Gödel's Incompleteness Theorems: The Plaintext Distillation

Edward C. Larson
eclars1071@gmail.com

Abstract

Gödel's intuition accurately zeroed in on a core truth: that the expressiveness of non-trivial mathematical theories outpaces the efficacy of any inference engine. His paradoxical form of argumentation, however, begat immense confusion and pessimism, which this essay examines critically and aims to undo. His actual technical analysis exposed a historic oversight in classical set theory: the lack of internal structural variation within the Cantor sets $\aleph_0$, $\aleph_1$, etc. The paradoxical argumentation amounts to an indirect proof (reductio ad absurdum) of internal differentiation within $\aleph_0$.

This paper introduces a new concept of *structural grade separation* (SGS) within the Cantor sets and examines their foundational role and implications within formal mathematical frameworks, challenging the traditional conception of coarse set cardinality. By introducing a comprehensive SGS scheme, we demonstrate how structural stratification illuminates transfinite set theory without sacrificing logical consistency. The theoretical bounds of this construction against established axioms of arithmetic and set theory are analyzed, illustrating how explicit structural hierarchy resolves ambiguities inherent in standard set theory and cardinal assignments. The broader mathematical significance of graded set structures for decision algorithms and formal reasoning engines are explored.

Contents

Introduction

Gödel's seminal 1931 paper, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme* [3], investigated a centrally important topic at the heart of mathematical foundations. The half-century that preceded him had made immense strides in applying, building upon, and generalizing the axiomatic and deductive framework of the ancient Greek geometers (*e.g.*, Euclid, Pythagoras, Eratosthenes) to much of modern mathematics. Hilbert, in 1900 [6], set forth an optimistic and ambitious overarching programmatic goal, which, in essence, sought to apply mechanized theorem proving machinery to the whole of mathematics. It was an early-day vision that coincides closely with modern-day inference engines and reasoning systems that originated in the second half of the 20th century and have become immensely important in contemporary AI engine backends.

1 Formal Theory Architecture and Construction

The subject matter of mathematics proper can be formalized rigorously and modeled, as a knowledge engineering endeavor of meta-mathematics, as a directed acylic graph (DAG) of THEORY modules[1].

The material introduced in this subsection encompasses, to a moderate degree of comprehensiveness, the vocabulary elaboration, grammar, and production rules - in a word, the *expressiveness* - of formal theories of pure mathematics.

1.1 TYPE Lists

Formal theories are strongly typed. A THEORY always features a fixed finite list of TYPE declarations, which begins with three universally recognized types central to Boolean logic, shown in Table 1.

Table 1: Universal Types of Boolean Logic Preamble

Index	Name	Form	Definition
0	BOOLEAN (B)	atomic	-
1	unary_swFn	function	[0] → 0
2	binary_swFn	function	[0, 0] → 0

The first TYPE is *atomic*, meaning that objects of the type (constants FALSE and TRUE) have no internal substructure. BOOLEAN (B) is a special type of universal significance, recognition of which is built into the machinery of all inference engines.

The next two type entries represent unary and binary combinational switching functions respectively. The form label *function* indicates that they are multi-input single-output (MISO) functions, each with a fixed finite number of input arguments. The definition notation indicates the indices of the input and output types, which necessarily precede the function type entry in the list. The minimal extension of Table 1 required to encompass Peano's axiomatic construction of the natural numbers (*i.e.*, successor arithmetic) is in Table 2.

Table 2: Essential Types of Successor Arithmetic

Index	Name	Form	Definition
3	NATURAL (N)	atomic	-
4	unary_natPred	function	[3] → 0
5	unary_natFunc	function	[3] → 3

Atomic and *function* are the most common TYPE forms, but are by no means all-encompassing. Two useful additions include: (*i*) restricted function types, and (*ii*) stack-based type schemata.

[1] THEORY, all capitalized, connotes a declarative keyword for knowledge development modules.

The basic function types in Tables 1-2 are total functions. That is, they are guaranteed to yield a well-defined output object for any combination of valid objects of designated input type in each argument slot. For restricted function types, the output object is well-defined if and only if the inputs collectively satisfy a certain condition. The type definition specifies the condition as a constant (CONST list entry) of type $[T_1, \ldots] \to B$, whose input signature, $[T_1, \ldots]$, is the same as that of the restricted function type. Conditional constants can also be used to provide split IF-ELSE definition of total functions.

Stack-based type schemata are similar to basic function types, but with the crucial difference that the input signatures are indefinitely long. A schema is actually a sequence of function types, as indicated in Table 3.

Table 3: Type Schema for Input Type T_1, Output Type T_2

Notation	Interpretation
TUPLE(0)$[T_1] \to T_2$	T_2
TUPLE(1)$[T_1] \to T_2$	$[T_1] \to T_2$
TUPLE(2)$[T_1] \to T_2$	$[T_1, T_1] \to T_2$
TUPLE(3)$[T_1] \to T_2$	$[T_1, T_1, T_1] \to T_2$

1.2 CONST Lists

A THEORY always features a fixed finite list of CONST declarations, which represent typed object instances (constants) of particular importance in the theory. The CONST list header, which goes hand-in-hand with Table 1, is shown in Table 4.

Table 4: Universal Constants of Boolean Logic Preamble

Index	Name	Type	Definition
0	FALSE	0	-
1	TRUE	0	-
2	NOT	1	-
3	AND	2	-
4	OR	2	-
5	IMPLIES	2	$\lambda\, p, q : \mathrm{OR}(\mathrm{NOT}(p), q)$

The first five entries, whose names are self-explanatory, are implicitly defined, based on their integral roles in the inference engine machinery. The sixth entry, IMPLIES, comes with an explicit definition in the form of a lambda functional expression. IMPLIES is not strictly necessary, but it is customarily included in the universal preamble because it is so centrally important in traditional logic and modern reasoning systems.

1.3 Peano Axioms

The minimal extension of Table 4 required to encompass Peano's axiomatic construction of the natural numbers is in Table 5.

The three Peano axioms, A1-A3, are explicitly defined. The constants 0 and S are defined implicitly based on the references to them in the axiom definitions. Note that the predicate variable, P, in A3 is of type $[N] \to B$ (index 4 in Table 2).

1.4 Theory of Equality

Axioms A1-A2 contain an equality operator ('='), which is defined in a simple generic theory of equality, with TYPE and CONST list extensions of Tables 1 and 4 respectively:

Table 5: Constants of Successor Arithmetic

Index	Name	Type	Definition
6	ZERO (0)	3	-
7	SUCCESSOR (S)	5	-
8	A1	0	$\forall n : \mathrm{NOT}(\mathrm{S}(n) = 0)$
9	A2	0	$\forall n_1, n_2 : (\mathrm{S}(n_1) = \mathrm{S}(n_2)) \Rightarrow (n_1 = n_2)$
10	A3	0	$\forall P : \mathrm{AND}(P(0), \forall n : P(n) \Rightarrow P(S(n))) \Rightarrow (\forall n : P(n))$

Table 6: TYPE List for Generic Theory of Equality

Index	Name	Form	Definition
3	arb_type	atomic	-
4	unary_arbPred	function	[3] $\rightarrow$ 0
5	binary_arbPred	function	[3, 3] $\rightarrow$ 0

Table 7: CONST List for Generic Theory of Equality

Index	Name	Type	Definition
6	EQ (=)	5	$\lambda\, x_1, x_2 : \forall p : p(x_1) \Rightarrow p(x_2)$
7	LEMMA_reflexivity	0	$\forall\, x : x = x$
8	LEMMA_symmetry	0	$\forall\, x, y : (x = y) \Rightarrow (y = x)$
9	LEMMA_transitivity	0	$\forall\, x, y, z : \mathrm{AND}(x = y, y = z) \Rightarrow (x = z)$

The first row of Table 7 contains the Leibniz definition of equality. The TYPE and CONST lists in Tables 1, 4, 6, and 7 provide the vocabulary and grammar foundation on which the reflexivity, symmetry, and transitivity lemmae can be posed - and proved as theorems. Once the lemmae have been proved, the theory of equality can be "exported" as a THEORY module, with arb_type instantiated by a user-defined type in an importing THEORY (*e.g.*, NATURAL[2] in the basic theory of successor arithmetic).

1.5 Recursive Function Definitions

The simple Peano theory of successor arithmetic, in Tables 2-5, uniquely determines the size and structure of the natural numbers. NAT provides the logical foundation of inductive reasoning and expand the rules of grammar to encompass recursive function definition.

For an object, F, of type [N] $\rightarrow$ T, a recursive function definition can be supplied by defining a base (b) and step generator (g), *viz.*,

$$\mathrm{F}(0) = \mathrm{b} \tag{1a}$$

$$\mathrm{F}(\mathrm{S}(n)) = \mathrm{g}(\mathrm{F}(n), n) \tag{1b}$$

The base, b, is a constant of type T. The generator, g, is of type [T] $\rightarrow$ T or [T, N] $\rightarrow$ T. The second input argument, which is optional, provides explicit index dependence.

Ordinary addition ('+') is defined recursively, with T as [N] $\rightarrow$ N. The base and generator definitions are:

$$\mathrm{b} = \lambda\, n : n \tag{2a}$$

$$\mathrm{g} = \lambda\, f : \lambda\, n : \mathrm{S}(f(n)) \tag{2b}$$

[2]NATURAL is abbreviated as NAT or N, for convenience as needed.

