Peano Arithmetic and Incompleteness: A Guided Exploration

Adapted and Annotated from Stephen Cook's CSC 438F/2404F Notes (2008)

Contents

Preamble

This document is a guided, pedagogically optimized exploration of **Peano Arithmetic (PA)**, its foundational role in mathematical logic, and the implications of its **incompleteness** and **undecidability**. It is adapted from Stephen Cook's 2008 lecture notes and presented with detailed commentary and unpacking.

Audience and Learning Profile

This is intended for curious learners with:

- A first-year undergraduate background in mathematics, computer science, or logic
- A strong preference for both **logical structure** (Abstract Sequential) and **creative experimentation** (Concrete Random)

Learning Style Alignment

This guide is structured to:

- Present material in rigorous, layered progression: *motivation* $\rightarrow$ *formalism* $\rightarrow$ *implications*
- Embed explorations, what-if prompts, and edge-case analyses to support discovery-based understanding
- Avoid oversimplification while scaffolding complex ideas clearly and carefully

1 Motivation and Background

Peano Arithmetic (PA) is a formal theory intended to capture the properties of the natural numbers $\mathbb{N}$ using first-order logic and a minimal set of arithmetic operations. But PA is not just a formalization: it is a deep window into the limits of logic itself.

1.1 Why Study PA?

There are three central motivations for studying Peano Arithmetic:

M1. Logical Foundations of Arithmetic: PA is a precise framework in which to ask and answer questions about numbers, addition, multiplication, and induction.

M2. Gödel's Incompleteness Theorems: PA plays a starring role in the incompleteness results — famously, that any sound, consistent theory expressive enough to include PA cannot prove all truths about the natural numbers.

M3. Formalization Power: Despite its limitations, PA is strong enough to formalize essentially all known theorems of elementary number theory — including, according to some researchers, even Wiles' proof of Fermat's Last Theorem.

1.2 Historical Perspective

- In 1889, Giuseppe Peano introduced a set of axioms for the natural numbers. These are now known as the **Peano Postulates**.
- In the 20th century, logicians sought to recast these postulates inside formal logic.
- This led to the development of PA — a *first-order axiomatic system* meant to capture the arithmetic of $\mathbb{N}$.

1.3 But Something Strange Happens...

PA is a powerful system — and yet, due to Gödel's Theorems, we know that:

- PA is **incomplete** — there exist true arithmetic sentences that PA cannot prove.
- PA cannot even prove its own **consistency**, if it is consistent.

This tension — between formal rigor and logical limits — is what makes the study of PA so intellectually rich.

AS Learning Anchor: Layered Concept Map

Peano Arithmetic (PA) is:

- A first-order theory over the language $\mathcal{L}_A = \{0, s, +, \cdot; =\}$
- Built from axioms P1–P6 (next section) and an Induction Schema
- Powerful enough to formalize arithmetic theorems
- Provably incomplete and undecidable

CR Prompt: What If...

What happens if we remove induction from PA? Could the theory still prove anything useful? Spoiler: This leads us to a weaker system — Robinson Arithmetic (RA) — and a surprising window into undecidability. Stay tuned in Section 5.

---

2 The Peano Postulates (Set-Theoretic Form)

Let us begin with the classic, informal axioms for $\mathbb{N}$, which serve as the foundation for all formal systems that model natural number arithmetic.

2.1 Set-Theoretic Formulation

Let N be a set with a distinguished element $0 \in N$, and a function $S : N \to N$, called the **successor function**. The **Peano Postulates** are:

GP1. $S(x) \neq 0$, for all $x \in N$

GP2. $S(x) = S(y) \Rightarrow x = y$, for all $x, y \in N$

GP3. Let $A \subseteq N$ be such that:

- $0 \in A$

- $x \in A \Rightarrow S(x) \in A$

Then $A = N$

GP3 is the principle of mathematical induction.

2.2 Implications of the Peano Postulates

Any two structures $\langle N, 0, S\rangle$ and $\langle N', 0', S'\rangle$ satisfying GP1–GP3 are **isomorphic**. That is, there exists a bijection $\varphi : N \to N'$ such that:

$$\varphi(0) = 0', \quad \varphi(S(x)) = S'(\varphi(x))$$

Interpretation: These axioms uniquely characterize the structure of the natural numbers — up to isomorphism.

CR Prompt: Can We Break the Rules?

What if we define a structure where $S(x) = 0$ for some x? Or where $S(x) = S(y)$ but $x \neq y$? Try constructing such a toy universe. How do these violations break the idea of a "natural number"?

2.3 From Set Theory to First-Order Logic

While GP1–GP3 are powerful, they are formulated in the language of set theory. To work within a purely logical framework (e.g., first-order logic), we need to formalize these axioms in a more restricted setting.

