Is P Versus NP Formally Independent?

Scott Aaronson∗
University of California, Berkeley

Abstract

This is a survey about the title question, written for people who (like the author) see logic as for-
bidding, esoteric, and remote from their usual concerns. Beginning with a crash course on Zermelo-
Fraenkel set theory, it discusses oracle independence; natural proofs; independence results of Razborov,
Raz, DeMillo-Lipton, Sazanov, and others; and obstacles to proving P vs. NP independent of strong
logical theories. It ends with some philosophical musings on when one should expect a mathematical
question to have a definite answer.

1 Introduction

The P vs. NP problem has been called “one of the most important problems in contemporary mathematics
and theoretical computer science” [60]. That is an understatement. Not only is P vs. NP the defining
question of our field; it’s one of the deepest questions ever asked for which we’d know how to recognize an
answer.1 (In other words, one of the deepest questions in NP.) If you doubt this, read the Clay Math
Institute’s list of million-dollar prize problems [12], and notice how P vs. NP stands out, not merely as the
only problem of the seven relevant practically, but as the only one pregnant philosophically. Does the ability
to recognize an answer to the other six questions—or to P vs. NP, or to any question—entail the ability to
find an answer? We are after not projective algebraic varieties or zeros of the Riemann zeta function, but
the nature of mathematical thought itself.

And therein lies the subject of this survey. If P vs. NP asks something so ineluctably metamathematical,
then should we be surprised if the tools of mathematics don’t suffice to answer it? In other words, could P
vs. NP be independent of the standard axiom systems, such as Zermelo-Fraenkel set theory? If P vs. NP
were independent, then there would be possibilities: either there would be no polynomial-time algorithm for
SAT, but it would be impossible to prove that in ZF; or else there would be such an algorithm, but it would
be impossible to prove that it works. Whether one judges this situation plausible seems related to one’s
view on another question: is P vs. NP “really” a question about logic, or about combinatorics?

Atitudes toward the last question have shifted over the past fifty years. In Gődel’s famous 1956 letter
to von Neumann (translated in [60]), where P vs. NP was first posed, Gődel apparently saw the problem as
a finitary analogue of the Hilbert Entscheidungsproblem:2

∗Email: aaronson@cs.berkeley.edu. Supported by an NSF Graduate Fellowship, by NSF ITR Grant CCR-0121555, and by
the Defense Advanced Research Projects Agency (DARPA).

1We are taking P vs. NP here as representative of a whole constellation of related questions, which are arguably just as
important: NP vs. P/poly, P vs. PSPACE, NP vs. coNP, BPP vs. BQP, the existence of public-key cryptography, and so on.
For simplicity, however, we focus on P vs. NP.

2The Entscheidungsproblem asks for a procedure that, given a mathematical statement, either finds a proof or tells us
It is evident that one can easily construct a Turing machine which, for each formula \( F \) of the predicate calculus and for every natural number \( n \), will allow one to decide if \( F \) has a proof of length \( n \). Let \( \Psi(F, n) \) be the number of steps that the machine requires for that and let \( \varphi(n) = \max_F \Psi(F, n) \). The question is, how fast does \( \varphi(n) \) grow for an optimal machine. One can show that \( \varphi(n) \geq Kn \). If there actually were a machine with \( \varphi(n) \sim Kn \) (or even only with \( \sim Kn^2 \)), this would have consequences of the greatest magnitude. That is to say, it would clearly indicate that, despite the unsolvability of the Entscheidungsproblem, the mental effort of the mathematician in the case of yes-or-no questions could be completely (footnote: apart from the postulation of axioms) replaced by machines. One would indeed have to simply select an \( n \) so large that, if the machine yields no result, there would then also be no reason to think further about the problem.

In the seventies, when serious work on \( P \) vs. \( NP \) began following the discovery of \( NP \)-completeness by Cook [14], Karp [30], and Levin [36], the problem continued to be seen in logical terms. People tried to separate \( P \) and \( NP \) using the same ideas that worked in recursion theory; this is the context of the Baker-Gill-Solovay [4] result that \( P \) vs. \( NP \) admits contradictory relativizations. In Section 3 we’ll see other logic-oriented work from this period, which explicitly exploited the possibility that \( P \neq NP \) might go the way of the Continuum Hypothesis—its truth forever unknowable, or dependent upon which axioms we choose.

In the eighties, the combinatorial view became dominant, in the wake of the circuit lower bounds of Furst-Saxe-Sipser [18], Ajtai [2], Razborov [50, 52], and Smolensky [61]. On this view, \( P \) vs. \( NP \) is no more likely to be formally independent than Goldbach’s Conjecture, the Riemann Hypothesis, or any other ‘natural’ mathematical problem. Proving \( P \neq NP \) (and indeed the stronger statement \( NP \nsubseteq P/poly \)) is really just a matter of rolling up our sleeves and analyzing increasingly powerful kinds of circuits.

The nineties saw a return to the logicism and brooding pessimism of the seventies. As discussed in Section 4, Razborov and Rudich [56] pointed out a basic limitation of the circuit lower bound techniques from the eighties: these techniques would (almost) argue against themselves, yielding efficient ways to solve the very problems they were supposed to prove intractable! Subsequently, as we’ll see in Section 4.1, Razborov [55] used this idea to prove, under cryptographic assumptions, that \( P \neq NP \) is independent of certain theories of ‘bounded arithmetic.’

I don’t know if it’s possible any longer to discern a unifying trend. Mulmuley and Sohoni [39, 57] have proposed to tackle \( P \) vs. \( NP \) using algebraic geometry—an approach that, they think, might escape the jaws of Razborov and Rudich. Using the deep result that \( \text{MIP} = \text{NEXP} \) [3], Buhrman, Fortnow, and Thierauf [8] showed that \( \text{MA}^{\text{EXP}} \), the exponential-time analogue of \( \text{MA} \), doesn’t have polynomial-size circuits. This result is notable because it avoids both the Baker-Gill-Solovay and the Razborov-Rudich limitations. So perhaps there’s hope after all.

In any event, however, it seems likely that future work on \( P \) vs. \( NP \) will be inseparable from work on the logical status of the question. Again and again in this survey, we’ll encounter the bizarre self-referential nature of \( P \neq NP \)—a conjecture that all but asserts the titanic difficulty of finding its own proof. For this reason, I think that anyone interested in complexity—even those who, like me, have no logic background whatsoever—ought to learn something about what it would mean for \( P \) vs. \( NP \) to be formally independent, and what can be said for and against that possibility. Hence this survey. Although it does contain the recommended daily allowance of theorems (for an average-weight theoretician), its central concern is that none exists. \( P = NP \) asks for an efficient procedure that finds a short proof. In one sense, however, \( P \) vs. \( NP \) is not a finitary analogue of the Entscheidungsproblem. For once we require all branches of a computation to halt eventually, the ‘nondeterministically recursive’ languages are just the recursive languages themselves, not the recursively enumerable languages. So the distinction between finding and verifying a bounded-size object is one that doesn’t exist in the recursion-theory world.
philosophical:

How could such a clear mathematical question—a question about finite machines and problems, a question of great importance for science and industry—be forever unanswerable?

Thus I’ll start, as I’ll end, by talking not about \( P \) vs. \( \text{NP} \), but about logic, provability, and truth.

2 Logic Primer

Have you ever lay awake at night, terrified of what would happen were the Zermelo-Fraenkel axioms found to be inconsistent the next morning? Would bridges collapse? Would every paper ever published in STOC and FOCS have to be withdrawn? (“The theorems are still true, but so are their negations.”) Merely contemplating such possibilities is enough to cause feelings of vertigo. How could a theorem be invalidated because of axioms that seem to have nothing to do with it? It would be as if you added up a restaurant bill, and then a physicist explained that your total was wrong because of the expansion of the universe.

This is the sort of issue philosophers like Wittgenstein talked about a lot. I won’t try to summarize what they said, for the same reason that Turing, after sitting in on Wittgenstein’s seminar, became exasperated and dropped out [38]. My concern is more practical: should we, who dwell in the tower of mathematics, worry about a collapse of the foundations?

