Karlsruhe Institute of Technology

Formale Systeme II: Theorie

Gödel's Incompleteness Theorem

SS 2022

Prof. Dr. Bernhard Beckert · Dr. Mattias Ulbrich

Literature

Dirk W. Hoffmann
(Hochschule Karlsruhe)

Die Gödel'schen Unvollständigkeitssätze

Springer-Verlag, 2013

`http://link.springer.com/book/10.1007%2F978-3-8274-3000-7`

Available for download in KIT network

Gödel's (In)completeness Theorems

Completeness Theorem, 1929

There is a complete and correct calculus for validity in first order logic.

First Incompleteness Theorem, 1931

There is **no** complete and correct calculus for validity in arithmetic over natural numbers.

Second Incompleteness Theorem, 1931

No consitent formal system which contains natural arithmetic can be used to prove its own consistency.

Starting Point

- **Logic basis** $\{\forall, \rightarrow, \neg\}$ for first order predicate logic
- **Signature** $\Sigma_{\mathcal{N}} = (\{0, 1, +, \cdot\}, \emptyset, \alpha)$ with equality
- **Standard model** $\mathcal{N} = (\mathbb{N}, I_{\mathcal{N}})$ of natural arithmetic
  (no approximation, no non-standard models)
- **Calculus** C: Any first-order calculus for $\mathcal{N}$.
  For instance: Hilbert Calculus + (first order) Peano axioms
  (axiom formulas, axiom schemas and the modus ponens rule)

Goal: Keep the vocabularies as small as possible.

Formal Preliminaries

Variable Assignment

$\mathcal{N}, x \mapsto v \models \varphi$ means "φ is true in natural numbers if free variable x has the value v."

Substitution

For a formula $\varphi \in Fml$ with exactly one free variable x and a term t, we write $\varphi[t]$ to denote the formula arising if x is substituted by t in φ. If this substitution is collision-free.

Literals

The signature $\Sigma_{\mathcal{N}}$ does not contain the natural numbers (but 0 and 1). Any number $n \in \mathbb{N}$ can be used in formulas via its syntactical embedding $\bar{n}$:

$$\bar{n} := \underbrace{1 + 1 + \cdots + 1}_{n \text{ times}} \text{ for all } n \in \mathbb{N}$$

Proof Plan

1. **Arithmetic Representation:**
   There is a formula *bwb* (*beweisbar*) over $\Sigma_{\mathcal{N}}$ that formalises provability of any formula $\varphi \in Fml_{\mathcal{N}}$.

   $$\mathcal{N}, x \mapsto \ulcorner\varphi\urcorner \models bwb \iff \vdash_C \varphi$$

   We say: Provability is arithmetically representable.

2. **Diagonalisation:**
   Formalise "I cannot be proved!"
   and have $\mathcal{N} \models \vartheta \iff \nvdash_C \vartheta$

Arithmetic Representation

Gödelisation (sequences)

Godelisation

There is a computable, injection $\ulcorner\cdot\urcorner : \mathbb{N}^* \to \mathbb{N}$
with computable inverse function $\alpha : \mathbb{N} \times \mathbb{N} \nrightarrow \mathbb{N}$.

Let $\pi(k)$ denote the k-th prime number. ($\pi(1) = 2$, $\pi(20) = 71$, …)

Define

$$\ulcorner(a_1, a_2, \ldots, a_n)\urcorner := \pi(1)^{1+a_1} \cdot \ldots \cdot \pi(n)^{1+a_n}$$

(injection because of uniqueness of prime factorisation).

Inverse function: Retrieving the k-th entry in $s \in \mathbb{N}$:

$$\alpha(s, k) = min\{y \leq s \mid \pi(k)^{y+1} | s \ \wedge \ \pi(k)^{y+2} \not| s\}$$

(if the minimum exists. Undefined at indices beyond the length of s)

Gödelisation (formulas)

Godelisation of formulas

There is a computable injection $\ulcorner\cdot\urcorner : Fml_{\mathcal{N}} \to \mathbb{N}$

Gödel's encoding:

Any formula $f \in Fml$ is a sequence of symbols from a syntactic vocabulary, i.e.,

$$\begin{array}{lllll} Fml \subseteq \{ & 0, & 1, & +, & =, \to, \neg, (,), \forall, x_1, x_2, \ldots\}^* \\ & \downarrow & \downarrow & \downarrow & \ldots \\ & 1 & 3 & 5 & \ldots \end{array}$$

and f can be represented as a finite sequence of natural numbers $seq(f)$.

