Arithmetic and Gödel's theorem

Ivano Ciardelli

Introduction to Computability

The problem of foundations: some background

- In the beginning of the 20th century, there was much turmoil concerning the foundations of mathematics.
- Geometry and calculus were reduced to the newborn theory of sets.
- But what are sets? How can we know which sets exist?
- Mathematicians disagreed about these questions and about what set-theoretic principles were acceptable in a proof.
- Worse, contradictions were discovered in early versions of set theory.
- Example: Russell's paradox. Let $R = \{x \mid x \notin x\}$.

$$R \in R \iff R \notin R$$

- Mathematics is supposed to be the realm of certainty.
  But now it appeared to rest on shaky and unclear foundations.

- Hilbert's solution: mathematics is not concerned with what objects are, but only with what properties they have.

  > *One must be able to say at all times—instead of points, straight lines, and planes—tables, chairs, and beer mugs.*

- Mathematics proceeds by postulating some statements (axioms), and deriving other statements (theorems).

- We never manipulate infinite objects (sets, real numbers, etc.) but only statements about these objects, which are finite entities.

- It is at this higher level that certainty can be recovered.

Mathematical certainty recovered

- thanks to mathematical logic, we can formalize mathematical statements without ambiguity;
- we can also lay out precise rules for making inferences;
- setting the axioms of a theory is a matter of definition;
- so, there is no room for disagreement over whether something counts as a theorem or not.

The completeness question

Can we make our axiomatization just rich enough to capture all the properties of the structure that it is supposed to characterize?

- how do we know that we have enough axioms?
- how do we know that we don't have too many?

We can consider this question in many settings.

The most fundamental mathematical setting is arithmetic,
the theory of natural numbers.

We are going to explore the above question for arithmetic; first, we need to see how arithmetic can be regimented in a formal language.

The language of arithmetic

Terms

- A term is an expression formed by means of 0, variables $x, y, z \ldots$, a unary function symbol S, and binary function symbols $+$ and $\cdot$.
- Notation: $\overline{n} := S(\ldots S(0))$ where S occurs n times.
- Examples: $\overline{3}$, $\overline{2} + \overline{1}$, $x \cdot y$, $\overline{5} + (\overline{3} \cdot z)$

Atomic formulas

- An atomic formula is an expression $t = t'$, where t and t' are terms.
- Examples: $\overline{1} = \overline{2}$, $\overline{1} \cdot x = \overline{5}$, $x + y = y + x$

Formulas

A formula is an expression built up from atomic formulas by means of connectives ($\neg, \wedge, \vee, \rightarrow$) and quantifiers ($\forall x, \exists x$, for x a variable).

Examples

- $\neg(x = 0)$
- $\forall x(x = 0 \ \rightarrow \ \neg(x = \overline{1}))$
- $\forall x \exists y(x \cdot y = 1)$

Notation We write $t \neq t'$ for $\neg(t = t')$

Sentences

A sentence is a formula that contains no free variables.

- By definition, a sentence is merely a string of symbols.
- However, a sentence can be said to be true or false when we fix an interpretation for the symbols that occur in it.
- We say that a sentence ψ is true in $\mathbb{N}$ (notation: $\mathbb{N} \models \psi$) if it is true when:
  - 0 is interpreted as the number zero
  - $S, +, \cdot$ are interpreted as successor, sum, and product on $\mathbb{N}$
  - quantifiers are interpreted as ranging over $\mathbb{N}$
- Examples:
  - $\mathbb{N} \models \exists x(x + x = 0)$
  - $\mathbb{N} \not\models \exists x(S(x) = 0)$

Looking for truth

- Our goal is to find out which sentences are true in $\mathbb{N}$.
- How can we do this?
- Axiomatic method: identify some sentences that are clearly true, take them as axioms, and use logic to infer consequences.
- If A is a set of sentences, let us write $A \vdash \psi$ if ψ can be inferred from the sentences in A by first-order logic (more later).

