

On the Broader Epistemological Significance of Self-Justifying Axiom Systems

Dan E.Willard*

State University of New York at Albany

Abstract

This article will be a continuation of our research into self-justifying systems. It will introduce several new theorems (one of which will transform our previous infinite-sized self-verifying logics into formalisms of purely finite size). It will explain how self-justification is useful, even when the Incompleteness Theorem clearly does sharply limit its scope.

Keywords and Phrases: Gödel's Second Incompleteness Theorem, Hilbert's Second Open Question, Semantic Tableaux Deduction, Consistency.

*This research was partially supported by the NSF Grant CCR 0956495.
Email = dan.willard.albany@gmail.com

1 Introduction

Gödel's Incompleteness Theorem has two parts. Its first half indicates no decision procedure can identify all of arithmetic's true statements. Its "Second Incompleteness" result specifies sufficiently strong logics *cannot* verify their own consistency. Gödel was careful to insert a caveat into his historic paper [11], indicating a *diluted* form of Hilbert's Consistency Program might reach some levels of partial success:

* *"It must be expressly noted Proposition XI (e.g. Gödel's "Second" Incompleteness Result) represents no contradiction of the formalistic standpoint of Hilbert. For this standpoint presupposes only the existence of a consistency proof by finite means, and there might conceivably be finite proofs which cannot be stated in P or in ..."*

Some scholars have interpreted * as, possibly, anticipating attempts to confirm Peano Arithmetic's consistency, via either Gentzen's formalism or Gödel's Dialetica interpretation. On the other hand, the Stanford's Encyclopedia's entry about Gödel quotes him, in its Section 2.2.4, stating he was hesitant to view the Second Incompleteness Theorem as fully ubiquitous, until learning of Turing's work. Moreover, Yourgrau [51] states von Neumann "*argued against Gödel himself*" in the early 1930's, about the definitive termination of Hilbert's consistency program, which "*for several years*" after [11]'s publication, Gödel "*was cautious not to prejudge*". Also, it is known [6, 13, 51] that Gödel initially presumed the second theorem was false, before proving his stunning result.

In any case several year after he wrote *'s initial statement, Gödel gave a 1933 lecture [12], where he told his audience that Hilbert's initial 1926 objectives, summarized formally by ** below, had "*unfortunately*" no "*hope of succeeding along*" its originally intended plans.

** (Hilbert [17] 1926): "*Where else would reliability and truth be found if even mathematical thinking fails? The definitive nature of the infinite has become necessary, not merely for the special interests of individual sciences, but rather for the honor of human understanding itself.*"

Our research, in both the current article and the prior papers [38]-[49] was stimulated by the prospect that we find ** enticing, even though the Second Incompleteness Theorem *unequivocally* demonstrates that logics *cannot* recognize their own consistency *in a robust sense*. Accordingly, we have studied *both* generalizations and boundary-case exceptions for the Second Incompleteness Theorem in [38]-[49]. The current article will seek to *both* strengthen these prior results, in the context of axiom systems with *strictly finite cardinalities*, and to also provide a more intuitive explanation of the meaning behind [38]-[49]'s results.

The thesis of this article will be delicate because there can be no doubt that the Second Incompleteness Theorem is sharply robust, when viewed from a conventional purist mathematical

perspective. On the other hand, we will argue that there are certain facets of a “Self-Justifying Logics”, that are tempting under a hard-nosed engineering perspective, contemplating sharply *curtailed forms* of Hilbert’s goals. These results will be fragile *but not fully immaterial*.

In other words, this article will offer a somewhat complicated 2-part interpretation of the Second Incompleteness Theorem where:

1. The Second Incompleteness Theorem is seen as being 100 % robust from a mathematical perspective because of the widely encompassing nature of the 1939 Hilbert-Bernays analysis [18] (centering around their three well-known “Derivability Conditions” [25]).
2. On the other hand, our discourse will partially appreciate Hilbert’s reluctance to fully embrace the Second Incompleteness Theorem, despite his joint work with Bernays [18] generalizing the Second Incompleteness Effect. (This is because it is awkward to explain how human beings can acquire the mental energy for motivating themselves to cogitate, without possessing some type of instinctive faith in their own self-consistency.)

