Gödel's 1st Incompleteness Theorem

Let $N$ be a first-order formal theory of arithmetic that is recursively axiomatizable. If $N$ is consistent, then it is negation incomplete.

Questions:
1. What is a "first-order formal theory of arithmetic"?
2. What does it mean to say a first-order formal theory of arithmetic is "consistent" and "negation incomplete"?
3. What does it mean to say a first-order formal theory of arithmetic is "recursively axiomatizable"?
1. First-order Formal Theory

A **formal theory** $T$ consists of:
(a) a *formal language* $L_T$ (alphabet, grammar, semantics),
(b) a set of axioms (a set of wffs of the language),
(c) a *proof system* (a method that allows derivations of more complex wffs from the axioms).

- $T$ is *first-order* if $L_T$ only contains variables for individuals, and not variables for predicates (2nd-order), or variables for predicates of predicates (3rd-order), etc.

- A *formal theory of arithmetic* is a formal theory whose language can express all the claims made about natural numbers in simple arithmetic (addition, subtraction, multiplication, division).

- **Idea**: To formalize arithmetic, we want to demonstrate how all of its true claims ("theorems") can be derived from a set of basic truths (axioms).
2. Consistency and Negation Completeness

- **Theorem**: A theorem of $T$ is a wff of $L_T$ that is provable in $T$'s proof system.
  - **Notation**: $T \vdash \varphi$ means "$\varphi$ is a theorem of $T$".

- **Logically Valid**: A logically valid wff of $T$ is a wff of $L_T$ that is true in all interpretations.
  - **Notation**: $T \models \varphi$ means "$\varphi$ is a logically valid wff of $T$".

- **Sound**: $T$ is sound just when every theorem of $T$ is logically valid:
  For any wff $\varphi$ of $L_T$, if $T \vdash \varphi$, then $T \models \varphi$.

- **Semantically Complete**: $T$ is semantically complete just when every logically valid wff of $T$ is a theorem of $T$:
  For any wff $\varphi$ of $L_T$, if $T \models \varphi$, then $T \vdash \varphi$.

**Two more syntactic notions**:

- **Consistent**: $T$ is consistent just when, for any wff $\varphi$ in $L_T$, it's not the case that both $T \vdash \varphi$ and $T \vdash \neg \varphi$.

- **Negation Complete**: $T$ is negation complete just when, for any wff $\varphi$ in $L_T$, either $T \vdash \varphi$ or $T \vdash \neg \varphi$.

**Motivations**:

**Consistency**: We don't want our theory of arithmetic to make contradictory claims.
- We don't want to be able to prove that 2 is both even and not even.

**Negation Completeness**: We want our theory of arithmetic to have something to say about any claim made about natural numbers.
- We want to be able to either prove or refute any such claim.
Example:

- Let $L$ consist of the alphabet $P, Q, R, \land, \lor, \neg, (, )$ and the grammar and semantics of $\text{PL}$.
- Let the proof system be the $\text{PL}$ tree rules.

- Consider two theories:
  - $T_1$, with one axiom: $\{\neg P\}$.
  - $T_2$, with three axioms: $\{\neg P, Q, \neg R\}$.

- Both $T_1$ and $T_2$ are sound and semantically complete (since $\text{PL}$ is).
- Both $T_1$ and $T_2$ are consistent.

Negation complete?

- $T_1$: No! There are wffs $\varphi$ of $L$ such that neither $\varphi$ nor $\neg \varphi$ is a theorem of $T_1$.
  - Ex. $(Q \land R)$. Trees for $\neg P \vdash (Q \land R)$ and $\neg P \vdash \neg (Q \land R)$ do not close.
  - Which means: The "given" $\neg P$ doesn't entail either $(Q \land R)$ or $\neg (Q \land R)$.

- $T_2$: Yes! For any wff $\varphi$ of $L$, there is a closed tree for either $\neg P, Q, \neg R \vdash \varphi$, or $\neg P, Q, \neg R \vdash \neg \varphi$.
  - The "given" $\neg P, Q, \neg R$ entail any wff formed from $P, Q, R$, via $\text{PL}$ connectives.

