Gödel's and Post's Proofs of Incompleteness

Máté Szabó and Patrick Walsh

Department of Philosophy
Carnegie Mellon University

Midwest PhilMath Workshop 17, 2016

Historical Context

- Gödel proved his incompleteness theorems in 1931 for *Principia Mathematica* and extensions thereof.
- Emil Post worked on incompleteness and undecidability in the 1920s.
  - In 1941, Post attempted to publish this work, adding an introduction and footnotes. It was not published.
  - This work was published only much later as *Absolutely Unsolvable Problems and Relatively Undecidable Propositions – Account of an Anticipation* in 1965.

Historical Context

- Post had no interest in priority:
  "there would be little point in publicizing the writer's anticipation [. . .] merely as a claim to unofficial priority."
- Instead, Post emphasized that:
  "with the Principia Mathematica as a common starting point, the road followed towards our common conclusions are so different that much may be gained from a comparison of these parallel evolutions."
- We take up this comparison

Outline

Gödel's System P

1. *Constants*:

$$\sim \quad \vee \quad \prod \quad 0 \quad f \quad (\quad)$$

2. *Variables*: Simply typed variables with natural number subscripts, level 1 corresponding to natural numbers and level 2 ranging over classes of natural numbers.
3. *Signs* of type 1 are of the form

$$a, fa, ffa, fffa, \ldots$$

where a is either 0 or a variable of type 1. Signs of higher types are the variables of that type.

Gödel's System P

4. *Elementary formula* have the form $a(b)$ with b of type n and a of type $n+1$.
5. The class of formulae is then constructed as the smallest class containing all the elementary formulas and such that if a and b are formulae, then so are

$$\sim (a)$$

$$(a) \vee (b)$$

$$x \prod (a)$$

(with x a variable).

Gödel's System P

- Gödel believed that his incompleteness results hold "for a wide class of formal systems", not only the extensions of System P by a finite number of axioms.
- However, the proof itself does not go beyond these extensions of System P, as there is no available abstract characterization of what a 'formal system' is, in 1931.
- In his 1964 Postscriptum to the Princeton Lectures, Gödel mentions both Post and Turing in coming up with a "unquestionably adequate definition of the general concept of formal system."

Post's Early Work

- *Introduction to a General Theory of Elementary Propositions* (1921) examines the propositional subsystem $(\sim, \vee)$ of *Principia Mathematica* and shows it to be:
  - *complete* (by use of truth-tables)
  - *decidable* (decision procedure: truth-table method)
  - *consistent*

Post's Early Work

- According to Post, Russell and Whitehead did not reach the highest generality possible in *PM*, because

"owing to the particular purpose the authors had in view they decided not to burden their work with more than was absolutely necessary for its achievement, and so *gave up the generality of outlook* which characterized symbolic logic. It is with the recovery of this generality that the first portion of our paper deals."

Generalized Outlook of Propositional Logic

Post, influenced by C.I. Lewis, suggests we define a general notion of symbolic logic as systems with the following form:

I. Fixed set of n-many functions

$$f_1(p_1, p_2, \ldots, p_{m_1}), \ldots, f_n(p_1, p_2, \ldots, p_{m_n}).$$

II. If $p_1, \ldots p_{m_i}$ are elementary propositions, then so is $f_i(p_1, \ldots p_{m_i})$.

III. Substitution

Generalized Outlook of Propositional Logic

IV.

$$\vdash g_{i1}(P_1, P_2, \ldots, P_{k_i})$$
$$\vdots$$
$$\vdash g_{i\kappa_i}(P_1, P_2, \ldots, P_{k_i})$$
produce
$$\vdash g_i(P_1, P_2, \ldots, P_{k_i})$$

P's are combinations of f's including the special case of an unmodified variable; the g's are combinations of f's.

