Gödel's 1st Incompleteness Theorem

Kurt Gödel

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 langauge),

(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

- A ***theorem*** of T is a *wff* of L_T that is provable in $T's$ proof system.
  - *Notation*: $T \vdash \varphi$ means "φ is a theorem of T".
- A ***logically valid*** *wff* of T is a *wff* of L_T that is true in all interpretations.
  - *Notation*: $T \vDash \varphi$ means "φ is a logically valid *wff* of T".
- T is ***sound*** just when every theorem of T is logically valid: For any *wff* φ of L_T, if $T \vdash \varphi$, then $T \vDash \varphi$.
- T is ***semantically complete*** just when every logically valid *wff* of T is a theorem of T: For any *wff* φ of L_T, if $T \vDash \varphi$, then $T \vdash \varphi$.

Two more syntactic notions:

- T is ***consistent*** just when, for any *wff* φ in L_T, it's not the case that both $T \vdash \varphi$ and $T \vdash \neg\varphi$.
- T is ***negation complete*** just when, for any *wff* φ 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, $\wedge$, $\vee$, $\neg$, (,) and the grammar and semantics of **PL**.
- Let the proof system be the **PL** tree rules.

> *Idea*: *Wffs* of L are only those *wffs* of **PL** that can be formed from P, Q, R using the **PL** connectives.

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

> *Idea*: In T_1, all tree proofs begin with ¬P at the top as given; in T_2, all tree proofs begin with ¬P, Q, ¬R at the top as givens.

- Both T_1 and T_2 are *sound* and *semantically complete* (since **PL** is).
- Both T_1 and T_2 are *consistent*.

Negation complete?

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

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

> *Moral*: *Semantic completeness* is distinct from *negation completeness.*
> - *T_1 is not negation complete, but uses a proof system (**PL** trees) that is semantically complete.*

Example:

- Let L consist of the alphabet P, Q, R, $\wedge$, $\vee$, $\neg$, (,) and the grammar and semantics of **PL**.
- Let the proof system be the **PL** tree rules.

Idea: *Wffs* of L are only those *wffs* of **PL** that can be formed from P, Q, R using the **PL** connectives.

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

Idea: In T_1, all tree proofs begin with $\neg$P at the top as given; in T_2, all tree proofs begin with $\neg$P, Q, $\neg$R at the top as givens.

- 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 $x^n + y^n = z^n$.

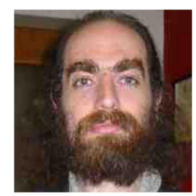
Pierre de Fermat

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



Supposedly proven by Grigori Perelman in 2003 after a century and $1million 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. $S(x)$ = successor of x.

(ii) Zero function. $Z(x) = 0$.

(iii) k-place identity function. $I^k{}_i(x_1, ..., x_k) = x_i \quad 1 \leq i \leq k$.

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:

Sum function. $+(x, y)$	*Product function.* $\times(x, y)$	*Factorial function.* $!(x)$
$+(x, 0) = x = I^1{}_1(x)$	$\times(x, 0) = 0 = Z(x)$	$!(0) = 1 = S(0)$
$+(x, S(y)) = S(+(x, y))$	$\times(x, S(y)) = +(\times(x, y), x)$	$!(S(y)) = \times(!(x), S(x))$

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.

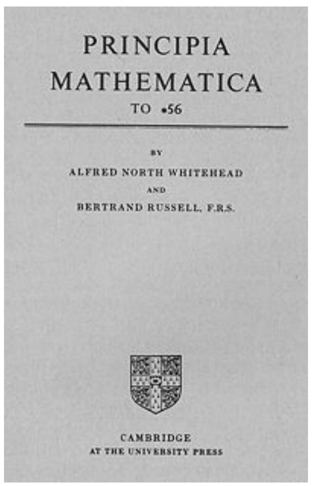
Alonzo Church

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.



A. N. Whitehead

Bertrand Russell

4. Aspects of the Proof

Giuseppe Peano

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

The Alphabet of L_N

0	individual constant
x, y, z, ..., v_k	individual variables ($k \geq 0$)
$=$	2-place predicate (identity)
S	1-place function (successor)
$+$, $\times$	2-place functions (sum, product)
$\wedge, \vee, \neg, \supset, \forall, \exists$, (,)	connectives, quantifiers, punctuation

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

- *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 $\mathbf{QL}^f$.