Moral: Semantic completeness is distinct from negation completeness.

- $T_1$ is not negation complete, but uses a proof system ($\text{PL}$ trees) that is semantically complete.
Example:
- Let $L$ consist of the alphabet $P, Q, R, \wedge, \lor, \neg, (, )$ and the grammar and semantics of PL.
- Let the proof system be the PL tree rules.
- Consider two theories:
  - $T_1$, with one axiom: $\{\neg P\}$.
  - $T_2$, with three axioms: $\{\neg P, Q, \neg R\}$.
- Both $T_1$ and $T_2$ are sound and semantically complete (since PL is).
- Both $T_1$ and $T_2$ are consistent.
- $T_1$ is not negation complete, $T_2$ is negation complete.

Note: We can "mechanically decide" what is a wff in $T_1$ and $T_2$, and hence what wffs are axioms.
- There is a mechanical, step-by-step process in $L$ of building complex wffs from atomic wffs, and atomic wffs from terms.

And: We can also "mechanically decide" what counts as a proof (a closed tree) in $T_1$ and $T_2$, and hence, for any wff, whether it is a theorem of $T_1$ or $T_2$.

Question: Can we make the notion of "mechanical decision procedure" more precise?
3. Recursively Axiomatizable Formal Theory

A formal theory $T$ is recursively axiomatizable just when its axioms can be encoded as recursive properties of natural numbers.

- **Motivation**: Makes possible a mechanical decision procedure (algorithm) that can decide for any wff of $L_T$, whether it is an axiom of $T$.
- **Holy Grail**: To construct a mechanical decision procedure that would decide for any wff of $L_T$, whether it is a theorem of $T$.

---

**Is Fermat's Last "Theorem" really a theorem?**

For $n \geq 3$, there are no whole numbers $x, y, z$ such that $x^n + y^n = z^n$.

![Pierre de Fermat](image)

Proven by Andrew Wiles in 1993 after 3 centuries of work.

---

**Is the Poincaré Conjecture a theorem?**

Every simply connected closed 3-manifold is homomorphic to the 3-sphere. (Or: the 3-sphere is the only type of bounded 3-dim space that contains no holes.)

![Henri Poincaré](image)

Supposedly proven by Grigori Perelman in 2003 after a century and $1$ million prize.

---

*Wouldn't it be easier if there were a program that decided which statements were theorems and which weren't?*
**Link between mechanical ("effective") decidability and recursive properties.**

- A recursive property can be encoded in a *primitive recursive* (p.r.) function.
- **And:** P.r. functions are generated by a mechanical algorithm.

**Idea:** Start with three simple functions as your "starter pack":

(i) Successor function. \( S(x) = \text{successor of } x. \)
(ii) Zero function. \( Z(x) = 0. \)
(iii) \( k \)-place identity function. \( I^k_i(x_1, \ldots, x_k) = x_i \quad 1 \leq i \leq k. \)

**Now:** Generate more complex functions from starter pack by one of two methods:

(a) **Primitive recursion:** Specify value of function for 0, then specify value for a given argument in terms of its value for smaller arguments.

(b) **Composition:** Generate a new function by composing two already-generated functions.

**Examples:**

<table>
<thead>
<tr>
<th>Sum function. ( +(x, y) )</th>
<th>Product function. ( \times(x, y) )</th>
<th>Factorial function. ( !(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( +(x, 0) = x = I^1_1(x) )</td>
<td>( \times(x, 0) = 0 = Z(x) )</td>
<td>( !(0) = 1 = S(0) )</td>
</tr>
<tr>
<td>( +(x, S(y)) = S(+(x, y)) )</td>
<td>( \times(x, S(y)) = +\big(\times(x, y), x\big) )</td>
<td>( !(S(y)) = \times( !(x) , S(x) ) )</td>
</tr>
</tbody>
</table>

**Claim (Church's Thesis):**

A (partial) function on the natural numbers is computable by algorithm (mechanically computable) if and only if it is a recursive (partial) function.
So: Gödel's 1st Incompleteness theorem says:
"Any attempt to consistently formalize arithmetic as a first-order theory with
"mechanically" recognizable axioms will be negation incomplete: There will be some
claim about natural numbers that is neither provable nor refutable in the theory."

