Simons Institute Research Vignette: Hard Problems All The Way Up

June 21, 2015

Computational complexity theory is the branch of computer science that studies the nature and limits of efficient computation; an overarching goal in this area is the classification of computational problems according to their inherent difficulty. One of the most intensively studied such classifications is provided by the *Polynomial Hierarchy*, which was introduced by Albert Meyer and Larry Stockmeyer in 1972. This hierarchy classifies problems according to a natural notion of logical complexity, and is defined with an infinite number of levels: problems at the zeroth level are the "easiest", and for every integer k, problems at the $(k+1)$-st level have logical complexity "one notch higher" than those at level k. In this vignette we describe some recent work done at the Simons Institute, in collaboration with Rocco Servedio of Columbia University, which sheds new light on the structure of this hierarchy.

The polynomial hierarchy begins at the zeroth level with the class P of problems solvable in polynomial time — the easiest problems. In theoretical computer science P captures an elegant and robust notion of computational efficiency: polynomial-time algorithms are considered efficient, and problems in P are considered tractable. Examples of fundamental problems in P include Linear Programming, Matching, and Primality. At the first level above P in the hierarchy is NP, the class of problems solvable in *nondeterministic* polynomial time, meaning that a "yes" answer can be verified efficiently. Well-known examples of problems in NP include the Boolean Satisfiability and Traveling Salesman problems, among countless others that arise in surprisingly diverse contexts.

Clearly $\mathsf{P} \subseteq \mathsf{NP}$, since the answer to a computational problem can be verified efficiently if it can be determined efficiently. But is the converse true? No efficient algorithms are known for the hardest problems in NP, the so-called NP-complete problems: while a "yes" answer to these problems can be verified in polynomial time, to date our best algorithms for solving them — determining whether the answer is "yes" or "no" — all run in exponential time. Indeed, the famous $\mathsf{P} \neq \mathsf{NP}$ conjecture asserts that there do not exist efficient algorithms for solving NP-complete problems: any algorithm for these problems must run in super-polynomial time. In other words, the $\mathsf{P} \neq \mathsf{NP}$ conjecture asserts that first two levels of the Polynomial Hierarchy are distinct.

To Infinity and Beyond

The Polynomial Hierarchy extends beyond P and NP to include an infinite number of classes of increasing logical complexity: just as NP generalizes P, the second level of the hierarchy generalizes

NP, the third generalizes the second, and so on ad infinitum. To understand this hierarchy it is convenient to think of problems in NP as asking questions with a single existential quantifier. For example, the BOOLEAN SATISFIABILITY problem asks if there exists a satisfying assignment to a Boolean formula, and the TRAVELING SALESMAN problem asks if there exists a short tour visiting every city exactly once. Problems at the k-th level of the hierarchy allow for questions with not just one existential quantifier, but k alternating quantifiers:

Does there exist X, such that for all Y, there exists Z such that ...?

As a simple example, consider the problem of determining if a Boolean formula ϕ is the smallest one among all that compute the same function. This problem, called FORMULA MINIMIZATION, is in the second level of the hierarchy (right above NP) since it asks a question with two quantifiers: "Does there exist a formula φ of smaller size than ϕ, such that for all assignments X, we have $\varphi(X) = \phi(X)$?" Intuitively, the second quantifier makes the problem more complex than problems in NP, which have only one quantifier. While an affirmative answer to an NP problem like BOOLEAN SATISFIABILITY ("Does there exist X such that $\phi(X) = 1$?") naturally admits a short proof ("Here is X such that $\phi(X) = 1$"), it is widely conjectured that there is no easy way to certify the minimality of ϕ in FORMULA MINIMIZATION — how does one succinctly prove that no smaller φ computes the same function as ϕ? — and hence that FORMULA MINIMIZATION is not in NP. So just as we believe that the zeroth and first levels of the hierarchy are distinct (i.e. $\mathsf{P} \neq \mathsf{NP}$), we also believe that first and second levels are distinct.

A central conjecture in computational complexity takes the above to its logical extreme and posits a far-reaching generalization of $\mathsf{P} \neq \mathsf{NP}$: *all* the infinitely many levels of the polynomial hierarchy are distinct. Just as BOOLEAN SATISFIABILITY is conjectured to not be in P, and FORMULA MINIMIZATION is conjectured to not be in NP, it is conjectured that for every k, there are problems in the $(k+1)$-st level that are not in the k-th level.

The hierarchy has infinitely many distinct levels in *some* world

More than forty years after Meyer and Stockmeyer's paper, we remain far from separating even the zeroth and first levels of the hierarchy (showing $\mathsf{P} \neq \mathsf{NP}$), much less showing that the hierarchy has infinitely many distinct levels. However, there has been significant success in attacking an important variant of these conjectures that was put forth by Meyer himself. Drawing inspiration from mathematical logic, we imagine a world in which algorithms have access to an *oracle* $\mathcal{O}_A$ giving out answers to a computational problem A "for free": an algorithm may query $\mathcal{O}_A$ with an input X to A, and is instantly given the answer without having to spend any time solving A on X itself. We may then ask: does $\mathsf{P} \neq \mathsf{NP}$ in this hypothetical world? And if so, what about the stronger conjecture that the hierarchy has infinitely many distinct levels?