From the definitions in Eqs. 2, it follows that:

$$+(x)(0) = x \tag{3a}$$
$$+(x)(\mathrm{S}(y)) = \mathrm{S}(+(x)(y)) \tag{3b}$$

which can readily be converted to familiar form with '+' as an infix operator.

1.6 Recursive Function Definitions on Type Schemata

Recursive definition can be provided for a sequence of constants F(0), F(1), F(2), ..., in which F(n) is of type TUPLE(n)[T_1] $\rightarrow \mathrm{T}_2$. Eqs. 1 now take the modified form:

$$\mathrm{F}(0)() = \mathrm{b} \tag{4a}$$
$$\mathrm{F}(\mathrm{S}(n))(\underline{x}) = \mathrm{g}(\mathrm{F}(n)(\underline{x}_{.\mathrm{rest}}), \underline{x}_{.\mathrm{last}}) \tag{4b}$$

in which b is a constant of type T_2, just as in Eq. 1a. Eq. 4b, however, is substantially different from Eq. 1b. $\underline{x}$ denotes a list of length S(n), which is treated on the right-hand side as a stack, whose last (top-most) element is denoted as $\underline{x}_{.\mathrm{last}}$. $\underline{x}_{.\mathrm{rest}}$ denotes what remains of the stack - a list of length n - after the last element is removed. The empty pair of parentheses on the left-hand side of Eq. 4a is a *pro forma* placeholder for an empty stack argument. Explicit index dependence has been omitted, for simplicity.

As an example, this form of recursive function definition can be used to define distinctness among elements of a list, which is a key definitional criterion in set theory. This proceeds in two steps. First, a list membership function, $\in$, which is of type TUPLE(n)[T] $\rightarrow$ [T $\rightarrow$ B], is defined, in which T is the set element type. The membership function is defined recursively, *viz.*,

$$\in(0)() = \lambda\, y : \mathrm{FALSE} \tag{5a}$$
$$\in(\mathrm{S}(n))(\underline{x}) = \lambda\, y : \mathrm{OR}(\in(n)(\underline{x}_{.\mathrm{rest}})(y), y = \underline{x}_{.\mathrm{last}}) \tag{5b}$$

The left-hand sides of Eqs. 5 essentially denote a method that operates on an instantiated list, $\underline{x}$. The right-hand sides indicate the result obtained when the method is interrogated with an arbitrary value, y, of type T. Eq. 5a always returns FALSE, because the zero-length list is empty. Eq. 5b returns TRUE if y coincides with the last element or is in the rest of the list.

With the membership function in hand, distinctness can be defined as a function of type TUPLE(n)[T] $\rightarrow$ B, *viz.*,

$$\Delta(0)() = \mathrm{TRUE} \tag{6a}$$
$$\Delta(\mathrm{S}(n))(\underline{x}) = \mathrm{AND}(\Delta(n)(\underline{x}_{.\mathrm{rest}}), \mathrm{NOT}(\in(n)(\underline{x}_{.\mathrm{rest}})(\underline{x}_{.\mathrm{last}}))) \tag{6b}$$

Eq. 6b indicates that the elements of $\underline{x}$ are distinct if (*i*) distinctness has already been established for $\underline{x}_{.\mathrm{rest}}$, and (*ii*) $\underline{x}_{.\mathrm{last}}$ does not coincide with any elements of $\underline{x}_{.\mathrm{rest}}$.

2 Inference Engines

The preceding subsection provides an extensive framework through which strongly-typed expressions can be constructed, through combination of vocabulary (TYPE and CONST lists) and grammar (production rules). Attention now turns to inference engines, which are tasked with classifying Boolean-valued expressions as TRUE or FALSE.

2.1 Sequent Preparation

The inference engine is tasked with executing a query, which is presented as a *sequent*, *viz.*,

A_1
A_2
A_3
...
$\vdash$
C_1
C_2
...

The A's and C's respectively represent antecedents and consequents, or axioms and conjectures, separated by a turnstile ($\vdash$). The sequent represents a proof challenge, which is discharged in the affirmative (marked QED) unless it is possible for all antecedents to be TRUE and all consequents to be FALSE. The following are thus *sufficient* conditions to claim QED: (*i*) any antecedent is FALSE, (*ii*) any consequent is TRUE, or (*iii*) an antecedent matches (*i.e.*, is syntactically identical to) a consequent.

In general, the closer to TRUE the antecedents and the closer to FALSE the consequents all are, the more boldly the sequent exposes itself to falsification. The greater the exposure, the more interesting, impressive, and powerful a successfully met proof challenge can be said to be. Antecedents close to TRUE signify parsimonious unrestrictive assumptions; consequents close to FALSE signify strong highly specific claims.

The initially prepared sequent equates to the Boolean-valued expression

$$\mathrm{AND}(A_1, \ldots) \Rightarrow \mathrm{OR}(C_1, \ldots)$$

in which the A's and C's are all "finished" expressions, in which all variable tokens within the expression syntax are bound to $\forall$, $\exists$, or λ qualifiers. Only finished expressions[3] are permitted in CONST list definitions.

The initial sequent usually consists of just one consequent, C_1, which is interpreted as a *conjecture* that a mathematician seeks to prove. The axioms of the formal theory (*e.g.*, Peano A1-A3) must be explicitly supplied as antecedents, unless they are effectively implicit in the built-in operating rules of the inference engine.

2.2 Sequent Grinding with Fixed Variables

Once an initial sequent has been prepared, it is submitted to an inference engine (IE) as a query. The IE embarks on an executional process, which can be called *sequent grinding*. The process can be introduced most simply in the context of combinational logic analysis. Consider, for example, the following sequent:

$\vdash$
$\forall x_1, x_2, x_3 : \mathrm{ONE}(x_1, x_2, x_3) \Leftrightarrow \mathrm{AND}(\mathrm{OR}(x_1, x_2, x_3), \sim\mathrm{MAJ}(x_1, x_2, x_3))$

which asserts an identity of 3-ary combinational logic. ONE and MAJ are named functions[4], which would be entries in the CONST list with explicit definitions in terms of Boolean primitives (AND, OR, NOT).

The first sequent grinding step is to dissolve the $\forall$ qualifier by replacing the bound variables with *fixed variables*: $x_1 \rightarrow \xi_{0,0}$, $x_2 \rightarrow \xi_{0,1}$, $x_3 \rightarrow \xi_{0,2}$. The sequent becomes:

$\vdash$
$\mathrm{ONE}(\xi_{0,0}, \xi_{0,1}, \xi_{0,2}) \Leftrightarrow \mathrm{AND}(\mathrm{OR}(\xi_{0,0}, \xi_{0,1}, \xi_{0,2}), \sim\mathrm{MAJ}(\xi_{0,0}, \xi_{0,1}, \xi_{0,2}))$

In the notation, $\xi_{t,i}$ denotes an arbitrary non-descript instance of type t (index into the TYPE list); i is a serial number determined at the time of creation in the IE execution.

The sequent is now in a syntactically decomposable form, and the main work commences. In reckoning with a work-in-process sequent, such as the last above, the IE generally has a number of incremental progress steps at its disposal. The simplest such steps are: (*i*) NOT dissolution and movement of the argument to the opposite side of the turnstile, (*ii*) antecedent-side AND flattening, and (*iii*) consequent-side OR flattening. As a general rule,

[3] In the terminology of natural language grammar, finished expressions equate to independent clauses.
[4] ONE outputs TRUE if and only if exactly one of the inputs is TRUE. MAJ is the majority function.

these simplification steps, when available, are always made, as they guarantee riskless progress without complexity proliferation.

Further incremental progress steps include:

- **Definitional Substitution**: If the top-most token of an antecedent or consequent expression is a CONST list entry with an explicit definition, the token may be replaced with the definitional expression.
- **Antecedent-Side OR Splitting**: If the top-most token of an antecedent is an OR functor, the sequent is split into a set of parallel proof challenges. In each of those, the antecedent expression is replaced by one of the OR arguments. A more restrictive condition is thus introduced on the antecedent side, making the proof challenge easier. The pre-split sequent is discharged if all of the parallel proof obligations are discharged.
- **Consequent-Side AND Splitting**: If the top-most token of a consequent is an AND functor, the sequent is split into a set of parallel proof challenges. In each of those, the consequent expression is replaced by one of the AND arguments.

When all of the aforementioned options have been exhausted, the results, in the combinational logic application, are sequents whose assertion components are all bare fixed variables on the antecedent or consequent side. The last and only remaining option is then cross-matching. A reduced sequent is discharged if and only if at least one fixed variable occurs on both the antecedent and consequent sides.

Combinational logic is a particularly simple kind of theory, such that all sequents prepared on conjectures expressible within the theory simplify in accordance with a common algorithmic prescription. An IE strategy that systematically implements the repertoire of incremental progress steps introduced thus far will always succeed[5] in returning QED, provided that the submitted conjecture is, in fact, correct. If the conjecture is not correct, the IE will instead return INCONCLUSIVE. Combinational logic is completely decidable, meaning that for any conjecture (C), either C or its negation, NOT(C), yields QED.