Thus, the current article will seek to separate a “*mathematical*” from what perhaps should be “*engineering-style*” appreciation of one’s internal consistency. We will seek to define and explore the latter (with the hope that it will help formalize how future 21st century computers can benefit from its engineering-style perspective, while still respecting the strict prohibitions formalized by Gödel’s millennial result.)

As the reader examines this paper, it should be kept in mind that it does focus on semantic tableaux deduction (similar to the earlier discussion that had appeared in [49]’s more abbreviated conference-style summary of our results). A second paper, currently under preparation, will examine Hilbert-style deductive systems (whose self-justification properties are partially analogous and partly quite different from tableaux-style systems). The combination of these two results will formally define both the potential of self-justifying logics and the limitations which the Second Incompleteness Theorem imposes upon them.

2 Background Setting

Let (α, d) denote any axiom system and deduction method satisfying the simple “**Split Rule**” below¹. This pair will be called “**Self Justifying**” when:

- i one of α ’s theorems will state that the deduction method d , applied to the system α , will produce a consistent set of theorems, and

¹Our “Split Rule” is the trivial requirement that all the axiom sentences in α are technically *proper axioms*, and that deduction method d is required to include **BOTH** a finite number of rules of inference and whatever “logical axioms” are needed (*if any ?*) by d ’s methodology. (This trivial Split-Rule notation convention will help us to provide a precisely formalized statement of our results. .)

- ii** the axiom system α is in fact consistent.

For any (α, d) , it is easy to construct a second $\alpha^d \supseteq \alpha$ that satisfies the Part-i requirement. For instance, α^d could consist of all of α 's axioms plus an added “**SelfRef**(α, d)” sentence, defined as stating:

- There is no proof (using d 's deduction method) of $0 = 1$ from the *union* of the system α with *this* sentence “**SelfRef**(α, d)” (looking at itself).

Kleene [20] noted how to encode rough analogs of “**SelfRef**(α, d)”. Each of Kleene, Rogers and Jeroslow [20, 31, 19] noted α^d may, however, be inconsistent (despite **SelfRef**(α, d)’s assertion), thus causing it to violate Part-ii’s requirement.

This problem arises in many contexts besides Gödel’s paradigm, where α was an extension of Peano Arithmetic (see [1, 2, 3, 4, 5, 7, 9, 11, 14, 15, 16, 18, 21, 23, 24, 27, 28, 29, 30, 31, 32, 34, 35, 36, 37, 41, 43, 47]). Such results formalize paradigms where self-justification is infeasible, due to diagonalization issues. (It should, perhaps, be added that among this lengthy list of articles, it was especially [1, 4, 11, 24, 29, 34, 37]’s incompleteness results that influenced our work in [38]-[49].) In any case, the main point is that most logicians have hesitated to employ an analog of a **SelfRef**(α, d) axiom because $\alpha^d = \alpha + \text{SelfRef}(\alpha, d)$ is typically inconsistent.

Our research in [38, 40, 44, 46, 45] focused on paradigms where self-justification is feasible. It involved weakening the properties a logic can prove about addition and/or multiplication (to avoid potential difficulties). To be more precise, let $Add(x, y, z)$ and $Mult(x, y, z)$ denote 3-way predicates specifying $x + y = z$ and $x * y = z$. Then a logic will be said to **recognize** successor, addition and multiplication as **Total Functions** iff it includes sentences 1-3 as axioms.

$$\forall x \exists z \quad Add(x, 1, z) \tag{1}$$

$$\forall x \forall y \exists z \quad Add(x, y, z) \tag{2}$$

$$\forall x \forall y \exists z \quad Mult(x, y, z) \tag{3}$$

A logic α will be called **Type-M** iff it contains 1-3 as axioms, **Type-A** iff it contains only (1) and (2) as axioms, **Type-S** iff it contains only (1) as an axiom, and **Type-NS** iff it contains none of these axioms. The relationship of these constructs to self-justification is explained by items (a) and (b):