V.

$$\vdash h_1(p_1, p_2, \cdots, p_{l_1})$$
$$\vdash h_2(p_1, p_2, \cdots, p_{l_2})$$
$$\vdots$$
$$\vdash h_\lambda(p_1, p_2, \cdots, p_{l_\lambda})$$

where the h's are particular combinations of the f's.

The Syntactic Approach of the *Anticipation*

> "Perhaps the chief difference in the method between the present development and its more complete successors [Gödel, Church, Turing] is its *preoccupation with the* ***outward forms*** *of symbolic expressions, and possible operations thereon*, rather than with logical concepts as clothed in, or reflected by, correspondingly particularized symbolic expressions, and operations thereon." [Our emphasis]

- To solve the Entscheidungsproblem, simpler and simpler formal systems are introduced — sequentially reducible.

Canonical Form A (and B)

I. Fixed set of n-many functions

$$f_1(p_1, p_2, \ldots, p_{m_1}), \ldots, f_n(p_1, p_2, \ldots, p_{m_n}).$$

II. If $p_1, \cdots p_{m_i}$ are elementary propositions, then so is $f_i(p_1, \cdots p_{m_i})$.

III. Substitution

Canonical Form A (and B)

IV.

$$\begin{gathered}\vdash g_{i1}(P_1, P_2, \ldots, P_{k_i}) \\ \vdots \\ \vdash g_{i\kappa_i}(P_1, P_2, \ldots, P_{k_i}) \\ \text{produce} \\ \vdash g_i(P_1, P_2, \ldots, P_{k_i})\end{gathered}$$

P's are combinations of f's including the special case of an unmodified variable; the g's are combinations of f's.

V.

$$\begin{gathered}\vdash h_1(p_1, p_2, \cdots, p_{l_1}) \\ \vdash h_2(p_1, p_2, \cdots, p_{l_2}) \\ \vdots \\ \vdash h_\lambda(p_1, p_2, \cdots, p_{l_\lambda})\end{gathered}$$

where the h's are particular combinations of the f's.

Canonical Form *C*

- Alphabet: a fixed finite number of distinct primitive symbols

$$a_1, a_2, \ldots, a_n$$

- Well-formed formulas: every string composed of a finite number of letters from the alphabet. (i.e. $a_2 a_2 a_n a_3$)
- Axioms: a fixed finite set of strings constructed from the alphabet.
- Production Rules. . .

Canonical Form *C*

Production rules: a finite set of inference rules of the following form to yield new theorems from old:

$$g_{11}\mathbf{P}_{i_1^1}g_{12}\mathbf{P}_{i_2^1}\ldots g_{1m_1}\mathbf{P}_{i_{m_1}^1}g_{1(m_1+1)}$$
$$g_{21}\mathbf{P}_{i_1^2}g_{22}\mathbf{P}_{i_2^2}\ldots g_{2m_2}\mathbf{P}_{i_{m_2}^2}g_{2(m_2+1)}$$
$$\vdots$$
$$g_{k1}\mathbf{P}_{i_1^k}g_{k2}\mathbf{P}_{i_2^k}\ldots g_{km_k}\mathbf{P}_{i_{m_k}^k}g_{k(m_k+1)}$$

produce

$$g_1\mathbf{P}_{i_1}g_2\mathbf{P}_{i_2}\ldots g_m\mathbf{P}_{i_m}g_{m+1}$$

g's are fixed sequences of *a*'s **P**'s are variables for strings.