Problem: The induction axiom GP3 quantifies over all subsets of N. But in first-order logic, we can't quantify over sets — only over elements of the domain.

Solution: We simulate subsets using formulas. That is, a formula $A(x)$ is treated as the characteristic property of a set:

$$\{x \in N \mid A(x) \text{ is true}\}$$

This leads to the **Induction Schema** in PA: a set of axioms, one for every formula $A(x)$.

2.4 The Language of PA

We now define the first-order language used to formalize PA:

$$\mathcal{L}_A = \{0, s, +, \cdot; =\}$$

- 0: a constant symbol for the number zero
- s: a unary function symbol (successor)
- $+$: binary function symbol (addition)

- $\cdot$: binary function symbol (multiplication)
- $=$: binary relation symbol (equality)

Important: In this setup, the universe of discourse is assumed to be $\mathbb{N}$, but this is not stated in the theory. The axioms define the intended structure.

2.5 Axioms of PA

We now introduce the axioms $P1$ through $P6$, along with the **Induction Schema**.

Successor Axioms

P1. $\forall x\ (s(x) \neq 0)$

P2. $\forall x\ \forall y\ (s(x) = s(y) \to x = y)$

Addition Axioms (Recursive Definition)

P3. $\forall x\ (x + 0 = x)$

P4. $\forall x\ \forall y\ (x + s(y) = s(x + y))$

Multiplication Axioms (Recursive Definition)

P5. $\forall x\ (x \cdot 0 = 0)$

P6. $\forall x\ \forall y\ (x \cdot s(y) = (x \cdot y) + x)$

Induction Schema

Let $A(x)$ be any formula in $\mathcal{L}_A$ (possibly with parameters $y_1, \ldots, y_k$). Then the axiom:

$$\forall y_1 \cdots \forall y_k\ \left[(A(0) \wedge \forall x(A(x) \to A(s(x)))) \to \forall x A(x)\right]$$

is part of PA. There is one such axiom for every formula $A(x)$. This is not a single axiom, but an infinite family — hence a schema.

Axiom Set Summary

Define:

$$\Gamma_{PA} = \{P1, P2, \ldots, P6\} \cup \{\text{All Induction Axioms}\}$$

Then PA is the set of all sentences provable from Γ_{PA}. That is:

$$PA = \{A \in \Phi_0 \mid \Gamma_{PA} \vdash A\}$$

AS Reflection: Hierarchy of PA

- **P1–P2:** Describe properties of the successor function
- **P3–P4:** Define addition recursively
- **P5–P6:** Define multiplication recursively
- **Induction:** Captures the essence of natural number reasoning

CR Prompt: Schema vs. Axiom

What if we replaced the entire Induction Schema with a single "Induction Axiom"? Try constructing such an axiom. What goes wrong? (Hint: First-order logic can't quantify over formulas!)

---

3 Examples: Formal Proofs in PA

To build comfort with reasoning inside PA, we now present some example theorems — starting with intuitive truths, then formalizing their derivations using only the axioms of PA.

3.1 Example 1: Every Nonzero Number Has a Predecessor

Let:

$$A(x) := (x = 0) \lor (\exists y\ x = s(y))$$

Claim: $PA \vdash \forall x\ A(x)$
Proof: Use induction on x.

- **Base Case:** $A(0) = (0 = 0) \lor (\exists y\ 0 = s(y))$, which is true.
- **Inductive Step:** Suppose $A(x)$. Then $A(s(x)) = (s(x) = 0) \lor (\exists y\ s(x) = s(y))$
  Since $s(x) \neq 0$ (by P1), we must show $\exists y\ s(x) = s(y)$, which holds with $y = x$.

Thus, by the Induction Schema, we conclude:

$$PA \vdash \forall x\ [(x = 0) \lor (\exists y\ x = s(y))]$$

3.2 Example 2: Associativity of Addition

We will prove the following theorem:

$$PA \vdash \forall x\ \forall y\ \forall z\ ((x + y) + z = x + (y + z))$$

Let:

$$A(z) := ((x + y) + z = x + (y + z))$$

Note that $A(z)$ is a formula with **free parameters** x and y. We will apply the Induction Schema to $A(z)$, treating x and y as constants.

Step 1: Base Case

Prove:

$$PA \vdash A(0) \quad \text{(i.e., } (x + y) + 0 = x + (y + 0)\text{)}$$

$$\begin{aligned}
(x + y) + 0 &= x + y && \text{(by Axiom P3)} \\
x + (y + 0) &= x + y && \text{(by Axiom P3)} \\
\Rightarrow (x + y) + 0 &= x + (y + 0)
\end{aligned}$$

Conclusion: $A(0)$ is provable using P3.