What's the Big Deal?
• Big Deal if you think there is a formal theory that captures all the claims of
  arithmetic.
4. Aspects of the Proof

**Peano Arithmetic**: A first-order recursively axiomatizable formal theory of arithmetic; call it $N$, with language $L_N$.

**The Alphabet of $L_N$**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>individual constant</td>
</tr>
<tr>
<td>$x, y, z, \ldots, v_k$</td>
<td>individual variables ($k \geq 0$)</td>
</tr>
<tr>
<td>$=$</td>
<td>2-place predicate (identity)</td>
</tr>
<tr>
<td>$S$</td>
<td>1-place function (successor)</td>
</tr>
<tr>
<td>$+, \times$</td>
<td>2-place functions (sum, product)</td>
</tr>
<tr>
<td>$\land, \lor, \neg, \supset, \forall, \exists, (, )$</td>
<td>connectives, quantifiers, punctuation</td>
</tr>
</tbody>
</table>

**Grammar of $L_N$**: Same as $\text{QL}^f$.

- **Convention**: Write $t_1 + t_2$ and $t_1 \times t_2$, instead of $+(t_1, t_2)$ and $\times(t_1, t_2)$.

**Semantics of $L_N$**: Same as $\text{QL}^f$.

- Intended domain of all $q$-valuations is the set of natural numbers.

- On this domain:

  - The $q$-value of the constant $0$ is the number $0$.
  - The $q$-value of $=$ is the set of all 2-tuples of numbers of the form $\langle m_1, m_2 \rangle$ where $m_1 = m_2$.
  - The $q$-value of $S$ is the set of 2-tuples of numbers $\{\langle 0,1 \rangle, \langle 1,2 \rangle, \langle 2,3 \rangle, \langle 3,4 \rangle, \ldots \}$.
  - The $q$-value of $+$ is the set of all 3-tuples of numbers of form $\langle m_1, m_2, m_3 \rangle$ where $m_1 + m_2 = m_3$.
  - The $q$-value of $\times$ is the set of all 3-tuples of numbers of form $\langle m_1, m_2, m_3 \rangle$ where $m_1 \times m_2 = m_3$. 
The axioms of $N$

(N1) $\forall x (0 \neq Sx)$
(N2) $\forall x \forall y (Sx = Sy \supset x = y)$
(N3) $\forall x (x + 0 = x)$
(N4) $\forall x \forall y (x + y = S(x + y))$
(N5) $\forall x (x \times 0 = 0)$
(N6) $\forall x \forall y (x \times Sy = (x \times y) + x)$
(N7) $\{ \varphi(0) \wedge \forall x (\varphi(x) \supset \varphi(Sx)) \} \supset \forall x \varphi(x)$, for $\varphi(x)$ an open wff with $x$ free.

- (N7) is the Axiom of Mathematical Induction.
  - It says: "For any property of natural numbers $\varphi$, if 0 has it, and if, for any number $n$, if $n$ has it entails that the successor of $n$ has it, then all numbers have it."

- Now: Let's show that $N$ is recursively axiomatizable.
  - Which means: Its axioms can be encoded in recursive functions.

- To do this, we'll first code the wffs and sequences of wffs of $L_N$ as numbers.
Gödel Numbering

- Let the symbols in the alphabet of $L_N$ be encoded by numbers by:

| ∧ | ∨ | ¬ | ⊃ | ∀ | ∃ | ( | ) | 0 | = | S | + | × | x | y | z | ... |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 | 23 | 25 | 2 | 4 | 6 | ... |

- Let expression $e$ in $L_N$ be the sequence of $k+1$ symbols $s_0, s_1, \ldots, s_k$.

**Algorithm to go from an expression $e$ to its Gödel number (g.n.)**

1. Take the code number $c_i$ for each $s_i$.
2. Use $c_i$ as an exponent for the $(i+1)$th prime number $\pi_i$.
3. Multiply the results to get $\pi_0^{c_0} \pi_1^{c_1} \pi_2^{c_2} \cdots \pi_k^{c_k}$.