In 1986 breakthrough results of Andrew Yao and Johan Håstad answered Meyer's question in the affirmative: there is an oracle relative to which the polynomial hierarchy has infinitely many distinct levels. Yao and Håstad proved this by leveraging a crucial connection, established by Merrick Furst, James Saxe, and Michael Sipser a few years earlier, between the polynomial hierarchy and Boolean circuits . To illustrate this connection, let's sketch the proof of the existence of an oracle relative to which $\mathsf{P} \neq \mathsf{NP}$. Somewhat surprisingly, it follows from an elementary fact from Boolean circuit complexity:

The function $\mathrm{OR}(Y) = Y_1 \vee \cdots \vee Y_N$ *cannot be computed by a decision tree of depth* $\mathrm{polylog}(N)$*.*

How does this relate to P and NP? We first observe that the OR function defines, for every oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}$, a computational problem $L_{\mathcal{O}}$ in NP: viewing $\mathcal{O}$ as a bit string $\mathcal{O}^*$ of length $N = 2^n$ (encoding its truth table), the answer to $L_{\mathcal{O}}$ is "yes" if and only if $\mathrm{OR}(\mathcal{O}^*) = 1$; that is, the answer is "yes" iff there exists an X such that $\mathcal{O}(X) = 1$. Now $L_{\mathcal{O}}$ is easily seen to be in NP thanks to the hallmark existential quantifier, and indeed, $L_{\mathcal{O}}$ remains in NP even if the N-variable OR function is replaced with depth-two circuits satisfying certain technical conditions (the OR of $\exp(\mathrm{poly}(n)) = \mathrm{quasipoly}(N)$ many ANDs of fan-in $\mathrm{poly}(n) = \mathrm{polylog}(N)$). Next, we note that any algorithm in P making $\mathrm{poly}(n) = \mathrm{polylog}(N)$ many queries to $\mathcal{O}$ can be represented as a $\mathrm{polylog}(N)$-depth decision tree querying the coordinates of the bit string $\mathcal{O}^*$. And since decision trees of depth $\mathrm{polylog}(N)$ cannot compute the N-variable OR function (by the elementary fact stated above), this implies the existence of an oracle $\mathcal{O}$ such that $L_{\mathcal{O}}$ is in NP but not P.

This correspondence extends to all higher levels of the hierarchy: depth-$(k+1)$ circuits satisfying certain technical conditions define for every oracle $\mathcal{O}$ a problem $L_{\mathcal{O}}$ in the k-th level of the hierarchy. By an extension of the argument above, to separate all levels of the hierarchy it suffices to establish a *depth hierarchy theorem* for Boolean circuits:

For every integer k, there is a polynomial-size depth-$(k+1)$ circuit C such that any depth-k circuit computing the same function as C requires super-quasipolynomial size.

Yao and Håstad proved precisely such a hierarchy theorem for circuit depth, and as a corollary confirmed the existence of an oracle relative to which the Polynomial Hierarchy has infinitely many distinct levels. In sharp contrast with the elementary fact underlying the oracle separation of P from NP, Yao and Håstad's proof is a technical tour de force, culminating a long line of work on the problem. At the heart of their proof is a delicate application of the *method of random restrictions*, a technique that dates back to Bella Subbotovskaya in the 1960's. Very roughly speaking, this method employs probabilistic arguments to show that Boolean circuits can be dramatically simplified by substituting randomly chosen constant values for a randomly chosen subset of their input variables.

Our work: The hierarchy has infinitely many distinct levels in *almost every world*

Yao and Håstad's celebrated result is an important piece of evidence in favor of the conjecture that the hierarchy comprises infinitely many levels in our actual, oracle-less world. However, it does not provide much information about the oracle witnessing this separation…could it be that this oracle is a particularly degenerate one, craftily engineered so that the hierarchy has infinitely many distinct levels in this atypical world? Could the truth actually be the opposite relative to most other oracles, in most other worlds? Håstad and many other researchers have asked whether his result can be strengthened to address these concerns: is the hierarchy infinite relative to not just *some* oracle, but a *typical* oracle? Does the hierarchy have infinitely many distinct levels in not just one world, but *almost every* world?

With Rocco Servedio we answer Håstad's question in the affirmative. Perhaps unsurprisingly, we do so via the Furst–Saxe–Sipser connection to Boolean circuits, by proving an *average-case* extension of Yao and Håstad's depth hierarchy theorem for Boolean circuits:

For every integer k, there is a polynomial-size depth-$(k+1)$ circuit C such that any depth-k circuit that agrees with C **on at least 51% of all inputs** *requires super-quasipolynomial size.*

This is a significant strengthening of the Yao–Håstad lower bound, which only shows that depth-k circuits of quasipolynomial size cannot agree with C on *every* input. Note that a constant function (which is entirely trivial from the point of view of circuit complexity) achieves 50% agreement; we show that depth-k circuits of quasipolynomial size can barely do any better.

A key component of our proof is an extension of the method of random restrictions, which we call the *method of random projections.* While restrictions work by fixing variables to 0, to 1, or leaving them unchanged, projections work by fixing variables to 0, to 1, or *identifying* groups of many variables — "projecting" them all to the same new variable, so that they must all take the same value. Very roughly speaking, we show that (like random restrictions) random projections simplify Boolean circuits, but the identification of variables helps maintain "useful structure" that we exploit in our lower bound. An exciting goal for future work is to find other applications of the method of random projections — perhaps, even, to give additional insights about the polynomial hierarchy.