In my opinion, no. If you don’t believe Euclid’s proof that there are infinitely many primes, then I don’t know why seeing a formal proof in ZF set theory should quell your doubts. You could deny Modus Ponens, you could demand a formal proof for the assertion “The infinitude of primes is a theorem of ZF” itself (ad infinitum), you could refuse to identify the formal notions of ‘prime’ or ‘infinity’ with the informal ones, and so on. Therefore the informal proof already yields as much certainty about the infinitude of primes as it’s humanly possible to have. If we try to place the proof on a “more secure foundation,” then the foundation is as open to doubt as the positive integers are.

If you accept this, I think it becomes easier to understand what a formal system is. It’s just a way to codify a form of reasoning—which presumably we’ve already accepted as valid—so that (1) we can program a computer to reason for us, and (2) we can explore the ultimate limitations of that form of reasoning.

2.1 The First-Order Sentences Game

As an example, consider the definition of a group. We have a ternary predicate \( M(x, y, z) \) (intuitively \( x \cdot y = z \)) that satisfies a set of three axioms, called a theory:

(1) **Uniqueness.** \( \forall x, y \exists z \ (M(x, y, z) \land \forall w \ (M(x, y, w) \implies (w = z))) \)

(2) **Associativity.** \( \forall x, y, z, w \ (\exists v \ M(x, y, v) \land M(v, z, w)) \implies (\exists v \ M(y, z, v) \land M(x, v, w)) \)

(3) **Identity and Inverse.** \( \exists x \forall y \ (M(x, y, y) \land M(y, x, y) \land \exists z \ (M(y, z, x) \land M(z, y, x))) \)

Then a model for the theory is just a set \( U \) of objects (called a universe), together with an assignment of ‘true’ or ‘false’ to \( M(x, y, z) \) for every \( x, y, z \) in \( U \), such that the axioms hold when the quantifiers range over \( U \). In general, an axiom can be any first-order sentence—meaning that it can have variable, constant, and predicate symbols; Boolean connectives (\( \land, \lor, \neg, \implies \), etc.); quantifiers (\( \forall, \exists \)); and equal signs (=). The equal sign is not shorthand for a binary predicate \( E(x, y) \); it really means that two objects are the same.
Crucially, the quantifiers can range only over objects in \( U \), not sets of objects. That’s what ‘first-order’ means.

How much can we say using these first-order sentences? To get some intuition, let’s ask how well we can control the cardinality of the universe \( U \). Groups can be finite, countable, or uncountable, so of course the axioms for a group have finite, countable, and uncountable models. If we want, though, we can force \( |U| \) to be at most 3:

- \( \exists x, y, z \forall w \ (w = x \lor w = y \lor w = z) \)

We can also force \( |U| \) to be infinite. Peano Arithmetic uses a binary predicate \( S(x, y) \) (intuitively \( y = x + 1 \)) that satisfies these three axioms:

1. **Zero.** \( \exists y \forall x \neg S(x, y) \)
2. **Unique Successor.** \( \forall x \exists y (S(x, y) \land y = x) \land \forall z (S(x, z) \implies (z = y)) \)
3. **Unique Predecessor.** \( \forall x, y, z (S(x, y) \land S(z, y)) \implies (x = z) \)

Can we force \( |U| \) to be uncountable? We might try to by writing down axioms that encompass all of mathematics. That sounds like a tall order, but as is well known, it’s enough to describe a universe whose objects are sets.\(^3\) What do these sets contain? Other sets! Zermelo-Fraenkel (ZF) set theory involves a binary predicate \( S(x, y) \) (intuitively \( x \in y \)), that satisfies the following axioms. (I’ll write the axioms in English, since it should be obvious by now how to convert them to first-order notation.)

1. **Empty Set.** There exists a set (denoted \( \emptyset \)) that does not contain any members.
2. **Extensionality.** If two sets contain the same members then they are equal.
3. **Pairing.** For all sets \( x \) and \( y \) there exists a set whose members are \( x \) and \( y \).
4. **Union.** For all sets \( x \) and \( y \) there exists a set (denoted \( x \cup y \)) that contains \( z \) if and only if \( z \in x \) or \( z \in y \).
5. **Infinity.** There exists a set \( x \) that contains \( \emptyset \) and that contains \( y \cup \{y\} \) for every \( y \in x \).
6. **Power Set.** For all sets \( x \) there exists a set (denoted \( 2^x \)) that contains \( y \) if and only if \( y \subseteq x \).
7. **Replacement for Predicate \( A \).** For all sets \( u \), if for all \( x \in u \) there exists a unique \( y \) such that \( A(x, y) \), then there exists a \( z \) such that for all \( x \in u \), there exists a \( y \in z \) such that \( A(x, y) \).
8. **Foundation.** All nonempty sets \( x \) contain a member \( y \) such that for all \( z \), either \( z \notin x \) or \( z \notin y \).

There are actually infinitely many axioms above, since we have an Axiom of Replacement for every binary predicate \( A(x, y) \). This doesn’t cause problems (though if we don’t like it, it turns out that there’s an equivalent, more complicated system with finitely many axioms, called Gödel-Bernays set theory).

So, do all models of ZF have uncountably many sets? One would think so—since after all, the Infinity and Power Set Axioms imply the existence of uncountable sets! To get a better handle on the question, though, let’s examine the inference rules of first-order logic, by which we prove theorems. We say a first-order sentence is valid if it can be obtained by the following rules:

---

\(^3\)How can we use axioms to define sets, if we need the concept of set even to say what we mean by a model for axioms? This is the kind of paradox that I think vanishes once we give up the idea of putting mathematics on a ‘more secure foundation,’ and see formal systems as just codifying forms of reasoning we’ve already accepted as valid.
(1) **Propositional Rule.** Any propositional tautology is valid.

(2) **Modus Ponens.** If $A$ and $A \Rightarrow B$ are valid then $B$ is valid.

(3) **Equality Rules.** The following are valid: (a) $x = x$, (b) $x = y \Rightarrow y = x$, (c) $x = y \land y = z \Rightarrow x = z$, (4) $x = y \Rightarrow (A(x) \Rightarrow A(y))$.

(4) **Change of Variables.** Changing variable names leaves a statement valid.

(5) **Quantifier Elimination.** If $\forall x A(x)$ is valid then $A(y)$ is valid.

(6) **Quantifier Addition.** If $A(y)$ is valid where $y$ is an unrestricted constant then $\forall x A(x)$ is valid.

(7) **Quantifier Rules.** The following are valid: (a) $\forall x A(x) \iff \exists x \neg A(x)$, (b) $(B \land \forall x A(x)) \iff \forall x (B \land A(x))$, (c) $(B \land \exists x A(x)) \iff \exists x (B \land A(x))$.

Obviously, if a set of axioms has a model, then applying the inference rules above can never lead to a contradiction. G"odel's Completeness Theorem says the converse: we didn't accidentally leave any rules out of the list, in the sense that if you can't get a contradiction by applying the rules, then the axiom set has a model. Equivalently, any sentence that's true in all models is provable. The Completeness Theorem is confusing for two reasons: on the one hand, it sounds like a tautology ("that which is consistent, is consistent")—what could it possibly mean to prove such a thing? And on the other hand, it seems to contradict the Incompleteness Theorem.

We're going to clear up this mess, and as a bonus, answer our question about whether all models of ZF are uncountable. The best way to understand the Completeness Theorem is to make up a consistent axiom set that you'd guess doesn't have a model. Given a theory $T$, let $\text{Con}(T)$ be the assertion that $T$ is consistent. We know from G"odel's Incompleteness Theorem that $\text{Con}(ZF)$ can be expressed in ZF, and also that $\text{Con}(ZF)$ can't be proved in ZF, assuming ZF is consistent.\(^4\) It follows that assuming ZF is consistent, the "self-hating theory" $ZF + \neg \text{Con}(ZF)$, or ZF plus the assertion of its own inconsistency, must also be consistent. So by the Completeness Theorem, $ZF + \neg \text{Con}(ZF)$ has a model. What on earth could it be? We'll answer this question via a fictional dialogue between you and the axioms of $ZF + \neg \text{Con}(ZF)$.

