Proofs in Three Bits or Less

Jacob Denson

November 24rd, 2017

John von Neumann is reported to have said "if people do not believe that mathematics is simple, it is only because they do not realize how complicated life is". But some mathematics is labyrinthian! Andrew Wiles's proof of Fermat's last theorem is 109 pages long. And dont remind me of those gigabyte-long computer-made proofs! Realizing my discontent, a friend referred me to a radically concise method of argument, known as an *interactive proof*, which has revolutionized our understanding of the complexity of mathematical proofs.

Think of the way your most irritating classmate follows a proof; every 30 to 40 seconds they're popping in with suspected flaws in your teacher's argument. Like your classmate, in interactive proofs an *interrogator* asks a series of yes/no questions to a *prover* who attempts to demonstate some claim. After a fixed number of questions, the interrogator decides whether they're convinced by the answer's recieved. The fewer questions, the more comprehensible the proof!

From your classmate's perspective, your teacher is probably just making an mistake. From cryptography's point of view, where interactive proofs originated, a prover might lie maliciously to trick the interrogator into believing a false claim. From either perspective, the interrogator must ask 'checkable' questions so they can be sure responses are correct. It is mathematically interesting to attempt to minimize the number of questions while preserving several criteria:

- *Perfect Completeness*: For a true statement, there exists a series of answers to our questions which will convince us the statement is correct.
- *Perfect Soundness*: For a false statement, no series of answers to our questions will convince us the statement is true.

The cost of perfection is efficiency. The statement

" This statement can't convince a perfect prover in fewer than a googolplex questions "

is true (how could it be false?) but is *very* slow to check. The main improvement interactive proofs give us is the ability to weaken our requirements. If the interrogator chooses their questions randomly over a distribution of possible question, we can introduce weaker constraints:

- *Imperfect Completeness*: For a true statement, there exists a series of answers to our randomly chosen questions which convince the prover with probability $\geq 1/2$.
- *Imperfect Soundness*: For a false statement, every answer to the prover's questions fail to convince the prover with probability $\geq 1 - \varepsilon$.

Here's an example of an imperfect interactive proof. Given two graphs G_0 and G_1, an *isomorphism* is a map between their vertices preserving adjacency of vertices. Consider determining whether two graphs G_0 and G_1 with n nodes are *not* isomorphic to one another. While proving two graphs *are* isomorphic is easy (just give me an isomorphism!), it is *not* easy to show that two graphs are not isomorphic (a problem familiar to anyone who's had the headache of proving two groups aren't isomorphic). However, we provide an interactive proof which asks only a single random question. Given H isomorphic either to G_0 or G_1, we ask a '0/1' question of the form

"If H is isomorphic to G_0, output zero.
If H is isomorphic to G_1, output one."

To choose our question, we fix an index $i \in \{0, 1\}$, and a permutation $\nu \in S_n$, chosen uniformly at random. We form the graph $\nu(G_i)$ by permuting the indices of the nodes of the graph G_i. Now we ask the question above for $H = \nu(G_i)$. The prover should give the answer i, since H is isomorphic to G_i, and if the prover fails to give this answer, we remain unconvinced of the claim. If the graphs are not isomorphic, and the prover always answers correctly, we will always be convinced, implying perfect completeness. On the other hand, if the graphs are isomorphic, the random graph $\nu(G_i)$ we obtain is independent of the

index i, and so for any answer the prover gives, there is a 50% chance of being caught out, giving imperfect soundness.

If we are willing to forgive some margin of error, we obtain a most magical result from the mathematical field of complexity theory, whose goal is to analyze the difficulty of mathematical problems.

Theorem 1 (The PCP Theorem)**.** *For $\varepsilon > 0$, every feasibly checkable*[1] *theorem is checkable with imperfect soundness and imperfect completeness in only 3 questions!*

Here's some examples of this remarkable theorem at work. Pick your favourite integer with 100 prime factors, and give me a *proof* that the integer has 100 prime factors. Giving the prime factorization is sufficient, because an interrogator can check that multiplying the prime factors together gives your original number. But the PCP theorem guarantees that you can give me a 'random' argument that's only three bits long. It shows that languages are incredibly efficient at communicating true or false facts. But make sure you understand the limitations of the theorem: I won't be able to know *what* the factorization is, just that such a factorization is.

Here's a more colorful example. A graph is a collection of dots connected by lines. The three coloring problems asks whether we can color the dots of a graph with three different colors, so no two adjacent dots have the same color. It seems that for a large graph, a proof that a graph is three colorable is quite large. But the PCP theorem says that you can communicate that you have found a three coloring on the graph (but not *the* three coloring) in three questions or less!

The theorem is remarkable. The main way the theorem is useful

(and this is one of the nice cases pure mathematics actually having *some* application to real life, since people have been discussing using PCP technologies to simplify transaction verification in certain digital currencies, like Bitcoin).

It says that languages provide a way to efficiently encode *local* properties of mathematical objects *globally*, where they can be easily checked, and I'm sure these ideas can be made useful in a large number of mathematical areas.

An interesting way to view the problem is that certain 'local' properties of mathematical objects can be encoded in such a way that these local properties become global. Here is an analogy from Dinur's talk on the subject. Imagine if we have a piece of toast, and, by sampling three points at random on the piece of toast, we want to determine whether the toast has jam on it. The encoding process as above can be compared to taking a knife, and smothering it all over the piece of toast, not looking at the bread. If there is any jam on the bread, then the knife will smother the jam all over the piece of toast, and after this, if we have any local jam on the piece of toast, it will now become a global property of the bread, and this can be very easily obtained by sampling three points on the piece of toast.

Example. Three colorability of graphs is a very local property. In particular, a graph can appear to be three colorable, but fail to be three colorable at a single position. If someone said to us they had a three coloring of a graph, and we could only ask them the color of individual vertices, then it would take $|V|$ questions to determine whether the person actually had a three coloring. If you had access to the entire three coloring, it would only take you $O(|V| + |E|)$ steps to check whether a given coloring was actually a three coloring, so the problem of checking a three coloring is computationally feasible, and the PCP theorem guarantees that we can select three questions at random about the prover's proof of the three colorability of the graph, and to a high degree of accuracy, determine whether the prover's proof given is false, or whether the proof is correct. Note that "Do you have a three coloring", is NOT a good question to ask in this context, because we have no reason to trust the prover, and if he said "Yes" we have no information about his proof. Thus there is a way of encoding the three colorability of a graph in such a way that the local property of failing to be three colorable becomes global.

Example. The mathematical proof of a mathematical theorem is a *very* local property. Proofs can seem correct, but have a single 'bug' at a single location which invalidates the entire proof. The PCP theorem says that there is a new way of encoding these proofs such that bugs are globally expressed, and become easy to find.

[1] In the language of complexity theorem, this means the theorem is $\in \mathbf{NP}$