Canonical Form *C*

Production rules: a finite set of inference rules of the following form to yield new theorems from old:

$$g_{11}\mathbf{P}_{i_1^1}g_{12}\mathbf{P}_{i_2^1}\ldots g_{1m_1}\mathbf{P}_{i_{m_1}^1}g_{1(m_1+1)}$$
$$g_{21}\mathbf{P}_{i_1^2}g_{22}\mathbf{P}_{i_2^2}\ldots g_{2m_2}\mathbf{P}_{i_{m_2}^2}g_{2(m_2+1)}$$
$$\vdots$$
$$g_{k1}\mathbf{P}_{i_1^k}g_{k2}\mathbf{P}_{i_2^k}\ldots g_{km_k}\mathbf{P}_{i_{m_k}^k}g_{k(m_k+1)}$$

produce

$$g_1\mathbf{P}_{i_1}g_2\mathbf{P}_{i_2}\ldots g_m\mathbf{P}_{i_m}g_{m+1}$$

$$\frac{a\mathbf{P}_1b\mathbf{P}_2c \qquad abac\mathbf{P}_3}{\mathbf{P}_2c\mathbf{P}_3c\mathbf{P}_1}$$

Example of Canonical Form C

- Alphabet: a, b, c
- Production Rule:

$$\frac{a\mathbf{P}_1 b\mathbf{P}_2 c \qquad abac\mathbf{P}_3}{\mathbf{P}_2 c\mathbf{P}_3 c\mathbf{P}_1}$$

- Applying the rule, *indeterministically*:

$$\frac{a\mathbf{b}b\mathbf{b}c \qquad abac\mathbf{aa}}{\mathbf{b}ca\mathbf{a}c\mathbf{b}} \qquad P_1 = b, P_2 = b, P_3 = aa$$

$$\frac{a\mathbf{bb}bc \qquad abac\mathbf{aa}}{c\mathbf{aa}c\mathbf{bb}} \qquad P_1 = bb, P_2 = \emptyset, P_3 = aa$$

Modus Ponens in Canonical Form *C*

- The familiar natural deduction rules can be represented as a production rules of Canonical Form *C* by considering the logical connectives as the fixed strings *g*.
- Production Rule:

$$\frac{\neg\ \mathbf{P_1} \vee \mathbf{P_2} \qquad \mathbf{P_1}}{\mathbf{P_2}}$$

Normal Form

- Finite alphabet
- Only **one** axiom (a string consisting only of letters of the alphabet)
- Production rules in the form:

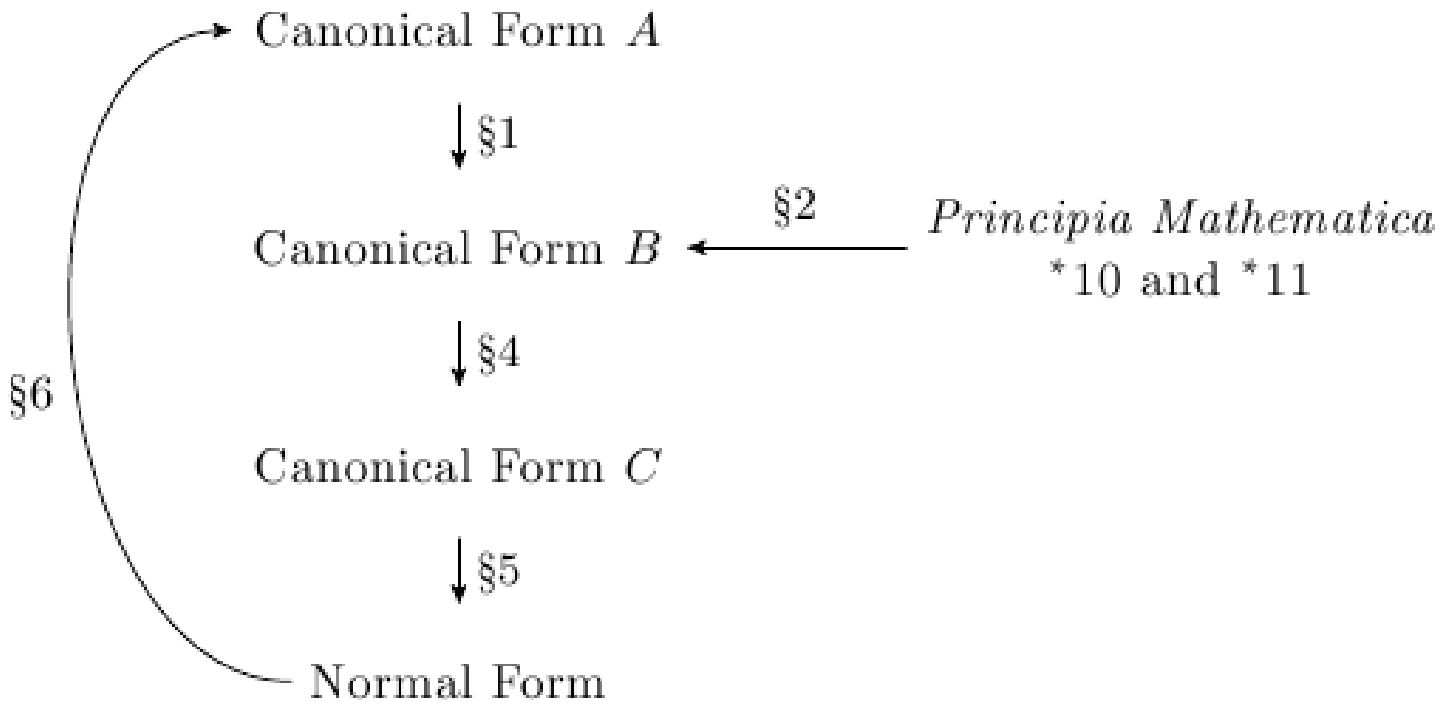
$$gP$$
produces
$$Pg'$$

Example of Normal System

Alphabet: a, b, c
Axiom: *aabcccac*
Production rule:

Of the form:	Using $g = aabc$ and $g' = bccc$	Applied to axiom with $P = ccac$
gP produces Pg'	$aabcP$ produces $Pbccc$	$aabc\mathbf{ccac}$ produces $\mathbf{ccac}bccc$

Reductions in *Anticipation*



Post's unsuccessful attempts to show Normal Systems, and in turn, *Principia Mathematica*, to be decidable led him to anticipate the analogues of undecidability and incompleteness results.

Post's Thesis

Identification of the intuitive notion of *generated set of sequences* with a precise notion:

> *"Every generated set of sequences on a given set of letters $a_1, a_2, ..., a_\mu$ is a subset of the set of assertions of a system in normal form with primitive letters $a_1, a_2, ..., a_\mu, a'_1, a'_2, ..., a'_\mu$, i.e., the subset consisting of those assertions of the normal system involving only the letters $a_1, a_2, ..., a_\mu$."*

Paraphrase

Every generated set is generated by a Normal System.

Post's Thesis

- By diagonalizing, a seeming counterexample can be 'constructed' for the Thesis.
- However,
  "we have merely *defined* a set [. . .] whereas to yield a true counter-example we must show how to *generate* that set"
- Exactly how Kleene convinced himself of Church's Thesis.

Complete Normal System K

- There exists a Normal System K and an effective representation with the following properties:
  - The representation takes a Normal System S and a string P to a string of K, denoted by (S, P).
  - $S \vdash P \iff K \vdash (S, P)$
    (K internalizes claims that strings P are derivable in a normal system S.)

- "The 'complete normal system' would thus correspond to Turing's 'universal computing machine'." [footnote from 1941]

Finite-Normal-Test and Undecidability

Let M and N be Normal Systems. N is a Finite-Normal-Test for M if:

- the alphabet of N includes the alphabet of M and at least one additional letter, say b
- if the string P is derivable in M, it is derivable in N as well

$$M \vdash P \iff N \vdash P$$

- if the string P is not derivable in M, then bP is derivable in N

$$M \nvdash P \iff N \vdash bP$$

Corresponding concepts

N is a Finite-Normal Test:

$$M \vdash P \iff N \vdash P$$
$$M \nvdash P \iff N \vdash bP$$

- b behaves as a negation sign.
- Definition of Finite-Normal-Test N implies that we cannot have both $N \vdash P$ and $N \vdash bP$. Consistency

Undecidability

Theorem

There is no Finite-Normal-Test for the Complete Normal System K.