**You:** Look, you say ZF is inconsistent, from which it follows that there's a proof in ZF that $1 + 1 = 3$. May I see that proof?

**Axioms:** Of course! The integer encoded by a set in the universe that I describe.

**You:** But what is it, as an ordinary integer?

**Axioms:** No, no, no! Talk to the axioms.

**You:** Alright, let me ask you about $X$. Is it greater or smaller than a billion?

**Axioms:** Greater.

**You:** The $10^{10^{10000000000}}$th Ackermann number?

**Axioms:** Greater than that too.

---

\(^4\)For suppose $\text{Con}(ZF)$ could be proved in ZF. It’s easy to prove in ZF that $\text{Con}(ZF)$ implies $G(ZF)$, where $G(ZF)$ is the famous G"odel sentence that says “I am not provable in ZF.” But then we would have a proof in ZF of $G(ZF)$, so ZF would be inconsistent. (Indeed, $\text{Con}(ZF)$ and $G(ZF)$ are easily seen to be equivalent.)
You: What’s $X^2 + 100$?
Axioms: Hmm, let me see… Y.
You: Why can’t I just add an axiom to rule out these weird ‘nonstandard integers?’ Let me try: for all integers $X$, $X$ belongs to the set obtained by starting from 0 and...
Axioms: Ha ha! This is first-order logic. You’re not allowed to talk about sets of objects—even if the objects are themselves sets.
You: Argh! I know you’re lying about this proof that $1 + 1 = 3$, but I’ll never catch you.
Axioms: That right! What Gödel showed is that we can keep playing this game forever. What’s more, the infinite sequence of bizarre entities you’d force me to make up—$X$, $Y$, and so on—would then constitute a model for the preposterous theory $\text{ZF} + \neg\text{Con (ZF)}$.
You: But how do you know I’ll never trap you in an inconsistency?
Axioms: Because if you did, the Completeness Theorem says that we could convert that into an inconsistency in the original axioms, which contradicts the obvious fact that $\text{ZF}$ is consis—no, wait! I’m not supposed to know that! Aaahh! (The axioms melt in a puddle of inconsistency.)

As a corollary of Gödel’s Completeness Theorem, we get the Löwenheim-Skolem Theorem, which actually predates Gödel: if a theory $T$ has a model, then it has a model of at most countable cardinality. Why? Because the game above—where we keep challenging $T$ to name the objects it says exist, and $T$ responds by ‘cooking new objects to order’—lasts at most countably many steps, since each ‘challenge’ can be expressed as a finite string. And the Completeness Theorem guarantees that the final result will be a model for $T$, assuming $T$ was consistent.

In fact, it’s known that any theory that has arbitrarily large finite models has an infinite model, and any theory that has an infinite model has models of whatever infinite cardinality we want. This is already a tip-off that the first-order sentences game can’t answer any question we might ask. It can’t even tell us how many objects are in the universe!

### 2.2 Proving Independence

Moving closer to the subject of the survey, how does one prove a statement independent of a first-order theory? There are actually two methods: consistency strength, and relative consistency.

Let’s start with consistency strength. If we remove the Axiom of Infinity from $\text{ZF}$, we get a theory equivalent to Peano Arithmetic (PA). Now, it’s not hard to see that $\text{ZF} \vdash \text{Con (PA)}$; that is, $\text{Con (PA)}$ is a theorem of $\text{ZF}$. The reason is that in $\text{ZF}$, there exist infinite sets—for example, the set of all finite sets—that we can take as models for PA. On the other hand, Gödel tells us that $\text{ZF} \not\vdash \text{Con (ZF)}$. If we want to prove $\text{Con (ZF)}$, we have to postulate an infinity bigger than anything definable in $\text{ZF}$—a so-called ‘large cardinal.’ (When set theorists say large, they mean large.) If LC asserts the existence of such a cardinal, then $\text{ZF + LC} \vdash \text{Con (ZF)}$. So we can rank the theories in a ‘consistency strength hierarchy,’ from PA to ZF to ZF + LC. Notice that in PA, we can’t even prove that $\text{Con (PA)} \implies \text{Con (ZF)}$. For then we could also prove that in ZF, and since $\text{ZF} \models \text{Con (PA)}$, we’d have $\text{ZF} \models \text{Con (ZF)}$, contradiction.

In general, to show that a statement is unprovable in some theory, we argue that if it wasn’t, then we could collapse two levels of the consistency strength hierarchy—and thereby violate the Incompleteness Theorem. As another application, you might wonder: suppose $P \neq \text{NP}$ were unprovable in a formal system such as $\text{ZF}$. Maybe we could still prove $P \neq \text{NP}$ by assuming from the outset (that is, adding as an axiom) that $\text{ZF}$ does prove $P \neq \text{NP}$? A result called Löb’s Theorem uses consistency strength to rule out this strange possibility. (I prefer to call it the “You-Had-The-Mojo-All-Along Theorem.”)

**Theorem 1 (Löb)** If sentence $A$ can be proven in $\text{ZF} + (\text{ZF} \vdash A)$, then it can also be proven in $\text{ZF}$ alone.
Proof. If ZF is inconsistent then it can prove anything, so assume Con(ZF). Suppose ZF proves (ZF ⊨ A) ⇒ A. Then ZF + ¬A ⊨ (ZF ⊬ A). Suppose also that we need the axiom (ZF ⊨ A) to prove A—that is, that ZF ⊬ A. Then since ZF is consistent, ZF + ¬A is also consistent. But ZF + ¬A proves ZF ⊬ A, which is equivalent to Con(ZF + ¬A). So ZF + ¬A ⊨ Con(ZF + ¬A), which contradicts the consistency of ZF + ¬A. ■

The other method for proving independence is relative consistency. As Shelah [59] put it, consistency strength makes the universe of logical theories ‘taller,’ while relative consistency makes it ‘fatter.’ Relative consistency was what Gödel [20] and Cohen [13] used to prove the independence of the Axiom of Choice (AC) and Continuum Hypothesis (CH) from ZF. Recall that AC is the assertion that, given a set x of nonempty, pairwise disjoint sets, there exists a set that shares exactly one element with each set in x. CH is the assertion that there’s no set of intermediate cardinality between the integers and the sets of integers. Then Gödel proved that

\[ \text{Con}(ZF) \Rightarrow (\text{Con}(ZF + AC) \land \text{Con}(ZF + CH)) , \]

while Cohen proved that

\[ \text{Con}(ZF) \Rightarrow (\text{Con}(ZF + \neg AC) \land \text{Con}(ZF + \neg CH)) . ^5 \]

In other words, by starting with a model for ZF, we get another model for ZF with specific properties we want—for example, that CH is true, or that AC is false. For this reason, it’s clear that Con(ZF) is not a theorem of, say, ZF + CH—for if it were, then we’d have

\[ ZF + CH \vdash \text{Con}(ZF + CH) , \]

therefore ZF + CH would be inconsistent, therefore ZF itself would be inconsistent. So unlike with the method of consistency strength, adding CH doesn’t lead to a ‘stronger’ theory—just a different one.

In summary, then, if we wanted to prove that \( P \neq NP \) (or \( P = NP \)) is unprovable in some theory, there are two ways we might go about it: consistency strength or relative consistency.

3 Early Results

With the logic background out of the way, it’s finally time to say something about \( P \) vs. \( NP \). We’ll start with some results from the seventies and eighties: in Section 3.1, oracle independence theorems due to Hartmanis and Hopcroft [26] and Hartmanis [25]; and in Section 3.2, non-oracle independence theorems (concerning weak logical theories) due to DeMillo and Lipton [15] and Sazanov [58].

3.1 Oracles

Given a Turing machine \( M \), let \( L (M) \) be the language accepted by \( M \). The following was shown in [24, 26].

