A Computational and Abstract Approach to Gödel's First Incompleteness Theorem

First Bachelor seminar talk

Benjamin Peters

Advisor: Dominik Kirst
Supervisor: Professor Gert Smolka

Universität des Saarlandes

December 15, 2021

Theorem (Gödel's first incompleteness theorem)

Any consistent and sufficiently powerful formal system is incomplete.

- Similar statement first shown by Gödel 1931
- Idea: Use logical formulas to represent provability
- Strengthened by Rosser 1936 to this modern form
- There is a folklore proof of a weaker theorem using computability theory
- Can this be strengthened?

Theorem (Gödel's first incompleteness theorem)

$\forall T \supseteq Q.$	T is powerful enough
$\quad \mathbb{N} \vDash T \;\longrightarrow$	T is sound
$\quad T$ enumerable $\;\longrightarrow$	T is reasonable
$\quad (\forall \varphi.\, T \vdash \varphi \vee T \vdash \neg\varphi) \;\longrightarrow$	T is complete
$\quad \mathrm{dec}\, H_{\mathrm{TM}}$	falsity

Proof has been mechanized in the Coq Library of Undecidability Proofs[1] (CLUP) by Kirst and Hermes 2021 using synthetic computability

[1] `https://github.com/uds-psl/coq-library-undecidability`

How can we strengthen this?

$$\forall T \supseteq Q.$$
$$\cancel{\mathbb{N} \vdash T}\ T \nvdash \bot \longrightarrow$$
$$T \text{ enumerable} \longrightarrow$$
$$\cancel{(\forall \varphi.\, T \vdash \varphi \lor T \vdash \neg\varphi) \longrightarrow}$$
$$\cancel{\operatorname{dec} H_{\mathrm{TM}} \perp}$$
$$\exists \varphi.\, T \nvdash \varphi \land T \nvdash \neg\varphi$$

- Actual falsity instead of $\operatorname{dec} H_{\mathrm{TM}}$
- Require consistency instead of soundness
- Explicitly construct independent sentence

Computational proof from Kleene 1967

We will do this computationally and abstractly!

Definition (Formal system)

A formal system $\mathrm{FS} = (S, \neg, \vdash)$ such that:

- $S : \mathbb{T}$ is an enumerable and discrete type of sentences
- $\neg : S \to S$ is a negation function
- $\vdash\ : S \to \mathbb{P}$ is an enumerable provability predicate
- FS is consistent: $\forall s.\, \neg(\vdash s \land \vdash \neg s)$

Definition (Completeness)

$\mathrm{FS} = (S, \neg, \vdash)$ is complete, if $\forall s. \vdash s \lor \vdash \neg s$.

Lemma (Decidability)

In a complete formal system, provability is decidable.

Proof.

Enumerate all provable sentences and search for a proof or refutation. $\square$

Definition (Weak representability)

A formal system $\mathrm{FS} = (S, \vdash, \neg)$ weakly represents a predicate $P : X \to \mathbb{P}$ if there is a representation function $r : X \to S$ such that

$$\forall x.\, P\, x \;\longleftrightarrow\; \vdash r\, x.$$

Weak representability transfers along *sound* extensions.

Lemma (Decidability of predicates)

Any predicate that can be weakly represented in a complete formal system is decidable.

Definition (Partial functions)

A function $f : X \rightharpoonup Y$ is a partial function, e.g. implemented using step-indexing.
A function application $f\,x$ can

- evaluate to y, written $f\,x \downarrow y$
- diverge

We say that $f\,x$ halts, if $\exists y.\, f\,x \downarrow y$.

Assumption (Church's thesis[2][3])

There is a function $\theta : \mathbb{N} \to \mathbb{N} \rightharpoonup \mathbb{B}$, such that

$$\forall (f : \mathbb{N} \rightharpoonup \mathbb{B}).\, \exists c.\, \forall xy.\, f\, x \downarrow y \;\longleftrightarrow\; \theta_c(x) \downarrow y.$$

Lemma (Special halting problem)

The special halting problem for θ, that is

$$H_0\, c := \theta_c(c) \text{ halts},$$

is undecidable.

[2]Troelstra, Dalen, and Beklemishev 1988

[3]Formulation in constructive type theory by Forster 2022

Theorem (Gödel's first incompleteness theorem)

There is no complete formal system that can weakly represent H_0.

There is a mechanized proof that Q weakly represents H_{TM}.

Theorem (Gödel's first incompleteness theorem)

$$\forall T \supseteq Q.\, \mathbb{N} \vDash T \;\longrightarrow\; T \text{ enumerable } \;\longrightarrow\; (\forall \varphi.\, T \vdash \varphi \vee T \vdash \neg\varphi) \;\longrightarrow\; \cancel{\mathrm{dec}\, H_0}\ \bot$$

Proof.