- Post's sketch of a proof:

  Assume L is a Finite-Normal-Test for K. This means that

$$L \vdash (S, P) \iff K \vdash (S, P)$$
$$L \vdash b(S, P) \iff K \nvdash (S, P)$$

Undecidability

L is a Finite-Normal Test for K:

$$L \vdash (S, P) \iff K \vdash (S, P)$$
$$L \vdash b(S, P) \iff K \nvdash (S, P)$$

- Restrict strings:
  - We call a string an a-sequence, if it involves only the letter a

    $$a, aa, aaa, aaaa, aaaaa, \ldots$$

  - The string consisting of m-many a's, will be denoted by a^m.

Undecidability

L is a Finite-Normal Test for K:

$$L \vdash (S, a^m) \iff K \vdash (S, a^m)$$
$$L \vdash b(S, a^m) \iff K \nvdash (S, a^m)$$

- Coding Normal Systems
  - There are only countably many Normal Systems.
  - Let us fix an enumeration σ, i.e. the i-th Normal System, system $\sigma(i)$, is denoted by S_i.

Undecidability

L is a Finite-Normal Test for K:

$$L \vdash (S_i, a^m) \iff K \vdash (S_i, a^m)$$
$$L \vdash b(S_i, a^m) \iff K \nvdash (S_i, a^m)$$

- Define L', diagonally, such that

$$L' \vdash a^m \iff L \vdash b(S_m, a^m).$$

If $L' = S_{m'}$, we get a contradiction. □

Normal-Deductive-System

Let N be a Normal System. N is a Normal-Deductive-System if:

- the alphabet of N includes the alphabet of K and at least one additional letter, say b
- the string (S, P) is derivable in K if and only if it is derivable in N

$$N \vdash (S, P) \iff K \vdash (S, P)$$

- if the string $b(S, P)$ is derivable in N then (S, P) is not derivable in K

$$N \vdash b(S, P) \implies K \nvdash (S, P)$$

Post's Incompleteness

- Let N be a Normal-Deductive-System.
- Define a normal system N' by the following property

$$a^m \text{ is derivable in } N' \iff b(S_m, a^m) \text{ is derivable in } N.$$

- Suppose that $N' = S_{m'}$.

Theorem:
$(S_{m'}, a^{m'})$ is independent of N.

Post's Incompleteness

Theorem
$(S_{m'}, a^{m'})$ *is independent of* N.

Case 1: assume $N \vdash (S_{m'}, a^{m'})$

$$
\begin{aligned}
N \vdash (S_{m'}, a^{m'}) &\implies K \vdash (S_{m'}, a^{m'}) && (N \text{ is normal deductive}) \\
&\implies S_{m'} \vdash a^{m'} && (K \text{ is complete}) \\
&\implies N' \vdash a^{m'} && (N' = S_{m'}) \\
&\implies N \vdash b(S_{m'}, a^{m'}) && (\text{def of } N') \\
&\implies K \nvdash (S_{m'}, a^{m'}) && (N \text{ is normal deductive})
\end{aligned}
$$

So $N \nvdash (S_{m'}, a^{m'})$, which implies $S_{m'} \nvdash a^{m'}$.