**Theorem 2 (Hartmanis-Hopcroft)** There exists a Turing machine \( M \) that halts on every input, such that relative to the oracle \( L (M) \), neither \( P = NP \) nor \( P \neq NP \) is provable in ZF, assuming ZF is consistent.

^5 Amusingly, both of these implications are actually theorems of PA—since to prove them, one ultimately just talks about axioms, which are finite strings of symbols.
The language \( L(M) \) will turn out to be the empty set (that is, \( M \) always rejects). The catch now is that the function \( f \) isn’t known to be independent of ZF. So you might wonder: can we construct a computable oracle \( O \)?

Skipping technicalities, the intuition is as follows. We construct \( O \) that \( L(M) = \{ \} \), beyond some finite point, the oracle \( L(M) \) equals \( B \), and thus \( P^{L(M)} \neq NP^{L(M)} \). Similarly, if there exists a ZF proof that \( P^{L(M)} = NP^{L(M)} \), then beyond some finite point \( L(M) \) equals \( A \), and thus \( P^{L(M)} = NP^{L(M)} \). We conclude that assuming ZF is consistent, there is no ZF proof of either statement.

The Hartmanis-Hopcroft theorem is called ‘representation-dependent,’ since the independence really relies on the Turing machine \( M \) by which we’re told about the oracle, not the oracle itself. If someone told us that \( L(M) \) was empty, then we’d be back to the standard (unrelativized) \( P \) vs. \( NP \) question, which of course isn’t known to be independent of ZF. So you might wonder: can we construct a computable oracle \( O \), such that \( P^O \) vs. \( NP^O \) is independent of ZF, no matter which Turing machine is used to specify \( O \)?

The answer turns out to be yes, as shown by Hartmanis [25] and also by Kurtz, O’Donnell, and Royer [33]. Skipping technicalities, the intuition is as follows. We construct \( O \) so that for almost all input lengths, \( O \) collapses \( P \) and \( NP \). But for a few, widely-separated input lengths, call them \( f(1), f(2), f(3), \ldots \), \( O \) separates \( P \) and \( NP \), and that is enough to guarantee that \( P \neq NP \) relative to \( O \), since there are infinitely many such lengths. The catch now is that the function \( f \), though computable, grows so quickly that one can’t prove in ZF that \( f \) is total, or even that \( f \) is defined for infinitely many values of \( n \). (We’ll have more to say about such fast-growing computable functions in Section 5.) If \( f \) were defined for only finitely many \( n \), then \( O \) would differ only finitely from an oracle that collapses \( P \) and \( NP \), so of course we’d have \( P^O = NP^O \). That’s what makes \( P^O \) vs. \( NP^O \) independent of ZF.

### 3.2 Weak Independence

If we want to study the provability of \( P \) vs. \( NP \), a natural idea is to show the problem independent of extremely weak logical theories. If nothing else, that would at least rule out any approach to \( P \) vs. \( NP \) that can be formalized within those theories. Already in the seventies and early eighties, there was a good deal of work in this direction [15, 28, 58]; we’ll look at two examples.

DeMillo and Lipton [15] show \( P \neq NP \) unprovable in a fragment of number theory they call ET. The objects of ET are integers, while the language consists of the functions \( x+y, x-y, x \cdot y, \min \{x, y \}, \max \{x, y \} \), and \( c^x \) (where \( c \) is a constant); as well as all polynomial-time computable predicates (but not functions).

---

6 Basically, \( A \) is just a PSPACE-complete language, which collapses both \( P \) and \( NP \) to PSPACE; while \( B \) is an infinite sequence of hard search problems, constructed by diagonalizing over all polynomial-time Turing machines.
For example, given integers $x, y, z$, we can write a predicate that tests whether $2^x + 3^y = 5^z$. We can do this even though $|2^x| + |3^y| + |5^z|$ is exponentially larger than $|x| + |y| + |z|$—since it’s only the predicates (in this case equality testing), not the arithmetic operations, that need to be polynomial-time. Finally, the axioms of ET are all true sentences of the form $\forall x \ A(x)$, where $A$ is a quantifier-free predicate. (Here ‘true’ means true for the ordinary integers.) In one sense the axioms are absurdly powerful—among other things, they give us Fermat’s Last Theorem for free (for each fixed value of $n$)—but in another sense the axioms are weak, since they have only one universal quantifier. The proof that $P \neq NP$ is unprovable in ET would take us too far afield. I’ll just say that a key part is to bound the possible growth rates of functions expressible with the allowed arithmetic operations.

Sazanov [58] shows exponential lower bounds to be unprovable in a quite different theory, by using the method of consistency strength discussed in Section 2.2. He defines a theory $T$ in which the objects are finite binary strings, and we have available all polynomial-time computable functions and predicates. Call a sentence ‘true’ if it’s true in the standard model. Then the axioms consist of all true first-order sentences, with all quantifiers bounded (except for an initial universal quantifier). A bounded quantifier has the form $\forall x \leq a$ or $\exists x \leq a$, where $a$ doesn’t depend on $x$ and $\leq$ denotes lexicographic ordering of strings. This is a richer axiom set than that considered by DeMillo and Lipton; on the other hand, the arithmetic operations—and in particular the exponential function—aren’t available.

Given strings $x$ and $y$, we can define a predicate $\text{EXP}(x, y)$, that tests whether $2^{|x|} \leq |y|$ (where $|x|$ is the length of $x$). However, we don’t have an axiom $E$ saying that $\forall x \exists y \ \text{EXP}(x, y)$. In other words, if our quantifiers range over strings of length $n$, then strings of length $2^n$ are outside our “universe of discourse.” Indeed Sazanov shows that $T + E \equiv \text{Con}(T)$. Intuitively, once we have an exponentially-long string $y$, we can construct the set of all polynomial-size strings, and thereby obtain a model for $T$. This is analogous to the fact that $ZF \equiv \text{Con}(PA)$—the only difference being that in Sazanov’s case, the Axiom of Infinity is scaled down to the “Axiom of Exponentiality” $E$. Continuing the analogy, we can conclude that $T \nvdash E$—since otherwise we’d have $T \equiv \text{Con}(T)$, contradicting the Incompleteness Theorem. Indeed, letting $\text{ACCEPT}$ be the assertion that there exists an algorithm for SAT, Sazanov shows further that $T + \text{ACCEPT} \nvdash E$. Now let $\text{ELB}$ (Exponential Lower Bound) be the assertion that any algorithm for SAT requires exponential time. Then

$$T + \text{ACCEPT} + \text{ELB} \equiv E.$$  

For $\text{ACCEPT}$ tells us that for all SAT instances $x$, there exists a $y$ that is the tableau of a computation of $\text{SAT}(x)$ by some fixed algorithm. Meanwhile $\text{ELB}$ tells us that $|y|$ is exponentially larger than $|x|$ infinitely often. We conclude that $T \nvdash \text{ELB}$; that is, $T$ is unable to prove exponential lower bounds for SAT. It’s interesting that this proof rules out only lower bounds of the form $2^{|x|}$ for $\varepsilon > 0$, since we need an exponentially long string to get a model for $T$. For all we know, $T$ does prove that SAT requires $n^{\log n}$ time.

All in all, we depart these early results feeling we learned a bit about logic, independence, and relativization, but less sure that we learned anything about $P$ vs. $NP$.\footnote{As a further result, DeMillo and Lipton show that assuming $P \neq NP$, the statement $NP \neq \text{coNP}$ is consistent with another fragment of number theory called PT. This represents a direction on which there hasn’t been much work to my knowledge: providing “evidence” for widely-believed complexity assumptions (such as $NP \neq \text{coNP}$, or $P \neq NP$), by showing that the assumptions are consistent with weak logical theories (or equivalently, that their negations are unprovable in those theories).}
4 Natural Proofs

In 1993, Razborov and Rudich [56] wrote a paper that contains, arguably, the most important insight anyone has thus far had into the \( P = NP \) question. Loosely speaking, they said that the reason \( P = NP \) is so difficult to prove is that \( P = NP \)!

