Is P Versus NP Formally Independent?

Scott Aaronson

University of California, Berkeley

(Lecture notes edited by John Reif from article by same name by Scott Aaronson)
History of P vs NP Problem:

Gödel’s 1956 letter to von Neumann where P vs. NP was first posed, Gödel apparently saw the problem as a finitary analogue of the Hilbert Entscheidungsproblem

(The Entscheidungsproblem asks for a procedure that, given a mathematical statement, either finds a proof or tells us it is not true.)

“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 $\phi(n) = \max_F \Psi(F,n)$. The question is, how fast does $\phi(n)$ grow for an optimal machine. One can show that $\phi(n) \geq Kn$. If there actually were a machine with $\phi(n) \sim Kn$ (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 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. “
History of P vs NP Problem, Cont:

- **NP Completeness of SAT:**

- **NP Reductions from SAT to Various Combinatorial Optimization Problems:**

- **Universal NP Search Algorithm:**
Importance of P vs NP Problem:

- **The P vs. NP problem has been called “one of the most important problems in contemporary mathematics and theoretical computer science”** (M. Sipser. The history and status of the P versus NP question, in Proceedings of ACM STOC’92, pp. 603–618, 1992. www.cs.berkeley.edu/~luca/cs278/papers/sipser.ps.)

- **Clay Math Institute’s list of million-dollar prize problems includes P vs. NP**
Logic Primer

First Order Logical Sentences $f(x_1, \ldots x_n)$

- Variables $x_1, \ldots x_n$, constant 0,1, and predicate symbols

- Boolean connectives ($\land, \lor, =, \Rightarrow$ etc.)

- quantifiers ($\exists, \forall$)
  (The quantifiers can range only over objects in U, not sets of objects. That’s what ‘first-order’ means.)

- equal signs ($=$)
  (The equal sign is not shorthand for a binary predicate $E(x,y)$; it means that two objects are the same.)

A model $M$ for a theory is:

- A set $U$ of objects (called a universe), together with
- An assignment of ‘true’ or ‘false’ to each logical formula $f(x_1, \ldots x_n)$ for every $x_1, \ldots x_n$ in $U$, such that the axioms hold when the quantifiers range over $U$.

Axioms: An axiom can be any first-order sentence
Example Definition of a group \( G \):

Theory: We have a ternary predicate \( G(x, y, z) \) (intuitively \( x \cdot y = z \)) that satisfies a set of three axioms:

1. **Uniqueness:** \( \forall x, y \exists z \ (G(x, y, z) \land \forall w \ (G(x, y, w) \Rightarrow (w=z))) \)

2. **Associativity:** \( \forall x, y, z, w \ (\exists v \ G(x, y, v) \land G(v, z, w)) \Rightarrow (\exists v \ G(y, z, v) \circ G(x, v, w)) \)

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

Controlling the cardinality of the universe \( U \):
Groups can be:

- finite,
- countable, or
- uncountable.

So the axioms for a group have finite, countable, and uncountable models.

Forcing $|U|$ to be at most 3:

- $\exists x, y, z \forall w (w=x \lor w=y \lor w=z)$
Forcing \(|U|\) to be infinite:

Peano Arithmetic uses a binary predicate \(S(x,y)\) (intuitively \(y = x + 1\)) that satisfies these three axioms:

1. Zero: \(\exists y \forall x \lnot S(x, y)\)

2. UniqueSuccessor: \(\forall x \exists y (S(x, y) \land (y = x) \land \forall z (S(x, z) \Rightarrow (z = y)))\)

3. Unique Predecessor: \(\forall x, y, z (S(x, y) \land S(z, y)) \Rightarrow (x = z)\)
Can we force $|U|$ to be uncountable?

Describing a universe whose objects are sets:

**Zermelo-Fraenkel (ZF) set theory:** Has binary predicate $S(x, y)$ (intuitively $x \in y$), that satisfies the following **axioms:** (writing 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 $A(x,y)$.