Complete axiomatizations?

We would like to identify a set of sentences A that satisfies:

- soundness: $A \vdash \psi \ \Rightarrow\ \mathbb{N} \models \psi$
- completeness: $\mathbb{N} \models \psi \ \Rightarrow\ A \vdash \psi$

The main candidate: Peano Arithmetic

The axioms of Peano Arithmetic (PA) are the following sentences.

- Zero is not a successor: $\forall x(0 \neq Sx)$
- Successor is injective: $\forall x \forall y((x \neq y) \rightarrow (Sx \neq Sy))$
- Recursive definition of sum:

$$\forall x(x + 0 = x)$$
$$\forall x(x + Sy = S(x + y))$$

- Recursive definition of product:

$$\forall x(x \cdot 0 = 0)$$
$$\forall x(x \cdot Sy = (x \cdot y) + x)$$

- Induction scheme. For any formula $\psi(\overline{x}, y)$, we have the axiom:

$$\forall \overline{x}(\psi(\overline{x}, 0) \wedge \forall y(\psi(\overline{x}, y) \rightarrow \psi(\overline{x}, Sy)) \rightarrow \forall y \psi(\overline{x}, y))$$

PA is quite powerful

If we try to build up arithmetic by making logical deductions in PA, we find that theorem after theorem, all the traditional results can be derived.

Question: is PA complete?

Can all true sentences be deduced by logic from the axioms of PA?
If not, what axioms need to be added to obtain a complete system?

We will see that PA is not complete. Worse, it cannot be completed! To see why, let us go back to our main question above.

The axiomatization question

Is it possible to identify a set of sentences A that satisfies:

- soundness: $A \vdash \psi \;\Rightarrow\; \mathbb{N} \models \psi$
- completeness: $\mathbb{N} \models \psi \;\Rightarrow\; A \vdash \psi$

A trivial answer

Sure! Just take $A = \{\psi \,|\, \mathbb{N} \models \psi\}$

- This is pointless: we don't know what the true sentences are.
  The whole point of an axiomatization is finding out.
- We want to know what the axioms of our system are.
- I.e., we should be able to tell whether or not a sentence is an axiom.
- Intuitively, the predicate "being an axiom" must be decidable.
- But what does this mean, exactly?

Coding formulas

Similarly to what we did for programs, we can define a coding of formulas as natural numbers:

- given a formula ψ, we can compute its code $\ulcorner \psi \urcorner$
- given a number n, we can compute the formula ψ_n that it encodes

The idea of coding formulas as numbers is due to Gödel.
For this reason, $\ulcorner \psi \urcorner$ is also called the Gödel number of ψ

Coding predicates

With a predicate P of formulas we associate a predicate M_P of numbers such that:

$$P(\psi) \iff M_P(\ulcorner \psi \urcorner)$$

Computability on formulas

We can then transfer notions of computability theory to formulas:

- we say that predicate $P(\psi)$ is decidable if $M_P(x)$ is decidable
- we say that predicate $P(\psi)$ is p.d. if the predicate $M_P(x)$ is p.d.
- we call a set of formulas $\mathcal{F}$ recursive if "$\psi \in \mathcal{F}$" is decidable
- we call a set of formulas $\mathcal{F}$ r.e. if "$\psi \in \mathcal{F}$" is p.d.

Now we can formulate the relevant constraint on axiomatization.

The axiomatization question, revisited

Is it possible to identify a recursive set of sentences A that satisfies:

- soundness: $A \vdash \psi \ \Rightarrow \ \mathbb{N} \models \psi$
- completeness: $\mathbb{N} \models \psi \ \Rightarrow \ A \vdash \psi$

We can also ask other crucial questions about arithmetic truth.

How computable is arithmetic truth?
Consider the predicate: *True*$(\psi) =$ "$\mathbb{N} \models \psi$"

- is *True* decidable?

  Is there a procedure to tell if a statement of arithmetic is true?