I’ll first review their argument; then, in Section 4.1, I’ll discuss how it can be used to show the unprovability of circuit lower bounds in certain logical theories.

As mentioned in Section 1, in the 1980’s people developed many ‘combinatorial’ techniques for proving circuit lower bounds [2, 18, 52, 61]. These techniques showed, for example, that the PARITY function requires constant-depth circuits of exponential size, provided the only gates allowed are \( \lor, \land, \neg \) (with unbounded fanin). What Razborov and Rudich noticed was that these techniques all follow the same basic strategy. To show that a Boolean function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) (say, PARITY) is not in complexity class \( C \), we do the following.

1. Invent a ‘complexity measure’ \( \mu \) for Boolean functions. (In [61], for example, \( \mu \) is the minimum degree of an approximating polynomial over a finite field.)

2. Show that for any function \( f^* \in C \), \( \mu(f^*) \) is small. (Say, because each gate in a circuit can only increase \( \mu \) by a small amount.)

3. Show that \( \mu(f) \) is large—from which it follows that \( f \notin C \).

The point is that, if the strategy above works, then it should work not only for the function \( f \) we’re interested in, but for many other functions as well. For under any ‘reasonable’ measure of complexity, a random function is extremely complex. Intuitively, if ‘simple’ means having a short description of some form, then most functions can’t be simple just by a counting argument! Indeed, this is how one shows that almost all Boolean functions have exponential circuit complexity, and that almost all strings have large Kolmogorov complexity. Furthermore, assuming we could analyze \( \mu(f) \) well enough to carry out step (3), we presumably have an efficient procedure to estimate \( \mu \)—where ‘efficient’ means using time polynomial in \( 2^n \), the size of the truth-table of \( f \).

Assuming our proof is ‘natural’ in the above senses, Razborov and Rudich show that we could turn the proof on its head, and use it to break pseudorandom functions computable by the class \( C \). The idea is simple: if \( f^* \in C \), then \( \mu(f^*) \) must be small, while if \( f \) is a genuinely random function, then \( \mu(f) \) is large with non-negligible probability. A subtlety is that the procedure to estimate \( \mu \) took time polynomial in \( 2^n \), not in \( n \). To deal with this, we choose the seed for the pseudorandom function \( f^* \in C \) to have length \( n^c \), where \( c \) is a large constant. Then distinguishing \( f^* \) from random by testing every seed exhaustively would take time roughly \( 2^{n^c} \). We succeeded instead in time \( 2^{O(n^\varepsilon)} \), or \( 2^{N^{1/c}} \) in terms of the seed length \( N = n^c \).

Why is this considered implausible? In the case \( C = P/poly \) (that is, the lower bound proof works against polynomial-size circuits), the evidence comes cryptographic reductions. Håstad et al. [27] showed that given any one-way function, we can construct a pseudorandom generator that’s roughly as hard to break. Earlier, Goldreich, Goldwasser, and Micali [21] had shown that given any pseudorandom generator, we can construct a pseudorandom function that’s roughly as hard to break. Putting these together, we find that our lower bound proof would give us a way to break any one-way function (including factoring, discrete logarithm, etc.), in time \( 2^{O(n^\varepsilon)} \) for any \( \varepsilon > 0 \). By comparison, the best known factoring algorithm is conjectured to run in time roughly \( 2^{n^{1/3}} \) [35].

\*\*\*Of course, assuming \( P = NP \), one could cite that as the reason why anything is hard to prove! But Razborov and Rudich connected the hardness of \( NP \) problems to the hardness of proving their hardness in a more interesting way.\*\*\*
If C is a smaller class, for instance the class TC^0 of constant-depth polynomial-size threshold circuits, then the reductions of [27] and [21] can’t be carried out. However, Naor and Reingold [40] gave a direct construction of a pseudorandom function in TC^0, which is provably as hard as factoring and discrete logarithm. It follows that any natural proof that a function isn’t in TC^0 would yield an algorithm for these problems much faster than any known today.

A final observation is that the stronger the lower bound we obtain for f, the stronger the upper bound it implies for breaking pseudorandom functions. For example, if our lower bound puts f outside of C = \text{DTIME}(t(n)) (where t is superpolynomial), then we can take the seed for the pseudorandom function to have length about t(n). So in terms of the seed length, we get a roughly 2^{t−1(\log n)} upper bound on the time needed to distinguish the function from random. Razborov and Rudich credit Wigderson with the amusing observation that no natural proof can show a better than half-exponential\(^9\) lower bound for the discrete logarithm function, \textit{unconditionally}. For as we show this function to be harder, we simultaneously show it to be easier!

\section{4.1 Natural Proofs and Independence}

In Section 3.2, we saw what might be called the ‘first generation’ of P vs. NP independence results. These sought to show P \neq NP unprovable in weak logical theories, by (say) analyzing the possible growth rates of functions provably total in those theories. Razborov and Rudich opened the way to a ‘second generation’ of independence results. These focus on a finite statement, such as

\[ \forall \text{Circuit}_n := \forall \text{SAT}_n \text{ requires circuits of size } n^{\log n} \]

where SAT\(_n\) is the set of SAT instances of size \(n\). The goal is to show (perhaps under a complexity assumption) that in some proof system, \forall \text{Circuit}_n requires proofs of length superpolynomial in n. That is, we want to argue against the plausibility of proving lower bounds by proving a lower bound.

The difference between the two generations of independence result highlights an important fact about P vs. NP itself. When we meet a statement—Goldbach’s Conjecture, for example—that says something about every natural number \(n\), we usually assume the statement is trivial for any fixed \(n\). So the sole problem is to find an argument that works for all \(n\). For \forall \text{Circuit}_n, though, we have no idea how to prove even \forall \text{Circuit}_{50} or \forall \text{Circuit}_{100}. The resources needed to prove \forall \text{Circuit}_n grow too quickly as a function of \(n\). Both natural proofs and the second-generation independence results focus exclusively on this latter ‘meta-complexity’ problem—bypassing the issue of how we’d generalize to all \(n\), even if we \textit{could} prove \forall \text{Circuit}_{50}, \forall \text{Circuit}_{100}, and so on.\(^{10}\)

To do justice to the second-generation independence results would require another survey altogether, and indeed such surveys exist [10, 46, 49]. Here I’ll just sketch the main ideas. Many of the results involve a set of theories called \textit{bounded arithmetic} [9, 10, 11]. In these theories, the objects are natural numbers, and we have available the constants 0 and 1 and the functions +, ×, ≤, \lfloor x/2 \rfloor, |x|, and \#, where |x| = \lfloor \log_2 (x + 1) \rfloor.

\(^{9}\)For \(t(n)\) to be half-exponential means that \(t(t(n))\) grows like \(2^n\). Such functions exist, but are difficult to describe (see Miltersen, Vinodchandran, and Watanabe [37]).

\(^{10}\)So the results really talk about \textit{proof-sequences} \(\{P_n\}_{n \geq 1}\), only indirectly about proofs. The notion of a proof-sequence helps to resolve the following puzzle. Suppose any pseudorandom function family in TC^0 could be broken by a quantum computer (not so implausible, given that the only candidate family, due to Naor and Reingold [40], is based on factoring and discrete logarithm). Might we then have a “quantum natural proof” that TC^0 \neq P? The answer is that at best, we’d get a quantum Turing machine that on input n, verifies in time \(2^{O(n)}\) (say) that CVF\(_n\), the Circuit Value Problem of size n, requires TC^0 circuits of size \(n^{\log n}\). For a natural proof of TC^0 \neq P, we’d still need a \textit{classical} algorithm to break pseudorandom functions in TC^0.
and \( x \# y = 2^{(x|y|)} \). We can use the bounded quantifiers \( \forall x \leq t \) and \( \exists x \leq t \), where \( t \) doesn’t depend on \( x \). There are standard axioms defining the functions, as well as two types of induction axiom:

\[
\text{IND : } [A(0) \land \forall x (A(x) \implies A(x+1))] \implies \forall x A(x)
\]
\[
\text{PIND : } [A(0) \land \forall x (A(\lfloor x/2 \rfloor) \implies A(x))] \implies \forall x A(x)
\]

Here \( A \) is a sentence that may contain additional variables. Clearly the IND axioms imply the PIND ones, but the converse is believed to be false. You can think of PIND as ‘lazy induction’: it forces us, the theorem provers, to do the hard work of showing that \( A(\lfloor x/2 \rfloor) \implies A(x) \) for all \( x \), so that it can get by with an exponentially ‘shallower’ induction than IND.