Step 2: Induction Step

Assume:

$$PA \vdash A(z) \quad \text{(Inductive Hypothesis)}$$

We must show:

$$PA \vdash A(s(z)) \quad \text{(i.e., } (x + y) + s(z) = x + (y + s(z))\text{)}$$

$$\begin{aligned}
(x + y) + s(z) &= s((x + y) + z) && \text{(by Axiom P4)} \\
&= s(x + (y + z)) && \text{(by IH)} \\
&= x + s(y + z) && \text{(by Axiom P4)} \\
&= x + (y + s(z)) && \text{(by Axiom P4)}
\end{aligned}$$

Thus:

$$(x + y) + s(z) = x + (y + s(z)) = A(s(z))$$

Conclusion: The induction step is provable using P4 and the induction hypothesis.

Final Conclusion

By the Induction Schema applied to $A(z)$, we conclude:

$$PA \vdash \forall z \; ((x + y) + z = x + (y + z))$$

Since this holds for arbitrary x, y, we generalize:

$$PA \vdash \forall x \; \forall y \; \forall z \; ((x + y) + z = x + (y + z))$$

□

AS Anchor: Key Proof Strategy

- Define property $A(z)$ with free parameters.
- Prove $A(0)$ directly from axioms.
- Assume $A(z)$ holds.
- Prove $A(s(z))$ using the recursive definitions (P3–P4).
- Apply the Induction Schema to generalize over all z.

CR Prompt: Recursive Reasoning Play

Try proving the following using PA axioms and induction:

1. Commutativity of addition: $x + y = y + x$
2. Associativity of multiplication: $(x \cdot y) \cdot z = x \cdot (y \cdot z)$
3. Distributivity: $x \cdot (y + z) = x \cdot y + x \cdot z$

Which axioms are needed? Can you identify patterns or "hidden symmetries" that simplify the inductive steps?

---

4 Robinson Arithmetic (RA): A Finitely Axiomatized Subtheory of PA

4.1 Motivation

PA is not finitely axiomatizable because of its Induction Schema — which includes an infinite family of axioms, one for each formula $A(x)$.

Question: What happens if we drop induction entirely?

Answer: We get a weaker theory known as **Robinson Arithmetic (RA)**.

4.2 Definition of RA

RA uses the same language as PA:

$$\mathcal{L}_A = \{0, s, +, \cdot; =\}$$

Its axioms include:

$$P1, P2, \ldots, P6 \quad \text{(Same as in PA)}$$

RA then adds three more axioms about the ordering relation $\leq$, defined syntactically via:

$$t_1 \leq t_2 \quad \text{stands for} \quad \exists z\ (t_1 + z = t_2)$$

RA includes:

P7. $\forall x\ (x \leq 0 \rightarrow x = 0)$

P8. $\forall x\ \forall y\ (x \leq s(y) \rightarrow (x \leq y \vee x = s(y)))$

P9. $\forall x\ \forall y\ (x \leq y \vee y \leq x)$

Summary

RA = $\{P1, \ldots, P9\}$

Important: RA has no induction. That makes it finitely axiomatized — but much weaker than PA.

AS Insight: Inclusion Relationship

$$RA \subset PA$$

All axioms of RA are provable in PA using induction. But RA is weaker — yet still powerful enough to represent every **recursively enumerable (r.e.) relation**.

CR Prompt: What Can RA Prove?

RA seems crippled without induction. And yet — it can prove surprising things. Try encoding simple computations (e.g., is a number even?) into RA's language. How far can you get before hitting a wall?

5 Representability and Undecidability in RA

5.1 Representability of Relations

Let $R(\vec{x}) \subseteq \mathbb{N}^n$ be an n-ary relation. We say a formula $A(\vec{x})$ **represents** R in a theory Σ if:

$$\forall \vec{a} \in \mathbb{N}^n : \quad R(\vec{a}) \iff \Sigma \vdash A(s\vec{a})$$

Where $s\vec{a}$ denotes the encoding of natural number tuples using successor terms (e.g., $2 \to ss0$).

Interpretation: The formula A "picks out" precisely the tuples in R, via provability in Σ.

5.2 Main Result: RA Representation Theorem

Theorem: Every recursively enumerable (r.e.) relation is representable in RA (and hence in every sound extension of RA) by a formula of the form $\exists y\, \Delta_0$.

- Δ_0: A bounded formula — all quantifiers are bounded (e.g., $\forall x \leq t$, $\exists x \leq t$)
- $\exists\Delta_0$: Existential closure over such bounded formulas

Key Implication: RA can encode any computation that a Turing machine can semi-decide — even though RA lacks induction!

5.3 Proof Sketch

Let $R(\vec{x})$ be r.e. Then there exists a formula $\exists y\, B(\vec{x}, y)$ such that:

$$R(\vec{a}) \iff TA \vdash \exists y\, B(s\vec{a}, y)$$

Using a key lemma (MAIN LEMMA) we will prove later, we can show that such $\exists\Delta_0$ formulas are also provable in RA whenever they are true in the standard model $\mathbb{N}$.