Define $$\ulcorner\varphi\urcorner := \ulcorner seq(f)\urcorner$$

Gödelisation – modern viewpoint

Godelisation of formulas

There is a computable injection $\ulcorner \cdot \urcorner : \mathit{Fml}_{\mathcal{N}} \to \mathbb{N}$

Today digital encoding is standard:

Let $\varphi \in \mathit{Fml}_{\mathcal{N}}$.

$$\ulcorner \varphi \urcorner = \texttt{new BigInteger(1, } \varphi\texttt{.toString().getBytes());}$$

Inverse operation:

$$\varphi = \texttt{Formula.parse(new String(}\ulcorner \varphi \urcorner\texttt{.toByteArray()))}$$

Primitive Recursive Functions

Definition

Let g and h be terms over $\Sigma_{\mathcal{N}}$ with $n-1$ and $n+1$ free variables. A definition of $f : \mathbb{N}^n \to \mathbb{N}$ is by primitive recursion if it follows the scheme

$$\begin{array}{rcl} f(0, x_2, \ldots, x_n) & = & g(x_2, \ldots, x_n) \\ f(k+1, x_2, \ldots, x_n) & = & h(k, f(k, x_2, \ldots, x_n), x_2, \ldots, x_n) \; . \end{array}$$

Simple examples: $n + m$, $n \cdot m$, $X(n, m) := m^n$, $F(n) := n!$

$$X(0, m) = 1 \qquad\qquad g(m) = 1$$
$$X(k+1, m) = X(k, m) * m \qquad\qquad h(k, r, m) = r * m$$

$$F(0) = 1 \qquad\qquad g() = 1$$
$$F(k+1) = F(k) * (k+1) \qquad\qquad h(k, r) = r * (k+1)$$

By the way: Functions defined by primitive recursion are a uniquely defined conservative extension.

Primitive Recursive Functions used by Gödel

1.	x/y	x ist durch y teilbar
2.	$\mathrm{Prim}(x)$	x ist eine Primzahl
3.	$n\,Pr\,x$	n-te in x enthaltene Primzahl
4.	$n!$	Fakultät von n
5.	$Pr(x)$	n-te Primzahl
6.	$n\,Gl\,x$	n-tes Glied der Zahlenreihe x
7.	$l(x)$	Länge der Zahlenreihe x
8.	$x * y$	Verkettung von x und y
9.	$R(x)$	Zahlenreihe mit x als alleinigem Element
10.	$E(x)$	Formel x in Klammern
11.	$n\,\mathrm{Var}\,x$	x ist eine Variable n-ten Typs
12.	$\mathrm{Var}(x)$	x ist eine Variable
13.	$\mathrm{Neg}(x)$	Negation der Formel x
14.	$x\,\mathrm{Dis}\,y$	Disjunktion von x und y
15.	$x\,\mathrm{Gen}\,y$	Generalisierung von y bez. x
16.	$n\,N\,x$	Zeichenkette x mit n vorangestellten f's
17.	$Z(n)$	Zeichenkette $\overline{n}$
18.	$\mathrm{Typ}_1{}'(x)$	x ist ein Zeichen ersten Typs (Term)
19.	$\mathrm{Typ}_n(x)$	x ist ein Zeichen n-ten Typs
20.	$Elf(x)$	x ist eine Elementarformel (atomare Formel)
21.	$Op(x, y, z)$	Hilfsrelation für $Fr(x)$
22.	$Fr(x)$	Hilfsrelation für $\mathrm{Form}(x)$
23.	$\mathrm{Form}(x)$	x ist eine Formel
24.	$v\,\mathrm{Geb}\,n, x$	Variable v ist an der Stelle n gebunden
25.	$v\,Fr\,n, x$	Variable v ist an der Stelle n frei
26.	$v\,Fr\,x$	Variable v kommt in x an mindestens einer Stelle frei vor
27.	$Su\,x\,(^{n}_{y})$	Formel x, nachdem an der Stelle n y eingesetzt wurde

