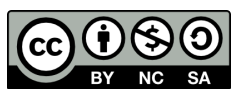
MIP$*$ = RE

Klaus Sutner

Carnegie Mellon University

Spring 2024



$\mathsf{MIP}^* = \mathsf{RE}$

Zhengfeng Ji[*1], Anand Natarajan[†2,3], Thomas Vidick[‡3], John Wright[§2,3,4], and Henry Yuen[¶5]

[1]*Centre for Quantum Software and Information, University of Technology Sydney*
[2]*Institute for Quantum Information and Matter, California Institute of Technology*
[3]*Department of Computing and Mathematical Sciences, California Institute of Technology*
[4]*Department of Computer Science, University of Texas at Austin*
[5]*Department of Computer Science and Department of Mathematics, University of Toronto*

January 14, 2020

Alas, the thing is **165 pages**, some highly technical.

Plus, this is really outside of my wheelhouse. Take everything I am going to say in section 3 with pounds of salt.

Prover and Verifier

Recall the interactive version of $\mathbb{NP}$, hard-to-solve but easy-to-verify problems. Suppose $x \in \mathbf{2}^\star$ is some instance.

Prover provides evidence (a "proof") that x is a yes-instance to the

Verifier checks the information provided by the prover, and announces a decision.

The prover will be allowed to have arbitrary computational resources, but the verifier must be polynomial time.

Completeness requires that for any yes-instance the prover can convince the verifier.

Soundness requires that for no-instances cheating on the provers' side will not help to convince the verifier.

Merriam-Webster

(1) The cogency of evidence that compels acceptance by the mind of a truth or a fact.

(2) The process or an instance of establishing the validity of a statement especially by derivation from other statements in accordance with principles of reasoning.

Hopelessly vague, though the second part points in the right direction.

Fortunately, in math we can do a whole lot better.

In the classical logic setting, and following Gödel, a proof is a sequence of formulae in some formal language (say, first-order logic) that uses only axioms, given assumptions and rules of inference:

$$\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_n \qquad\qquad \varphi_n \quad \text{target assertion}$$

A typical rule of inference is the cut rule:

$$\frac{\varphi \Rightarrow \psi \qquad \psi \Rightarrow \chi}{\phi \Rightarrow \chi}$$

Note that this is a tricky rule: the middle formula ψ vanishes without a trace. It is difficult to reverse engineer a proof using cuts; in general, we have no idea what ψ might be.

One of Hilbert's brilliant ideas was to think of proofs as just another kind of mathematical object that can be studied the same way as, say, vector spaces.

Classical proof theory has unearthed a wealth of information about the properties of various formal systems built around this notion of proof. Many of the arguments are quite technical, and seem to attract a certain type of temperament (just like the study of degrees of unsolvability).

Theorem (Gentzen's Hauptsatz, 1936)

In the sequent calculus, the cut rule can be eliminated.

The proof is requires induction on ordinals, not just the naturals.

One application of the theorem is to check a system for consistency.

And Computation?

In 1971 Stephen Cook studied the question whether an efficiently checkable proof of an assertion always leads to an efficient way of finding such a proof. On the face of it, checking seems far easier than finding.

> Stephen Cook
>
> The Complexity of Theorem Proving Procedures
>
> Proc. STACS 1971

This seminal paper is the beginning of the $\mathbb{P} = \mathbb{NP}$ saga.

At least if we disregard Gödel's letter to von Neumann the 1950s.

Here is a truly amazing fact: Kurt Gödel more or less introduced the $\mathbb{P}$ versus $\mathbb{NP}$ problem in a 1956 letter to von Neumann, who was dying of cancer at the time. Sadly, Gödel never got an answer (or it is lost)[†].

And, Gödel being Gödel, naturally never bothered to communicate his insights to anyone else.

See Gödel letter and Lipton Blog.

[†] The letter was discovered in 1989, almost 20 years after Cook's paper.

- For any first-order formula F and $n \in \mathbb{N}$, it is decidable if there is a proof of F of length at most n.
- Let $\Psi(F, n)$ be the time complexity of a corresponding Turing acceptor.
- Let $\varphi(n) = \max_F \Psi(F, n)$. This exists because F is also bounded by n.

So $\varphi(n)$ is the running time of a universal theorem prover looking for proofs of length n. How large could $\varphi(n)$ be?

Clearly, there are exponentially many potential proofs of length n. At first glance, the only obvious solution would be to search through all of them, so something like $2^{O(n)}$ seems quite reasonable.

Amazing Fact:

It is quite difficult to prove an apparently trivial result like $\varphi(n) \geq k \cdot n$.