Thus:

$$R(\vec{a}) \iff RA \vdash \exists y\, B(s\vec{a}, y)$$

□

5.4 Corollary 1: Every Sound Extension of RA is Undecidable

Proof: Suppose Σ is a sound extension of RA. Let $K = \{e \mid \varphi_e(e)\downarrow\}$ be the standard halting set.

Since K is r.e., it is representable in Σ by some formula $A(x)$ with:

$$a \in K \iff \Sigma \vdash A(sa)$$

Define:

$$f(a) = \#A(sa) \quad \text{(i.e., the Gödel number of the formula)}$$

Then f is computable. Hence, membership in K reduces to theoremhood in Σ. So if Σ were decidable, then K would be decidable — contradiction.

□

5.5 Corollary 2: Church's Theorem

Statement: The set of valid sentences in the language $\mathcal{L}_A$ is undecidable.

Proof: Let $RA = \{P1, \ldots, P9\}$. Form the conjunction:

$$\gamma = P1 \land P2 \land \cdots \land P9$$

Then:

$$A \in RA \iff (\gamma \to A) \text{ is valid}$$

So deciding validity would decide membership in RA, which is undecidable by Corollary 1.

$\square$

AS Anchor: Summary of Logical Consequences

- **RA is finitely axiomatized** but can represent all r.e. relations.
- **Any sound extension of RA is undecidable**.
- **The set of valid sentences of arithmetic is undecidable**.
- **Induction is not needed** to express rich computational properties — only to prove certain truths.

CR Prompt: Encoding Ideas into RA

Pick a relation like:

1. "x is even" — $\exists y(x = y + y)$
2. "$x < y$" — $\exists z(x + s(z) = y)$
3. "x is prime" — Use universal quantification over divisors

Try writing these relations as $\exists\Delta_0$ formulas in the language $\mathcal{L}_A$. Which ones are representable in RA? How do you know?

6 The MAIN LEMMA: Provability of Bounded Sentences in RA

6.1 Goal and Strategy

Goal: Prove that if a bounded sentence A is true in the standard model $\mathbb{N}$, then:

$$TA \vdash A \quad \text{and hence} \quad RA \vdash A$$

This tells us that bounded truths — those with limited quantifier scope — can be fully captured within RA.

Definition (Bounded Sentence): A sentence is **bounded** if all of its quantifiers are of the form:

$$\forall x \leq t \quad \text{or} \quad \exists x \leq t$$

where t is a term in the language $\mathcal{L}_A$, and x does not appear in t.

Examples

- Bounded: $\forall x \leq s(s0)\, \exists y \leq x\, (x = y + y)$
- Not bounded: $\forall x\, \exists y\, (x = y + y)$

Intuition: Bounded quantification means we can "unroll" the formula into a finite check.

6.2 Expanding the Language: $\mathcal{L}_{A,\leq}$

To simplify reasoning about bounded quantifiers, we extend our language by adding a primitive symbol $\leq$.

$$\mathcal{L}_{A,\leq} = \mathcal{L}_A \cup \{\leq\}$$

Then we define a companion theory:

$$RA_{\leq} = RA \cup \{P0\}$$

Where:

$$\textbf{P0:}\ \forall x\, \forall y\, (x \leq y \leftrightarrow \exists z (x + z = y))$$

Translation Lemma (Semantic Equivalence)

Let A be any formula over $\mathcal{L}_{A,\leq}$, and let A' be the formula in $\mathcal{L}_A$ obtained by replacing each instance of $t_1 \leq t_2$ with $\exists z(t_1 + z = t_2)$, where z is fresh.

Then:

$$RA_{\leq} \vdash A \iff RA \vdash A'$$

This allows us to freely work in $\mathcal{L}_{A,\leq}$ with $\leq$ as a primitive, knowing that every result is still valid in the original theory after translation.

6.3 Proof of the MAIN LEMMA

We proceed by **induction on the complexity** (number of connectives and quantifiers) of the bounded sentence A. Prior to that, we normalize A into negation-normal form.

Negation Normalization

Drive all negations inward using:

$$
\begin{gathered}
\neg\neg A \equiv A \\
\neg(\forall x \leq t)\, B(x) \equiv \exists x \leq t\, \neg B(x) \\
\neg(\exists x \leq t)\, B(x) \equiv \forall x \leq t\, \neg B(x)
\end{gathered}
$$

Thus, every sentence is equivalent to one built from atomic formulas and connectives $\wedge, \vee$, and bounded quantifiers.

6.4 Base Case: Atomic Formulas

There are four atomic sentence types in $\mathcal{L}_{A,\leq}$:

- $t = u$
- $t \neq u$
- $t \leq u$
- $\neg(t \leq u)$

Let us establish RA-provability of true instances of each.