28.	$k\,St\,v, x$	Hilfsfunktion für $A(v, x)$
29.	$A(v, x)$	Anzahl der Stellen, an denen v in x frei vorkommt
30.	$Sb_0\,(x\,^{n}_{y})$	Hilfsfunktion für $Sb\,(x\,^{v}_{y})$
31.	$Sb\,(x\,^{v}_{y})$	Substitution von v durch y
32.	$x\,\mathrm{Imp}\,y$, $x\,\mathrm{Con}\,y$, $x\,\mathrm{Aeq}\,y$, $v\,\mathrm{Ex}\,y$	Definition von ‚$\rightarrow$', ‚$\wedge$', ‚$\leftrightarrow$', ‚$\exists$'
33.	$n\,Th\,x$	n-te Typenerhöhung von x
34.	Z-$Ax(x)$	x ist eine Instanz des Axiomenschemas I.1, I.2 oder I.3
35.	A_i-$Ax(x)$	x ist eine Instanz des Axiomenschemas II.i
36.	A-$Ax(x), \ldots, A_4$-$Ax(x)$	x ist ein Axiom der Axiomengruppe II
37.	$Q(x, y, v)$	Hilfsprädikat zur Sicherstellung der Kollisionsfreiheit
38.	L_1-$Ax(x)$	x ist eine Instanz des Axiomenschemas III.1
39.	L_2-$Ax(x)$	x ist eine Instanz des Axiomenschemas III.2
40.	R-$Ax(x)$	x ist eine Instanz des Axiomenschemas IV.1
41.	M-$Ax(x)$	x ist eine Instanz des Axiomenschemas V.1
42.	$Ax(x)$	x ist ein Axiom
43.	$Fl(x, y, z)$	x lässt sich aus y und z ableiten
44.	$Bw(x)$	x ist eine formale Beweiskette des Systems P
45.	xBy	x ist ein Beweis für die Formel y

taken from [Hoffmann 2013]
All functions are explained in Chapter 5.2.

Prim. Recurs. Functions as Formulas

Satz VII (adapted)

A function $f : \mathbb{N}^n \to \mathbb{N}$ defined by primitive recursion can be represented as a formula $F(f) \in Fml_{\mathcal{N}}$ with $n+1$ free variables in the sense that

$$\mathcal{N}, Def_f \models f(x_1, ..., x_n) = y \leftrightarrow F(f)(x_1, ..., x_n, y)$$

Proof *Idea*: (oversimplified, for $n = 1$)
Let $Def_f = f(0) = u, \forall k. f(k{+}1) = g(f(k))$

$$Def_f \models f(x) = y \leftrightarrow (\forall s \in \mathbb{N}^x.\ s_0{=}u \land \forall k. s_{k+1}{=}g(s_k) \to s_x = y)$$

Non-trivial proof.
The function symbol can be replaced by a quantification $\forall s : \mathbb{N}^x$ for a bounded sequence. This can be replaced by $\forall n{:}\mathbb{N}.\ \forall d{:}\mathbb{N}$ of two numbers thanks to unique solutions in the Chinese Remainder Theorem $\to$ [Hoffmann 2013, Ch. 6.2]

Prim. Recurs. Functions as Formulas

In particular: $B(x, y)$ (*x ist Beweis für y*) is representable

$$\mathcal{N}, x \mapsto S, y \mapsto \ulcorner\varphi\urcorner \models F(B(x, y))$$
$$\iff S \text{ describes a proof in } C \text{ for } \varphi$$

$$bwb := \exists x. F(B(x, y)) \qquad \text{has one free variable } y$$

Provability can be arithmetically represented

$$\mathcal{N}, y \mapsto \ulcorner\varphi\urcorner \models bwb \quad \iff \quad \vdash_C \varphi$$

DL-Programs Arithmetically Representable

Recall: Programs are representable ($\rightarrow$ Ch. “Dynamic Logic”)

Every DL program π can be represented as a formula $\kappa(\pi) \in \mathit{Fml}_{\mathcal{N}}$

Proof Checker

Proof checker

A DL-program PC over $\mathcal{N}$ can be implemented that two single input variable S and n returns $r = 1$ if S is the godelisation of a proof for the formula ϕ with $\ulcorner \phi \urcorner = n$ in calculus C.