Instantiate abstract proof with first-order logic and Church's thesis for Turing machines. $\square$

What do we need to do to allow consistent extensions?

Definition (Weak representability)

A formal system $\mathrm{FS} = (S, \vdash, \neg)$ weakly represents a predicate $P : X \to \mathbb{P}$ if there is a representation function $r : X \to S$ such that

$$\forall x.\, P\, x \;\longleftrightarrow\; \vdash r\, x.$$

Weak representability transfers along *sound* extensions.

Lemma (Decidability of predicates)

Any predicate that can be weakly represented in a complete formal system is decidable.

Definition (Value-representability)

A formal system $\mathrm{FS} = (S, \vdash, \neg)$ value-represents a function $f : \mathbb{N} \rightharpoonup \mathbb{B}$ if there is a representation function $r : \mathbb{N} \to \mathbb{B} \to S$ such that

$$\forall xy.\, f\, x \downarrow y \;\longrightarrow\; \vdash r\, x\, y \wedge \vdash \neg r\, x\, (!y).$$

Value-representability transfers along *consistent* extensions.

Definition

A formal system value-represents all computable functions, if

$$\forall c.\, \Sigma r.\, r \text{ value-represents } \theta_c.$$

Definition (Consistent guessing)

A language $L \subseteq \mathbb{N}$ fulfills consistent guessing if

$$\{(c, x) \mid \theta_c(x) \downarrow \mathrm{true}\} \subseteq L \quad \wedge \quad \{(c, x) \mid \theta_c(x) \downarrow \mathrm{false}\} \cap L = \emptyset.$$

Lemma (Consistent guessing is undecidable)

Any language $L \subseteq \mathbb{N}$ that fulfills consistent guessing is undecidable.

Proof.

Let $f : \mathbb{N} \to \mathbb{N} \to \mathbb{B}$ be s.t. $\forall cx.\, f\, c\, x = \mathrm{true} \longleftrightarrow (c, x) \in L$.
Consider $g : \mathbb{N} \to \mathbb{B}, g\, c := \,!f\, c\, c$, let c be the code of g.
We now have

$$f\, c\, c = \mathrm{true} \longleftrightarrow f\, c\, c = \mathrm{false}.$$

$\square$

Theorem (Gödel's first incompleteness theorem)

Any formal system $\mathrm{FS} = (S, \neg, \vdash)$ that can value-represent all computable functions is incomplete.

Proof.

We write r_c for the value-representation of a code c. Let $h : \mathbb{N} \to \mathbb{N} \to \mathbb{B}$ be the following function:

$$h\, c\, x \coloneqq \begin{cases} \mathrm{true} & \text{if } r_c\, x\, \mathrm{true} \text{ is provable} \\ \mathrm{false} & \text{otherwise} \end{cases}$$

Assuming FS is complete, h is well-defined and decides

$$L = \{(c, x) \mid h\, c\, x = \mathrm{true}\},$$

which fulfills consistent guessing. $\square$

Theorem (Gödel's first incompleteness theorem)

$$\forall T \supseteq Q.\, \cancel{\mathbb{N} \vDash T}\ T \nvdash \bot \ \longrightarrow\ T \text{ enumerable } \longrightarrow$$
$$(\forall \varphi.\, T \vdash \varphi \vee T \vdash \neg\varphi) \ \longrightarrow\ \cancel{H_0}\ \bot$$

Theorem (Gödel's first incompleteness theorem)

In any formal system that can value-represent all computable functions there is an independent sentence.

Definition (Consistent guessing)

A language $L \subseteq \mathbb{N}$ fulfills consistent guessing if

$$\{(c, x) \mid \theta_c(x) \downarrow \mathrm{true}\} \subseteq L \quad \wedge \quad \{(c, x) \mid \theta_c(x) \downarrow \mathrm{false}\} \cap L = \emptyset.$$

Lemma (Consistent guessing is undecidable)

Any language $L \subseteq \mathbb{N}$ that fulfills consistent guessing is undecidable.

Proof.

Let $f : \mathbb{N} \to \mathbb{N} \to \mathbb{B}$ be s.t. $\forall cx.\, f\, c\, x = \mathrm{true} \longleftrightarrow (c, x) \in L$.
Consider $g : \mathbb{N} \to \mathbb{B}, g\, c := !f\, c\, c$, let c be the code of g.
We now have

$$f\, c\, c = \mathrm{true} \longleftrightarrow f\, c\, c = \mathrm{false}.$$

□

Proof.