- is *True* partially decidable?

  Is there a procedure to enumerate the true statements of arithmetic?

These are the questions that we will address first.

Definition

A predicate $M(x_1,\ldots,x_n)$ of numbers is defined by formula $\chi_M(x_1,\ldots,x_n)$ if for all numbers $n_1,\ldots,n_k$:

$$\mathbb{N} \models \chi_M(\overline{n_1},\ldots,\overline{n_k}) \iff M(n_1,\ldots,n_k) \text{ holds}$$

We say that M is definable if it is defined by some formula.

Predicate	Formula
$x \leq y$	

Definition

A predicate $M(x_1, \ldots, x_n)$ of numbers is defined by formula $\chi_M(x_1, \ldots, x_n)$ if for all numbers $n_1, \ldots, n_k$:

$$\mathbb{N} \models \chi_M(\overline{n_1}, \ldots, \overline{n_k}) \iff M(n_1, \ldots, n_k) \text{ holds}$$

We say that M is definable if it is defined by some formula.

Predicate	Formula
$x \leq y$	$\exists z(x + z = y)$

Definition

A predicate $M(x_1,\ldots,x_n)$ of numbers is defined by formula $\chi_M(x_1,\ldots,x_n)$ if for all numbers $n_1,\ldots,n_k$:

$$\mathbb{N} \models \chi_M(\overline{n_1},\ldots,\overline{n_k}) \iff M(n_1,\ldots,n_k) \text{ holds}$$

We say that M is definable if it is defined by some formula.

Predicate	Formula
$x \leq y$	$\exists z(x+z=y)$
even(x)	

Definition

A predicate $M(x_1,\ldots,x_n)$ of numbers is defined by formula $\chi_M(x_1,\ldots,x_n)$ if for all numbers $n_1,\ldots,n_k$:

$$\mathbb{N} \models \chi_M(\overline{n_1},\ldots,\overline{n_k}) \quad \Longleftrightarrow \quad M(n_1,\ldots,n_k) \text{ holds}$$

We say that M is definable if it is defined by some formula.

Predicate	Formula
$x \leq y$	$\exists z(x + z = y)$
even(x)	$\exists y(x = \overline{2} \cdot y)$

Definition

A predicate $M(x_1,\ldots,x_n)$ of numbers is defined by formula $\chi_M(x_1,\ldots,x_n)$ if for all numbers $n_1,\ldots,n_k$:

$$\mathbb{N} \models \chi_M(\overline{n_1},\ldots,\overline{n_k}) \iff M(n_1,\ldots,n_k) \text{ holds}$$

We say that M is definable if it is defined by some formula.

Predicate	Formula
$x \leq y$	$\exists z(x+z=y)$
even(x)	$\exists y(x=\overline{2}\cdot y)$
prime(x)	

Definition

A predicate $M(x_1,\ldots,x_n)$ of numbers is defined by formula $\chi_M(x_1,\ldots,x_n)$ if for all numbers $n_1,\ldots,n_k$:

$$\mathbb{N} \models \chi_M(\overline{n_1},\ldots,\overline{n_k}) \iff M(n_1,\ldots,n_k) \text{ holds}$$

We say that M is definable if it is defined by some formula.

Predicate	Formula
$x \leq y$	$\exists z(x + z = y)$
even(x)	$\exists y(x = \overline{2} \cdot y)$
prime(x)	$\neg\exists y\exists z((y \neq \overline{1}) \wedge (z \neq \overline{1}) \wedge (x = y \cdot z))$

Definition

We say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is defined by a formula $\chi_f(x_1, \ldots, x_k, y)$ in case it defines the predicate "$f(x_1, \ldots, x_k) = y$", that is, if:

$$\mathbb{N} \models \chi_f(\overline{n_1}, \ldots, \overline{n_k}, \overline{m}) \quad \Longleftrightarrow \quad f(n_1, \ldots, n_k) = m$$

