Logic, Computability and Incompleteness

Formal Arithmetic and the Diagonal Lemma

Hilbert's Program

- Beginning at the turn of the 20th century, Hilbert proposed a strategy for the foundation of classical mathematics that eventually developed into the so-called 'Formalist Program'.
- This program was in response to the foundational crisis prompted by the newly discovered inconsistency of 'naïve' set theory, in the form of Russell's paradox,

  which also infected Frege's 'Logicist' foundational system.
- Russell's paradox is famously formulated in terms of the set of all sets that are not members of themselves.
- It leads directly to a contradiction in naïve set theory, because this theory assumes the unrestricted **Comprehension Axiom**:

Hilbert's Program

- **Comprehension Axiom**: for <u>any</u> formula $\varphi(x)$ containing x as a free variable, there exists the **set** $\{x: \varphi(x)\}$ whose members are exactly those objects that satisfy $\varphi(x)$.
- Thus, if the formula $\varphi(x)$ stands for "x is prime", then $\{x: \varphi(x)\}$ will be the set of prime numbers.
- If $\varphi(x)$ stands for "$\neg(x = x)$", then $\{x: \varphi(x)\}$ will be the null set.
- But if we let $\varphi(x)$ stand for $x \in x$ and let $\boldsymbol{S} = \{x: \neg\varphi(x)\}$, then $\boldsymbol{S}$ is the set whose members are exactly those objects that are not members of themselves.
- Is $\boldsymbol{S}$ a member of itself?
- Can easily deduce $(\boldsymbol{S} \in \boldsymbol{S}) \leftrightarrow \neg(\boldsymbol{S} \in \boldsymbol{S})$

Hilbert's Program

- Russell's paradox corresponds to the fact that the FOL formula $\exists x \forall y(Rxy \leftrightarrow \neg Ryy)$ is **unsatisfiable** – there can be no such x.

 If we let Rxy mean *'y is an element of x'*, then in standard set theoretical notation this is the same as $\exists x \forall y(y \in x \leftrightarrow y \notin y)$

 If we assume the (intuitively plausible) **Comprehension Axiom**

 then we can prove that there is such an x, and hence our theory will be able to prove a contradiction…

 Hilbert sought to avoid such disasters by advocating an idealized foundational program in which all of mathematics is deducible in an axiomatizable formal theory where the axioms themselves are (independent and) provably consistent.

Representability in a Theory

- As we saw when revisiting FOL, a Formal Theory **T** is a set of sentences (in some formal language ***L***) which is closed under the relation of logical consequence. So for all sentences Φ of ***L***, if $\mathbf{T} \vdash \Phi$ then $\Phi \in \mathbf{T}$ in which case Φ is a **theorem** of **T**, written $\vdash_{\mathbf{T}} \Phi$

Representability in a Theory:

an n-place function of natural numbers f^n is representable in a theory **T** iff there is a formula $\boldsymbol{A}(x_1, \ldots, x_n, x_{n+1})$ in the language of **T** such that for any natural numbers $p_1, \ldots p_n, j$

if $f^n(p_1, \ldots p_n) = j$ then $\vdash_{\mathbf{T}} \forall x(\boldsymbol{A}(\mathbf{p}_1, \ldots \mathbf{p}_n, x) \leftrightarrow x = \mathbf{j}$

where **p** is the **numeral** for p, i.e. **o** followed by p applications of the successor function $'$

Robinson Arithmetic

- In this case $\boldsymbol{A}(x_1, \ldots, x_n, x_{n+1})$ represents $f^{\,n}$ in $\mathbf{T}$.

  Thus if $\boldsymbol{A}(x_1, \ldots, x_n, x_{n+1})$ represents $f^{\,n}$ in $\mathbf{T}$

  and $f^{\,n}(p_1,\ldots p_n) = j$ then both

  $\vdash_{\mathbf{T}} \boldsymbol{A}(\mathbf{p}_1,\ldots\mathbf{p}_n, \mathbf{j})$ and $\vdash_{\mathbf{T}} \forall x(\boldsymbol{A}(\mathbf{p}_1,\ldots\mathbf{p}_n, x) \longrightarrow x = \mathbf{j})$
- The formal theory of particular interest to us will be $\boldsymbol{Q}$, the theory of Robinson Arithmetic
- The language $\boldsymbol{L}$ of $\boldsymbol{Q}$ is FOL

  with $\mathbf{o}$, $'$, $+$, $\cdot$ as distinguished vocabulary items.
- $\boldsymbol{Q}$ is the set of all sentences in $\boldsymbol{L}$ which are **logically entailed** by the following 7 **axioms**:

Robinson Arithmetic

Q1: $\forall x \forall y (x' = y' \rightarrow x = y)$

Q2: $\forall x\, \mathbf{o} \neq x'$

Q3: $\forall x (x \neq \mathbf{o} \rightarrow \exists y\, x = y')$

Q4: $\forall x (x + \mathbf{o} = x)$

Q5: $\forall x \forall y (x + y') = (x + y)'$

Q6: $\forall x (x \cdot \mathbf{o} = \mathbf{o})$

Q7: $\forall x \forall y (x \cdot y') = (x \cdot y) + x$

Each axiom is a single sentence, so ***Q*** is finitely axiomatizable

Representability in Robinson Arithmetic

- Robinson Arithmetic ***Q*** differs from the stronger theory of Peano Arithmetic ***PA***, in that it *lacks* the **schema** of Mathematical Induction:

$$[\Phi\,(\mathbf{o}) \wedge (\forall x\,(\Phi\,(x) \rightarrow \Phi\,(x')))] \rightarrow \forall x\,\Phi\,(x)$$

  where $\Phi(\mathbf{v})$ is any formula in the language ***L***

  with the variable **v** free.

- The **schema** of Mathematical Induction introduces infinitely many axioms as <u>instances</u> of the schema.
- <u>**Very**</u> <u>**important**</u> <u>**property**</u> of ***Q***:

  All recursive functions are representable in ***Q***

Representability in Robinson Arithmetic

- Thus for every function f^n of natural numbers obtainable from the set of Base functions:

  1) zero function

  2) successor function

  3) projection functions

  through finite applications of Composition, Primitive recursion and Minimization,

  there is a formula $\boldsymbol{A}(x_1, \ldots, x_n, x_{n+1})$ in the language $\boldsymbol{L}$

  such that if $f^n(p_1,\ldots p_n) = j$ then both

  $$\vdash_{\boldsymbol{Q}} \boldsymbol{A}(\mathbf{p}_1,\ldots\mathbf{p}_n, \mathbf{j}) \text{ and } \vdash_{\boldsymbol{Q}} \forall x(\boldsymbol{A}(\mathbf{p}_1,\ldots\mathbf{p}_n, x) \longrightarrow x = \mathbf{j})$$

Arithmetization of Syntax

- We'll now look at Gödel numbering, which is the first ingredient needed to achieve formal 'self-reference' in arithmetic.
- Gödel numbering is a scheme for assigning natural numbers to *expressions* in a **formal object language**.
- Necessary characteristics of a Gödel numbering scheme:

1) different expressions get different numbers

2) given any expression can effectively calculate its Gödel number

3) given any number can effectively determine

(i) whether it's the Gödel number of an expression, and

Arithmetization of Syntax

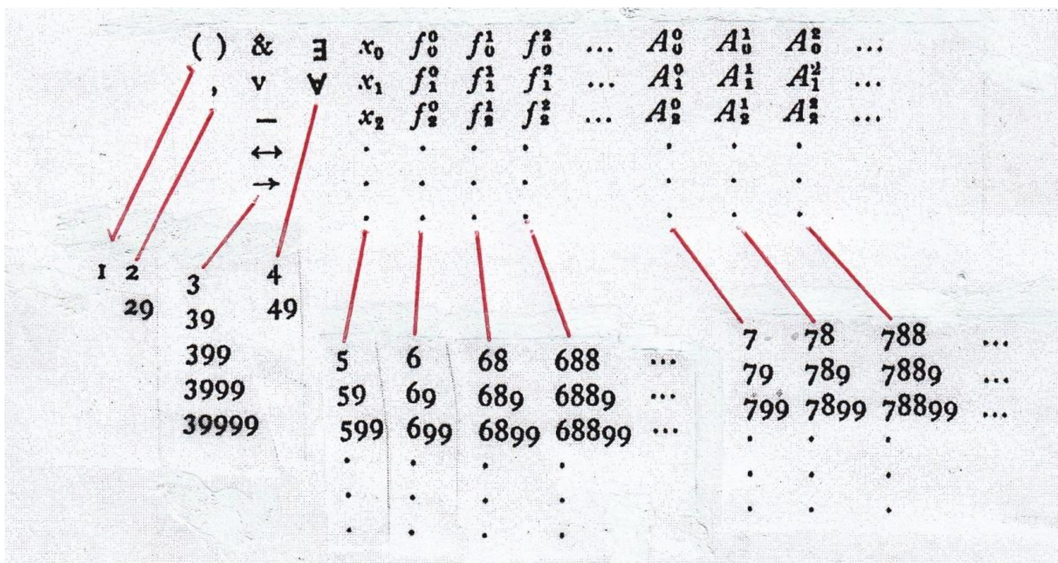
(ii) if so, can effectively recover the expression from the number.