We write r_c for the value-representation of a code c. Consider the following program $f(c, x)$:

1. enumerate all provable sentences s.
2. if $s = r_c\, x\, \mathrm{true}$, accept.
3. if $s = \neg r_c\, x\, \mathrm{true}$, reject.
4. otherwise, continue searching

and the function g:

$$g\, c := \begin{cases} \mathrm{false} & \text{if } f(c, c) \downarrow \mathrm{true} \\ \mathrm{true} & \text{if } f(c, c) \downarrow \mathrm{false} \\ \text{undefined} & \text{if } f(c, c) \text{ diverges} \end{cases}$$

Let c be the code of g. Now, $r_c\, c\, \mathrm{true}$ is independent in FS, that is $\nvdash r_c\, c\, \mathrm{true}$ and $\nvdash \neg r_c\, c\, \mathrm{true}$. $\square$

Theorem (Gödel's first incompleteness theorem)

$$\forall T \supseteq Q.\, \mathbb{N} \vDash T \;\longrightarrow\; T \text{ enumerable} \;\longrightarrow\; (\forall \varphi.\, T \vdash \varphi \lor T \vdash \neg\varphi) \;\longrightarrow\; \operatorname{dec} H_0$$

$$\forall T \supseteq Q.\, T \nvdash \bot \;\longrightarrow\; T \text{ enumerable} \;\longrightarrow\; \exists \varphi.\, T \nvdash \varphi \land T \nvdash \neg\varphi$$

I verified all of the abstract arguments using Coq.

Goals

- Complete instantiation of the abstract proof to first-order logic with Q, additionally assuming a form of value-representability
- Instantiate the proof using the halting problem with a proof of weak representability of Turing machines in Q from CLUP
- Attempt to investigate Gödel's second incompleteness theorem using the abstract approach
- Investigate using recursively inseparable sets for showing the abstract theorems
- Mechanize a proof of value-representability of Turing machines in Q

Theorem (Gödel's first incompleteness theorem)

$$\forall T \supseteq Q.\, \mathbb{N} \vDash T \;\longrightarrow\; T \text{ enumerable } \longrightarrow$$
$$(\forall \varphi.\, T \vdash \varphi \lor T \vdash \neg\varphi) \;\longrightarrow\; \mathrm{dec}\, H_0$$

$$\forall T \supseteq Q.\, T \nvdash \bot \;\longrightarrow\; T \text{ enumerable } \longrightarrow$$
$$\exists \varphi.\, T \nvdash \varphi \land T \nvdash \neg\varphi$$

I verified all of the abstract arguments using Coq.

Forster, Yannick (2022). “Parametric Church’s Thesis: Synthetic Computability without Choice”. In: *Logical Foundations of Computer Science: International Symposium, LFCS 2022, January 10-13, 2022.*

Gödel, K. (1931). “Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I”. In: *Monatshefte für Mathematik und Physik* 38, pp. 173–198.

- Kirst, Dominik and Hermes, Marc (2021). "Synthetic Undecidability and Incompleteness of First-Order Axiom Systems in Coq". In: *12th International Conference on Interactive Theorem Proving (ITP 2021)*. Ed. by Liron Cohen and Cezary Kaliszyk. Vol. 193. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 23:1–23:20. ISBN: 978-3-95977-188-7. DOI: 10.4230/LIPIcs.ITP.2021.23. URL: https://drops.dagstuhl.de/opus/volltexte/2021/13918.
- Kleene, Stephen Cole (1967). *Mathematical Logic*. Dover Publications.
- Rosser, Barkley (1936). "Extensions of Some Theorems of Gödel and Church". In: *The Journal of Symbolic Logic* 1.3, pp. 87–91. ISSN: 00224812. URL: http://www.jstor.org/stable/2269028.

References III

Troelstra, A.S., Dalen, D. van, and Beklemishev, L.D. (1988). *Constructivism in Mathematics, Vol 1*. Constructivism in Mathematics. Elsevier Science. ISBN: 9780444703583. URL: https://books.google.de/books?id=EubuAAAAMAAJ.

Halting problem is undecidable

Lemma

The predicate

$$H_0\, c := \theta_c(c) \text{ halts}$$

is undecidable.

Proof.

Let $f : \mathbb{N} \to \mathbb{B}$ be a function such that $\forall c.\, f\, c = \mathrm{true} \longleftrightarrow H_0\, c$.
Choose

$$g : \mathbb{N} \rightharpoonup \mathbb{B}, g\, c := \begin{cases} 0 & \text{if } f\, c = \mathrm{false} \\ \text{undefined} & \text{if } f\, c = \mathrm{true} \end{cases}$$

and let c be the code of g. We have

$$\begin{aligned} f\, c = \mathrm{false} &\longleftrightarrow g\, c = 0 \longleftrightarrow \theta_c(c) \text{ halts} \longleftrightarrow H_0\, c \\ &\longleftrightarrow f\, c = \mathrm{true} \end{aligned}$$

Therefore, H_0 is undecidable. □

Undecidability of CG

Lemma (Consistent guessing is undecidable)

Any language $L \subseteq \mathbb{N}$ that fulfills consistent guessing is undecidable.