Lemma A1: Arithmetic of Numerals

Let $m, n \in \mathbb{N}$. Then:

$$RA \vdash s^m + s^n = s^{m+n}, \quad RA \vdash s^m \cdot s^n = s^{m \cdot n}$$

Proof: By induction outside the system using Axioms P3–P6.

Lemma A: Term Equality for Closed Terms

Let t be a closed term (no variables) such that its standard interpretation is s^n. Then:

$$RA \vdash t = s^n$$

Proof: Structural induction on the term t, applying Lemma A1.

Lemma B: Inequality of Distinct Numerals

If $m < n$, then:

$$RA \vdash s^n \neq s^m$$

Proof: Induction on m, using Axioms P1 (successor not 0) and P2 (injectivity of s).

Lemma C: Bounded Enumeration of Numerals

For all $n \in \mathbb{N}$:

$$RA_{\leq} \vdash \forall x \ (x \leq s^n \to (x = 0 \lor x = s(0) \lor \cdots \lor x = s^n))$$

Proof: Induction on n using P7 and P8.

These lemmas suffice to handle atomic formulas and negated atomic formulas.

6.5 Inductive Step: Connective Cases

Suppose $A = B \land C$ or $A = B \lor C$. Since both B and C are simpler, we apply the induction hypothesis:

$$RA \vdash B, \quad RA \vdash C \Rightarrow RA \vdash B \land C$$

Trivial for conjunction and disjunction.

6.6 Inductive Step: Quantifier Cases

Suppose:

$$A = \forall x \leq t \, B(x)$$

Let t be a closed term; then by Lemma A, $RA \vdash t = s^n$. So we reduce to proving:

$$RA_{\leq} \vdash \forall x \ (x \leq s^n \to B(x))$$

Idea: Show that:

$$RA \vdash B(0), B(1), \ldots, B(n) \Rightarrow RA \vdash \forall x \leq s^n \, B(x)$$

By Lemma C and substitution properties.

Case: Existential Bounded Quantifier

If $A = \exists x \leq t \, B(x)$, and some instance $B(k)$ is true for $k \leq n$, then by the induction hypothesis:

$$RA \vdash B(k) \Rightarrow RA \vdash \exists x \leq t \, B(x)$$

(We'll formalize this equivalence shortly.)

Conclusion: MAIN LEMMA

If A is a true bounded sentence in the standard model, then:

$$RA \vdash A$$

$\square$

CR Prompt: What's Outside the MAIN LEMMA?

Construct a true sentence that is not bounded, such as:

$$\forall x \, \exists y \, (x < y \wedge \mathrm{Prime}(y))$$

Why doesn't the MAIN LEMMA apply? Can you bound the quantifiers somehow to fall under the lemma's reach?

6.7 Corollaries

Corollary 1. The set of bounded sentences of TA is decidable.

Corollary 2. Every true $\exists\Delta_0$ sentence is provable in RA.

Corollary 3. The set of $\exists\Delta_0$ theorems of RA is r.e.

AS Anchor: Concept Stack Summary

- **Bounded sentence:** All quantifiers are bounded by terms.
- **Translation Lemma:** Eliminates $\leq$ via encoding.
- **MAIN LEMMA:** All true bounded sentences are provable in RA.
- **Engine for:** RA Representation Theorem, Church's Theorem, Undecidability.

7 Formal Proof: RA Representation Theorem

7.1 Theorem (RA Representation Theorem)

Statement: Every recursively enumerable (r.e.) relation $R(\vec{x})$ is representable in RA by an $\exists\Delta_0$ formula.

That is, there exists a bounded formula $B(\vec{x}, y)$ such that:

$$\forall \vec{a} \in \mathbb{N}^n : \quad R(\vec{a}) \iff RA \vdash \exists y \, B(s\vec{a}, y)$$

7.2 Proof

Step 1: Exists-Delta Theorem (from prior work)

Since $R(\vec{x})$ is r.e., there exists a bounded formula $B(\vec{x}, y)$ such that:

$$\forall \vec{a} \in \mathbb{N}^n : \quad R(\vec{a}) \iff TA \vdash \exists y \, B(s\vec{a}, y)$$

Step 2: Apply MAIN LEMMA

If $R(\vec{a})$ holds, then $\exists y \, B(s\vec{a}, y) \in TA$. Since $\exists y \, B(s\vec{a}, y)$ is an $\exists\Delta_0$ formula (bounded), we conclude:

$$RA \vdash \exists y\, B(s\vec{a}, y)$$

Step 3: Definition of Representability
By definition, $\exists y\, B(\vec{x}, y)$ represents $R(\vec{x})$ in RA.

$\square$

CR Prompt: Turing Machines in RA?

Try designing a formula $B(e, x, y)$ to encode: "Turing machine e halts on input x in y steps."
Can you write this as an $\exists\Delta_0$ formula? How would you express the step-by-step transitions?