Predicate	Formula
Successor $S(x)$	

Definition

We say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is defined by a formula $\chi_f(x_1, \ldots, x_k, y)$ in case it defines the predicate "$f(x_1, \ldots, x_k) = y$", that is, if:

$$\mathbb{N} \models \chi_f(\overline{n_1}, \ldots, \overline{n_k}, \overline{m}) \iff f(n_1, \ldots, n_k) = m$$

Predicate	Formula
Successor $S(x)$	$y = S(x)$

Definition

We say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is defined by a formula $\chi_f(x_1,\ldots,x_k,y)$ in case it defines the predicate "$f(x_1,\ldots,x_k) = y$", that is, if:

$$\mathbb{N} \models \chi_f(\overline{n_1},\ldots,\overline{n_k},\overline{m}) \quad \Longleftrightarrow \quad f(n_1,\ldots,n_k) = m$$

Predicate	Formula
Successor $S(x)$	$y = S(x)$
Zero $Z(x)$	

Definition

We say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is defined by a formula $\chi_f(x_1, \ldots, x_k, y)$ in case it defines the predicate "$f(x_1, \ldots, x_k) = y$", that is, if:

$$\mathbb{N} \models \chi_f(\overline{n_1}, \ldots, \overline{n_k}, \overline{m}) \iff f(n_1, \ldots, n_k) = m$$

Predicate	Formula
Successor $S(x)$	$y = S(x)$
Zero $Z(x)$	$y = 0$

Definition

We say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is defined by a formula $\chi_f(x_1,\ldots,x_k,y)$ in case it defines the predicate "$f(x_1,\ldots,x_k) = y$", that is, if:

$$\mathbb{N} \models \chi_f(\overline{n_1},\ldots,\overline{n_k},\overline{m}) \quad \Longleftrightarrow \quad f(n_1,\ldots,n_k) = m$$

Predicate	Formula
Successor $S(x)$	$y = S(x)$
Zero $Z(x)$	$y = 0$
Projection $U_i^n(x_1,\ldots,x_n)$	$y = x_i$

Theorem

All computable functions are definable.

Proof idea

- any computable function is obtained from basic recursive functions by composition, recursion, and minimalization
- we proceed by induction on the construction of f
- we have seen that the basic recursive functions are definable
- we are left with the induction step:
  - if f is obtained by composition from defin. functions, it is definable
  - if f is obtained by recursion from definable functions, it is definable
  - if f is obtained by minimization from defin. functions, it is definable

Illustration: the case of composition

If $f = g \circ \langle h_1, \ldots, h_n \rangle$ and $g, h_1, \ldots, h_n$ are definable, then f is definable.

Suppose we have the following definitions:

- $g(\overline{x})$ defined by $\chi_g(\overline{x}, y)$
- $h_i(\overline{x})$ defined by $\chi_{h_i}(\overline{x}, y)$

Then, $f(\overline{x})$ is defined by the following formula $\chi_f(\overline{x}, y)$:

$$\exists z_1 \ldots \exists z_n (\chi_{h_1}(\overline{x}, z_1) \wedge \cdots \wedge \chi_{h_n}(\overline{x}, z_n) \wedge \chi_g(z_1, \ldots, z_n, y))$$

Corollary

All partially decidable predicates are definable.

Proof

- For simplicity, consider the unary predicate $M(x)$.
- If M is p.d., the semi-characteristic function s_M is computable.
- So by the previous theorem we have a formula $\chi_{s_M}(x, y)$ such that

$$s_M(n) = m \iff \mathbb{N} \models \chi_{s_M}(\overline{n}, \overline{m})$$

- This means that we have

$$M(n) \text{ holds } \iff s_M(n) = 1 \iff \mathbb{N} \models \chi_{s_M}(\overline{n}, \overline{1})$$

- Therefore, $M(x)$ is defined by the formula $\chi_M(x) := \chi_{s_M}(x, \overline{1})$.