Gödel suggested that it just might be the case that $\varphi(n)$ is linear, or maybe some very low degree polynomial. He felt that "trial and error" (or, in modern parlance, nondeterminism) might be replaced by a more systematic approach.

Linear time would be truly amazing: given F, we could pick n large enough so it covers the size of some very complex proofs currently known. Say, n could be a billion or a trillion. In a sense, we would have a solution to a truncated but still highly interesting version of the Entscheidungsproblem.

No Way?!?!

Is this simply all wishful thinking?

Well, there are other places in math where an exponential speedup is possible: instead of spending N steps on exhaustive search, we manage in just $\log N$ steps. Think primality testing. This is a much more limited problem, but Gödel did hold out some hope that a similar speed-up might happen here.

Of course, this would give us $\mathbb{P} = \mathbb{NP}$. So maybe not ...

It is rather ironic that Gödel, who single-handedly destroyed Hilbert's dream to find a universal solution to the Entscheidungsproblem, later came up with this idea. And that he asked von Neumann for help, since von Neumann spent the early part of his career working on Hilbert's program.

More Proofs

The connection between complexity theory and proof theory has lead to the consideration of various types of proofs that are not traditionally studied in classical proof theory:

- probabilistically checkable proofs
- zero-knowledge proofs
- interactive proofs
- quantum based proofs

And, some of the results of this new-fangled, CS-inspired proof theory are truly amazing.

Let's stay within the IP framework; there is a verifier $\mathcal{V}$ and a prover $\mathcal{P}$ that exchange messages. For emphasis, let's say that $\mathcal{P}$ sends a proof π.

We allow the use of random bits and require certain soundness and completeness conditions of the protocol.

Strange Idea: We allow the verifier to not read the whole proof.

This may sound strange, but since everything is probabilistic, the verifier might flip a coin to determine which parts of the proof to check.

Given two functions r and q define the class PCP(r, q) as follows:

- $\mathcal{V}$ uses $O(r(n))$ random bits and reads $O(q(n))$ bits of π.
- Completeness is 1.
- Soundness is $1/2$.

Similarly we write PCP$(\mathcal{F}_1, \mathcal{F}_2)$ for function classes.

It is easy to see that PCP$(0, \text{poly}) = \mathbb{NP}$.

Less obvious is the following.

Theorem

$\mathsf{PCP}(\log, \mathrm{poly}) = \mathbb{NP}$.

Here is the huge surprise, with a rather difficult proof, due to Arora-Safra 1992 and Arora-Lund-Motwani-Sudan-Szegedy 1992.

Theorem (PCP Theorem)

$\mathsf{PCP}(\log, 1) = \mathbb{NP}$.

Actually, the same constant works for all languages in $\mathbb{NP}$.

One way to strengthen IP-like protocols is to allow **multiple provers**. They can coordinate their strategy before the rounds start, but once the session is under way they cannot directly communicate with each other, only with the verifier.

At first glance, that may seem utterly useless: we do not constrain the computational power of the prover in an IP protocol, so whatever $\mathcal{P}_2$ can do, $\mathcal{P}_1$ can already do on her own.

True, but the verifier can cross-examine the provers and play one off against the other. This actually can help the verifier.

Theorem (1991)

$\mathsf{MIP} = \mathrm{NEXP}_1$

More amazing than anything else we have seen so far, is the fact that to really get power out of multiple provers we need to borrow an idea from quantum physics: entanglement.

Entanglement is a particularly bizarre aspect of an already hopelessly bizarre theory.

In fact, all the experts agree that no one really understands quantum physics.

Anyone who is not shocked by quantum theory has not understood a single word.

Niels Bohr

It is safe to say that nobody understands quantum mechanics.