Provability (Idea)

$$bwb := \exists S.\ \kappa(PC)(S, x, 1) \quad \text{with free var } x$$

Calculus representable

$$\mathcal{N}, x \mapsto \ulcorner \varphi \urcorner \models bwb \quad \iff \quad \vdash_C \varphi$$

Proof Plan

1. **Arithmetic Representation:**
   There is a formula *bwb* (*beweisbar*) over $\Sigma_{\mathcal{N}}$ that formalises provability of any formula $\varphi \in \mathit{Fml}_{\mathcal{N}}$.

   $$\mathcal{N}, x \mapsto \ulcorner\varphi\urcorner \models \mathit{bwb} \iff \vdash_C \varphi$$

   We say: Provability is arithmetically representable.

2. **Diagonalisation:**
   Formalise "I cannot be proved!"
   and have $\mathcal{N} \models \vartheta \iff \nvdash_C \vartheta$

Diagonalisation

The set of unprovable formulas

- The set of all formulas $Fml_{\mathcal{N}}$ with exactly one free variable is recursive; let $R : \mathbb{N} \to Fml_{\mathcal{N}}$ be an enumeration of all such formulas.
- We are interested in the diagonal elements, i.e. the application of the n-th formula to the value n:
$$R(n)[\bar{n}] \in Fml_{\mathcal{N}}$$
- Consider the set K of all unprovable diagonal elements
$$K := \{n \mid \nvdash_C R(n)[\bar{n}]\} = \{n \mid \mathcal{N}, x \mapsto \ulcorner R(n)[\bar{n}] \urcorner \models \neg bwb\}$$
- $$\begin{array}{lll} & R(n) & \ldots \text{ primitive recursive} \\ & \cdot[\cdot] & \ldots \text{ primitive recursive (substitution, No 31)} \\ & bwb & \ldots \text{ arithmetically representable} \\ \Longrightarrow & K & \ldots \text{ arithmetically representable} \end{array}$$
- There is $q \in \mathbb{N}$ with $R(q) = K$: $\mathcal{N} \models R(q)[\bar{n}] \iff n \in K$

“I cannot be proved”

- $K := \{n \mid \nvdash_C R(n)[\bar{n}]\}$
- $\mathcal{N} \models R(q)[\bar{n}] \iff n \in K$
- $R(q)$'s intuition is “Formula x cannot be proved”
- What about $\vartheta := R(q)[\bar{q}]$? Intuition ϑ says: “ϑ cannot be proved”, or **“I cannot be proved”**.

Assume $\vdash_C \vartheta$

$$\overset{C \text{ sound}}{\Longrightarrow} \mathcal{N} \models R(q)[\bar{q}] \implies q \in K \implies \nvdash_C R(q)[\bar{q}] \ \lightning$$

Assume $\vdash_C \neg\vartheta$

$$\overset{C \text{ sound}}{\Longrightarrow} \mathcal{N} \models \neg R(q)[\bar{q}] \implies q \notin K \implies \vdash_C R(q)[\bar{q}] \ \lightning$$

Contradiction

ϑ says "ϑ cannot be proved"

$$\mathcal{N} \models \vartheta \iff \nvdash_C \vartheta$$

$$\begin{array}{llll} \vdash_C \vartheta & \implies \mathcal{N} \models \vartheta & & \implies \nvdash_C \vartheta \\ \vdash_C \neg\vartheta & \implies \mathcal{N} \models \neg\vartheta & \implies \mathcal{N} \not\models \vartheta & \implies \vdash_C \vartheta \end{array}$$

There is at least one formula ϑ for which neither ϑ nor $\neg\vartheta$ can be proved.

$\implies$ **The calculus C must be incomplete** (if it is correct).

$\square$

Concluding remarks

- Note: Calculus C exists $\implies$ There is a decision procedure (Examine φ and $\neg\varphi$ in parallel.
  Since the theory $\mathcal{N}$ is complete, one must be true.)

- Reminder: Theoretical incompleteness is different from practical incompleteness ... but often a good indicator

- Limitation of ZFC: *Continuum Hypothesis independent*
  "There is no set whose cardinality lies strictly between that of the integers and its powerset."

Relation to Computability

- Programs (turing machines) arithmetically representable.
- *Halting Problem* $\implies$ Incompleteness
- Check proof for *Halting problem*: Similar diagonalisation idea