2.3 Sequent Grinding with Free Variables

Consider the symmetry lemma (Table 7) in the theory of equality. The initially prepared sequent is:

$$\vdash$$
$$\forall x_1, x_2 : (x_1 = x_2) \Rightarrow (x_2 = x_1)$$

Dissolution of the $\forall$ qualifier, followed by elementary simplification, yields:

$$\forall p : p(\xi_{3,0}) \Rightarrow p(\xi_{3,1})$$
$$\xi_{4,0}(\xi_{3,1})$$
$$\vdash$$
$$\xi_{4,0}(\xi_{3,0})$$

in which the type indices for the fixed variables are those in Table 6. Movement of the antecedent-side $\forall$ to the consequent side, through the DeMorgan relation, yields:

$$\xi_{4,0}(\xi_{3,1})$$
$$\vdash$$
$$\xi_{4,0}(\xi_{3,0})$$
$$\exists p : \mathrm{NOT}(p(\xi_{3,0}) \Rightarrow p(\xi_{3,1}))$$

Dissolution of the $\exists$ qualifier is similar to that for $\forall$, except that the bound variables are instead replaced with

[5] Provided that the number of variates is small. With moderately large numbers of variates, BSAT adversity sets in.

free variables: $p \rightarrow X_{4,0}$. The sequent becomes:

$X_{4,0}(\xi_{3,0}) \Rightarrow X_{4,0}(\xi_{3,1})$
$\xi_{4,0}(\xi_{3,1})$
$\vdash$
$\xi_{4,0}(\xi_{3,0})$

Whereas fixed variables are, in effect, opaque non-descript constants, free variables may be instantiated in any way that enables a proof challenge to be satisfied. When only fixed variables are involved, such as in combinational logic, results of proof challenges are either unconditionally affirmative (QED) or inconclusive. With free variables, on the other hand, proof obligations are instead discharged conditionally, with a certain set of potentially viable instantiations.

When sequents are split, the child sequents each return a set of instantiation candidates. Most of those are not compatible with instantiations returned by siblings and are thus eliminated. The reconciliation process is ultimately concerned only with whether there exists *at least one* viable instantiation compatible with all the child results. In the case of the last sequent above, the final instantiation result is

$$X_{4,0} = \lambda\, y : \mathrm{NOT}(\xi_{4,0}(y)) \tag{7}$$

The discovery of a viable instantiation converts the proof obligation result, in the sequent in which $\exists$ was dissolved, to unconditional QED. The symmetry lemma is proved.

2.4 Inferencing Outcomes

In the general scenario, a formal theory is developed, a conjecture (C) is posed, and the IE executes. If all goes well, the IE returns QED in a timely manner. Otherwise, it either returns INCONCLUSIVE or fails to terminate.

If C is discharged in the affirmative, the same must not happen for NOT(C). If it does, we are in trouble. The purported proofs of C and NOT(C) unmistakably indicate that the theory is inconsistent and thus immediately invalidated. Dunce cap issued.

Much more often in practice, C and NOT(C) both fail to discharge affirmatively. Several diagnoses are then possible:

- Whereas an inconsistent theory is *overconstrained* in its axiomatic stipulations, it may instead be that the theory is *underconstrained*. In such cases, the knowledge model under construction within the theory is vague and ill-developed, such that the indeterminacy of C and NOT(C) is symptomatic of an unsecured degree of freedom in the domain knowledge specification.

- The theory itself may be well-constrained, but the IE is too scantly implemented. Well-guardrailed nevertheless, it is self-aware of its limitations and promptly pleads ignorance upon encountering a sequent for which all easy incremental progress options have been exhausted.

- It may be that the theory is well-constrained and the IE is as well-developed, well-guardrailed, and as powerful as can reasonably be expected. However, the conjecture is inherently too difficult to prove, at least within the scope of the theory in which it is natively expressed.

The last possibility describes a theory that is *undecidable* with respect to a particular conjecture, C. That is, it is impossible to mark C or NOT(C) as QED, based on fundamental limitations of attempting to reason within the confines of the native theory. That implies either of two more profound diagnoses:

- It may be that a more powerful theory can come to the rescue. As a spectacular example, this was conclusively found to be the case for Fermat's Last Theorem, which had been an open question for some 350 years until it was resolved affirmatively in 1994. Wiles' proof of FLT [17] invoked advanced theory of modular forms and Galois representations, which substantially exceeds the inferential scope of the theory of ordinary standard arithmetic in which $x^n + y^n \neq z^n$ is natively expressed.

- Goldbach's conjecture [4], on the other hand, remains unresolved. It is an open question whether there exists a more powerful theory that can come to the rescue, or whether the conjecture amounts to a *floating island*. The latter signifies the possibility that there fundamentally does not exist any deductive path to axiomatic bedrock. The nonexistence of any deductive path, it is noted and emphasized, is a matter of objective Platonic reality, not epistemology (*i.e.*, pertaining to the sheer *difficulty* of finding a path).

A complicating issue, at the nexus of ontology and epistemology, is that conjectures may de decidable (because solutions *do* exist), but the decision *process* is not computationally feasible in any practical sense. Two examples:

- BSAT I (introduced further on) is a steganography problem that is fundamentally impervious to analytic methods and lacks any efficient solution technique.
- The possibility of Goldbach counter-examples. The first counter-example, for all we know, could be 2^{128}, 10^{316}, or Graham's number. The falsity of Goldbach's conjecture would imply that resolution of the question lies in the mundane realm of brute-force search, as opposed to advanced analytic reasoning (as in FLT). Littlewood's 1914 theorem [10], pertaining to the Riemann zeta function (RZF), demonstrates that such stupendously large numbers *cannot* be shrugged off.

2.5 Inferencing Strategy

It is apparent that inferencing strategy is an art more than a science. For any work-in-process sequent (exceedingly simple applications notwithstanding), the IE has a wide choice of incremental options at its disposal. All are valid inferencing rules that accord with what can fairly be called *generally accepted reasoning principles* (GARP[6]).

Generalists

The combinational logic application exemplifies a class of problems that can be solved using a static inferencing strategy, which can be summarized as follows:

- **Step 1**: Apply NOT dissolution, antecedent-side AND flattening, and consequent-side OR flattening, when available, to realize *it situ* sequent simplification.
- **Step 2**: Scan the resulting reduced sequent for antecendent-side FALSE, consequent-side TRUE, or cross-matching of fixed variables. Return QED or INCONCLUSIVE.
- **Step 3**: Apply definitional substitution, antecedent-side OR splitting, and consequent-side AND splitting, when available. Iteratively perform Steps 1-3, until exhausted.
- **Step 4**: Collect, reconcile, and merge results from all reduced sequents. Output the final result.

This mechanized recipe can be regarded as a generalist IE strategy, which aims to be all things to all mathematicians. As a practical matter, however, it can never realistically aspire to be more than a limited panoply of common basic utilities.

Specialists

A pervasive difficulty is that the IE, for a given work-in-process sequent, typically has a wide variety of choices for incremental progress steps, all GARP-compliant. However, the most propitious choice is seldom apparent in the local contexts of proof challenges. Except for the very few riskless steps already noted, most incremental step choices pose downside risk. As examples, sequent splitting expands the number of proof challenges, and it will be for naught if two or more cannot be met. Definitional substitution tends to proliferate complexity by replacing a CONST label with a composite expression, which would be unnecessary and wasteful if cross-matching of the CONST label would much more quickly discharge the sequent. In plainer terms, most incremental step options risk running down rabbit holes.

Beyond very simple applications at small scale, generalist strategies are inadequate. What is instead usually required are tailored inferencing strategies built around the specific nature of a particular conjecture and the

[6]GARP is only concerned with the *legality* of inferencing steps, not with whether they are practical, efficienct, or well-chosen.

specialized knowledge domain in which it is couched. They must often draw upon advanced theories far beyond the native theories in which the conjectural expressions originate. As a broad generality, specialist engines are designed pragmatically and narrowly, oftentimes exclusively for the purpose of trying to solve a single problem. The narrower, the more closely they approximate the asymptotic limit of idiot savant.

3 The Central Difficulty

Gödel had a flash of brilliant truthful insight, whose plaintext distillation is a concise maxim:

The expressiveness of non-trivial mathematical theories outpaces the efficacy of any inference engine.

It is never quite put that way, but it is the most direct and straightforward way of interpreting Gödel's work, which stands 95 years later as an intuitive attestation of conviction that Hilbert's optimistic ambitious vision that generalist proof engines could conquer all of mathematics was fundamentally doomed. Hilbert's *finitistic* program ([6], [7]) essentially envisioned a universal proof engine that could be all things to all mathematicians.

3.1 Expressiveness

Expressiveness is cheap and easy; analysis is costly and hard. Expressions readily spring forth from production rules, which are mechanical recipes prescribing how the rules of grammar combine with vocabulary to generate complexity within a formal language or theory. Rules, for the most part, are unrestricted and easy to apply.

In general, expression complexity scales exponentially, *viz.*, b^P, in which b signifies an alphabet size. Without loss of generality, $b = 2$. The exponent, P, then signifies information-theoretic entropy, or, equivalently, the minimum description length (MDL) of an expression in a Shannon-optimal bit encoding[7] of the formal language. In this respect, recursive function definition is extremely important, because it gives MDL-compact expression to infinitudes of objects and objects of arbitrarily large magnitude.