Theorem (arithmetic truth is not p.d.)
The predicate *True* is not partially decidable.

Proof

- We know that the predicate $M_K(x) =$ "$\varphi_x(x)\downarrow$" is p.d.
- Therefore there is a formula $\chi_K(x)$ that defines $M_K(x)$
- This means that $\varphi_n(n)\downarrow \iff \mathbb{N} \models \chi_K(\overline{n})$
- So, we have:

$$\begin{aligned} \varphi_n(n)\uparrow &\iff \mathbb{N} \models \neg\chi_K(\overline{n}) \\ &\iff \mathrm{True}(\neg\chi_K(\overline{n})) \iff M_{\mathrm{True}}(\ulcorner\neg\chi_K(\overline{n})\urcorner) \end{aligned}$$

Proof, continued

- We have established: $M_{\overline{K}}(n) \iff M_{\text{True}}(\ulcorner\neg\chi_K(\overline{n})\urcorner)$
- Now, define a function $k : \mathbb{N} \to \mathbb{N}$ by letting $k(n) := \ulcorner\neg\chi_K(\overline{n})\urcorner$
- k is total, and it is computable by the following procedure. Given n,
  - write down the sentence $\neg\chi_K(\overline{n})$
  - compute the corresponding code $\ulcorner\neg\chi_K(\overline{n})\urcorner$
- We have that: $M_{\overline{K}}(n) \iff M_{\text{True}}(k(n))$
- So, $M_{\overline{K}} \leq M_{\text{True}}$. Since $M_{\overline{K}}$ is not p.d., neither is M_{True}
- By definition, this means that the predicate *True* is not p.d.

This means that the set of true arithmetic statements is not r.e.

By the characterization of r.e. sets as the ranges of total computable functions, we have:

Corollary

There exists no procedure to enumerate all true statements of arithmetic.

Moreover, since $\mathrm{True}_{\mathbb{N}}$ is not p.d., it is not decidable either. So, we have:

Corollary

There exists no procedure to determine if a sentence of arithmetic is true.

These results set inherent limits to our knowledge of arithmetic truth.

Back to axiomatization

- Let us now come back to the issue of axiomatization.
- To tackle it, we need some notions about proofs in first-order logic.
- Many different proof systems for classical first-order logic exist.
- The simplest proof system to state (though not to work with) is the Hilbert-style proof system.
- This system has many logical axioms and only two inference rules.

Inference rules

Modus Ponens	Generalization
$\dfrac{\psi \quad \psi \rightarrow \chi}{\chi}$	$\dfrac{\psi}{\forall x \psi}$

Axioms for propositional connectives

- Implication:
  - $\psi \to (\chi \to \psi)$
  - $(\psi \to (\chi \to \rho)) \to ((\psi \to \chi) \to (\psi \to \rho))$
- Conjunction:
  - $(\psi \wedge \chi) \to \psi$
  - $(\psi \wedge \chi) \to \chi$
  - $\psi \to (\chi \to (\psi \wedge \chi))$
- Disjunction:
  - $\psi \to (\psi \vee \chi)$
  - $\chi \to (\psi \vee \chi)$
  - $(\psi \to \rho) \to ((\chi \to \rho) \to (\psi \vee \chi \to \rho))$
- Negation:
  - $(\psi \to \rho) \to ((\psi \to \neg\rho) \to \neg\psi)$
  - $\neg\psi \to (\psi \to \chi)$
  - $\neg\neg\psi \to \psi$

Axioms for quantifiers

- Universal quantifier:
  - $\forall x \psi(x) \to \psi(t)$
  - $\forall x(\chi \to \psi(x)) \to (\chi \to \forall x \psi(x))$ where x does not occur in χ
- Existential quantifier:
  - $\psi(t) \to \exists x \psi(x)$
  - $\exists x(\psi(x) \to \chi) \to (\exists x \psi(x) \to \chi)$ where x does not occur in χ

Axioms for quantifiers

- Universal quantifier:
  - $\forall x \psi(x) \to \psi(t)$
  - $\forall x(\chi \to \psi(x)) \to (\chi \to \forall x \psi(x))$ where x does not occur in χ
- Existential quantifier:
  - $\psi(t) \to \exists x \psi(x)$
  - $\exists x(\psi(x) \to \chi) \to (\exists x \psi(x) \to \chi)$ where x does not occur in χ

Proposition

The predicate $\mathrm{LogAx}(\varphi) =$ "φ is a logical axiom" is decidable.