(infinitely many Axiom of Replacements for every binary predicate $A \ (x, y)$. More complicated system with finitely many axioms, called Go¨del-Bernays set theory).

(8) **Foundation:** All nonempty sets $x$ contain a member $y$ such that for all $z$, neither $z \in x$ nor $z \in y$. 
Inference rules of first-order logic, by which we prove
Theorems: A first-order sentence is valid if can be obtained by the following rules:

(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) $x = y \land y = z \Rightarrow x = z$,
   (d) $x = y \Rightarrow (A(x) \Rightarrow A(y))$.

(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) $\neg \forall x A(x) \iff \exists x \neg A(x)$,
   (b) $(B \land \forall x A(x)) \iff \forall x (B \land A(x))$,
   (c) $(B \land \exists x A(x)) \iff \exists x (B \land A(x))$.
If a set of axioms has a model, then applying the inference rules above can never lead to a contradiction. Gödel’s Completeness Theorem says the converse:

- If you can’t get a contradiction by applying the rules, then the axiom set has a model.
  (So we didn’t accidentally leave any rules out of the above list.)
- Equivalently, any sentence true in all models is provable.

Consistency of a Theory:
- Given a theory $T$, let $\text{Con} (T)$ be the assertion that $T$ is consistent.
- $\text{Con} (\text{ZF})$ can be expressed in ZF

Gödel’s Incompleteness Theorem:
- $\text{Con} (\text{ZF})$ can’t be proved in ZF, assuming ZF is consistent
  - Assuming ZF is consistent, the “self-hating theory”: $\text{ZF}+\neg \text{Con} (\text{ZF})$ (this is ZF plus the assertion of its own inconsistency), must also be consistent.
- So by the Completeness Theorem: $\text{ZF}+\neg \text{Con} (\text{ZF})$ has a model.
What could be the model for: 
ZF+ ¬ Con(ZF) ?

Fictional dialogue between you and the axioms of ZF + 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 ZF+ Con(ZF): I prefer to talk about integers that encode proofs. (Actually sets that encode integers that encode proofs. But I’ll cut you a break—you’re only human, after all.)

You: Then show me the integer.

Axioms: OK, here it is: X.

You: What the hell is X?

Axioms: It’s just X, the integer encoded by a set in the universe that I describe. You: But what is X, as an ordinary integer?

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

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

Axioms: Greater.

You: The 10^{1,000,000,000} th Ackermann number?Axioms: Greater than that too.

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 Go¨del 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 ZF + 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 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, get the Loïwenheim-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.
**Size of First Order Logic Models:**

- 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!
A statement is independent of a first-order theory if it can neither be proved or disproved.

Methods for Proving Independence
How does one prove a statement independent of a first-order theory?

Show that there are two models:
M1 where the statement holds
M2 where the statement does not hold
Two further methods: consistency strength, and relative consistency.

Consistency Strength Method for Proving Independence:

- If we remove the Axiom of Infinity from ZF, we get a theory equivalent to Peano Arithmetic (PA).

- Not hard to see that $\text{ZF} \implies \text{Con (PA)}$; so Con (PA) is a theorem of ZF. (The reason is that in 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 $\text{ZF} \not\equiv \text{Con(ZF)}$.

(Notice that in PA, we can’t even prove that Con (PA) $\implies$ Con (ZF). For then we could also prove that in ZF, and since ZF $\implies$ Con (PA), we’d have ZF $\implies$ Con (ZF), contradiction.)
Relative Consistency Strength Method for Proving Independence:

- Used by Gödel and Cohen to prove the independence of the Axiom of Choice (AC) and Continuum Hypothesis (CH) from ZF.

Axiom of Choice (AC): 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 $x$.

Continuum Hypothesis (CH): The assertion that there’s no set of intermediate cardinality between the integers and the sets of integers.