Now, a \( \Sigma_1 \)-sentence is any sentence of the form \( \exists x P(x) \), where the quantifier \( \exists x \) is bounded. Likewise, a \( \Sigma_2 \)-sentence has the form \( \exists x \forall y P(x,y) \) where \( \exists x \) and \( \forall y \) are bounded, and so on, in analogy to the polynomial hierarchy \( \text{PH} \). Then the theory \( T_0 \) is bounded arithmetic with an IND axiom for every \( \Sigma_1 \)-sentence \( A \), and \( S_1 \) is bounded arithmetic with a PIND axiom for every \( \Sigma_1 \)-sentence \( A \). The theories can be ordered as follows from less to more powerful:

\[
S_1 \subseteq T_1 \subseteq S_2 \subseteq T_2 \subseteq \cdots
\]

A major open problem is whether the above inclusions are proper.

The bounded arithmetic theories are weaker than Sazanov’s theory \( T \), which we considered in Section 3.2. However, the independence results that can be shown are stronger: they rule out all superpolynomial lower bounds, not just exponential lower bounds as in Sazanov’s case. Moreover, as discussed earlier, for each particular \( n \) they give a lower bound on the length of a proof of \( \text{Circuit}_n \).

Razborov [55] showed that if there exists a pseudorandom generator that requires circuits of size \( \Omega \left( 2^{n^\varepsilon} \right) \) to break for some \( \varepsilon > 0 \) (the same assumption used for natural proofs), then \( \text{NP} \not\subseteq \text{P/poly} \) is unprovable in the theory \( S_1 \). Indeed, no class of functions definable in bounded arithmetic can then be proven to lie outside \( \text{P/poly} \), so in particular, \( \Sigma_k^p \not\subseteq \text{P/poly} \) is unprovable for all \( k \). Also, Razborov showed that \( \text{NP} \not\subseteq \text{P/poly} \) is unprovable in \( S_1 \), under the weaker assumption that there exists a pseudorandom generator computable in polynomial time that requires bounded fan-in circuits of depth \( \Omega(n^\varepsilon) \) to break for some \( \varepsilon > 0 \).

The key to the above results is what’s called the Efficient Interpolation Property (EIP). Suppose we have a conjunction \( \varphi = A(x,z) \land B(y,z) \), where \( x,y,z \) are sets of variables (\( z \) being a set of ‘shared’ variables). If \( \varphi \) is unsatisfiable, then for every fixed choice of \( z \), either \( A(x,z) \) or \( B(y,z) \) must be unsatisfiable. Given \( z \) as input, the interpolation problem is to return which of \( A \) or \( B \) is unsatisfiable—or if both are unsatisfiable, then either \( A \) or \( B \). We say a proof system \( S \) satisfies the EIP, if whenever \( \varphi \) has a polynomial-size refutation in \( S \), there’s a polynomial-time algorithm that solves the interpolation problem.

What Razborov did was to create a \( \varphi \) for which solving the interpolation problem is as hard as breaking a pseudorandom function. Let \( f \) be a pseudorandom function mapping \( n^c \) bits to \( 2^n \) bits, and let \( s \) be any explicit function such that

\[
\text{Range}(f) \cap (\text{Range}(f) \oplus s) = \emptyset.
\]

Then for \( z \in \{0,1\}^{2^n} \), define \( \varphi = A(z) \land B(z) \) where (informally, and suppressing variables besides \( z \))

\[
A(z) \iff z \text{ belongs to } \text{Range}(f),
\]
\[
B(z) \iff (z \oplus s) \text{ belongs to } \text{Range}(f).
\]

Clearly \( \varphi \) is unsatisfiable. On the other hand, if we could decide which of \( A \) or \( B \) is unsatisfiable (that is, solve the interpolation problem), then we could distinguish \( f \) from a random function in time \( 2^{O(n)} \). It follows that in any proof system \( S \) satisfying the EIP, \( \varphi \) can’t have a polynomial-size refutation assuming there exist
2^{n^2}$-secure pseudorandom generators. From there it’s easy to show that the tautologies $\{\neg\text{Circuit}_n\}_{n \geq 1}$ (assuming they are tautologies!) don’t have polynomial-size proofs in $\mathcal{S}$. As an example, Pudlák [45] showed that the Cutting Planes proof system satisfies the EIP, and this immediately implies that Cutting Planes can’t prove $\text{NP} \not\subseteq \text{P/poly}$, again under the pseudorandomness assumption.

So EIP is a useful property, since it immediately gives us conditional lower bounds on a proof system’s ability to prove circuit lower bounds. Thus, many researchers hoped to show the EIP for stronger proof systems, especially the so-called Extended Frege ($\mathcal{EF}$) system. Unfortunately, Krajíček and Pudlák [31] showed that the EIP fails for this system—ironically, under a cryptographic assumption like those used to prove lower bounds on proof size! The idea is simple: they created a disjunction $\varphi = A(z) \land B(z)$ where

\[
A(z) \iff z \text{ is an RSA encryption of a 0 bit,} \\
B(z) \iff z \text{ is an RSA encryption of a 1 bit.}
\]

Clearly $\varphi$ is unsatisfiable. Moreover, using some number-theoretic facts, Krajíček and Pudlák showed that this unsatisfiability can be proven in the theories $T^2_2$ and $S^1_2 + \Phi$, where $\Phi$ is the axiom that any integer having a Pratt certificate\footnote{Pratt [44] showed that $\text{PRIMES} \in \text{NP}$; that is, every prime has a succinct proof of primality. An obvious question is whether the deterministic primality test of Agrawal, Kayal, and Saxena [1] might eliminate the need for $\Phi$. This seems unlikely, since one would still need to prove the correctness of the AKS algorithm in $S^1_2$.} is prime. This implies in particular that $\varphi$ has a polynomial-size refutation in $\mathcal{EF}$. It follows that, if $\mathcal{EF}$ satisfied the EIP, then there would be a polynomial time algorithm to solve the interpolation problem and thereby break RSA. We conclude that $\mathcal{EF}$ doesn’t satisfy the EIP, assuming RSA is secure.

I’ll conclude with one example of an unconditional ‘second-generation’ independence result. In the well-known resolution proof system, we’re given a set of CNF clauses that we want to prove unsatisfiable. The allowed operation is to “smash together” two previous clauses, of which one contains a literal $a$ and the other contains $\neg a$, to derive a new clause. For example, given the clauses $(a \lor \neg b)$ and $(\neg a \lor c \lor d)$ we can derive $(b \lor c \lor d)$. The length of a resolution proof is the number of clauses it derives prior to the empty clause. Haken [23] showed the first superpolynomial lower bound on the lengths of resolution refutations. The unsatisfiable formulas he used were based on the Pigeonhole Principle; they encoded that $n + 1$ pigeons are each in one of $n$ holes, with no two pigeons in the same hole. Intuitively, with resolution we’re stuck reasoning locally: “let’s see, if we put this pigeon here, and moved that one there... darn, still doesn’t work!” Resolution isn’t powerful enough to count the pigeons and the holes, and see that the former are more numerous than the latter.