Definition (Logical proofs)
Let A be a set of formulas and ψ a formula. A proof of ψ from A is a sequence $\pi = \chi_1 \ldots \chi_n$ of formulas, such that:

- $\chi_n = \psi$
- for every $i \leq n$, either of the following holds:
  - χ_i is a logical axiom
  - $\chi_i \in A$
  - χ_i is obtained by modus ponens from χ_j, χ_k with $j, k < i$
  - χ_i is obtained by generalization from χ_j with $j < i$

We write $A \vdash \psi$ to mean that a proof of ψ from A exists.

Coding proofs as numbers

- Using our coding for formulas we can define a coding for proofs.
- If $\pi = \chi_1 \dots \chi_n$, then we let:

$$c_P(\pi) = c_* \langle \ulcorner \chi_1 \urcorner, \dots, \ulcorner \chi_n \urcorner \rangle$$

where c_* was our map for coding sequences of arbitrary length.

Proposition

If A is recursive, the following predicate is decidable:

$$\mathrm{Proof}_A(x, y) = \text{“}x \text{ codes a proof of } \psi_y \text{ from } A\text{”}$$

Proof. By Church's thesis. To decide the predicate, proceed as follows:

- decode y as a formula ψ_y
- decode x as a sequence $\pi = \xi_1 \ldots \xi_k$ of formulas
- check whether $\xi_k = \psi_y$
- for $i \leq k$, check if at least one of the following conditions hold:
  - ξ_i is a logical axiom
  - $\xi_i \in A$
  - χ_i is obtained by modus ponens from χ_j, χ_k with $j, k < i$
  - χ_i is obtained by generalization from χ_j with $j < i$
- if all these checks succeed, output 1; otherwise, output 0

Definition

If A is a set of formulas, let $\text{Thm}_A(\psi) =$ "$A \vdash \psi$"

Proposition

If A is recursive, Thm_A is partially decidable.

Proof

To show: the arithmetic predicate $M_{\text{Thm}_A}(x) =$ "$A \vdash \psi_x$" is p.d.
We have:

$$M_{\text{Thm}_A}(x) = \exists y \text{Proof}_A(y, x)$$

Since Proof_A is decidable, M_{Thm_A} is p.d., and so is Thm_A

Theorem (a version of Gödel's theorem)
No recursive set of axioms is sound and complete for arithmetic.
That is, there is no recursive set A such that $A \vdash \psi \iff \mathbb{N} \models \psi$.

Proof

- For a contradiction, suppose that such an A exists.
- This means that we have $\mathrm{Thm}_A(\psi) \iff \mathrm{True}(\psi)$.
- But this is impossible.
- Since A is recursive, we know that Thm_A is p.d.
- But we have proved above that True is not p.d.

Corollary (no recursive axiomatization decides all arithmetical statements)

Let A be a set of axioms which is recursive and sound.
Then there exists a sentence ψ such that $A \nvdash \psi$ and $A \nvdash \neg\psi$.

Proof

- Let A be recursive and sound.
- By the previous theorem, A is not complete.
- This means that for some ψ, $\mathbb{N} \models \psi$ but $A \nvdash \psi$.
- Since $\mathbb{N} \models \psi$, it follows that $\mathbb{N} \not\models \neg\psi$.
- By soundness, $A \nvdash \neg\psi$.
- We have thus concluded that $A \nvdash \psi$ and $A \nvdash \neg\psi$.