7.3 From Representability to Undecidability

We now generalize our conclusions to arbitrary sound extensions of RA.

Corollary: Every Sound Extension of RA is Undecidable

Let $\Sigma \supseteq RA$ be sound.
Let $R(x) = K(x)$ be the halting problem (r.e.). Then by the RA Representation Theorem:

$$\exists y\, A(x, y) \text{ represents } R(x) \text{ in } \Sigma$$

That is:

$$x \in K \iff \Sigma \vdash \exists y\, A(sx, y)$$

Let $f(x) = \#\exists y\, A(sx, y)$. Then f is computable. Hence:

$$x \in K \iff f(x) \in \Sigma^{\wedge}$$

So the halting problem reduces to the set of theorems of Σ. Therefore, Σ is undecidable.

$\square$

7.4 Church's Theorem (General Form)

Statement: The set `VALID` of all valid sentences in the language $\mathcal{L}_A$ is undecidable.
Proof: Let $\gamma = P1 \wedge \cdots \wedge P9$. Then for any sentence $A \in \mathcal{L}_A$:

$$RA \vdash A \iff (\gamma \to A) \text{ is valid}$$

Thus, deciding validity would allow us to decide RA, which is undecidable. Hence, `VALID` is undecidable.

$\square$

AS Anchor: Logical Landscape Recap

- **RA can represent any r.e. relation.**
- **Any sound extension of RA is undecidable.**
- **Validity over arithmetic is undecidable.**
- **RA is still a finite theory (no induction).**

CR Prompt: What if RA Were Decidable?

Imagine a world where RA *is* decidable. Could you then decide the halting problem? Construct a path of reasoning that leads to a contradiction.

8 Strong Representability and the Main Undecidability Theorem

8.1 Definition: Strong Representability

Let Σ be a theory and $R(\vec{x}) \subseteq \mathbb{N}^n$. A formula $A(\vec{x})$ **strongly represents** R in Σ if:

$$\begin{aligned}&\forall \vec{a} \in \mathbb{N}^n :\\ &\quad R(\vec{a}) \Rightarrow \Sigma \vdash A(s\vec{a})\\ &\quad \neg R(\vec{a}) \Rightarrow \Sigma \vdash \neg A(s\vec{a})\end{aligned}$$

Remarks:

- This is a stronger condition than ordinary representability.
- If Σ is consistent, then strong representability implies representability.
- The converse holds only if Σ is complete.

8.2 Strong RA Representation Theorem

Theorem: Every **recursive** relation $R(\vec{x})$ is strongly representable in RA by a formula of the form:

$$A(\vec{x}) = \exists y\, [B_1(\vec{x}, y) \wedge \forall z \leq y\, \neg B_2(\vec{x}, z)]$$

Where B_1 and B_2 are bounded formulas representing R and $\neg R$, respectively.

8.3 Proof

Let $R(\vec{x})$ be recursive. Then both R and $\neg R$ are r.e.

So:

- There exists a Δ_0 formula $B_1(\vec{x}, y)$ representing R
- There exists a Δ_0 formula $B_2(\vec{x}, z)$ representing $\neg R$

Define:

$$A(\vec{x}) := \exists y \, [B_1(\vec{x}, y) \wedge \forall z \leq y \, \neg B_2(\vec{x}, z)]$$

Case 1: $R(\vec{a})$ is true

Then $B_1(s\vec{a}, sb) \in TA$ for some b. Since $\neg R(\vec{a})$ is false, we know:

$$\forall z \leq sb \, \neg B_2(s\vec{a}, z) \in TA$$

By the MAIN LEMMA, both parts are provable in RA. Therefore:

$$RA \vdash A(s\vec{a})$$

Case 2: $\neg R(\vec{a})$ is true

Then $B_2(s\vec{a}, sc) \in TA$ for some c

By P9: $\forall y \; (y \leq sc \vee sc \leq y) \in RA$

We break into cases:

- If $y \leq sc$, then $B_2(s\vec{a}, y)$ is true for some y. So the negation of $\forall z \leq y \, \neg B_2(s\vec{a}, z)$ holds.
- If $sc \leq y$, then $\exists z \leq y \, B_2(s\vec{a}, z)$ is provable in RA

Hence:

$$RA \vdash \neg A(s\vec{a})$$

□

AS Anchor: Representation Notions Compared

- **Representability:** $R(\vec{a}) \Rightarrow \Sigma \vdash A(s\vec{a})$
- **Strong Representability:** Add $\neg R(\vec{a}) \Rightarrow \Sigma \vdash \neg A(s\vec{a})$
- Strong representability implies full syntactic alignment between truth and provability

CR Prompt: Build a Strong Representation

Try explicitly building $A(x)$ that strongly represents the relation:

$$R(x) := \text{"}x \text{ is even"} \Rightarrow \exists y(x = y + y)$$

Write $A(x) := \exists y[x = y + y \wedge \forall z \leq y\, x \neq z + z + 1]$
Can you justify why this meets both directions of strong representability?