Recently Raz [47] improved Haken’s result to show that even formulas based on the so-called ‘Weak Onto Pigeonhole Principle’—that $m \gg n$ pigeons can’t be assigned to $n$ holes, with exactly one pigeon in every hole—require resolution refutations of size $2^{n^\varepsilon}$ for some $\varepsilon > 0$. By a reduction originally due to Razborov [51], a corollary is that the set of formulas $\{\text{Circuit}_n\}_{n \geq 1}$ (under a natural propositional encoding), encoding that SAT has small circuits, don’t have polynomial-size resolution refutations. Intuitively, consider a small DNF formula where each clause contains every variable (either negated or non-negated). Though it seems obvious that such a formula can’t compute SAT, one does need to show that there are more ‘yes’ instances of SAT than there are clauses in the DNF—and that requires the Weak Onto Pigeonhole Principle!

Improving on Raz’s result, Razborov has shown [54] that $\neg\text{Circuit}_n$ has no succinct resolution proofs even when the circuits in the encoding have bounded fan-in. Using a different reduction—based on the Nisan-Wigderson pseudorandom generator [41], rather than the pigeonhole principle—Razborov has also shown [53]
that \( \forall \text{Circuit}_n \) has no succinct proofs in several extensions of resolution, including PCR (Polynomial Calculus and Resolution) and \( k \)-DNF resolution, or resolution with \( k \)-DNF’s instead of clauses, for \( k = \varepsilon \log n \).

Taking a step back, do these results tell us anything about \( P \) vs. \( \text{NP} \)—for example, whether the problem is likely to be independent of ZF or some other strong theory? In the appendix of [48], Razborov argues that all known “explicit” techniques for circuit lower bounds can be formalized in \( S^2_2 \). Therefore, the inability even of \( S^2_2 \) to prove \( \text{NP} \not\subseteq P/\text{poly} \) under a pseudorandomness assumption could be taken as strong evidence that those techniques are logically too weak to prove \( P \neq \text{NP} \).

On the other hand, a telling point, mentioned in [48], is that \( S^2_1 \) can’t even formalize the Shannon counting argument, which shows that almost all Boolean functions require exponential-size circuits. As a consequence, \( S^2_1 \) can’t formalize Kannan’s result [29] that \( \text{NP}^{\text{exp}} \not\subseteq P/\text{poly} \), which uses the Shannon counting argument; or the Buhrman-Fortnow-Thierauf result [8] that \( \text{MA}^{\text{exp}} \not\subseteq P/\text{poly} \), which in turn uses Kannan’s result. It’s clear, then, that there exist techniques not formalizable in \( S^2_1 \), that to us humans are almost the epitome of triviality (unlike many techniques that are formalizable in \( S^2_2 \)). Whether a theorem can be proven in a weak fragment of arithmetic might not say much about the theorem’s conceptual difficulty. In one of the many ironies of \( P \) vs. \( \text{NP} \), it’s conceivable that the independence results will ultimately contribute to proving \( P \neq \text{NP} \), not because of what they tell us, but because of the lower bound techniques developed to prove the independence results themselves!

\[ \text{5 Proving Independence} \]

Ultimately, of course, we’d want to prove \( P \) vs. \( \text{NP} \) independent of a strong theory, one that encompasses all “ordinary” mathematical reasoning (or at least a large chunk of it). But there’s a reason that’s probably impossible with current techniques, which was pointed out by Kurtz, O’Donnell, and Royer [33], and (independently) Ben-David and Halevi [6].

Let’s define a \( \Pi_1 \)-sentence to be any sentence of the form, “For all \( x \), \( P(x) \),” where \( P \) is function that can be proven to be recursive in Peano Arithmetic, PA. (That is, there’s a Turing machine \( M \) that computes \( P \), together with a proof in PA that \( M \) halts on every input.) Similarly, a \( \Pi_2 \)-sentence has the form \( \forall x \exists y P(x,y) \) where \( P \) is provably recursive, a \( \Pi_3 \)-sentence has the form \( \forall x \exists y \forall z P(x,y,z) \), and so on.

An amusing exercise is to classify famous unsolved math problems by ‘quantifier complexity.’ Goldbach’s Conjecture is \( \Pi_1 \), since we can write a simple predicate that checks whether an integer is a sum of two primes. The Twin Primes Conjecture is \( \Pi_2 \): for all \( n \), there exists a \( p > n \) such that \( p \) and \( p + 2 \) are prime. To make this a \( \Pi_1 \)-sentence, we’d need to strengthen the conjecture by including an explicit lower bound—say, that there are at least \( \log \log \log n \) twin primes less than \( n \). The complexity of the Riemann Hypothesis (RH) is less obvious. But a result of Lagarias [34] shows that RH is equivalent to the assertion that for all positive integers \( n \), the sum of the divisors of \( n \) is at most

\[ \alpha(n) = H_n + e^{H_n} \ln H_n, \]

where \( H_n = 1 + \frac{1}{2} + \cdots + \frac{1}{n} \) is the \( n^{th} \) harmonic number. This assertion is \( \Pi_1 \), since we can compute increasing good upper bounds on \( \alpha(n) \), dovetailing over all \( n \).

The problem that concerns us, \( P \) vs. \( \text{NP} \), is of course \( \Pi_2 \): for all Turing machines \( M \) and polynomial time bounds \( p \), there exists a SAT instance \( \varphi \) such that \( M \) doesn’t halt with the correct answer to \( \varphi \) in at most \( p(n) \) steps, where \( n \) is the size of \( \varphi \). The \( \text{NP} \) vs. \( P/\text{poly} \) problem is similarly \( \Pi_2 \).

What Kurtz et al. and Ben-David and Halevi noticed is that despite being a \( \Pi_2 \)-sentence, \( P \) vs. \( \text{NP} \) can be extremely well “approximated” by \( \Pi_1 \)-sentences. For example, let \( A \) be the assertion that “SAT does not have circuits of size \( n^{\log n} \)” Then \( A \) is \( \Pi_1 \), since for each \( n \), we can just search exhaustively through all
circuits of size $n^{\log n}$ and see whether any of them computes SAT. Moreover, $A$ implies $P \neq NP$, and is very likely true (we’d imagine) assuming $P \neq NP$.

Now assuming PA is consistent, we can’t prove a $\Pi_1$-sentence independent of PA without proving the sentence—since if the sentence were false, it would have a simple counterexample! (In Apostolos Doxiadis’ novel *Uncle Petros and Goldbach’s Conjecture* [17], the brilliant Uncle Petros misses this point completely while musing about whether Goldbach’s Conjecture is formally independent.)

There’s a further point, however. With the exception of Gödel’s Incompleteness Theorem, all the techniques we have for proving independence from PA actually prove independence from a stronger theory, PA+$\Pi_1$. This theory consists of PA augmented by the set of all true $\Pi_1$-sentences. Since the true $\Pi_1$-sentences can’t even be recursively enumerated, obviously PA+$\Pi_1$ is an unrealistically powerful theory. Despite its power, though, it turns out to be unable to prove even some simple $\Pi_2$-sentences.

As an example, consider Goodstein’s Theorem [22]. We write a positive integer $n$ (say 40) as a sum of powers of 2, then write the exponents as sums of powers of 2, and so on:

$$40 = 2^5 + 2^3 = 2^{2^2+2^0} + 2^{2^2+2^0}.$$  

(This is called the base-2 *hereditary representation.*) Then we change all the 2’s to 3’s to obtain $3^{3^3+3^0}+3^{3^3+3^0}$, then subtract 1 and re-express in base-3 hereditary representation:

$$3^{3^3+3^0}+3^{3^3+3^0} - 1 = 22,876,792,455,041 = 3^{3^3+3^0} + 2 \cdot 3^3 + 2 \cdot 3^{2\cdot3^0} + 2 \cdot 3^{3^0} + 2 \cdot 3^0.$$  

Then we change all the 3’s to 4’s, subtract 1, re-express in base-4 hereditary representation, and so on, increasing the base by 1 each time. Surprisingly, by an argument involving countable ordinals, Goodstein showed that no matter what $n$ we start with, this process will always converge to 0. Subsequently Paris and Harrington [42] showed that Goodstein’s Theorem is independent of PA—meaning that the use of countable ordinals was necessary.