This is how Gödel puts it:

> *The development of mathematics toward greater precision has led [. . .] to the formalization of large tracts of it, so that one can prove any theorem using nothing but a few mechanical rules. The most comprehensive formal systems that have been set up hitherto are the system of Principia Mathematica on the one hand and the Zermelo-Fraenkel axiom system for set theory [. . .] on the other.*

Continued:

> *These two systems are so comprehensive that in them all methods of proof today used in mathematics are formalized, that is, reduced to a few axioms and rules of inference. One might therefore conjecture that these axioms and rules of inference are sufficient to decide any mathematical question that can at all be formally expressed in these systems.*
>
> *It will be shown below that this is not the case, that on the contrary there are in the two systems mentioned relatively simple problems in the theory of integers which cannot be decided on the basis of the axioms. This situation is not in any way due to the special nature of the systems [. . .], but holds for a very wide class of formal systems*

Continued:

> *These two systems are so comprehensive that in them all methods of proof today used in mathematics are formalized, that is, reduced to a few axioms and rules of inference. One might therefore conjecture that these axioms and rules of inference are sufficient to decide any mathematical question that can at all be formally expressed in these systems.*
>
> *It will be shown below that this is not the case, that on the contrary there are in the two systems mentioned relatively simple problems in the theory of integers which cannot be decided on the basis of the axioms. This situation is not in any way due to the special nature of the systems [...], but holds for a very wide class of formal systems*

Incompleteness is not an accidental feature of some formal systems,
but an inherent limitation to the axiomatic method,
which stems from the fact that the set of arithmetic truths is not r.e.

For a brighter view of axiomatic theories of arithmetic, let us now see some results about what *can* be done in PA.

Definition

We say that a function $f : \mathbb{N}^k \to \mathbb{N}$ is represented by a formula $\chi_f(\overline{x}, y)$ in PA in case whenever $f(n_1, \ldots, n_k) = m$ we have:

$$PA \vdash \chi_f(\overline{n_1}, \ldots, \overline{n_k}, \overline{m}) \land \forall x(\chi_f(\overline{n_1}, \ldots, \overline{n_k}, x) \to x = \overline{m})$$

Example

Addition is represented by the formula $x_1 + x_2 = y$ in PA, that is:

$$n_1 + n_2 = m \quad \Rightarrow \quad PA \vdash \overline{n_1} + \overline{n_2} = \overline{m}$$

This is true because PA includes the recursive definition of addition.

Theorem
Every computable function f is representable in PA.

Proof. By induction on the definition of f as a recursive functions. (The details are quite complicated.)

This is a powerful result:

- we saw above that the language of PA has the resources to define all computable functions;
- now we know that the axioms of PA provide the resources to compute the values of these functions, whenever they are defined.

Definition

We say that a predicate $M(x_1, \ldots, x_k)$ of numbers is represented by a formula $\chi_M(x_1, \ldots, x_k)$ in PA if for all numbers $n_1, \ldots, n_k$:

- $M(n_1, \ldots, n_k)$ does hold $\Rightarrow$ $PA \vdash \chi_M(\overline{n_1}, \ldots, \overline{n_k})$
- $M(n_1, \ldots, n_k)$ doesn't hold $\Rightarrow$ $PA \vdash \neg\chi_M(\overline{n_1}, \ldots, \overline{n_k})$

Intuitively, M is representable in PA if PA decides M.

Definition

We say that a predicate $M(x_1, \ldots, x_k)$ of numbers is semi-represented by a formula $\chi_M(x_1, \ldots, x_k)$ in PA if for all numbers $n_1, \ldots, n_k$:

$$M(n_1, \ldots, n_k) \text{ holds} \iff PA \vdash \chi_M(\overline{n_1}, \ldots, \overline{n_k})$$

Intuitively, M is representable in PA if PA semi-decides M.

Theorem
All decidable predicates are representable in PA.