3.2 BSAT

The mismatch between expressive ease and analytic intractability is well-exemplified by the BSAT (binary satisfiability) problem. BSAT poses what is essentially a randomly generated amorphous combinational logic expression, consisting of P binary variates combined through conjunctive (AND), disjunctive (OR), and negation (NOT) operators. In general, P is moderately large, *i.e.*, such that P itself is tractable in magnitude for practical purposes of storage and scanning (*i.e.*, elementary computational operations), whereas 2^P is too big. If some variates are written more than once, the expression generally becomes impervious to simplification.

BSAT formalizes the inscrutability of the P-ary switching function that was generated and stands formulaically. It tells us that it is intractably hard to prove or disprove so much as an existentially qualified assertion about the function, *e.g.*, whether or not it is identically zero (*i.e.*, output column 0 in all rows). The only easy recourse is limited to point queries (*i.e.*, direct evaluation of the function for a given tuple of variate instantiations), but that is utterly futile. It merely provides inspection of a small sample of rows in a truth table with 2^P rows. If it were practically feasible to compute and scan all rows, any question of the kind posed by BSAT could be readily answered by brute force.

3.3 Austere BSAT

BSAT, as posed in that conventional way, does not readily click, as it does not initially make sense or seem at all probable that an arbitrary amorphous combinational logic expression could equate identically to zero. The confusion is well-taken and calls for a simplified austere problem formulation.

Suppose that a scribe secretly generates a P-ary switching function randomly at will, with (i) probability 0.5 of being identically zero, or (ii) probability 0.5 of being zero in all but one of 2^P rows. The scribe hands us a compiled implementation of the function, which can only be used to make point queries. We then have a needle-in-a-haystack conundrum and are not sure whether a needle exists in the first place. It is a steganographic challenge. Any reasonable number of queries is almost certain to yield all zeros, and we are at a loss to prove or disprove the existentially qualified conjecture.

[7]Entropy and MDL are precisely equivalent if and only if every bit has equal probability of being 0 or 1.

3.4 The Challenge of Making BSAT Challenging

The original BSAT problem formulation, with the arbitrary amorphous switching function and the question of whether it is identically zero, is almost certain to be answered quickly in the negative, as all that it takes is one query returning 0 and another query returning 1. How can a truly challenging BSAT question, for a given amorphous function, be composed?

The challenge of coming up with a difficult BSAT question itself exemplifies the intractability of analysis vis-à-vis the utter triviality of generating the function. The analysis strategy entails:

1. Identify not merely the one particular switching function (henceforth referred to as the *reference function*) that was generated, but the *topological family* of switching functions to which it belongs. This calls for partial mastery of the set of all P-ary functions, which is a space of size 2^{2^P}.

2. Consider next the set of 2^P neighboring functions, obtained by inverting the output column in a single row of the truth table. Any such steganographically perturbed function is virtually indistinguishable statistically from the reference.

3. The neighboring functions map to some number of topologies. For any one of those, come up with an MDL-concise question, pertaining to a property that holds for the reference but not the neighboring family. This becomes the BSAT challenge.

The first two steps are easy. Beginning with the moderately long amorphous expression of the reference function, the scribe secretly chooses, equiprobably, whether or not to apply a steganographic alteration. If the decision is made to perturb, the scribe generates a conjunctive (AND) combination of the P variates, some of which are inverted through a random bit mask of the scribe's choosing. The resulting AND expression is then XOR'ed with the reference expression.

The third step is the truly difficult part, as it requires identification and characterization of the topological families to which the reference and steganographic candidates belong. The complexity of this challenge, which pertains directly to the intrinsic complexity of the various families, scales as 2^{2^P}, whereas the intractability of austere BSAT, which involves just two simple known families, scales as 2^P.

3.5 Steganographic Asymmetry

In the austere BSAT problem formulation (BSAT I), only two simple topological families, known to both the scribe and us, are involved. One is the pair of functions in which the output column of the truth table is all zeros or all ones. The other is the set of functions - 2^{P+1} in all - in which the output column is uniform except for one row. Half of the functions can be expressed as AND combinations of the variates, inverted or uninverted; the others can be expressed as OR combinations.

The austere BSAT challenge has to do purely with informational asymmetry. The scribe applies a steganographic dithering (or perhaps not), to which we are not privy. We receive only a compiled implementation of the function, which cannot be reverse-engineered[8] via point queries.

The impracticality of reverse-engineering a steganographic dither has only to do with the lack of a flat numeric representation of the one function that the scribe chose. Were it feasible to perform point queries on all 2^P rows (*e.g.*, for $P \approx 100$), the challenge would be surmounted. Furthermore, there does not exist, as a matter of principle, any analytic technique to aid our search for the needle or to determine whether a needle exists. Brute force search is fundamentally the only option and fails by computational exhaustion except for small P.

3.6 Information-Theoretic Divergence

In the original BSAT formulation (BSAT II), the scribe applies the production rules of the formal language of switching function theory in essentially willy-nilly fashion, to generate an amorphous expression. We can stand over the scribe's shoulder as this as carried out. The secretive part comes afterwards, but is actually irrelevant to the fundamental considerations that make BSAT I and BSAT II altogether different problems.

[8] Reverse engineering would recover the source code, which would reveal the scribe's secret choices.

BSAT II is not about cryptographic asymmetry, as the scribe is not withholding any secrets. The scribe is no less at a loss than we are in wondering what he has wrought. The essential challenge of BSAT II, in plain simple terms, is to discern what there is to say about the function that is interesting. Even if it were feasible to scan all 2^P rows of the truth table, much of what could be readily ascertained would not qualify as "interesting", and it would certainly not be the whole story. The interesting properties are not of the particular function itself. but rather, of the *family* to which it belongs.

Topological familes are equivalence classes of switching functions, members of which are equivalent through transitive closure of the following operations: (*i*) permutations of the input variates, (*ii*) negation of any input variate, and (*iii*) negation of the function output. Examples of family properties include: (*i*) symmetries, (*ii*) tree decomposability of the function, (*iii*) separability or unateness, (*iv*) efficient representational structures (*e.g.*, feedforward neural networks), and (*v*) overall complexity of the family compared to others. Features readily ascertainable from scanning of particular functions, by contrast, are limited and superficial (*e.g.*, balance of zeros and ones in the output column).

The space of switching functions scales as 2^{2^P}, which underscores the intractability of the analysis challenge. The information-theoretic divergence between expressive ease and analytic intractability is at the heart of what Gödel investigated and claimed in the realm of meta-mathematics.

3.7 Hashing Asymmetry

The information-theoretic divergence can be thought of as a one-way ratchet, similar in certain respects to cryptographic hashing. In both cases, it is easy, in essence, to generate an amorphous expression willy-nilly. The resulting expression maps to an object in a highly-dimensional space of size 2^P and also - and more importantly - to an equivalence class of objects in a space of size 2^{2^P}.

A password or specific snapshot version of a file is run through a standard hashing function (*e.g.*, MD5, SH1, ...), yielding an amorphous bit string of fixed size. Like the functional expression generated by the BSAT scribe, the hash is moderately long (*e.g.*, 512 bits). The equivalence class, analogous to the topological family, is the set of all other files (of varying lengths and all imaginable content types) that map to the one particular reference hash. The reverse-hashing challenge can be posed as follows: (*i*) Given the hash, find *any* file or password, different from the original, that maps to the same hash. Or even more challenging: (*ii*) Given a second file (different from and unrelated to the original, but separately meaningful in its own right), find a small steganographic noise perturbation of the second file (which, by assumption, need not be bit-perfect) that maps to the target hash.

3.8 Complexity, Expressed Simply

The common theme of the three aforementioned examples - steganography, BSAT II, and hashing - is ease of generation and MDL-conciseness on the one hand, versus intractability of solution (by brute-force or analysis) on the other. Complexity theory, as a broad thematic realm, offers no shortage of closely related examples, *e.g.*, chaos theory, fractals, dynamic systems, cellular automata, chess, card games.

3.9 Implications for Meta-mathematics

Expressiveness is immensely powerful in pure mathematics, just as in all of the preceding contexts. It readily poses a limitless multitude of analytic challenges that are, in all but the simplest theories, intractable. From a meta-mathematics perspective, that translates to the meekness of inference engines, which are tasked with fathoming deep semantic structure in complex objects, which is ultimately about the depth of their footings down to axiomatic bedrock. Finding such footings is extremely difficult and generally not feasible, as it has directly to do with core information-theoretic divergence on orders of 2^P or 2^{2^P}.

Hilbert's original conception of finitistic inference engines may have been naively optimistic and overly restrictive, but it was fundamentally not ill-taken or arbitrary. Finitism, in properly generalized perspective, speaks accurately to the incremental step-by-step local topography of deductive paths connecting theorems to axiomatic bedrock.

4 Meta-mathematics

Meta-mathematics is the formal systematic study *of* mathematics proper, as a knowledge-building endeavor. It deals with the delicate matters of ontology (*i.e.*, absolute mathematical truth) and epistemology (*i.e.*, what inference engines are actually capable of proving).