- $S$ has g.n. $2^{21}$.
- $SS0$ has g.n. $2^{21}3^{21}5^{17}$.
- $\exists y(SS + y) = SS0$ has g.n. $2^{11}3^{14}5^{13}7^{21}11^{21}13^{23}17^{4}19^{15}23^{19}29^{21}31^{21}37^{17}$!

**Algorithm to go from a g.n. to an expression $e$**

(i) Calculate the (unique) prime factorization of the g.n.
(ii) Find the sequence of exponents of the prime factors.
A proof in $N$ can be written as a sequence of wffs, hence encoded in a g.n.

**Algorithm to go from a sequence of expressions $e_0, e_1, ..., e_n$ to a g.n.**

1. Calculate the g.n. of each $e_i$.
2. Use $g_i$ as an exponent for the $(i+1)$th prime number $\pi_i$.
3. Multiply the results to get $\pi_0^{g_0} \pi_1^{g_1} \pi_2^{g_2} ... \pi_n^{g_n}$.

**Algorithm to go from a g.n. to a sequence of expressions**

(i) Find the sequence of exponents of the prime factors of the g.n.
(ii) Treat these exponents as g.n.s and take their prime factors.

- A proof in $N$ can be written as a sequence of wffs, hence encoded in a g.n.

**Ex:** Algorithm for rewriting a tree proof as a linear sequence of wffs.

(i) List trunk wffs first.
(ii) At a fork, take left branch, and continue listing wffs that have not yet appeared in the sequence.
(iii) At the end of a branch, return to the last fork, take the right branch, and continue listing wffs.
(iv) Repeat (ii) and (iii) until all branches have been followed.
• Gödel numbers let us encode syntactic properties of the language $L_N$ in purely numerical properties of (relations between) of natural numbers.

<table>
<thead>
<tr>
<th>Syntactic property</th>
<th>Numerical relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Being a term of $L_N$.</td>
<td>$\text{Term}(n)$. Holds just when $n$ is the g.n. of a term of $L_N$.</td>
</tr>
<tr>
<td>Being an atomic wff of $L_N$.</td>
<td>$\text{Atom}(n)$. Holds just when $n$ is the g.n. of an atomic wff of $L_N$.</td>
</tr>
<tr>
<td>Being a wff of $L_N$.</td>
<td>$\text{Wff}(n)$. Holds just when $n$ is the g.n. of a wff of $L_N$.</td>
</tr>
<tr>
<td>Being a closed wff of $L_N$.</td>
<td>$\text{Sent}(n)$. Holds just when $n$ is the g.n. of a closed wff of $L_N$.</td>
</tr>
<tr>
<td>Being an axiom of $N$.</td>
<td>$\text{Ax}(n)$. Holds just when $n$ is the g.n. of an axiom of $N$.</td>
</tr>
<tr>
<td>Being a proof in $N$.</td>
<td>$\text{Prf}(m, n)$. Holds just when $m$ is the g.n. of a proof in $N$ of the closed wff with g.n. $n$.</td>
</tr>
</tbody>
</table>

**Claim 1:** All of the numerical relations in Table 1 are primitive recursive.

**What this means:**
• To say $\text{Term}(n)$ is primitive recursive is to say that there is a p.r. function that computes $\text{Term}(n)$; i.e., that tells us, for a given $n$, if $\text{Term}(n)$ holds.
• **Idea:** To show this, we have to find p.r. functions that encode the algorithm that goes from a g.n. to an expression of $L_N$, and we have to find p.r. functions that encode the algorithm that determines what a term is in $L_N$.
• **Note:** That $\text{Ax}(n)$ is primitive recursive demonstrates that $N$ is recursively axiomatizable.
Expressibility in $N$

- Let $\bar{n}$ be shorthand for the term $\text{SSS}...\text{S}0$ in $L_N$, where $\text{S}$ occurs $n$-times.