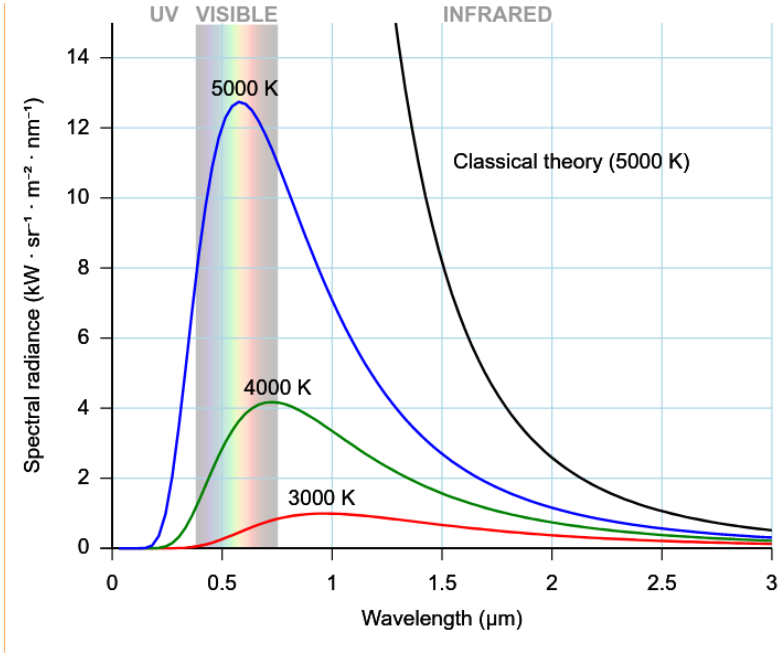
Richard Feynman

If you are not completely confused by quantum mechanics, you do not understand it.

John Wheeler

Quantum mechanics makes absolutely no sense.

Roger Penrose



The roots of quantum physics lie in the problem of describing black-body radiation, an enterprise that took up all of the 19th century.

The solution by Max Planck in 1900 introduced an unimaginable new idea: energy comes in discrete packets, not in some continuous form.

In the early days of quantum physics, mathematical foundations were a bit shaky–which was acceptable, since the results were spectacular otherwise. Together with relativity theory, quantum physics is the central accomplishment of physics in the 20th century.

And there is a noteworthy comment by Steven Weinberg in his book on quantum field theory:

> *. . . there are parts of this book that will bring tears to the eyes of the mathematically inclined reader.*

In other words, as long as the physics works well, ignore all problems with underlying math. Not my idea of a good time, but perfectly understandable.

In 1932 John von Neumann provided a rock-solid foundation for quantum physics, based essentially on two ideas:

- A quantum state can be thought of as a vector in some Hilbert space.
- Measurement then comes down to applying a linear operator.

This lead to the study of operator algebras as interesting mathematical entities, beyond just applications to quantum physics.

Von Neumann and his coworker Murray identified three types of operator algebras: type I, II and III. They analyzed type I in great detail.

Entanglement (Verschränkung according to Schrödinger who coined the term) is the bane of actual quantum computers: we would like to maintain lots of entangled qubits over long periods of time, but that is a huge technological challenge. Some people feel that it's simply impossible.

On the other hand, entanglement seems to be very popular in nature, so maybe there is a way to manage this.

Either way, entangled qubits do not enable faster-than-light communication, but there still is some sense of establishing an instantaneous connection between far away places.

Einstein famously was not too thrilled with quantum physics (and, no, this is not because he failed to understand it in great detail). In 1935 he co-authored a famous paper that proposes a Gedankenexperiment that seeks to use "local hidden-variables" to deal with annoying properties of entangled particles (spooky action at a distance).

A. Einstein, B. Podolsky, N. Rosen

Can Quantum-Mechanical Description of Physical Reality be Considered Complete?

Physical Review. 47 (10) 777–780.

In 1964 John Bell proposed a physically realizable (but rather delicate and difficult) experiment that could be used to dispel local theories, once and for all. Unfortunately, these experiments are very difficult to carry out, and it is even more difficult to make sure that there are no loopholes in the argument.

Still, starting in the 1980s, experimentalists managed to carry out this type of experiment, in ever increasing sophistication (see the CHSH game below). Alain Aspect, John Clauser and Anton Zeilinger won the Nobel prize in 2022 for their work. At this point, there is no reasonable doubt about the veracity of the claims.

The final result is: Einstein made a boo-boo, non-locality is a real thing, as spooky as it may seem.

What does quantum physics and entanglement have to do with proofs, or, more specifically, with multiple provers?

In 2003, a strange multiprover class was introduced: here the provers still must not communicate, but they are allowed to share entangled qubits.

It was utterly unclear what the computational strength of MIP$*$ would be, it looks like MIP$*$ might be weaker than MIP: the provers could use the entangled bits to cheat more efficiently.

But no ...

2012 MIP $\subseteq$ MIP$*$

2019 $\mathrm{NEXP}_2 \subseteq$ MIP$*$

We will only be interested in 2-prover, single-round protocols.

So $\mathcal{V}$ sends x and y to Alice and Bob, respectively.

Then Alice and Bob concoct their answers a and b, and send them to $\mathcal{V}$.

Lastly, $\mathcal{V}$ ponders x, y, a and b deeply, and makes a judgment. Unlike the provers, $\mathcal{V}$ is required to work in polynomial time.

Here is one way of interpreting Bell type experiments as a sort of MIP$*(2,1)$ protocol[†]:

- $\mathcal{V}$ sends a random bit x to Alice, and a random bit y to Bob.
- They send back bits a and b, respectively.