8.4 Undecidability Theorem (General Form)

Theorem: If every recursive relation is representable in Σ, then Σ is undecidable.

Proof Strategy (Diagonalization)

Suppose Σ is recursive. Then:

$$\text{Define } R(x) := \text{“x is the code of a formula not provable in } \Sigma\text{”}$$

Then R is recursive. Let $A(x)$ represent $R(x)$ in Σ
Let $e = \#A(x)$, and define:

$$S(x) := \neg R(d(x)) \quad \text{where } d(x) = \#A(sx)$$

By construction, we derive a contradiction akin to Gödel's theorem. Hence Σ is not recursive.

□

8.5 Main Theorem: Every Consistent Extension of RA is Undecidable

Proof:
Let $\Sigma \supseteq RA$ be consistent. Let $R(\vec{x})$ be recursive. Then:

$$\text{Strong RA Representation Theorem} \Rightarrow A(\vec{x}) \text{ strongly represents } R \text{ in RA}$$

Then A represents R in every consistent extension of RA, including Σ

$$\Rightarrow \text{By the Undecidability Theorem, } \Sigma \text{ is undecidable}$$

□

AS–CR Reflection: Why This Matters

This result shows that the limit is not about *truth*, but about *structure*. Even weak, syntactically defined theories like RA generate undecidability when they encode the arithmetic of computation.

CR Prompt: Make Your Own Theory

Try defining a new theory Σ by adding a (false) axiom to RA, like:

$$\Sigma = RA + \text{“There are only finitely many primes”}$$

Is Σ consistent? What does the Main Theorem say about its decidability?

9 Nonstandard Models and the Separation of RA and PA

9.1 Motivation

The previous sections have shown that:

- RA is a finite, weak theory
- PA is a strictly stronger theory (includes induction)
- But we haven't yet proven that $RA \neq PA$

We now provide a **model-theoretic separation**: we construct a model that satisfies all axioms of RA, but fails a sentence provable in PA. This proves:

$$RA \neq PA$$

9.2 The Structure $\mathbb{Z}[X]^+$

Let:

$$\mathbb{Z}[X]^+ = \{p(X) \in \mathbb{Z}[X] \mid p = 0 \text{ or leading coefficient of } p > 0\}$$

Universe: All integer-coefficient polynomials with either:

- Zero polynomial
- Leading coefficient positive

Operations:

- $+$: polynomial addition
- $\cdot$: polynomial multiplication
- $s(p(X)) := p(X) + 1$
- $0 :=$ the zero polynomial

This defines a structure over the language $\mathcal{L}_A = \{0, s, +, \cdot; =\}$

9.3 Claim: $\mathbb{Z}[X]^+ \models RA$

We must verify axioms P1–P9.

P1: $s(p) = p + 1 \neq 0$ for all $p \in \mathbb{Z}[X]^+$
If $p + 1 = 0$, then $p = -1 \notin \mathbb{Z}[X]^+$, since its leading coefficient is negative.

P2: $s(p) = s(q) \Rightarrow p + 1 = q + 1 \Rightarrow p = q$

P3–P6: Arithmetic is standard and obeys recursive definitions.

P7–P9: Use the syntactic definition of $\leq$: $p \leq q \iff \exists r(p + r = q)$. This holds due to closure of $\mathbb{Z}[X]^+$ under addition.

9.4 Sentence That Separates PA and RA

Define:

$$A := \exists x \, \forall y \, [x \neq y + y \wedge x \neq y + y + s(0)]$$

Interpretation: "There exists a number x that is neither even nor odd."
This is **false in** $\mathbb{N}$, so:

$$PA \vdash \neg A \quad \text{(provable by induction)}$$

However, in $\mathbb{Z}[X]^+$:

- Let $x = X \in \mathbb{Z}[X]^+$
- For any polynomial y, $y + y = 2y$, and $y + y + 1 = 2y + 1$
- So $X \neq 2y$, and $X \neq 2y + 1$, because their degrees differ

Hence:

$$\mathbb{Z}[X]^+ \models A \Rightarrow A \notin RA$$

Therefore:

$$RA \nvdash \neg A, \quad \text{but } PA \vdash \neg A \Rightarrow RA \neq PA \quad \square$$

9.5 Corollary: RA and PA Have Different Models

RA has "exotic" models like $\mathbb{Z}[X]^+$

PA forces models to behave like $\mathbb{N}$ at the level of induction — even if they are nonstandard in size or structure.