Independence of the Axiom of Choice (AC) and Continuum Hypothesis (CH) from ZF:

Gödel proved:

$$\text{Con}(\text{ZF}) \Rightarrow (\text{Con}(\text{ZF} + \text{AC}) \land \text{Con}(\text{ZF} + \text{CH}))$$

Cohen proved:

$$\text{Con}(\text{ZF}) \Rightarrow (\text{Con}(\text{ZF} + \neg \text{AC}) \land \text{Con}(\text{ZF} + \neg \text{CH}))$$

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 $\Rightarrow$ 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.
Summary:

If we wanted to prove that $P \neq NP$ (or $P = NP$) is unprovable in some theory, then show there are distinct models where the statement holds and does not hold.

If this is not possible, there are two further ways we might go about it:

- Consistency Strength or
- Relative Consistency.
Oracles

Given a Turing machine $M$, let $L(M)$ be the language accepted by $M$.

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

**Proof.**

The language $L(M)$ will turn out to be the empty set (that is, $M$ always rejects), but its emptiness can’t be proven in ZF.

By Baker, Gill, and Solovay, there exists the following oracles:

- A computable oracle $A$ relative to which $P = \text{NP}$, and
- Another computable oracle $B$ relative to which $P \neq \text{NP}$.

Let $M_1, M_2, \ldots$ be a standard enumeration of Turing machines.

Let $P_1, P_2, \ldots$ be a standard enumeration of ZF proofs.

Then we define $M$ as follows: given an integer $x$ as input, $M$ accepts if either

- There’s a proof that $P^{L(M)} = \text{NP}^{L(M)}$ among $P_1, \ldots, P_x$, and $x \in B$; or
- There’s a proof that $P^{L(M)} \neq \text{NP}^{L(M)}$ among $P_1, \ldots, P_x$, and $x \in A$. 
We used M in the definition of M, but this is justified by: **Recursion Theorem:** Can always assume without loss of generality that a program has access to its own code.

(The idea is a generalization of the famous self-printing program: Print the following twice, the second time in quotes."Print the following twice, the second time in quotes.")

If there exists a ZF proof that $P^L(M) = NP^L(M)$, then 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 $P^L(M) \neq NP^L(M)$, 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. **QED**
Theorem (Hartmanis, also Kurtz, O’Donnell, and Royer): Can 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 the Oracle O.

Proof: Construct Oracle O so that:

• For almost all input lengths, O collapses P and NP.
• But using a computable function f, for the input lengths $f(1), f(2), f(3), \ldots$, the Oracle O separates P and NP.

This guarantees that $P \neq NP$ relative to O, since there are infinitely many such lengths.

What makes $P^O$ vs. $NP^O$ independent of ZF:

• The function f 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.

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 $P^O = NP^O$. 
Independence in Weak Logical Theories

First Generation of P vs. NP independence results:
Showed P≠ NP unprovable in weak logical theories, by either

- analyzing the possible growth rates of functions provably total in those theories,
- or using the method of consistency strength
DeMillo and Lipton‘s **Theory ET:**

**ET Is a fragment of number theory:**

- The objects of ET are integers,
- Language consists of the functions \(x + y, x - y, x \cdot y, \min \{x, y\}, \max \{x, y\}, \text{ and } c^x\) (where \(c\) is a constant); as well as all polynomial-time computable predicates (but not functions).
- 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.)

**Example:** given integers \(x, y, z\), we can write a predicate that tests whether \(2^x + 3y = 5z\). (Although \(|2^x| + |3y| + |5z|\) is exponentially larger than \(|x| + |y| + |z|\)—since it’s only the predicates, not the arithmetic operations, that need to be polynomial-time.)

**Axioms are absurdly powerful:** they give us Fermat’s Last Theorem for free (for each fixed value of \(n\)).

**Axioms are Weak:** Have only one universal quantifier.

**Theorem** (DeMillo and Lipton): \(P \neq NP\) **unprovable in a ET**

**Proof Idea:** A key part is to bound the possible growth rates of functions expressible with the allowed arithmetic operations.
Sazanov’s Bounded Quantifier Theory T:

- The objects are finite binary strings, and
- Have available all polynomial-time computable functions and predicates.
- Call a sentence ‘true’ if it’s true in the standard model.
- 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.)