- a. The existence of Type-A systems that can recognize their own consistency under semantic tableaux deduction, while proving analogs of all Peano Arithmetic’s Π_1 theorems (in a slightly different language), were demonstrated in [42, 44]. Also, [40, 45] noted that some specialized forms of Type-NS systems can likewise recognize their own Hilbert consistency.
- b. The above evasions of the Second Incompleteness Theorem are known to be near-maximal in a mathematical sense. This is because the combined work of Pudlák, Solovay, Nelson

and Wilkie-Paris [26, 29, 34, 37] implied no natural Type-S system can recognize its Hilbert consistency, and Willard subsequently [41, 47, 48] hybridized their formalisms with some techniques of Adamowicz-Zbierski [1, 2] to establish that most Type-M systems cannot recognize their own semantic tableaux consistency.

Other fascinating efforts to evade the Second Incompleteness Theorem have used the Kreisel-Takeuti “CFA” system [23] or the the *interpretational framework* of Friedman, Nelson, Pudlák and Visser [10, 26, 29, 36]. These systems are unrelated to our approach because they do not use Kleene-like “*I am consistent*” axiom-sentences. Instead, CFA uses the special properties of “second order” generalizations of Gentzen’s *cut-free* Sequent Calculus, and the interpretational approach formalizes how some systems recognize their Herbrand consistency on localized sets of integers, which unbeknownst to themselves, includes all integers. (These alternate results are interesting but unrelated to our approach.)

3 Defining Notation and Earlier Results

A function F will be called **Non-Growth** iff $F(a_1 \dots a_j) \leq \text{Maximum}(a_1 \dots a_j)$ holds. Six examples of non-growth functions are

1. *Integer Subtraction* (where $x - y$ is defined to equal zero when $x \leq y$),
2. *Integer Division* (where $x \div y$ equals x when $y = 0$, and it equals $\lfloor x/y \rfloor$ otherwise),
3. $\text{Maximum}(x, y)$,
4. $\text{Logarithm}(x) = \lfloor \text{Log}_2(x) \rfloor$
5. $\text{Root}(x, y) = \lfloor x^{1/y} \rfloor$. and
6. $\text{Count}(x, j)$ designating the number of “1” bits among x ’s rightmost j bits.

The term **U-Grounding Function** referred in [44] to a set of primitives, which included the preceding functions plus the *growth operations* of addition and $\text{Double}(x) = x + x$. Our language L^* was built out of these symbols, plus the primitives of “0”, “1”, “=” and “ \leq ”.

In a context where t is any term in [44]’s language L^* , the quantifiers in $\forall v \leq t \Psi(v)$ and $\exists v \leq t \Psi(v)$ were called *bounded quantifiers*. Any formula in L^* , all of whose quantifiers are bounded, was called a Δ_0^* formula. The Π_n^* and Σ_n^* formulae were then defined by the usual rules that:

1. Every Δ_0^* formula is considered to be “ Π_0^* ” and also “ Σ_0^* ”.
2. A wff is called Π_n^* when it is encoded as $\forall v_1 \dots \forall v_k \Phi$ with Φ being Σ_{n-1}^*
3. Also, a wff is called Σ_n^* when it is encoded as $\exists v_1 \dots \exists v_k \Phi$, where Φ is Π_{n-1}^* .

Our articles [38, 42, 44] used the symbol D to denote a deduction method. They focused mostly around the semantic tableaux deductive methodology, whose formal definition can be found in the textbooks by Fitting and Smullyan [8, 33] and whose definition is also reviewed by Appendix A of the current article.

Our articles [43, 44] also considered an improved faster deductive technology, called **Tab-k deduction**, that consists of a speeded-up version of a tableaux, which permits a *limited analog* of Gentzen-style deductive cuts for Π_k^* and Σ_k^* formulae. Thus, if H denotes a sequence of ordered pairs $(t_1, p_1), (t_2, p_2), \dots (t_n, p_n)$, where p_i is a Semantic Tableaux proof of the theorem t_i , then H has been called a “**Tab-k Proof**” of a theorem T from α ’s axioms iff $T = t_n$ and also:

1. Each of the “intermediately derived theorems” t_1, t_2, \dots, t_{n-1} have a complexity no greater than that of either a Π_k^* or Σ_k^* sentence.
2. Each proper axiom in p_i ’s proof comes either from α or is one of t_1, t_2, \dots, t_{i-1} .

