSPACE.

Why is the Razborov–Rudich theorem considered such a big deal? After all, even if one believes in pseudorandom number generators, all their theorem says is that a property that distinguishes an NP-hard function from every P/poly function must be either hard to compute or not possessed by many functions. At first glance, this result may seem to be just providing some guidance to someone trying to construct a suitable property, rather than erecting a formidable barrier to this avenue of proof.

Indeed, this optimistic interpretation is perfectly reasonable and was suggested by Razborov and Rudich themselves in their paper. However, it turns out that in practice, it is not so easy to dream up candidate properties of NP-hard functions that are provably not possessed by easily computed functions and also not possessed by many functions. Razborov and Rudich devote many pages of their paper to analyzing proofs in the literature that various explicit functions are hard to compute, and they show that in example after example the proofs rely on exhibiting natural properties. (That so many proofs in the literature fall into this category is their motivation for calling such proofs *natural*.) Furthermore, they also show that any attempt to construct a property in a certain inductive manner, so as to produce something they call a *formal complexity measure*, is doomed to result in a property that is possessed by many functions.

A further point to consider is that one of the main techniques for producing properties that are not easy to compute is *diagonalization*. That is, one somehow enumerates all the easy-to-compute functions and uses a version of Cantor's diagonal argument to produce a function not in the list. While diagonalization arguments do indeed produce properties that are not easy to compute, they often suffer from another malady: they usually *relativize*. Space does not permit us to explain relativization in detail, but the main point is that it has been known since a 1975 paper by Baker, Gill, and Solovay [2] that relativizing arguments cannot possibly prove P $\neq$ NP. Thus Razborov–Rudich and Baker–Gill–Solovay collectively shut off many tempting routes to proving P $\neq$ NP.

Nevertheless, it is my personal opinion that the optimistic approach is the right one; that is, the Razborov–Rudich result should be regarded as a hint, and not a barrier, to separating complexity classes. The only real barrier is our lack of imagination.

For a more detailed exposition of natural proofs, the interested reader is referred to the excellent exposition by Steven Rudich [4, Lecture 8], as well as of course Razborov and Rudich's original paper [3]. Eric Allender's paper [1] is also highly recommended as a source of ideas for how to "think outside the box" of natural proofs.

References

[1] Eric S. Allender, Cracks in the defenses: Scouting out approaches on circuit lower bounds, in: Edward A. Hirsch, et al. (eds.), *Computer Science—Theory and Applications: Third International Computer Science Symposium in Russia, CSR 2008, Moscow, Russia, June 2008, Proceedings,* Lecture Notes in Computer Science, vol. 5010, Springer, 2008, pp. 3–10.

[2] Theodore Baker, John Gill, and Robert Solovay, Relativizations of the P =? NP question, *SIAM J. Comput.* **4** (1975), 431–442.

[3] Alexander A. Razborov and Steven Rudich, Natural proofs, *J. Comput. System Sci.* **55** (1997), 24–35.

[4] Steven Rudich and Avi Wigderson (eds.), *Computational Complexity Theory*, American Mathematical Society and IAS/Park City Mathematics Institute, 2004.