A $k$-place numerical relation $P$ is **expressible** in $N$ just when there is a wff $\varphi(v_1, \ldots, v_k)$ of $L_N$ with free occurances of $v_1, \ldots, v_k$, such that for any natural numbers $n_1, \ldots, n_k$,

if $n_1, \ldots, n_k$ stand in relation $P$ to each other, then $N \vdash \varphi(\bar{n}_1, \ldots, \bar{n}_k)$,

if $n_1, \ldots, n_k$ do not stand in relation $P$ to each other, then $N \vdash \neg \varphi(\bar{n}_1, \ldots, \bar{n}_k)$.

**Ex.** The 1-place numerical relation $ev(n)$ of being even is expressible in $N$.

- The wff of $L_N$ that expresses this is $\exists y (2 \times y = x)$, where $x$ occurs free.
- **Which means:** For any natural number $n$,

  if $n$ is even, then $N \vdash \exists y (2 \times v = \bar{n})$,

  if $n$ is not even, then $N \vdash \neg \exists y (2 \times v = \bar{n})$.

- So: To say $\text{Prf}(m, n)$ is expressible in $N$ is to say that there is a wff of $L_N$, call it $\mathcal{PF}(x, y)$ which says "$x$ is the g.n. of a proof in $N$ of the wff with g.n. $y$", such that, for any numbers $m, n$:

  if $\text{Prf}(m, n)$ holds, then $N \vdash \mathcal{PF}(\bar{m}, \bar{n})$,

  if $\text{Prf}(m, n)$ does not hold, then $N \vdash \neg \mathcal{PF}(\bar{m}, \bar{n})$.

**Claim 2:** A numerical relation is primitive recursive if and only if it is expressible in $N$. 
The Gödel Sentence of $N$

**Def.** The 2-place numerical relation $W(m, n)$ holds just when $m$ is the g.n. of a proof in $N$ of the wff $\varphi(\bar{n})$, obtained from the wff $\varphi(y)$ (in which $y$ occurs free) whose g.n. is $n$.

- **Claim:** $W(m, n)$ is primitive recursive.
  - **So:** There's a wff $\mathcal{W}(x, y)$ that expresses $W(m, n)$ in $N$.

**Def:** The **Gödel sentence** $\mathcal{G}$ is the wff $\forall x \neg \mathcal{W}(x, \bar{p})$, where $p$ is the g.n. of the wff $\mathcal{U}(y) =_{\text{def}} \forall x \neg W(x, y)$, in which $y$ occurs free.

$\mathcal{G}$ says: "There is no number $m$ such that $m$ is the g.n. of a proof in $N$ of $\mathcal{U}(\bar{p})$.

**But:** $\mathcal{U}(\bar{p})$ is just $\mathcal{G}!$

**So:** $\mathcal{G}$ says: "There is no proof in $N$ of $\mathcal{G}$.

**Claim 1:** $\mathcal{G}$ is true if and only if it is unprovable in $N$.

- If $\mathcal{G}$ is true, then "There is no proof of $\mathcal{G}$ in $N$" is true; hence $\mathcal{G}$ is unprovable in $N$.
- If $\mathcal{G}$ is unprovable, then there is no $m$ such that $m$ is the g.n. of a proof in $N$ of $\mathcal{G}$; so $\mathcal{G}$ is true.
Claim 2: If $N$ is sound, then $N$ is not negation complete.

- **Idea:** We will show that $\mathcal{G}$ is a wff of $L_N$ such that neither $N \vdash \mathcal{G}$ nor $N \vdash \neg \mathcal{G}$.

**Suppose:** $N$ is sound.

- **Then:** For any wff $\varphi$, if $N \not\models \varphi$, then $N \not\models \varphi$. 
  "If $\varphi$ is false, then $\varphi$ is not provable."
- **Now:** Suppose $N \vdash \mathcal{G}$.
- **Then:** $N \not\models \mathcal{G}$.
  *Suppose $\mathcal{G}$ could be proved in $N$. Since $\mathcal{G}$ is provable if and only if it is false (Claim 1.)*
- **So:** $N \not\models \mathcal{G}$.
- **Thus:** $N \not\models \mathcal{G}$.
  From soundness of $N$.
- **So:** $N \not\models \neg \mathcal{G}$.
  *Or $\neg \mathcal{G}$ is false.*
- **So:** $N \not\models \neg \mathcal{G}$.
  *From soundness of $N$.