Thus, a Tab-k proof is essentially a generalization of a classic semantic tableaux proof that essentially owns the equivalent of an extra specialized modus ponens rule for Π_k^* and Σ_k^* sentences.

Let us say an axiom system α has a **Level-J Understanding** of its own consistency under a deduction method D iff α can prove that there exists no proofs using its axioms and D ’s deduction of both a Π_j^* theorem and its negation. In this notation, items A and B summarize [39, 41, 43, 44, 47]’s main results:

- A. For any axiom system A using L^* ’s U-Grounding language, [44] showed its $IS_D(A)$ formalism could prove all A ’s Π_1^* theorems and simultaneously verify its Level-1 consistency under Tab-1 deduction.
- B. Two negative results, tightly complementing item A’s positive result, were exhibited in [39, 41, 43, 47]. The first was that [39, 41, 47] showed most systems are unable to verify their Level-0 consistency under semantic tableaux deduction, when they included statement (3)’s “Type-M” axiom that multiplication is a total function. Moreover, [43] offered an alternate form of this incompleteness result, showing statement (2)’s *far weaker* Type-A systems cannot verify their Level-0 consistency under Tab-2 deduction.

The contrast between these positive and negative results has led to our conjecture that automated theorem provers are likely to eventually achieve a fragmentary part of the ambitions that were suggested by Hilbert in **. This is because the question of whether a formalism can support an *idealized Utopian* conception of its own consistency is *different* from exploring the degrees to which theorem-provers can possess a *fragmentary knowledge* of their own consistency. The Incompleteness Theorem has demonstrated an Utopian idealized form of self-justification is

unobtainable, but our research has found some diluted cousins of this construct are feasible and warrant examination.

In summary, it should be kept in mind, during the remainder of this article, that the Hilbert-Bernays Derivability Conditions [16, 18, 25] impose severe limits upon any evasion of the Second Incompleteness Theorem. On the other hand, it appears that a human's faith in his own consistency is an essential prerequisite to gain the needed psychological motivation for stimulating cogitation? (This is why we suspect Hilbert was never willing to concede that all facets of his consistency program were hopeless.) A broad theme of this paper will, thus, be that it is helpful to distinguish between the goals of a theoretical-oriented study of arithmetic from that of a more engineering-styled approach, since the Second Incompleteness Theorem is a perfect result from the first perspective while it permits for well-defined limited-scale part-way exceptions from the second vantage point.

4 The $\text{IS}_D(A)$ Axiom System

In a context where A denotes any axiom system using L^* 's U-Grounding language, $\text{IS}_D(A)$ was defined in [44] to be an axiomatic formalism capable of recognizing all of A 's Π_1^* theorems and corroborating its own Level-1 consistency under D 's deductive method. It consisted of the following four groups of axioms:

Group-Zero: Two of the Group-zero axioms did define the constant-symbols, \bar{c}_0 and \bar{c}_1 , designating the integers of 0 and 1. The Group-zero axioms will also define the growth functions of addition and $\text{Double}(x) = x + x$. The net effect of these axioms will be to set up a machinery to define any integer $n \geq 2$ using fewer than $3 \cdot \lceil \log n \rceil$ logic symbols.

Group-1: This axiom group did consist of a finite set of Π_1^* sentences, denoted as F , which can prove any Δ_0^* sentence that holds true under the standard model of the natural numbers. (Any finite set of Π_1^* sentences F with this property may be used to define Group-1, as [44] noted.)