- Particular Gödel numbering scheme used in B&J:
- Numbers 1-7 are used to distinguish basic categories of symbols:

  1, 2 for punctuation symbols, 3 for truth-functional connectives, 4 for quantifiers, 5 for variables, 6 for function symbols and 7 for predicate symbols.
- Numbers 8, 9 used to make internal distinctions (e.g. for non-zero superscripts and subscripts).

Scheme is given in charts on p. 171:

Arithmetization of Syntax



Arithmetization of Syntax

- Each basic vocabulary **symbol** is thereby given a unique number. To do this, the actual coding must deal directly with official **object language symbols** systematically arranged.
- However, in practice it's *inconvenient* to work purely with object language symbols, so we will adopt some conventions:

official symbol	'informal' name
$x_0, x_1, x_2, \ldots$	$x, y, z, \ldots$
f_0^0	$\mathbf{0}$
f_0^1	$'$
f_0^2	$+$
f_1^2	$\cdot$
A_0^2	$=$

Arithmetization of Syntax

- Concatenation of basic vocabulary symbols to form complex expressions is reflected by concatenation of the numbers of the symbols involved and then read in **decimal notation**.

  The Gödel number of symbol '$\boldsymbol{A}$' is written $\boldsymbol{gn}[\boldsymbol{A}]$.

  So if $\boldsymbol{gn}[\boldsymbol{A}] = i$ and $\boldsymbol{gn}[\boldsymbol{B}] = j$

  then $\boldsymbol{gn}[\boldsymbol{AB}] =$ the number denoted by 'ij' in decimal notation.

  example:

Arithmetization of Syntax

Given the foregoing naming conventions:

gn[∀] = 49, ***gn***[x] = 5, ***gn***[y] = 59

gn[**o**] = 6, ***gn***[′] = 68, ***gn***[+] = 688, ***gn***[·] = 6889

gn[=] = 788

And given the method for determining the Gödel number of a contatenation of symbols

gn[$\forall x(x = x)$] = the concatenation of the **numbers** of the seven constituent **symbols**

$\forall \quad x \quad (\quad x \quad = \quad x \quad)$

/ | | | | | \

49 5 1 5 788 5 2 = 4951578852

Arithmetization of Syntax

- In this manner, the language ***L*** of the theory ***Q*** that is *intended* to be about the natural numbers can instead be interpreted as being about its own syntax (**!**)

  So there will be (unintended) interpretations $\mathcal{J}$ in which ***L*** can be seen as '*making assertions about itself*'
- Furthermore the sentences of ***L*** which are theorems of ***Q*** must be true in every model of ***Q***.
- In particular, all recursive operations on expressions and sequences of expressions can be represented in ***Q***,

  which means that the corresponding sentences are provable in the system.

Arithmetization of Syntax

- So *via* its theorems, the theory ***Q*** can be interpreted as proving things about itself, by associating expression in ***L*** with Gödel numbers and then proving assertions about these numbers.

  And these sentences must be **true** in *every model* of ***Q***.
- This possibility is realized by Gödel's ingenious version of diagonalization, which is the technical heart of the

  limitative metatheoretical results to follow….

Diagonalization

- Convention: if $\boldsymbol{gn}[\boldsymbol{A}] = n$, let $\ulcorner \boldsymbol{A} \urcorner = \mathbf{n}$

  i.e. $\mathbf{o}''^{\cdots}{}'$ with n applications of the successor function.
- So **n** is the **numeral** for the Gödel number of ***A***.