Clearly Goodstein’s Theorem is $\Pi_2$: for all $n$, there exists a $t$ such that the process converges in $t$ steps if we start with $n$. What Paris and Harrington actually showed was that Goodstein’s Theorem is independent of PA+$\Pi_1$—in other words, even if we threw in all true $\Pi_1$-sentences for free, the theorem would still be unprovable in PA. Indeed, Kreisel [32] showed the following (see also [6]).

**Lemma 3 (Kreisel)** A function is provably recursive in PA+$\Pi_1$ if and only if it’s provably recursive in PA.

Now suppose $P$ vs. $NP$ were independent of PA+$\Pi_1$. Then NP would need to have circuits of size $n^{\log n}$—for if it didn’t, then we’d have a true $\Pi_1$-sentence (and hence theorem of PA+$\Pi_1$) that implies $P \neq NP$. Indeed, by Kreisel’s lemma, NP would need to have circuits of size $n^\alpha(n)$, where $\alpha$ is any unbounded function such that $\alpha^{-1}$ is provably recursive in PA. For example, we could take $\alpha$ to be the inverse Ackermann function. One can obtain ridiculous consequences for uniform computation also—although there, to keep the sentences $\Pi_1$, one needs to say something like, “there exists in algorithm that decides SAT in time $n^{\alpha(n)}$ for infinitely many long stretches of $n$’s.”

### 6 Conclusion

It’s clear that a proof of $P \neq NP$, if it exists at all, would require utterly new techniques. The results of the previous section twist the knife further by showing that, if $P \neq NP$ is unprovable in a strong theory such as
Peano arithmetic, then utterly new techniques would be required to show that. These pessimistic results raise a disturbing possibility: could \( P \neq NP \) be somehow “neither true nor false?”

Before you dismiss this question as silly, consider the case of the continuum hypothesis CH. Unlike Gödel’s Incompleteness Theorem, which tells us that a sentence is true by the very fact of its unprovability, the independence results of Gödel and Cohen don’t point us one way or another regarding the “truth” of CH. Cantor thought CH was true. Gödel hoped that new “self-evident” axioms would someday be discovered, which would settle once and for all whether intermediate cardinalities exist. Cohen predicted that CH might come to be seen as *obviously* false—a prediction that’s partly come to pass (see, for example, a survey by Woodin [62]). But even if we reject CH, there remains the question: how many intermediate cardinalities are there? It turns out that *any* number would be consistent with ZF. Cohen thought perhaps there are uncountably many; Woodin thinks there’s exactly one.

Should we lose sleep over this situation? My own feeling is that we shouldn’t—but that perhaps we should readjust our intuitions regarding the continuum. You might think you know what an “arbitrary set of reals” means, but do you? Imagine you were a programmer, and your boss ordered you to write a C program that loops through all sets of reals, halting if it finds one of intermediate cardinality. You might be forgiven for protesting, with Kronecker, that “God created the integers; all else is the work of Man.”

David Deutsch, of quantum computing fame, has argued that even mathematical statements are ultimately about physical reality—since mathematics is rooted in computation, and the laws of physics determine what is and isn’t computable [16]. Whether or not you agree with this, it does suggest the following “physical process criterion” for mathematical truth:

> *We should expect a mathematical question to have a definite answer, if and only if we can phrase the question in terms of a physical process we can imagine.*

There might be questions that we can’t phrase in terms of a physical process, but that we can nevertheless answer—for example, whether \( |S| < |2^S| \) for every set \( S \). Someone who accepted the physical process criterion would see these cases as fortuitous accidents—and would predict that closely related questions can be found which are unanswerable.

That brings us to \( P \) vs. \( NP \). I’ve argued that, like the Continuum Hypothesis, \( P \neq NP \) is a statement as much about as of mathematics—one whose truth or falsehood is almost impossible to investigate without also investigating the means of investigation. But in one crucial respect, \( P \neq NP \) is not at all like CH. It’s a \( \Pi_2 \)-sentence. It refers, not to a transfinite realm beyond human experience, but to machines, Boolean circuits, 1’s and 0’s.

Here’s a simple argument for why, if you accept the physical process criterion, then you should accept that any \( \Pi_\alpha \)-sentence has a definite truth-value, where \( \alpha \) is a countable ordinal. Think of interactive proofs. Naively, one might say that whether a language belongs to \( PSPACE \) or \( NEXP \) is irrelevant, since those classes lie so far beyond what we could compute or even verify. But if we postulate a *prover* who can interact with us—or multiple provers, or a probabilistically-checkable proof—then these questions become relevant. Analogously, although a \( \Pi_2 \)-sentence might be beyond our capacity to verify if true or falsify if false in any finite amount of time, that no longer holds if we have an omniscient wizard around. We propose an efficient algorithm, the wizard responds with a SAT instance on which it fails, we propose another algorithm, the wizard responds with another instance, etc. This process will continue forever if and only if \( P \neq NP \).

The same argument works for a \( \Pi_3 \)-sentence, \( \forall x \exists y \forall z \ P(x, y, z) \). We just need to dovetail. We propose \( x_1 \), the wizard responds with \( y_1 \), we check \( P(x_1, y_1, z) \) for the first few values of \( z \). Then we propose \( x_2 \),

---

12Most computer scientists would agree that the “extended” (or polynomial-time) Church-Turing thesis depends on physical laws, and indeed is probably falsified by quantum mechanics. Deutsch, however, goes further, and says that under different laws of physics, even the set of computable functions would in general be different.
the wizard responds with $y_2$, we check $P(x_2, y_2, z)$ for the first few values of $z$, as well as $P(x_1, y_1, z)$ for a few more values of $z$. And so on. It’s not hard to show by induction that, for any $\Pi_\alpha$-sentence $A$ (where $\alpha$ is a countable ordinal), there exists an “interactive protocol” that continues forever if and only if $A$ is true.

It might be objected that an omniscient wizard doesn’t correspond to a physical process we can imagine. But provided we can imagine playing the game against an opponent with an arbitrary fixed strategy, it seems legitimate also—at least to me!—to imagine that the opponent uses an optimal strategy.

If you’re still reading this, perhaps you’ve really gotten into the spirit of it and asked: what if the universe is finite? Recent evidence for a positive cosmological constant (e.g. [43]), together with arguments from black hole thermodynamics [7], imply an upper bound of about $10^{122}$ on the maximum number of bits accessible by any one observer. Intuitively, even if the universe is spatially infinite, most of it recedes too quickly from any one observer to be seen by that observer. Now, what if your conversation with the wizard would terminate eventually in a proof that $P = NP$, but only after you’d exchanged strings much longer than $10^{122}$ bits? Would the conversation then satisfy the physical process criterion or not? I’ll leave the answer to this as a matter of taste. I’ll say only that my own “Platonic circle”—that is, the set of mathematical entities that I’m willing to regard as physically imaginable—doesn’t extend all the way to uncountable infinities, but it does extend to integers past $10^{122}$, and even $2^{10^{122}}$.

So I’ll state, as one of the few definite conclusions of this survey, that $P \neq NP$ is either true or false. It’s one or the other. But we may not be able to prove which way it goes, and we may not be able to prove that we can’t prove it.

7 Further Reading

For the current consensus on $P$ vs. $NP$, see William Gasarch’s “$P=\neg NP$ poll” [19]—or go to

http://www.ideosphere.com/fx-bin/Claim?claim=P\neq NP

to bet on whether $P \neq NP$ will be proved by 2010. See my Complexity Zoo at

http://www.cs.berkeley.edu/~aaronson/zoo.html

for more about the complexity classes mentioned in this survey. My favorite reference for the material of Section 2 is the book of Cohen [13]. For a definition of Cutting Planes and other proof systems mentioned in Section 4.1, see the survey of Beame and Pitassi [5]. Finally, there are interesting discussions about the logical status of $P$ vs. $NP$ on the Foundations of Mathematics (FOM) mailing list; see


8 Acknowledgments

I thank Lance Fortnow and Alexander Razborov for their comments (even though they don’t agree with everything I say here!), and Jonathan Shewchuk for pointing out some errors and omissions.

References


www.cse.iitk.ac.in/users/manindra/primality.ps.