4.1 Axiomatic Foundations

The progressive development of mathematics proper can be modeled as a directed acylic graph (DAG) of THEORY modules, ranging from simple to advanced frontiers. The development of a formal theory comes with its vocabulary and production rules. Those are, broadly speaking, mechanically facile and safe. As was said, expressiveness is cheap and easy. Axioms, on the other hand, tread a much finer line. They may assert too much, assert too little, or be just right.

4.2 Overconstrained Theories

Consider the sequent:

$$\begin{array}{l} \mathrm{A} \\ \vdash \end{array}$$

in which A denotes the conjunctive bundling of the axioms of a formal theory. If the sequent discharges, it follows that the axioms are collectively inconsistent. Any consequent, regardless of content or truth value, can be placed below the turnstile, and the sequent still discharges, formally implying that the axioms prove anything.

Axioms, which are typically user-crafted in the formulation of specialist theories, are risky, in that it is often extremely difficult to prove that they are certifiably free of contradiction. Especially tricky is how different axioms, such as some pertaining to addition and others pertaining to multiplication, can interact with one another.

Inconsistent theories are so because their axioms assert too much and end up treading on their own toes. In this sense, the theories can be regarded as *overconstrained*. In general, the challenge of pure mathematics is to come up with parsimonious unrestrictive axioms (*i.e.*, mild departures from TRUE[9]) in return for highly specific claims with far-reaching significance and utility.

From the ontic standpoint in meta-mathematics, NOT(A) for an inconsistent theory is a Platonic truth, irrespective of whether we can prove it - or whether it is provable *in principle*.

4.3 Underconstrained Theories

It may be that a theory has an insufficient bundle of axioms. An axiomatic bundle, A, is *underconstrained* if there exists an "orthogonal" supplementary bundle, B, such that AND(A, B) and AND(A, NOT(B)) are both consistent. The opposite supplementary bundles both strengthen the axiomatic foundation, but engender different theories.

Classical set theory, as it took root in the late 19th century after Cantor's work [1], exemplifies an underconstrained theory. Strong structural constraint, such as set forth in ZFC[10] and Gödel's constructible set theory, represents a supplementary bundle (B) that yields a well-constrained theory in which the continuum hypothesis holds. That is, the structure of the resulting theory excludes objects of intermediate size between the first and second Cantor estates ($\aleph_0$ and $\aleph_1$). The opposite bundle - NOT(B) - yields a different theory featuring large cardinal extensions (*e.g.*, Cohen's proper forcing axiom).

4.4 Well-Constrained Theories

A theory is *well-constrained* if its axiomatic foundation is neither overconstrained nor underconstrained. The axioms stake out a position that is just right: neither too diffident nor too aggressive.

Well-constrained theories are especially important and the only ones of true interest, because they exclusively define mathematical *ontology*. A mathematical assertion can be said to objectively either true or false, in the

[9] TRUE can be regarded as the default antecedent baseline in the absence of any explicit axioms in a formal theory.

[10] ZFC denotes Zermelo, Fraenkel, plus axiom of choice.

Platonic sense, provided that the assertion is well-posed. An assertion is well-posed, by definition, if and only if it is couched in the expressive space of a well-constrained theory.

4.5 Easy and Difficult Proofs

But how can we ever *know* whether a theory is well-constrained? This is the central epistemic question of meta-mathematics. The short answer: Only when a more powerful theory comes to the support of a less powerful theory, in which case the latter can potentially be certified as well-constrained. This follows from the main implication for meta-mathematics of the asymmetry between expressive ease and analytic intractability:

- In general, only a tiny minority of assertions expressible within a theory are decidable within the inferential scope of the theory. These are what can be regarded as "easy" proofs.
- The dominant majority of assertions, including most interesting ones, are *not* decidable by the inferencing apparatus of the native theory. They are fundamentally reliant on a greater theory to come to their rescue, such as exemplified by Wiles' proof of FLT. For unsolved problems (*e.g.*, for Goldbach, RZF), not only do no known proofs exist, but it is also unknown whether any greater theory that can do the job exists at all.

As a simple example of the distinction, consider standard arithmetic, which is the basic theory of ordinary addition and multiplication. Primeness is inextricably a feature to which the theory gives expression. Consider the following four assertions:

- **4A**: There exist infinitely many 1 mod 4 primes.
- **4B**: There exist infinitely many 3 mod 4 primes.
- **6A**: There exist infinitely many 1 mod 6 primes.
- **6B**: There exist infinitely many 5 mod 6 primes.

4B and 6B are both easy to prove, using simple analysis[11] within the basic theory. No new concepts need be invented. 4A and 6A, on the other hand, are substantially more difficult to prove and require additional resources, *i.e.*, more TYPE and CONST definitions, and new theoretical development. They depend on a greater theory - specifically, the theory of Pythagorean numbers. The Pythagorean theory proves the following theorems, where m and n are co-prime positive integers of opposite parity[12]:

- **4C**: All prime factors of $m^2 + n^2$, which is itself 1 mod 4, are 1 mod 4.
- **6C**: All prime factors of $3m^2 + n^2$, which is itself 1 mod 6, are 1 mod 6.

The proof of 4A easily follows from the availability of 4C as a lemma, and proof of 6A easily follows from the availability of 6C. The greater theory proves the assertions expressible within the lesser theory.

4.6 Certification of Theories

The preceding subsection focuses on the decidability of particular assertions in a lesser theory, with the help of specialized insight from a greater theory. *Certifiability* signifies the more general question of whether the entire expression space of a lesser theory can be proved systematically to be consistent (*i.e.*, not overconstrained) or secured (*i.e.*, not underconstrained).

Gentzen's analysis [2] was historically one of the most significant milestones on this front. He proved that standard arithmetic, with ordinary addition and multiplication, is consistent. Together with the axiom of choice for finite sets, standard arithmetic is provably well-constrained, from which it follows that Goldbach's conjecture is Platonically certified as either true or false.

[11] Both are similar to Euclid's proof of primes infinitude.

[12] Opposite parity means that m is even and n is odd, or vice versa.

4.7 Complete Decidability

Complete decidability is the question of whether all expressible conjectures (or certain large classes of conjectures) in a theory can be proved decidable, with the help of insights from a greater theory. Gödel's First Incompleteness theorem (G1) directly addresses this topic, which touches the most interesting, important, and mysterious question of all: Do floating islands exist? Do there exist well-posed assertions that, by definition as such, are Platonically objectively true, yet are fundamentally detached from axiomatic bedrock, such that they are not provable using the full resources of mathematics?

Presburger's analysis of addition-only arithmetic [13] was historically one of the most significant milestones on this front. He proved that addition-only arithmetic, with multiplication deliberately excluded, is consistent and completely decidable, like combinational logic. That is, any conjecture, C, expressible within the theory - essentially, involving algebraic formulae consisting of finitely many integer summation terms - can be decided. That is C or NOT(C) - one or the other, but obviously not both - can be marked QED in a finite amount of time[13], using a standard repertoire of IE tailoring techniques. A specialist IE tailored to Presburger arithmetic is guaranteed to terminate in a reasonable amount of time for any conjecture, provided that the number of terms (P) is small, but BSAT adversity sets in as P becomes moderately large.

The complete decidability of standard arithmetic remains an unresolved open question, because of such difficult problems as Goldbach. The introduction of multiplication causes arithmetic to cross a critical threshold of complexity, opening a Pandora's box of analytic intractability (*e.g.*, prime number theory, Diophantine equations) that fundamentally precludes what Presburger was able to establish for addition-only arithmetic. Ring theory, which formalizes the interactions of addition and multiplication at the most basic level, is ground zero for exploring the essential nature of complications.

5 Gödel's Analysis

At this juncture, we turn attention to Gödel's historic seminal work. It is a complex topic, which is best approached and treated through three perspectives:

- Gödel's two incompleteness theorems, at face value as they were stated and have been generally understood for the last 95 years. They were, in historical actuality, attestations of intuitive conviction on Gödel's part about the implications of the core asymmetry for meta-mathematics. These represent the greatest and most lasting significance of his work.
- What Gödel actually addressed and proved, in his technical analysis of the famously self-referential assertion.
- The mythos, confusion, and intellectual defeatism that took root in the wake of his work.

5.1 Gödel's Incompleteness Theorems

The claims that Gödel made - and that came to be heard widely - are bold, powerful, and far-reaching:

- **G1**: Gödel's First Incompleteness Theorem pertains to complete decidability. He maintained, as an inferential leap from his analysis of the self-referential assertion, that standard arithmetic *does* feature floating islands. And indeed, that is generally understood to be what G1 claims.
- **G2**: Gödel's Second Incompleteness Theorem pertains to the question of consistency. He maintained that large classes of theories encompassing standard arithmetic cannot be certified for consistency.

G1 and G2 touch two of the three core bases at the heart of meta-mathematics. A third claim can be added:

- **G3**: Historically, there was no Third Incompleteness Theorem, but G3 can be thought of as the niche represented by the question of certifying theories as adequately secured (*i.e.*, not underconstrained).

G1-G3, as a trio, are the central questions that speak to how ontology and epistemology jointly shape the overall architecture of mathematics proper.

[13] Decidability is concerned with whether the IE *in principle* terminates, not with whether proof times scale well.

5.2 Demonstrations of Incompleteness

Gödel was the first to recognize that the fundamental mismatch between expressiveness and inferencing efficacy has profound implications for meta-mathematics. His intuition was accurate and later vindicated by two key discoveries, in combinatorics and number theory.