  Hence $\ulcorner \boldsymbol{A} \urcorner$ is the Gödel numeral of ***A***,

  in which case **n** can be construed as a name in the object language ***L*** denoting the object language expression '***A***'.
- This is the second step in achieving formal 'self-reference'.
- Now let the diagonalization of ***A*** be defined as the sentence

$$\exists x\, (x = \ulcorner \boldsymbol{A} \urcorner \wedge \boldsymbol{A})$$

Diagonalization

- If $\boldsymbol{A}$ has just the variable x free, written $\boldsymbol{A}(x)$, then the diagonalization of $\boldsymbol{A}$ is logically equivalent to $\boldsymbol{A}(\ulcorner \boldsymbol{A} \urcorner)$

$$\exists x\,(x = \ulcorner \boldsymbol{A} \urcorner \wedge \boldsymbol{A}) \quad \equiv \quad \boldsymbol{A}(\ulcorner \boldsymbol{A} \urcorner)$$

- **Lemma**: there is a recursive function **diag** such that

  diag(n) = the Gödel number of the <u>diagonalization</u> of the expression with Gödel number n.

- **Proof**: by construction.

1) Let **lh**(n) = $\mu m(0 < m \wedge n < 10^m)$ [read 'the least m such that (…)]

   So **lh**(n) = the number of digits in the decimal notation for the number n.

Diagonalization

2) Let $m * n = m \cdot 10^{\mathbf{lh}(n)} + n$

$m * n$ is the number denoted by the arabic numeral formed by <u>concatenating</u> the arabic numeral for m with the numeral for n

3) Define the function **num**(x) such that

$$\mathbf{num}(0) = 6 \qquad\qquad \mathbf{num}(n+1) = \mathbf{num}(n) * 68$$

So **num**(n) = the Gödel number of the numeral **n** (!)

4) The diagonalization of formula $\boldsymbol{A}$ was defined as

$$\exists x\,(x = \ulcorner \boldsymbol{A} \urcorner \wedge \boldsymbol{A}).$$

And if $\boldsymbol{gn}[\boldsymbol{A}] = n$, then $\ulcorner \boldsymbol{A} \urcorner = \mathbf{n}$.

Hence the diagonalization of $\boldsymbol{A}$ is the formula $\exists x(x = \mathbf{n} \wedge \boldsymbol{A})$.

So let $\mathbf{diag}(n) = 4515788 * (\mathbf{num}(n) * (3 * (n * 2)))$

Diagonalization

$$\mathbf{diag}(n) \;=\; 4515\underbrace{788}\; * \;(\underbrace{\mathbf{num}(n)} * (3 * (n * 2)))$$

$$\exists\, x\, (x \;=\; \mathbf{n} \;\wedge\; A\,)$$

The Diagonal Lemma

- Hence **diag**(*n*) is the Gödel number of the diagonalization of the expression with Gödel number *n*,

  and **diag** is recursive by construction □
- Since all recursive functions are representable in ***Q***,

  diag is representable in ***Q***.
- **Diagonal Lemma**: Let **T** be a theory in which **diag** is representable. Then for **any** formula ***B***(*y*) in the language of **T** with just the variable *y* free, there is a sentence ***G*** such that

$$\vdash_{\mathbf{T}} \boldsymbol{G} \leftrightarrow \boldsymbol{B}(\ulcorner \boldsymbol{G} \urcorner)$$

- **Proof:** exhibit a procedure for constructing such a ***G***

  for any given ***B***(*y*).

The Diagonal Lemma

Let the formula $\boldsymbol{A}_{\mathbf{d}}(x, y)$ represent **diag** in **T**.

Then for any numbers n, k,

if $\mathbf{diag}(n) = k$ then $\vdash_{\mathbf{T}} \forall y(\boldsymbol{A}_{\mathbf{d}}(\mathbf{n}, y) \leftrightarrow y = \mathbf{k})$

Let $\boldsymbol{F}$ be defined as the formula $\exists y(\boldsymbol{A}_{\mathbf{d}}(x, y) \land \boldsymbol{B}(y))$.

$\boldsymbol{F}$ contains just the variable x free.

Now let $\boldsymbol{G}$ be defined as the diagonalization of $\boldsymbol{F}$ (!)

i.e. $\boldsymbol{G}$ is the <u>sentence</u> $\exists x(x = \ulcorner \boldsymbol{F} \urcorner \land \exists y(\boldsymbol{A}_{\mathbf{d}}(x, y) \land \boldsymbol{B}(y)))$.