Post's Incompleteness

Case 2: assume $N \vdash b(S_{m'}, a^{m'})$

$$N \vdash b(S_{m'}, a^{m'}) \implies K \nvdash (S_{m'}, a^{m'}) \quad (N \text{ is normal deductive})$$
$$\implies S_{m'} \nvdash a^{m'} \quad (K \text{ is complete})$$

$$N \vdash b(S_{m'}, a^{m'}) \implies N' \vdash a^{m'} \quad (\text{def of } N')$$
$$\implies S_{m'} \vdash a^{m'} \quad (S_{m'} = N')$$

□

Gödel's Incompleteness

- First we order the one-place predicates of system P, called *class expressions*, so that the n-th class expression is $R_n(x)$.
  - $R_n(x)$ corresponds to Post's (S_n, P).
- We can represent provability in *consistent* system P:

$$P \vdash \phi \iff P \vdash Bew(\phi)$$

$$P \vdash \overline{Bew(\phi)} \implies P \nvdash \phi$$

- Define the set K of natural numbers in system P by

$$n \in K \iff \overline{Bew}(R_n(n)).$$

Gödel's Incompleteness

- Represent this definition of K inside of P by a class expression $S = R_q$:

$$P \vdash S(q) \iff q \in K \iff \overline{Bew}(R_q(q)).$$

Theorem

$R_q(q)$ is neither provable nor refutable in system P (assuming consistency):

Proof:

$$\begin{aligned} P \vdash R_q(q) &\implies P \vdash S(q) && (S = R_q) \\ &\implies P \vdash q \in K && (S(x) \leftrightarrow x \in K) \\ &\implies P \vdash \overline{Bew}(S(q)) \\ &\implies P \nvdash S(q) \end{aligned}$$

Gödel's Incompleteness

Proof (cont'd)

$$
\begin{aligned}
P \vdash \overline{R_q(q)} &\implies P \vdash \overline{S(q)} && (S = R_q) \\
&\implies P \vdash \overline{q \in K} && (\text{def of } S) \\
&\implies P \vdash Bew(R_q(q)) && (\text{def of } K) \\
&\implies P \vdash R_q(q) && (\text{representability of Bew})
\end{aligned}
$$

Contradiction with P being consistent. □

Gödel	Post
Predicate R(x)	Normal System (S,P)
Bew: Semi-representation of theorem predicate	Normal-Deductive-System
Diagonalization $n \in K \leftrightarrow \overline{Bew}(R_n(n))$	Diagonalization $N' \vdash a^m \leftrightarrow N \vdash b(S_m, a^m)$
Truth and Semantics	Derivability (syntax only)
Extensions of System P	Any symbolic logic
ω-consistency	*n/a* – No universal quantifier
Consistency	not both $N \vdash P$ and $N \vdash bP$

Thank You!

Reduction

Canonical Form X is reduced to Canonical Form Y if

- to any system S_1 in Canonical Form X with alphabet A
- there exists a system S_2 in Canonical Form Y with alphabet A', where $A \subseteq A'$

  and the theorems of S_2 involving only the letters of A are the same as the theorems of S_1

Thus, a solution of the finiteness problem for Canonical Form Y(or S_2) can easily be transformed into one for Canonical Form X (or S_1).

Post's Thesis - A Counterexample?

- Now we can define the following 'diagonal' set D of a-sequences:

  a^m is in D iff it is not an assertion of the m-th Normal System.

$$a^m \in D \iff a^m \text{ is } \textit{not} \text{ an assertion of } \sigma(m)$$

- By definition, this set differs from each of the subsets of assertions of Normal Systems: it disagrees with the m-th Normal System on a^m.

Post's Thesis - A Counterexample?

- According to the Thesis, *D* must be generated by some Normal System. So it *seems* we have a counterexample to Post's Thesis.
- However, Post emphasizes

> *"we have merely defined a set of a-sequences, whereas to yield a true counter-example we must show how to generate that set, i.e., set up a system of "combinatory iteration" whose operations would at some time yield each and every a-sequence in that set, but would never yield an a-sequence not in the set."*

Post's Thesis - A Counterexample?

- So we have that

  Post's Thesis holds $\iff$ Normal Systems are *undecidable*.

- Post's difficulty in giving a decision procedure for Normal Systems committed him more to his thesis.
- So now we prove undecidability of Normal Systems based on the Thesis.