AS Anchor: Semantic View of Theories

- **PA:** All models satisfy induction — eliminates certain counterexamples
- **RA:** More permissive — satisfied by wider class of structures
- **Model-theoretic separation:** One sentence provable in PA, but not in RA

CR Prompt: Build a Fake Arithmetic

Try defining a model like $\mathbb{Z}[X]^+$ with strange properties. For example:

- Use matrices or functions as numbers
- Define "successor" as adding a fixed function
- What RA axioms hold? What PA sentences fail?

Challenge: Can you build a consistent model of RA in which some basic arithmetic truths from $\mathbb{N}$ fail?

10 Synthesis: The Structure of Arithmetic Theories

10.1 Axiomatizability, Consistency, and Completeness

Let Σ be a theory over the language $\mathcal{L}_A$. Consider the following properties:

Axiomatizable: Σ is recursively enumerable

Sound: All theorems of Σ are true in $\mathbb{N}$

Complete: For every sentence A, either $\Sigma \vdash A$ or $\Sigma \vdash \neg A$

Decidable: There is an algorithm to decide whether $A \in \Sigma$

10.2 The Hierarchy of Results

T1. (Gödel's First Incompleteness) If Σ is sound, axiomatizable, and contains PA, then it is incomplete.

T2. (Gödel's Second Incompleteness) Such a Σ cannot prove its own consistency.

T3. (Church's Theorem) The set of valid sentences in $\mathcal{L}_A$ is undecidable.

T4. (RA Representation Theorem) Every r.e. relation is representable in RA by an $\exists\Delta_0$ formula.

T5. (Strong RA Representation Theorem) Every recursive relation is strongly representable in RA.

T6. (Main Undecidability Theorem) Every consistent extension of RA is undecidable.

T7. (Separation Theorem) There exist sentences (e.g., $\neg A$) provable in PA but not in RA.

10.3 Diagram: Semantic vs Syntactic Boundaries

Semantic Truths (TA)
$\searrow$ provable bounded fragment
$\swarrow$
RA
$\hookrightarrow$ finitely axiomatized, undecidable, incomplete
$\searrow$
PA
$\hookrightarrow$ axiomatizable, undecidable, incomplete
$\searrow$
Sound Extensions of PA
$\hookrightarrow$ still incomplete by Gödel 2

AS Summary Table: Theory Properties

Theory	**Finitely Axiomatized**	**Decidable**	**Complete**	**Sound**
RA	Yes	No	No	Yes (in $\mathbb{N}$)
PA	No	No	No	Yes
TA	N/A	N/A	Yes	Yes

CR Prompt: Build the Boundary

Suppose you define a theory $\Sigma := RA + A$ for some undecidable sentence $A \notin RA$. Can you make Σ consistent but incomplete? What are its models?

Try this for different kinds of A:

- A sentence false in $\mathbb{N}$
- A sentence true but not provable in RA
- A sentence unprovable in PA (if one exists)

11 Exercises for Mastery and Discovery

11.1 Abstract Sequential (AS) Exercises

A1. Prove that RA proves the commutativity of addition: $x + y = y + x$. State which axioms you use.

A2. Use the Induction Schema to prove: $PA \vdash x \cdot (y + z) = x \cdot y + x \cdot z$

A3. Translate the sentence $x \leq y \to x = y$ into pure $\mathcal{L}_A$ and prove/disprove it in RA.

A4. Prove Lemma C: $\forall x (x \leq s^n \to x = 0 \vee x = 1 \vee \cdots \vee x = s^n)$

A5. Show that each of P7, P8, and P9 can be proved in PA using induction.

11.2 Concrete Random (CR) Explorations

D1. Construct a nonstandard model of RA using:

- Matrices with non-negative integer entries
- Successor defined as $M \mapsto M + I$
- Addition and multiplication as matrix ops

Which axioms hold? Which fail?

D2. Try creating a bounded formula $B(x, y)$ that encodes: *"y is the smallest prime factor of x"* Can you express this entirely in Δ_0 notation?

D3. Write a program that takes a sentence in $\mathcal{L}_A$ and checks whether it is syntactically bounded. Then auto-generate brute-force proofs for the first 20 true bounded sentences.

D4. Explore the theory $\text{Th}(\mathbb{Q}^+)$ under $s(x) = x + 1$, with rational numbers. Which RA axioms hold?

D5. Invent your own symbolic arithmetic universe using ASCII strings:

- "0" := empty string
- "s" := string concatenation
- "+" := string doubling

Define an RA-like theory. What kind of arithmetic emerges?

Closing Note

The study of RA and PA is not only about arithmetic — it is a deep probe into the limits of formal reasoning, the boundary between computation and logic, and the architecture of provability. You have now traversed a complete ladder:

$$\text{Computation} \to \text{Arithmetic} \to \text{Proof Theory} \to \text{Metamathematics}$$

There is much more to explore. And your tools — structure and curiosity — are exactly what's needed to go further.