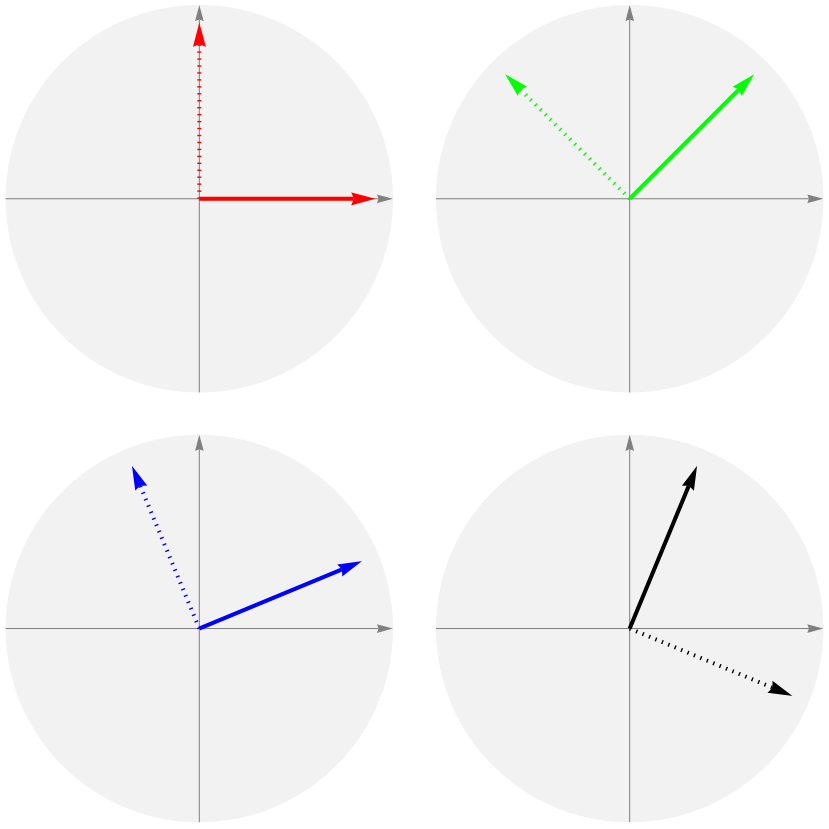
  The goal is to have $x \wedge y = a \oplus b$.

If Alice and Bob are classical, albeit all powerful, they can do no better than getting the right answer 75% of the time (assuming a uniform distribution for the verifier $\mathcal{V}$). Randomness would not help in this situation (either private or shared)

y \ x	0	1
0	$0,0$	$a,0$
1	$0,b$	a,b

[†]The Clauser-Horne-Shimony-Holt game, proposed in 1969.

But if funky provers can share entangled qubits, they can do better: the success rate goes up to slightly more than 85% (actually, $\cos^2(\pi/8)$, which is optimal according to Tsirelson using only one pair of entangled qubits).



The specific kind of problem we want to tackle with a MIP$*$ protocol is a so-called non-local game, modeled after the CHSH game from above.

Definition

A non-local game has the form $\mathfrak{G} = \langle \mu, D \rangle$ where

- μ is a sample distribution over query pairs (x, y) sent by the verifier, and
- D is a decision function: the verifier accepts iff $D(x, y, a, b) = 1$.

So this is really a description of the verifier: $\mathcal{V}$ samples query pairs according to μ, and decides according to D.

For CHSH games we are dealing with single bits, more generally we consider bitstrings $x, y \in [n]$ and $a, b \in [k]$.

Key Question:
What is the best strategy for Alice and Bob in a non-local game $\mathfrak{G}$?

Write $\mathsf{val}^\star(\mathfrak{G})$ for the optimal success probability of Alice and Bob.
Unsurprisingly, a real definition of the optimal value involves quantum physics (or rather: math that is derived from the physics) and looks like this:

$$\mathsf{val}^\star(\mathfrak{G}) = \sup_{p \in C_{\mathrm{qs}}(n,k)} \sum_{x,y,a,b} \mu(x,y)\, D(x,y,a,b)\, p_{x,y,a,b}$$

$C_{\mathrm{qs}}(n,k)$ is the "quantum spatial correlation set." Don't ask.

At first glance, all we can say about $\mathsf{val}^\star(\mathfrak{G})$ is that it's a real number between 0 and 1. And presumably a fairly complicated number, nothing like $1/3$ or so.

Approximation?