Ramsey Combinatorics

Paris and Harrington [12] investigated finite Ramsey theory, a branch of combinatorics pertained to order in large-scale structures. They came across a combinatorial assertion that can be expressed within the vocabulary and grammar of Peano arithmetic[14] and is certifiably Platonic, but yet is unprovable within Peano arithmetic.

According to the finite Ramsey baseline theory, which can be thought of as a generalization of the pigeonhole principle, coloration of small subsets of a sufficiently large finite set implies that a monochromatic subset of specified size (m) must exist. Paris and Harrington modified the Strengthened Finite Ramsey Theorem by adding a seemingly trivial "size rule", which holds that m must equal or exceed the least element value (infimum) of the subset. They found that the size rule causes the growth rate of the bounding function[15] exceeds that of any function whose totality can be proved within Peano arithmetic. The analysis implied the combinatorial assertion was fundamentally not provable within the confines and scope of Peano arithmetic, and that to break the impasse would require appealing to stronger axioms in a more advanced theory. The finding was highly noteworthy as a tangible example of G1 in mathematics proper - in contradistinction to the highly paradoxical self-referential mapping that Gödel had devised.

Goodstein Sequences

Goodstein [5] devised an arithmetical game, which involved starting with a number written in hereditary base-k notation, incrementing the base to $k+1$, subtracting 1, and repeating. In general, the sequence increases stupendously rapidly, but Goodstein conjectured that every such sequence eventually returns to zero. Kirby and Paris [9] later proved that the conjecture is factually true, but fundamentally unprovable within conventional number theory. The proof of the unprovability within Peano arithmetic relies on mapping each hereditary expression to a strictly decreasing sequence of transfinite ordinal numbers. However, because Peano arithmetic cannot prove the ordering of ordinals up to ϵ_0, the conjecture is formally unprovable within that scope. To break the impasse would require appealing to a more advanced theory of infinite ordinals.

5.3 What Gödel Actually Exposed

Gödel, in his seminal 1931 paper, investigated the correspondence between the natural numbers (NAT) and the address space of all expressions of type [N] $\rightarrow$ B expressible in standard arithmetic. A key assumption in the argumentation is that because all predicates in the expression space are of finite MDL, the expression space is countably finite (like NAT). The predicates can accordingly be listed, *viz.*,

$\lambda n : \phi_1(n)$
$\lambda n : \phi_2(n)$
$\lambda n : \phi_3(n)$
$\ldots$

in which the ϕ_i's are ordinary functions expressible in the vocabulary and grammar of arithmetic.

Suppose that there exists a bijective mapping, α, that maps NAT to the address space (*i.e.*, ϕ indexing), and consider the predicate

$$\lambda n : \mathrm{NOT}(\phi(\alpha(n))(n)) \tag{8}$$

wherein ϕ is now of type [N] $\rightarrow$ ([N] $\rightarrow$ B). Assuming, as Gödel did, that ϕ is a valid object of standard arithmetic, it follows that there exists an integer, n_{G}, such that the assertion predicate in Eq. 8 is at address $a_{\mathrm{G}} \equiv \alpha(n_{\mathrm{G}})$ in the list.

[14] *Peano arithmetic* is the technically precise terminology for what been called standard arithmetic.
[15] The bounding function corresponds to the fast-growing hierarchy (specifically at the level of ϵ_0).

Now consider the assertion

$$\phi(a_{\mathrm{G}})(n_{\mathrm{G}}) \tag{9}$$

Is this statement true or false? Suppose that it is true. Definitional substitution of Eq. 8 for $\phi(a_{\mathrm{G}})$ yields $\mathrm{NOT}(\phi(a_{\mathrm{G}})(n_{\mathrm{G}}))$, which directly contradicts what Eq. 9 ostensibly maintains. The opposite contradiction arises from supposition that the statement in Eq. 9 is false.

Implications of the Paradox

The assertion in Eq. 9 is a genuine paradox. It amounts to an indirect proof, in the sense of the Greek geometers, that the hypothesis of bijective correspondence between NAT and the assertion predicate address space is *false*.

This conclusion is jarring against the backdrop of classical set theory, which was traditionally understood to hold that any two countably infinite sets are equinumerous and thus must be isomorphic to one another. The contradiction, however, demonstrates that there is a real structural difference between (*i*) "generic" countably infinite sets (*e.g.*, NAT), which can be thought of as representing "baseline infinity", and (*ii*) sets of all finite subsets of NAT. The latter is isomophic to the address space.

The paradox exposes a historical oversight in classical set theory, as it had reached fruition in the half-century that preceded him. Most notably, Cantor established a Peano-like succession of cardinally separated infinite *estates*, *viz.*,

$$\aleph_0 \rightarrow \aleph_1 \rightarrow \aleph_2 \rightarrow \ldots \tag{10}$$

The first estate, $\aleph_0$, encompasses all countable infinities, whose structure is defined by the Peano axioms (A1-A3). Jumps between successive estates are defined by power set relations. The set of all subsets, finite and infinite, of one estate defines the structure of the next estate.

Cantor did not define and elaborate any internal structure within the estates, however. The omission led to a prevailing assumption - largely unnoticed and unexamined - that equinumerous sets (*i.e.*, both lying within the same estate) were on an equal footing in all respects and that any isomorphism (bijective mapping) between them was necessarily valid. There was no conception of or perceived need for semantically mandatory type separation, as there is nowadays in all modern computing architectures. A framework for identifying and elaborating internal structural differentiation within the Cantor estates is introduced in the next section.

Type-Theoretic ASLR

The assertion predicate in Eq. 8 is self-vitiating because the very supposition of mapping between address space and object space (NAT) is flawed. The expression space encoding, *as an encoding*, is arbitrary. The address space is a TYPE of its own. Like NAT, it is countably infinite and resides in $\aleph_0$, but its type-theoretic isolation, in formal terms, means that any mapping between the address space and NAT is indeterminate and unspecifiable. Equivalently interpreted, the address space type comes with a built-in address space layout randomization (ASLR) stipulation. A mapping could be scrambled, and it would have no bearing on the content of mathematics proper.

5.4 Gödel Numbering

Gödel devised a numbering system, which amounted to a specific mapping proposal from the address space to NAT. It can be visualized as follows:

- Encode the expression lexically as a Shannon-efficient bit string (*e.g.*, Morse code).
- Treat the encoded bits as prime exponents to obtain a *Gödel number* assigned to the expression. The expression can readily be recovered by prime factorization of the Gödel number and decoding of the resulting bit string.

Morse code had been around for a century when Gödel penned his paper, and he should have used it directly for the encoding without the primes exponentiation layer.

A corollary of the paradox is that Gödel numbering is ultimately irrelevant - not because it is profligately inefficient, but because it is a detailed elaboration of a sacrificial strawman in an indirect proof. That would be true even of a Shannon-optimal encoding.

5.5 The Self-Referential Assertion

Gödel's analysis traversed the address space of expressions of type [N] → B. However, he introduced a peculiar predicate expression, *viz.*,

$$\lambda n : \text{INCONCLUSIVE}(n) \tag{11}$$

which returns TRUE if a static generalist IE returns an inconclusive result for the assertion at address n, or FALSE if the IE returns QED.

The expression in Eq. 11 is at some address, a'_{G} (different from that in Eq. 9). What happens when we submit to the IE the assertion INCONCLUSIVE(a'_{G})? That assertion is self-referential and states that it cannot be proved. If the IE returns QED, the assertion is falsified - an untenable contradiction. It follows that the IE must return an inconclusive result, validating the assertion. We thus have a truthful statement that the IE fails to mark QED.

By this form of argumentation, Gödel claimed definitive proof of G1, as a general sweeping claim about meta-mathematics. Should we buy it?

5.6 Gödel's Fallacy

The vocabulary and grammar of formal theories of mathematics proper - TYPE and CONST lists - is blandly simple and built from the ground up. Which raises a basic question: How can a dryly technical formal theory, with strictly limited vocabulary and well-disciplined production rules, end up talking about itself? How can the functor name INCONCLUSIVE possibly make its way into a formal theory of mathematics proper, as a CONST list entry?

The notion that INCONCLUSIVE can be considered a valid functor of mathematics proper conceivably makes sense only under the supposition that the lineup of provable assertions or the loop invariant structures underlying them forms an elegant mathematical pattern in its own right, like a tiling tessalation or the Mandelbrot set. That probably was close to Hilbert's aesthetic conception of what proof engine operation would resemble.

Eq. 11 signifies a dramatic escalation from Eq. 8. Both rest on the same address space mapping fallacy, but Eq. 11 boldly posits that INCONCLUSIVE has a valid place in mathematics proper. If it did, it would require a constructive definition, which would reveal its footing down to the bedrock of primitives in the TYPE and CONST lists. That, however, is conceivably possible only if proof of C or NOT(C) can be traced entirely within the inferential scope of the theory at hand. But it has become evident that that is generally not the case, as most conjectures rely on support extrinsic to the theories in which they originate.

From a 21st-century perspective, Gödel's argumentation was historically the world's first remote code execution exploit. It was conceptually analogous to a cyber hack that exploited the lack of separation between DATA and CODE address spaces. The reference to INCONCLUSIVE in Eq. 11, in effect, enables malware code to be injected at address a'_{G}, as in a SQL injection. The implanted code drives the IE into a self-vitiating state that validates the absurdist statement "I am not provable!".