- 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 $\{\langle 0,1\rangle, \langle 1,2\rangle, \langle 2,3\rangle, \langle 3,4\rangle, ...\}$.
- 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)	$\forall x \neg(0 = Sx)$
(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 φ, if 0 has it, and if, for any number n, if n has it entails that the succesor 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:

$\wedge$	$\vee$	$\neg$	$\supset$	$\forall$	$\exists$	(	)	0	=	S	+	$\times$	x	y	z	$\cdots$
1	3	5	7	9	11	13	15	17	19	21	23	25	2	4	6	$\cdots$

- Let expression e in L_N be the sequence of k+1 symbols s_0, s_1, ..., 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 π_i.
3. Multiply the results to get $\pi_0^{c_0} \pi_1^{c_1} \pi_2^{c_2} \ldots \pi_k^{c_k}$.

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

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.

Algorithm to go from a sequence of expressions $e_0, e_1, \ldots, e_n$ 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 π_i.
3. Multiply the results to get $\pi_0^{g_0}\pi_1^{g_1}\pi_2^{g_2}\ldots\pi_n^{g_n}$.

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 1: Important Examples

Syntactic property	*Numerical relation*
Being a term of L_N.	$Term(n)$. Holds just when n is the g.n. of a term of L_N.
Being an atomic *wff* of L_N.	$Atom(n)$. Holds just when n is the g.n of an atomic *wff* of L_N.
Being a *wff* of L_N.	$Wff(n)$. Holds just when n is the g.n. of a *wff* of L_N.
Being a closed *wff* of L_N.	$Sent(n)$. Holds just when n is the g.n. of a closed *wff* of L_N.
Being an axiom of N.	$Ax(n)$. Holds just when n is the g.n. of an axiom of N.
Being a proof in N.	$Prf(m, n)$. Holds just when m is the g.n. of a proof in N of the closed *wff* with g.n. n.

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

What this means:

- To say $Term(n)$ is primitive recursive is to say that there is a p.r. function that computes $Term(n)$; i.e., that tells us, for a given n, if $Term(n)$ 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 $Ax(n)$ is primitive recursive demonstrates that N is recursively axiomatizable.

Expressibility in N

- Let $\bar{\mathsf{n}}$ be shorthand for the term $\mathsf{SSS...S0}$ in L_N, where S occurs n-times.

A k-place numerical relation P is ***expressible*** in N just when there is a *wff* $\varphi(\mathsf{v}_1, ..., \mathsf{v}_k)$ of L_N with free occurances of $\mathsf{v}_1, ..., \mathsf{v}_k$, such that for any natural numbers $n_1, ..., n_k$,

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

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

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

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

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

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

- *So*: To say $Prf(m, n)$ is expressible in N is to say that there is a *wff* of L_N, call it $\mathcal{PF}(\mathsf{x}, \mathsf{y})$ 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 $Prf(m, n)$ holds, then $N \vdash \mathcal{PF}(\bar{\mathsf{m}}, \bar{\mathsf{n}})$,

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

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* $\varphi(\bar{\mathsf{n}})$, obtained from the *wff* $\varphi(\mathsf{y})$ (in which y occurs free) whose g.n. is n.

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

Def: The ***Gödel sentence*** $\mathcal{G}$ is the *wff* $\forall \mathsf{x} \neg \mathcal{W}(\mathsf{x}, \bar{\mathsf{p}})$, where p is the g.n. of the *wff* $\mathcal{U}(\mathsf{y}) =_{\text{def}} \forall \mathsf{x} \neg \mathrm{W}(\mathsf{x}, \mathsf{y})$, 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 $\mathcal{U}(\bar{\mathsf{p}})$."
But: $\mathcal{U}(\bar{\mathsf{p}})$ 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* φ, if $N \nvDash \varphi$, then $N \nvdash \varphi$.	*"If φ is false, then φ is not provable."*
- *Now*: Suppose $N \vdash \mathcal{G}$.	*Suppose $\mathcal{G}$ could be proved in N.*
- *Then*: $N \nvDash \mathcal{G}$.	*Since $\mathcal{G}$ is provable if and only if it is false (Claim 1.)*
- *So*: $N \nvdash \mathcal{G}$.	*From soundness of N.*
- *Thus*: $N \vDash \mathcal{G}$.	*Claim 1.*
- *So*: $N \nvDash \neg\mathcal{G}$.	*Or $\neg\mathcal{G}$ is false.*
- *So*: $N \nvdash \neg\mathcal{G}$.	*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* φ of L_N such that $N \nvdash \varphi$; and if N is *ω-consistent*, then $N \nvdash \neg\varphi$.