Proof

- For simplicity, suppose $M(x)$ is unary.
- If M is decidable, then c_M is computable.
- So, we have a formula $\chi_{c_M}(x, y)$ which represents c_M.
- We claim that the formula $\chi_{c_M}(x, \overline{1})$ represents M.
- If $M(n)$ holds, $c_M(n) = 1$; since χ_{c_M} represents c_M, $PA \vdash \chi_{c_M}(\overline{n}, \overline{1})$
- If $M(n)$ doesn't hold, $c_M(n) = 0$. We reason as follows:
  - since χ_{c_M} represents c_M, $PA \vdash \chi_{c_M}(\overline{n}, \overline{0}) \wedge \forall x(\chi_{c_M}(\overline{n}, x) \rightarrow x = \overline{0})$
  - by the rules of logic, this implies $PA \vdash \neg(\overline{1} = \overline{0}) \rightarrow \neg\chi_{c_M}(\overline{n}, \overline{1})$
  - since $PA \vdash \overline{1} \neq \overline{0}$, it follows $PA \vdash \neg\chi_{c_M}(\overline{n}, \overline{1})$
- Thus, M is representable in PA.

Theorem
All partially decidable predicates are semi-representable in PA.

Proof
Similar to the one of the previous theorem.

Again, these results tell us that PA is quite powerful, having the tools to:

- decide every decidable predicate
- semi-decide every partially decidable predicate.

Theorem

The predicate $\mathrm{Thm}_{\mathrm{PA}}(\psi) =$ "$\mathrm{PA} \vdash \psi$" is undecidable.

Proof

- We have to show that $M_{\mathrm{Thm}_{\mathrm{PA}}}(x) =$ "$\mathrm{PA} \vdash \psi_x$" is undecidable.
- We know that $M_K(x) =$ "$\varphi_x(x)\downarrow$" is p.d.
- So, we have a formula $\chi_K(x)$ which semi-represents M_K in PA.
- Let $k : \mathbb{N} \to \mathbb{N}$ be the function $k(n) = \ulcorner \chi_K(\overline{n}) \urcorner$.
- k is total and computable (see the argument by CT given above)
- Claim: $M_K(n) \iff M_{\mathrm{Thm}_{\mathrm{PA}}}(k(n))$
  - If $M_K(n)$ holds, then by semi-representability $PA \vdash \chi_K(\overline{n})$; since $k(n) = \ulcorner \chi_K(\overline{n}) \urcorner$, the predicate $M_{\mathrm{Thm}_{\mathrm{PA}}}(k(n))$ holds.
  - If $M_K(n)$ doesn't hold, then by semi-representability $PA \nvdash \chi_K(\overline{n})$; since $k(n) = \ulcorner \chi_K(\overline{n}) \urcorner$, the predicate $M_{\mathrm{Thm}_{\mathrm{PA}}}(k(n))$ doesn't hold.
- Hence, $M_K \leq M_{\mathrm{Thm}_{\mathrm{PA}}}$. Since M_K is undecidable, so is $M_{\mathrm{Thm}_{\mathrm{PA}}}$.

A summary of the situation

What we can do in PA, and in any recursive extension of PA

- decide whether something counts as a proof
- systematically enumerate all the theorems
- compute any computable function
- decide any decidable predicate
- semi-decide any partially decidable predicate

What we cannot do in PA, or in any recursive extension of PA

- deduce all true statements of arithmetic
- decide all statements of arithmetic
- decide whether a statement is a theorem

Some concluding remarks

On the axiomatic method

- The axiomatic method is still our best tool to achieve clarity and certainty in mathematics.
- But the incompleteness theorem forces us to work with systems which are always partial—which always leave some issues undecided.
- Of course, these issues will be decided by stronger systems. But how to choose between these systems?

On the role of computability

- The notions of computability theory are central to logic and to the foundations of mathematics.
- By applying techniques from computability theory, such as reduction or the diagonal method, some deep results can be readily obtained.