- **Thus:** $\mathcal{G}$ is a wff of $L_N$ such that neither $N \vdash \mathcal{G}$ nor $N \vdash \neg \mathcal{G}$.

**Thus:** $N$ is not negation complete.

- **Note:** This is a "semantic" proof of $N$'s negation incompleteness (it relies on the notion of soundness).

- What about a purely "syntactic" proof of $N$'s negation incompleteness?
Claim 3: If $N$ is consistent, then there is a wff $\varphi$ of $L_N$ such that $N \nvdash \varphi$; and if $N$ is $\omega$-consistent, then $N \nvdash \neg \varphi$.

- **First**: Show that if $N$ is consistent, then $N \nvdash G$.

  **Suppose**: $G$ is provable in $N$.  
  Or $N \vdash \forall x \neg W(x, \bar{p})$.

  - **Then**: There is a natural number $m$ such that $m$ is the g.n. of a proof in $N$ of $G$.

  - **So**: The 2-place numerical relation $W(m, p)$ holds, where $p$ is the g.n. of the wff $U(y)$.

  - **So**: $N \vdash W(\overline{m}, \bar{p})$.

  - **Now**: $G$ entails $\neg W(\overline{m}, \bar{p})$.

  - **So**: Since $N \vdash G$, we have $N \vdash \neg W(\overline{m}, \bar{p})$.

  **Thus**: $N$ is inconsistent. (There is a wff $W(\overline{m}, \bar{p})$ such that both it and its negation are theorems of $N$.)
Claim 3: If $N$ is consistent, then there is a wff $\varphi$ of $L_N$ such that $N \not\vdash \varphi$; and if $N$ is $\omega$-consistent, then $N \not\vdash \neg \varphi$.

Def: A theory $T$ with $L_N$ as its language is **$\omega$-inconsistent** just when, for some open wff $\varphi(x)$, $T$ can prove each $\varphi(\bar{m})$ and $T$ can also prove $\neg \forall x \varphi(x)$ (i.e., $\exists x \neg \varphi(x)$).

- **Or**: $T$ can prove $\varphi$ for each natural number, and it can also prove $\neg \varphi$ for some natural number.
- **Now**: Show that if $N$ is $\omega$-consistent, then $N \not\vdash \neg G$.

**Suppose**: $N$ is $\omega$-consistent and $\neg G$ is provable in $N$.
- **Then**: $N \vdash \neg \forall x \neg W(x, \bar{p})$. Or: $N \vdash \exists x \neg \neg W(x, \bar{p})$. (*)
- **Now**: If $N$ is $\omega$-consistent, then it is consistent.
- **So**: $G$ is not provable.
- **So**: For any number $m$, $m$ is not the g.n. of a proof in $N$ of $G$.
- **So**: The 2-place numerical relation $W(m, p)$ does not hold, where $p$ is the g.n. of the wff $U(y)$.
- **Which means**: $N \vdash \neg W(\bar{m}, \bar{p})$. (Since $W(m, n)$ is expressible in N.) (**)
- **Note**: (*) and (**) entail $N$ is $\omega$-inconsistent.
- **Thus**: $\neg G$ must be unprovable in $N$.

**But**: Claim 3 still doesn't quite say, "If $N$ is consistent, then $N$ is negation complete."
• Can show the following:

I. If $N$ is consistent, recursively axiomatizable, and negation complete, then it is recursively decidable.

II. If $N$ is consistent and recursively axiomatizable, then it is not recursively decidable.

So: If $N$ is consistent and recursively axiomatizable, then it is not negation complete.

Proof of (I). Show how to construct a mechanical procedure that decides, for any wff $\varphi$ of $L_N$, whether $\varphi$ is a theorem of $N$.