Proof.

Let $f : \mathbb{N} \to \mathbb{N} \to \mathbb{B}$ be s.t. $\forall cx.\, f\,c\,x = \text{true} \longleftrightarrow (c, x) \in L$.
Consider $g : \mathbb{N} \to \mathbb{B}, g\,c := !f\,c\,c$, let c be the code of g. We have:

$$f\,c\,c = \text{true} \longrightarrow g\,c = \text{false} \longrightarrow \theta_c(c)\downarrow \text{false} \longrightarrow (c, c) \notin L \\ \longrightarrow f\,c\,c = \text{false}$$

$$f\,c\,c = \text{false} \longleftrightarrow g\,c = \text{true} \longleftrightarrow \theta_c(c)\downarrow \text{true} \longrightarrow (c, c) \in L \\ \longleftrightarrow f\,c\,c = \text{true}.$$

$\square$

Undecidability of CG

Lemma (Consistent guessing is undecidable)

Any language $L \subseteq \mathbb{N}$ that fulfills consistent guessing is undecidable.

Proof.

Let $f : \mathbb{N} \to \mathbb{N} \to \mathbb{B}$ be s.t. $\forall cx.\, f\,c\,x = \mathrm{true} \longleftrightarrow (c, x) \in L$.
Consider $g : \mathbb{N} \to \mathbb{B}, g\,c := !f\,c\,c$, let c be the code of g. We have:

$$\begin{aligned} f\,c\,c = \mathrm{true} &\longleftrightarrow g\,c = \mathrm{false} \longleftrightarrow \theta_c(c)\downarrow \mathrm{false} \longrightarrow (c, c) \notin L \\ &\longleftrightarrow f\,c\,c = \mathrm{false} \\ f\,c\,c = \mathrm{false} &\longleftrightarrow g\,c = \mathrm{true} \longleftrightarrow \theta_c(c)\downarrow \mathrm{true} \longrightarrow (c, c) \in L \\ &\longleftrightarrow f\,c\,c = \mathrm{true}. \end{aligned}$$

$\square$

h computes consistent guessing

Let $h : \mathbb{N} \to \mathbb{N} \to \mathbb{B}$ be the following function:

$$h\,c\,x := \begin{cases} \text{true} & \text{if } r_c\,x\,\text{true is provable} \\ \text{false} & \text{otherwise} \end{cases}$$

To show: $L = \{(c, x) \mid h\,c\,x = \text{true}\}$ fulfills consistent guessing.

We have:	To show:
$\theta_c(x)\downarrow\text{true}$	$(c, x) \in L$
$\vdash r_c\,x\,\text{true}$ by value-representability	$h\,c\,x = \text{true}$

We have:	To show:
$\theta_c(x)\downarrow\text{false}$	$(c, x) \notin L$
$\vdash \neg r_c\,x\,\text{true}$ by value-representability	$h\,c\,x = \text{false}$
$\nvdash r_c\,x\,\text{true}$ by consistency	

Proof.

We write r_c for the value-representation of a code c. Consider the following program $f(c, x)$:

1. enumerate all provable sentences s.
2. if $s = r_c\, x\, \mathrm{true}$, accept.
3. if $s = \neg r_c\, x\, \mathrm{true}$, reject.
4. otherwise, continue searching

and the function g:

$$g\, c := \begin{cases} \mathrm{false} & \text{if } f(c, c) \downarrow \mathrm{true} \\ \mathrm{true} & \text{if } f(c, c) \downarrow \mathrm{false} \\ \text{undefined} & \text{if } f(c, c) \text{ diverges} \end{cases}$$

Let c be the code of g. Now, $r_c\, c\, \mathrm{true}$ is independent in FS, that is $\nvdash r_c\, c\, \mathrm{true}$ and $\nvdash \neg r_c\, c\, \mathrm{true}$. $\square$

Independence in FS

We have: To show:

$\vdash r_c\, c\, \mathrm{true}$

$$\bot$$

STS: $\vdash \neg r_c\, c\, \mathrm{true}$

$$\theta_c(c) \downarrow \mathrm{false}$$

$$g\, c = \mathrm{false}$$

$$f(c, c) \downarrow \mathrm{true}$$

We have: To show:

$\vdash \neg r_c\, c\, \mathrm{true}$

$$\bot$$

STS: $\vdash r_c\, c\, \mathrm{true}$

$$\theta_c(c) \downarrow \mathrm{true}$$

$$g\, c = \mathrm{true}$$

$$f(c, c) \downarrow \mathrm{false}$$

Versions of Gödel's first incompleteness theorem

	No explicit sentence	Explicit sentence
Soundness	H_0, KH2021	
ω-consistency		Gödel's proof
Consistency	CG 1	CG 2, Rosser's trick