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

Suppose: $\mathcal{G}$ is provable in N.	*Or* $N \vdash \forall \mathsf{x} \neg \mathcal{W}(\mathsf{x}, \bar{\mathsf{p}})$.
- *Then*: There is a natural number m such that m is the g.n. of a proof in N of $\mathcal{G}$.	
- *So*: The 2-place numerical relation $W(m, p)$ holds, where p is the g.n. of the *wff* $\mathcal{U}(\mathsf{y})$.	*Recall that* $\mathcal{U}(\bar{\mathsf{p}})$ *is* $\mathcal{G}$.
- *So*: $N \vdash \mathcal{W}(\bar{\mathsf{m}}, \bar{\mathsf{p}})$.	*Since* $W(m, n)$ *is expressible in* N.
- *Now*: $\mathcal{G}$ entails $\neg\mathcal{W}(\bar{\mathsf{m}}, \bar{\mathsf{p}})$.	*Universal instantiation.*
- *So*: Since $N \vdash \mathcal{G}$, we have $N \vdash \neg\mathcal{W}(\bar{\mathsf{m}}, \bar{\mathsf{p}})$.	
Thus: N is inconsistent. (There is a *wff* $\mathcal{W}(\bar{\mathsf{m}}, \bar{\mathsf{p}})$ such that both it and its negation are theorems of N.)	

Claim 3: If N is *consistent*, then there is a *wff* φ of L_N such that $N \nvdash \varphi$; and if N is ω-*consistent*, then $N \nvdash \neg\varphi$.

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

- *Or*: T can prove φ for each natural number, and it can also prove $\neg\varphi$ for some natural number.
- *Now*: Show that if N is ω-*consistent*, then $N \nvdash \neg\mathcal{G}$.

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

Thus: $\neg\mathcal{G}$ 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*, *recursviely 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* φ of L_N, whether φ is a theorem of N.

Suppose: N is consistent, recursively axiomatizable, and negation complete.

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

How to mechanically generate a list of N's theorems
- For each number n, check all numbers m to see if *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 φ, 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:

	0	1	2	$\cdots$
0	no	yes	no	$\cdots$
1	yes	yes	yes	$\cdots$
2	no	yes	yes	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

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

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* $\neg\mathcal{P}_n(\bar{\mathsf{n}})$ is a theorem in N, where $\mathcal{P}_n(\mathsf{x})$ 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 $\neg\mathcal{P}_n(\bar{\mathsf{n}})$ 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:

	0	1	2	⋯
0	no	yes	no	⋯
1	yes	yes	yes	⋯
2	no	yes	yes	⋯
⋮	⋮	⋮	⋮	⋱

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

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* $\neg\mathcal{P}_n(\bar{\mathsf{n}})$ is a theorem in N, where $\mathcal{P}_n(\mathsf{x})$ 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 $\neg\mathcal{P}_m(\bar{\mathsf{m}})$ is a theorem in N. — By definition of D.

   (b) If $D(m)$ holds, then $\mathcal{P}_m(\bar{\mathsf{m}})$ is a theorem in N.

   (c) If $D(m)$ doesn't hold, then $\neg\mathcal{P}_m(\bar{\mathsf{m}})$ is a theorem in N.

   (b), (c): Because $D(n) = P_m(n)$ is recursive, and hence expressible in N by the *wff* $\mathcal{P}_m(\mathsf{x})$

- *Now*: (a) and (c) entail that $\neg\mathcal{P}_m(\bar{\mathsf{m}})$ is a theorem in N.
- *So* (a) entails that $D(m)$ holds.
- *But*: (b) then entails that $\mathcal{P}_m(\bar{\mathsf{m}})$ is a theorem in N.

Thus: There's a *wff* $\mathcal{P}_m(\bar{\mathsf{m}})$ of L_N such that both it and its negation are theorems in N.