5.7 Exploit Takes Root

Gödel was of no malicious intent, but his argumentation took root and reverberated sociologically like an intellectual malware implant. In the first place, how could a dense academic paper by a little-known author in Vienna writing in an obscure field have become an instant hit so quickly in the world of 1931? That normally would have been exceedingly improbable, except for the happenstance that Gödel's argumentation made its debut at the epicenter of the mathematical foundations community, at the Königsberg Epistemology Conference in September 1930. Gödel voiced the gist of his idea[16] to von Neumann, who instantly recognized and concurred with it. Gödel had not actually penned the paper at that time, but the intuitive instincts of Gödel, von Neumann, and Bernays strongly aligned and readily saw that it was the death knell for Hilbert's grand optimism.

There was also an asymmetry of expectations. Hilbert, as the stodgy conservative defender of the traditional establishment, had to be right every time, whereas Gödel, in the role of bomb-thrower or hacker, had to be right only once.

The normal channels of tedious withering peer review were circumvented, and G1, taken at face value, became the accepted wisdom and dominant consensus almost overnight. Why? In large part, because nobody seriously

[16] Most importantly, that the function form in Eq. 8 is reminiscent of the Cantor diagonal.

disbelieved the substantive gist of what G1 conveyed. The mismatch between expressiveness and proof difficulty was *felt* by mathematicians.

Reaction to Gödel's work engendered a pervasive climate of intellectual defeatism and gloom. It became a powerful cultural narrative - indeed, almost a theological tenet - that Gödel had shown that human reasoning itself is fundamentally limited and meek. It is noteworthy that Gentzen's constructive analysis, which showed that arithmetic with addition and multiplication is consistent, did little to reverse the pessimism that had taken over by the mid-1930's. By that time, the academic and philosophical dispensation that had sprung from Gödel's paper had hardened. Incompleteness had become a dramatic unifying mythos, which would become widely disseminated and popularized. The methodical inductive analysis that Gentzen presented was less cultually provocative than a grand declaration of tragedy and human ineptitude.

6 Structural Grade Separations within Infinity

Gödel's exploit ultimately demonstrates how reasoning systems can be crashed, when the infinite estates are unfurnished with internal structure. This section proposes a framework that models the estates as finite-dimensional discrete Cartesian grids, points in which represent *structural grades* at which reasoning systems can perform incremental steps. $\aleph_0$ contains an internal 1-dimensional lattice of grid points, $\aleph_1$ contains a two-dimensional lattice, $\aleph_2$ a three-dimensional lattice, and so forth.

6.1 Intra-Estate Grids

Inference engines rely on dense grids, movement along which enables them to progress in incremental "wormholing" fashion. Grid navigation consists of alternating sequences of "horizontal" and "vertical" steps, like a staircase. Horizontal movement between grid points can be thought of as *inductive treading* - essentially tasks of proving that properties that hold for n also hold for $S(n)$. Vertical movement within estates can be thought of as *inductive graduation*. The navigation process generalizes Hilbert's finitistic program, which remains apt in concept, because reasoning steps necessarily work with finite amounts of information that do not, in isolation, see the big picture.

6.2 Internal Structure of $\aleph_0$

Consider a generic type, T, defined solely by the Peano axioms. It represents the baseline of countable infinity, denoted as $\aleph_0[0]$. The baseline is the appropriate setting for reasoning about countably infinite types whose internal structure (if any) is irrelevant to the argumentation.

The next level of reasoning is a half-step up and involves reasoning about finite subset of T, whose size is specifically limited (capped). Reasoning about capped finite subsets amounts to an increase in complexity that scales polynomially. The next half-step entails inductive closure over *all* finite sets, with the cap restrictions removed. The reasoning process then "graduates" to the next grid point, denoted as $\aleph_0[1]$.

Continuing in this fashion enables the inference engine to traverse the one-dimensional lattice within $\aleph_0$. The full sequence of steps is as follows. Note that type T is renamed to $\mathrm{T}_0[0]$.

- $\aleph_0[0]$: Baseline infinity.
  - $\aleph_0[0].P$ signifies finite P-ary sets of objects of type $\mathrm{T}_0[0]$.
- $\aleph_0[1]$: Inductive closure of $\aleph_0[0].P$.
  - Type $\mathrm{T}_0[1]$ signifies arbitrary finite sets of objects of type $\mathrm{T}_0[0]$, uncapped.
  - $\aleph_0[1].P$ signifies finite P-ary sets of objects of type $\mathrm{T}_0[1]$.
- $\aleph_0[2]$: Inductive closure of $\aleph_0[1].P$.
  - Type $\mathrm{T}_0[2]$ signifies arbitrary finite sets of objects of type $\mathrm{T}_0[1]$, uncapped.
  - $\aleph_0[2].P$ signifies finite P-ary sets of objects of type $\mathrm{T}_0[2]$.
- $\cdots$

6.3 Beyond $\aleph_0$

We next navigation to higher estates. This involves alternating steps of power set jumps between estates and navigation within the higher estates.

Power Set Jump from $\aleph_n$ to $\aleph_{n+1}$

Transitions from $\aleph_n$ to $\aleph_{n+1}$ take place in parallel, on a discrete Cartesian grid of dimension $n+1$:

- $\aleph_{n+1}[i_0]\cdots[i_n][0]$ signifies arbitrary sets, finite or infinite, of objects of type $\aleph_n[i_0]\cdots[i_n]$.

Internal Structure of $\aleph_n$

$\aleph_n$ consists of a discrete n-dimensional Cartesian grid of basepoints. Each basepoint is elaborated horizontally, adding a grid dimension internal to $\aleph_n$. The horizontal elaboration consists of two components:

- $\aleph_n[i_0]\cdots[i_n].P$ signifies P-ary finite sets of objects of type $\aleph_n[i_0]\cdots[i_n]$, capped.
- $\aleph_n[i_0]\cdots[i_n+1]$ signifies arbitrary finite sets of objects of type $\aleph_n[i_0]\cdots[i_n]$, uncapped. It is the inductive closure of $\aleph_n[i_0]\cdots[i_n]$.

The first component entails analysis of finite subsets, capped at specified size P. The second component is the inductive closure of the analysis, resulting in promotion to the next level of reasoning at $\aleph_n[i_0]\cdots[i_n+1]$.

6.4 SGS Interpretation of the Meta-Mathematical Challenges

In the SGS framework, the meta-mathematical challenges introduced earlier can be formalized as in terms of $\aleph_0[k]$, which represents the lesser theory under study, and $\aleph_0[k+1]$, which represents the greater theory that provides the necessary inferencing strength.

Addition-Only Arithmetic

For additional-only arithmetic, $k=0$ identifies the adjacent loci. The lesser theory engenders expressions consisting of finitely many addition ($+$) and succession (S) operations and is thus couched in $\aleph_0[0]$. Presburger showed that: (i) complete decidability can be established within the scope of the lesser theory, using quantifier elimination (exhaustive sequent grinding), and (ii) inductive graduation to $\aleph_0[1]$ reaches the analytic arena in which the consistency of the lesser theory can be proved. Expressions in addition-only arithmetic, like combinational logic, are sufficiently tame that methodical sequent grinding can mark any assertion or its negation QED.

It is noted that addition-only arithmetic is decidable within the lesser theory ($\aleph_0[0]$), whereas it takes the greater theory ($\aleph_0[1]$) to adjudicate consistency. This is atypical, as decidability is generally the more difficult question. That difficult problems (*e.g.*, Goldbach, Riemann Hypothesis) at the frontier remain unsolved pertains to decidability (G1), whereas their Platonic certification pertains to consistency (G2) and sufficiency (G3).

Standard Arithmetic

The introduction of multiplication complicates arithmetic immensely, beyond a critical threshold such that the meta-mathematical challenge is raised to $k=1$. Standard arithmetic achieves inductive closure over the finite integers[17], and so is couched in $\aleph_0[1]$. However, consistency cannot be established within that scope, providing a concrete example of G2.

Gentzen [2] proved the consistency of standard arithmetic through inductive treading through the transfinite ordinals[18] ($\aleph_0[1].P$), followed by inductive graduation to $\aleph_0[2]$. The inductive treading in Gentzen's analysis involved well-ordered trees capped at a definite ordinal level, which he denoted as a succession of exponential towers, *viz.*,

$$\epsilon_0 = \omega^{\omega^{\omega^{\cdots}}} \tag{12}$$

[17] The meta-mathematics literature (*e.g.*, in Gentzen's notation) customarily denotes the finite integers as ω.

[18] In the SGS framework, transfinite ordinals are finite sets of arbitrary (uncapped) finite sets of integers.

Consistency is proved once the $\aleph_0[2]$ arena is reached. Complete decidability, however, is too difficult and remains an open question.

Paris-Harrington-Ramsey Combinatorics

The combinatorial assertion that Paris and Harrington investigated is expressible within the scope of $\aleph_0[1]$. However, the extension of the Ramsey baseline with the size rule cannot be proved within that scope, providing a concrete example of G1.