If we want to anything computational, we need to make do with approximations: given some accuracy $\varepsilon > 0$, find r such that

$$|\mathsf{val}^\star(\mathfrak{G}) - r| < \varepsilon.$$

Given the formal definition, it is far from obvious how one might do this. In fact, a crucial parameter here is the dimension d, the number of entangled qubits Alice and Bob can use: the more, the merrier.

One standard approximation method is to compute a sequence of upper/lower bounds α_d/β_d for $\mathsf{val}^\star(\mathfrak{G})$ that converges to the actual value when $d \to \infty$.

More precisely, we can **compute**

- an increasing sequence (α_n) such that $\alpha_n \leq \mathsf{val}^\star(\mathfrak{G})$,
- a decreasing sequence (β_n) such that $\beta_n \geq \mathsf{val}^\star(\mathfrak{G})$.

The lower bounds are not too bad; one can prove that $\lim \alpha_n = \mathsf{val}^\star(\mathfrak{G})$.

Alas, the β sequence is much harder to define, and it is absolutely unclear that it converges (in fact, it does not in general).

Recall von Neumann's operator algebras? In the 1970s, Fields medalist Alain Connes handled type III and made a comment about type II_1.

The comment is now known as the Connes' Embedding Conjecture (CEC).

Ignoring all details, CEC claims that type II_1 yields to descriptions in terms of finite-dimensional matrices (recall, we are dealing with Hilbert spaces, this is all functional analysis).

Until recently, CEC was open, but it was known that there are equivalent conjectures in other areas:

- Kirchberg's conjecture (C^* algebras)
- Tsirelson's problem (quantum information theory)

Surprisingly, Connes' embedding conjecture implies that $\lim \beta_n = \mathsf{val}^\star(\mathfrak{G})$.

It follows that CEC implies that there is an approximation algorithm to compute $\mathsf{val}^\star(\mathfrak{G})$: for any $\varepsilon > 0$ we can effectively find n such that

$$\beta_n - \alpha_n < 2\varepsilon.$$

So we have

$$|\mathsf{val}^\star(\mathfrak{G}) - (\alpha_n + \beta_n)/2| < \varepsilon.$$

Alas, and rather surprisingly, we'll see that this just can't be true.

Theorem (12.10)

Given a Turing machine $\mathcal{M}$, one can in polynomial time construct a non-local game $\mathfrak{G}_{\mathcal{M}}$ such that

- *$\mathcal{M}$ halts implies* $\mathrm{val}^{\star}(\mathfrak{G}_{\mathcal{M}}) = 1$,
- *$\mathcal{M}$ diverges implies* $\mathrm{val}^{\star}(\mathfrak{G}_{\mathcal{M}}) \leq 1/2$.

The Turing machine $\mathcal{M}$ in the theorem is entirely unconstrained, absolutely anything goes. So it may take an absurd amount of time for the machine to halt.

The proof is rather complicated and rests on iterating the PCP theorem. Informally, the verifier checks that a more powerful verifier is correct, repeatedly. By the way, it also uses Kleene's recursion theorem.

RE $\subseteq$ MIP$*$ follows from the theorem: Halting is complete for RE and the game $\mathfrak{G}_{\mathcal{M}}$ can be computed from $\mathcal{M}$.

MIP$*$ $\subseteq$ RE is essentially the lower bound strategy explained above: for every instance x, one can in polynomial time compute a good verifier $\mathcal{V}_z$ and then lower-approximate val$^\star(\mathcal{V}_z)$. This produces a semidecision procedure.

This is a huge surprise: MIP$*$ is tantalizingly close to what one might consider a physically realizable form of computation. Thus, it is not unreasonable to expect that it should be too weak to handle undecidability. There is no counterpart to this in classical physics (though one can pick counter-realistic subtheories of physics where Halting is doable; Newtonian mechanics, for example).

This must not be misconstrued as meaning that we could use a quantum computer to solve the Halting problem by calculating $\mathrm{val}^\star(\mathfrak{G})$.

Also, there is the pesky question of how many entangled qubits a real quantum computer can handle. Recall, α_d is directly defined as the success probability of the best strategy using only d entangled qubits. Large numbers of entangled qubits may well turn out to be a real bottleneck in quantum computation.

And CEC?

Amazingly, the theorem also demolishes Connes' embedding conjecture, a claim in functional analysis that seems to have nothing to do with Turing machines, computability or complexity.

For if CEC were true, this would mean that we can solve the Halting Problem: we then have an approximation algorithm for $\mathrm{val}^\star(\mathfrak{G}_{\mathcal{M}})$. Just run that algorithm for a sufficiently small ε, say, $\varepsilon = 1/8$.

Again, this is utterly amazing, no one expected Connes' conjecture to be resolved via an excursion into complexity theory and computability.