Suppose $gn[\boldsymbol{F}] = n$, so $\ulcorner \boldsymbol{F} \urcorner = \mathbf{n}$

As noted above, $\boldsymbol{G}$ is logically equivalent to the result of instantiating the variable x with $\ulcorner \boldsymbol{F} \urcorner$ which is $\mathbf{n}$:

The Diagonal Lemma

To repeat, $\boldsymbol{G}$ is the <u>sentence</u> $\exists x(x = \ulcorner \boldsymbol{F} \urcorner \wedge \exists y(\boldsymbol{A}_{\mathbf{d}}(x, y) \wedge \boldsymbol{B}(y)))$

and $\qquad \boldsymbol{G} \equiv \exists y(\boldsymbol{A}_{\mathbf{d}}(\mathbf{n}, y) \wedge \boldsymbol{B}(y))$

so $\qquad$ (i) $\quad \vdash_{\mathbf{T}} \boldsymbol{G} \leftrightarrow \exists y(\boldsymbol{A}_{\mathbf{d}}(\mathbf{n}, y) \wedge \boldsymbol{B}(y))$

Next, suppose $\mathbf{diag}(n) = k$.

Then, since $\boldsymbol{gn}[\boldsymbol{F}] = n$ and $\boldsymbol{G}$ is the diagonalization of $\boldsymbol{F}$,

$\qquad \boldsymbol{gn}[\boldsymbol{G}] = k$ and $\ulcorner \boldsymbol{G} \urcorner = \mathbf{k}$

Now, since $\boldsymbol{A}_{\mathbf{d}}(x, y)$ represents **diag** in **T** and $\mathbf{diag}(n) = k$

we get (ii) $\vdash_{\mathbf{T}} \forall y(\boldsymbol{A}_{\mathbf{d}}(\mathbf{n}, y) \leftrightarrow y = \mathbf{k})$

Taking (i) and substituting provable equivalents from (ii)

we get (iii) $\quad \vdash_{\mathbf{T}} \boldsymbol{G} \leftrightarrow \exists y(y = \mathbf{k} \wedge \boldsymbol{B}(y))$

The Diagonal Lemma

Taking (iii) $\vdash_{\mathbf{T}}$ $\boldsymbol{G} \leftrightarrow \exists y(y = \mathbf{k} \wedge \boldsymbol{B}(y))$

and applying the same strategy used to get (i) yields

$$\vdash_{\mathbf{T}} \boldsymbol{G} \leftrightarrow \mathbf{B}(\mathbf{k})$$

And since $\ulcorner \boldsymbol{G} \urcorner = \mathbf{k}$

it's now immediate that

$$\vdash_{\mathbf{T}} \boldsymbol{G} \leftrightarrow \mathbf{B}(\ulcorner \boldsymbol{G} \urcorner) \blacksquare$$

Conceptual Overview

- It’s fitting at this point to step back for a moment and reflect on the evolution of our theoretical perspective.
- Leibniz (1646-1716) speculated about the development of a precise artificial language, a ‘*calculus ratiocinator*’,

  in which all of human thought could be reduced to calculation.
- Frege’s *Begriffsschrift* (1879) is the first actual instance of an **artificial language** constructed according to exact rules of syntax,

  and the *Begriffsschrift’*s system of first-order logic was powerful enough to formalize all the reasoning ordinarily used in mathematics.
- So (in principle at least) all of mathematics could be carried out ***inside*** this formal system.

Conceptual Overview

- Whitehead and Russell then succeeded in developing all of classical mathematics *within* the artificial logical system of *Principia Mathematica* (PM, 1910).
- Beginning around the turn of the century, Hilbert proposed a perspective in which we abstract away from proofs *inside* such formal systems

and instead look at them from the **outside**

from a **metamathematical** perspective in which we prove

higher-level metalogical results ***about*** these object level systems.

Thus in 1930 Gödel proved the **completeness** of first-order logic,

and in 1936 Church established its **undecidability**.

Conceptual Overview

- In his 1931work on the **incompleteness** of PM,
  Gödel took the level of abstraction a step further
  by embedding these metamathematical concepts
  inside the object level system itself to attain yet new results.
- By using a scheme for numerically coding the syntax of a formal theory of arithmetic,
  it was possible to interpret sentences in the formal object language as making assertions about various properties of sentences in the formal object language .
- In particular, as we'll soon see, a property of special interest is
  'sentence Φ is **provable** in formal arithmetic'.

Conceptual Overview

- Thus, given a numerical coding scheme, Gödel was able to construct a sentence Ψ in formal arithmetic,

  which can be interpreted as asserting that

  ‘sentence Φ is **provable** in formal arithmetic’.
- And using the diagonal lemma, Gödel was able to construct a case where

  Ψ is identical to Φ **!**
- Thus, *via* this employment of Cantor’s diagonal method, matters can be arranged such that

  the object language sentence asserted to be provable

  and the object language sentence making the assertion

  are one and the same…