Suppose: $N$ is consistent, recursively axiomatizable, and negation complete.
- Let $\varphi$ be an arbitrary wff of $L_N$.
- Generate a list of $N$'s theorems. \textit{Since $N$ is recursively axiomatizable.}
- Either $\varphi$ or $\neg \varphi$ must appear. \textit{Because $N$ is negation complete.}
- If $\varphi$ appears, then $\varphi$ is a theorem. \textit{Because $N$ is consistent.}
- If $\neg \varphi$ appears, then $\varphi$ is not a theorem.

How to mechanically generate a list of $N$'s theorems
- For each number $n$, check all numbers $m$ to see if $\text{Prf}(m, n)$ holds.
- If it does hold, add the wff whose g.n. is $n$ to the list.

Note: This is different from having a mechanical procedure that determines, for any $\varphi$, whether it will ever turn up in the list!
**Proof of (II) If N is consistent and recursively axiomatizable, then it is not recursively decidable.**

**Suppose:** $N$ is recursively decidable. Then $N$ is recursively axiomatizable.

- **Now:** Show that $N$ is not consistent.

1. List all the 1-place recursive properties of numbers $P_0(n)$, $P_1(n)$, ... as recursive sets of numbers:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

   Each row represents the extension of the property labeled by that row:
   - Extension of $P_0$ is $\{1, \ldots\}$
   - Extension of $P_1 = \{0, 1, 2, \ldots\}$
   - Extension of $P_2 = \{1, 2, \ldots\}$

2. Define a 1-place property $D(n)$ by: $D(n)$ holds if and only if $P_n(n)$ does not hold.
   - Or: $D(n)$ holds if and only if $\neg P_n(\bar{n})$ is a theorem in $N$, where $P_n(x)$ expresses $P_n(n)$ in $N$.

3. **Claim:** $D(n)$ is a recursive property, so it must be in the list, say $D(n) = P_m(n)$.

   **Proof:** The following is a mechanical procedure that decides if a number $n$ has the property $D$:
   - (i) For any number $n$, check if $\neg P_n(\bar{n})$ is a theorem of $N$ (possible since $N$ is recursively decidable).
   - (ii) If so, then $D(n)$ holds.
   - (iii) If not, then $D(n)$ doesn't hold.
**Proof of (II)** If $N$ is consistent and recursively axiomatizable, then it is not recursively decidable.

**Suppose:** $N$ is recursively decidable. Then $N$ is recursively axiomatizable.

- **Now:** Show that $N$ is not consistent.

1. List all the 1-place recursive properties of numbers $P_0(n)$, $P_1(n)$, ... as recursive sets of numbers:

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>no</td>
<td>yes</td>
<td>no</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>...</td>
</tr>
<tr>
<td>2</td>
<td>no</td>
<td>yes</td>
<td>yes</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   *Each row represents the extension of the property labeled by that row:*
   - Extension of $P_0$ is $\{1, \ldots\}$
   - Extension of $P_1$ is $\{0, 1, 2, \ldots\}$
   - Extension of $P_2$ is $\{1, 2, \ldots\}$

2. Define a 1-place property $D(n)$ by: $D(n)$ holds if and only if $P_n(n)$ does not hold.
   - Or: $D(n)$ holds if and only if $\neg P_n(n)$ is a theorem in $N$, where $P_n(x)$ expresses $P_n(n)$ in $N$.

3. **Claim:** $D(n)$ is a recursive property, so it must be in the list, say $D(n) = P_m(n)$.

4. **Question:** Does $D(m)$ hold? (Does the number $m$ have the property $D$ that it labels?)
   - (a) $D(m)$ holds if and only if $\neg P_m(m)$ is a theorem in $N$.
   - (b) If $D(m)$ holds, then $P_m(m)$ is a theorem in $N$.
   - (c) If $D(m)$ doesn't hold, then $\neg P_m(m)$ is a theorem in $N$.

   - **Now:** (a) and (c) entail that $\neg P_m(m)$ is a theorem in $N$.
   - **So** (a) entails that $D(m)$ holds.
   - **But:** (b) then entails that $P_m(m)$ is a theorem in $N$.

   **Thus:** There's a wff $P_m(m)$ of $L_N$ such that both it and its negation are theorems in $N$. 