Defining Predicates in T:
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 $x$).

Addition Axiom “of Exponentiality” $E$ (not in T): $\forall x \exists y \text{EXP}(x, y)$.

Our bounded quantifiers in $T$ range over strings of length $n$, and strings of length $2^n$ are outside our “universe of discourse” unless we assume axiom $E$. 
**Theorem** (Sazanov) $T + E \Rightarrow \text{Con} (T)$.

**Proof Idea:** Using the method of consistency strength.

Once we have an exponentially-long string $y$, we can construct the set of all polynomial-size strings, and thereby obtain a model for $T$.

**Note:** This is analogous to the fact that ZF 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 $T \not\models E$—since otherwise we’d have $T \Rightarrow \text{Con} (T)$, contradicting the Incompleteness Theorem.
Let **ACCEPT** be the assertion that there exists an algorithm for SAT,

**Theorem (Sazanov)** $T + \text{ACCEPT} \not\equiv E$.

Let ELB (Exponential Lower Bound) be the assertion that any algorithm for SAT requires exponential time. Then

**Theorem (Sazanov)** $T + \text{ACCEPT} + \text{ELB} \Rightarrow E$.

**Proof Idea:**

- ACCEPT tells us that for all SAT instances $x$, there exists a $y$ that is the tableau of a computation of SAT($x$) by some fixed algorithm.
- Meanwhile ELB tells us that $|y|$ is exponentially larger than $|x|$ infinitely often.
- We conclude that $T \not\equiv \text{ELB}$; that is, $T$ is unable to prove exponential lower bounds for SAT.

**QED**
**Natural Proofs** (Razborov and Rudich)

In the 1980’s people developed many ‘combinatorial’ techniques for proving circuit lower bounds:

**Example:** the PARITY function requires constant-depth circuits of exponential size, in un-bounded fanin circuits with gates ∨,∧).

All techniques all follow the same basic “Natural Proof” strategy:

To show that a Boolean function \( f : \{0, 1\}^n \rightarrow \{0, 1\} \) is not in **complexity class** \( C \), we do the following.

1. Define a ‘complexity measure’ \( \mu \) for Boolean functions. (for example, \( \mu \) can be 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 \( f \notin C \)) using a time polynomial in \( 2^n \) procedure to calculate \( \mu(f) \).
Theorem (Razborov and Rudich) If a “Natural Proof” predicate for Boolean function $f$ exists, then we can break (with some probability $1/n^{O(1)}$) any pseudorandom functions computable by the class $C$ with seed length $n^c$, where $c$ is a large constant.

Proof Idea:

- Let $f^* \in C$ be a pseudorandom function with seed length $n^c$, where $c$ is a large constant.
  - Since $f^* \in C$, then $\mu(f^*)$ is small.
  - Note that distinguishing $f^*$ from a truly random number generator by testing every seed exhaustively would appear to take time roughly $2^{n^c}$ by enumerating all seeds of length $n^c$.
- Can show that if a Boolean function $f$ of $n$ inputs has large $\mu(f)$ and so is not in complexity class $C$, then there are a large number of random Boolean functions of $n$ inputs that have large $\mu(f)$ and are not in complexity class $C$.
- Hence a random Boolean function on $n$ inputs with non-negligible probability of having large $\mu$, so can be distinguished from pseudorandom function $f^*$ with non-negligible probability.
- Distinguishing takes time $2^{O(n)}$, or $2^{N^{1/c}}$ in terms of the seed length $N = n^c$.

QED
**Prior Theorem** (Goldreich, Goldwasser, and Micali): Given any pseudorandom generator, we can construct a pseudorandom function that’s roughly as hard to break.

**Conclusion:** This 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 $\varepsilon > 0$. (Whereas the best known factoring algorithm runs in time roughly $2^{n^{1/3}}$).