The bounding function for the Paris-Harrington theorem grows at a rate indexed by F_{ϵ_0} in the fast growing hierarchy of the $\aleph_0[1].P$ inductive tread. Because standard arithmetic can only prove the totality of functions up to levels F_α for $\alpha < \epsilon_0$, to establish the general existence of bounds, uncapped, requires inductive graduation to $\aleph_0[2]$.

Goodstein-Kirby-Paris Number Theory

The Goodstein sequences, defined in terms of hereditary base incrementation and subtraction, is expressible within the scope of $\aleph_0[1]$. To show that the sequences terminate (at zero) requires mapping the hereditary representations to strictly decreasing sequences of transfinite ordinals below ϵ_0. This too entails inductive treading through $\aleph_0[1].P$ and graduation to $\aleph_0[2]$.

6.5 Gödel's Exploit as an Indirect Proof of the Existence of Internal $\aleph_0$ Structure

Gödel's seminal analysis is well-known, in profession circles and popularizations (*e.g.*, [11]) alike, for its paradoxical form of argumentation. It gives the impression - and is conventionally taken to imply, quite literally - that any serious mathematical theory, despite its ostensible limitation to bland technical parlance, becomes self-referential. Which raises a basic question: Was Gödel's work itself serious meta-mathematics, or was it just a facetious parlor trick? The self-referential assertion is not grounded obviously in a concrete mathematical problem domain, as are Ramsey combinatorics and the Goodstein sequences. So what, then, does Gödel's argumentation have to do with the real world of mathematics? Is it just an amusing curio, or does it have important practical bearing on mathematics proper?

In truth, the middle decades of the 20th century had no good answers to these questions. Common refrain held that Gödel's incompleteness theorems mattered "in principle", but there was never any clarification or elaboration of said principle. It was not until a half-century later that the work of Paris, Harrington, and Kirby demonstrated undecidability within $\aleph_0[1]$. That is, tangible concrete examples of how inferencing efficacy is fundamentally unable to keep up with expressiveness. Their work, however, demonstrated that the difficulty gets "kicked upstairs" to $\aleph_0[2]$. Nobody has yet found any concrete examples of true floating islands, *i.e.*, expressible assertions not possible in $\aleph_0[2]$ or in any higher grid locus. Nor has any established mathematical theory been definitively shown *not* to be Platonically certifiable. In other words, absolute proof of G1-G3 remains elusive.

Substantive credit and lasting value can be ascribed to Gödel's argumentation as follows: The exploit served as an indirect proof of structural grade separation within $\aleph_0$. The argumentation is essentially as follows: Assume the contrary hypothesis[19] that the $\aleph_0$ estate is devoid of any internal differentiation. It then follows that any two countably infinite sets, such as NAT and the expression address space, are isomorphic. A non-descript isomorphism can be pulled out of thin air, given a name, and treated as a valid mapping of full stature, without sowing any seeds of semantic turmoil. The assumed validity of the isomorphism has two insidious implications in Gödel's argumentation: (*i*) It makes the isomorphism itself (*i.e.*, the particularism of the expression encoding) and the efficacy of the inference engine valid subjects of mathematics proper. (*ii*) Self-reference comes about not because of paradoxical seeds lurking in the semantics of the formal language itself, but as a side effect of coincidence stemming from NAT and expression encodings cohabiting the same address space.

The contrary hypothesis leads to the formulation of an absurd function. The hypothesis is thus shown to be false. Grade differentiation within the $\aleph_0$ estate does exist; QED.

[19] This "contrary" hypothesis is what the orthodoxy of Gödel's time seriously believed.

6.6 Set Cardinality

The structural differentiation internal to the estates overturns the traditional set-theoretic conception of cardinality. Cantor had identified only coarse cardinality differences across power set jumps. He left the estates internally undifferentiated.

Historically, Cantor used the diagonal argument in two distinct contexts: (*i*) intra-estate uncountability of real numbers (*i.e.*, showing that any attempted enumeration of $\mathbb{R} \in (0, 1)$ yields an unlisted real number), and (*ii*) inter-estate power set jumps. Cantor's Theorem proved that $|A| < |\mathcal{P}(A)|$ for *any* set A, by showing no surjection exists from A to $\mathcal{P}(A)$. Because Cantor used diagonalization to establish the power set jump ($\aleph_0 \to \aleph_1$), classical literature has frequently confused the mechanism of diagonalization with inter-estate cardinal expansion.

The analysis in Sec. 5.3, however, demonstrates that cardinal inequality applies to intra-estate inductive graduations, the first of which is $\aleph_0[0] \to \aleph_0[1]$. This implies that Cantor diagonal arguments, such as in Eq. 8, are necessary, but not sufficient, to prove cardinal inequality. Under the SGS framework, diagonalization is fundamentally an indicator of structural grade separation between neighboring grid points, *i.e.*, $\aleph_n[i_0]\cdots[i_n]$ and $\aleph_n[i_0]\cdots[i_n + 1]$.

The SGS framework makes absolute distinction between mild (intra-estate) and steep (inter-estate) grade separations. For mild separations, the diagonal argumentation involves indexed sequences of inductively bounded (capped) finite sets, Turing machine state transitions, or total functions in the fast-growing hierarchy below ϵ_0. It exposes the inability of a formal language couched at the lower grid point ($\aleph_0[k]$) to represent its own meta-level evaluation predicate (operative at $\aleph_0[k + 1]$) using only its native object-level bounded terms. The diagonal argumentation constructs an element that escapes the provability closure of $\aleph_0[k]$. Steep grade separation across power set jumps is fundamentally different in that it encompasses infinite subsets, which cannote be systematically enumerated by any discrete inductive process.

Classical set theory assumed that any two sets in the same Cantor estate are equinumerous and could therefore be freely interconverted. It was precisely that naive conception of equinumerosity that enabled Gödel's exploit. The SGS framework replaces the traditional conception of cardinality with equivalence relations between pairs of grid points.

The strong type-theoretic constructive definitions ensure that any grid point inherently "knows" its own indexing, *viz.*, $\aleph_n[i_0]\cdots[i_n]$. Given any two grid loci, the nearest common ancestor can thus be readily ascertained. Intra- and inter-estate distances can then be defined separately. Traditional cardinality is reinterpreted as equivalence classes based on zero inter-estate distance.

References

[1] G. Cantor. *Über eine elementare Frage der Mannigfaltigkeitslehre*. Vol. 1. 1891, pp. 75–78.

[2] G. Gentzen. "Die Widerspruchsfreiheit der reinen Zahlentheorie". In: *Mathematische Annalen* 112.1 (1936), pp. 493–565.

[3] K. Gödel. "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I". In: *Monatshefte für Mathematik und Physik* 38.1 (1931), pp. 173–198.

[4] C. Goldbach. "Brief (Nr. XII) an Leonhard Euler vom 7. Juni 1742". In: *Korrespondenz mit L. Euler*. Ed. by P. H. Fuss. Vol. 1. Correspondance mathématique et physique de quelques célèbres géomètres du XVIIIème siècle. St. Petersburg: Imperial Academy of Sciences, 1742, pp. 125–129.

[5] R. L. Goodstein. "On the restricted ordinal theorem". In: *The Journal of Symbolic Logic* 9.2 (1944), pp. 33–41.

[6] D. Hilbert. "Mathematische Probleme". In: *Nachrichten von der Königlichen Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*. Göttingen: Vandenhoeck & Ruprecht, 1900, pp. 253–297.

[7] D. Hilbert. "Über das Unendliche". In: *Mathematische Annalen* 95.1 (1926), pp. 161–190.

[8] D. Hilbert and P. Bernays. *Grundlagen der Mathematik*. Vol. 1. Berlin: Springer, 1934.

[9] L. Kirby and J. Paris. "Accessible independence results for Peano arithmetic". In: *Bulletin of the London Mathematical Society*. Vol. 14. 4. Wiley Online Library, 1982, pp. 285–293.

[10] J. E. Littlewood. "Sur la distribution des nombres premiers". In: *Comptes Rendus Hebdomadaires des Séances de l'Académie des Sciences* 158 (1914), pp. 1869–1872.

[11] E. Nagel and J. R. Newman. *Gödel's Proof*. New York: New York University Press, 1958.

[12] J. Paris and L. Harrington. "A Mathematical Incompleteness in Peano Arithmetic". In: *Handbook of Mathematical Logic*. Ed. by Jon Barwise. Amsterdam: North-Holland, 1977, pp. 1133–1142.

[13] M. Presburger. "Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchem die Addition als einzige Operation hervortritt". In: *Sprawozdanie z I Kongresu Matematyków Krajów Słowiańskich [Comptes Rendus du I Congrès des Mathématiciens des Pays Slaves]*. Warsaw, 1930, pp. 92–101.

[14] B. Riemann. "Über die Anzahl der Primzahlen unter einer gegebenen Grösse". In: *Monatsberichte der Königlichen Preußischen Akademie der Wissenschaften zu Berlin* (1859), pp. 671–680.

[15] A. Tarski. *Der Wahrheitsbegriff in den formalisierten Sprachen*. Vol. 1. 1936, pp. 261–405.

[16] A. N. Whitehead and B. Russell. *Principia Mathematica*. Vol. 1. Cambridge: Cambridge University Press, 1910.

[17] A. Wiles. "Modular elliptic curves and Fermat's Last Theorem". In: *Annals of Mathematics* 141.3 (1995), pp. 443–551.