Group-2: Let $\Gamma\Phi\Box$ denote Φ 's Gödel Number, and $\text{HilbPrf}_A(\Gamma\Phi\Box, p)$ denote a Δ_0^* formula indicating p is a Hilbert-styled proof of theorem Φ from axiom system A . For each Π_1^* sentence Φ , the Group-2 schema of [44] did contain an axiom of form (4). (Thus $\text{IS}_D(A)$ can trivially prove all A 's Π_1^* theorems.)

$$\forall p \quad \{ \text{HilbPrf}_A(\Gamma\Phi\Box, p) \Rightarrow \Phi \} \quad (4)$$

Group-3: This final part of the $\text{IS}_D(A)$ essentially represented a self-referencing Π_1^* axiom, indicating $\text{IS}_D(A)$ meets §3's criteria of being "Level-1 consistent" under deductive method D . It amounts, thus, to the following declaration:

No two proofs exist for a Π_1^* sentence and its negation, when D 's deductive method is applied to an axiom system, consisting of the union of Groups 0, 1 and 2 with **this sentence** (looking at itself).

One encoding of #, as a self-referencing Π_1^* axiom, appears in [44]. Thus, the below sentence (5) represents [44]'s Π_1^* styled encoding for # in a context where:

- i. $\text{Prf}_{\text{IS}_D(A)}(a, b)$ is a Δ_0^* formula indicating that b is a proof of a theorem a under $\text{IS}_D(A)$'s axiom system and D 's deduction method, and
- ii. $\text{Pair}(x, y)$ is a Δ_0^* formula indicating that x is a Π_1^* sentence and y represents x 's negation.

$$\forall x \forall y \forall p \forall q \quad \neg [\text{Pair}(x, y) \wedge \text{Prf}_{\text{IS}_D(A)}(x, p) \wedge \text{Prf}_{\text{IS}_D(A)}(y, q)] \quad (5)$$

Remark 4.1 A fully formal summary of the techniques that [44] used to encode sentence (5) is provided by the combination of Appendices B and C. The former appendix summarizes our methods for generating the Gödel numbers of semantic tableaux and Tab- k proofs in an optimally compressed manner. The latter appendix explores how sentence (5)'s self-referencing statement is precisely encoded.

Notation. An operation $I(\bullet)$ that maps an initial axiom system A onto an alternate system $I(A)$ will be called **Consistency Preserving** iff $I(A)$ is consistent whenever all of A 's axioms hold true under the standard model of the natural numbers. In this context, [44] demonstrated:

Theorem 4.2 Suppose the symbol D denotes either semantic tableaux deduction or its Tab-1 generalization. Then the $\text{IS}_D(\bullet)$ mapping operation is consistency preserving (e.g. $\text{IS}_D(A)$ will be consistent whenever all of A 's axioms hold true under the standard model of the natural numbers).

We emphasize the most difficult part of [44]'s result was neither the definition of its $\text{IS}_D(A)$'s axiom system nor the Π_1^* fixed-point encoding of (5)'s Group-3 axiom. Instead, the key challenge was the confirming of Theorem 4.2's "Consistency Preservation" property.

The confirming of this property is subtle because its invariant breaks down when D is a deduction method only slightly stronger than either semantic tableaux or Tab-1 deduction. Thus, Pudlák's and Solovay's work [29, 34] implies Theorem 4.2's analog fails when D represents Hilbert deduction, and [43] showed its generalization fails even when D represents Tab-2 deduction.

5 A Finitized Generalization of Theorem 4.2's Methodology

One difficulty with $\text{IS}_D(A)$ was is that it employed an infinite number of different incarnations of sentence (4) in its Group-2 schema (since it contained one incarnation of this sentence for each Π_1^* sentence Φ in L^* 's language). Such a Group-2 schema is awkward because it simulates A 's Π_1^* knowledge almost via a brute-force enumeration.

Our Definition 5.1 and Theorems 5.2 and 5.7 will show how to mostly overcome this problem by compressing the infinite number of instances of sentence (4) in $\text{IS}_D(A)$'s Group-2 schema into a purely finite structure.

Definition 5.1 Let β denote any finite set of axioms that have Π_1^* encodings. Then $\text{IS}_D^\#(\beta)$ will denote an axiom system, similar to $\text{IS}_D(A)$, except its Group-2 scheme will employ β 's set of axioms, instead of using an infinite number of applications of statement (4)'s scheme. (Thus, the “*I am consistent*” statement in $\text{IS}_D^\#(\beta)$'s Group-3 axiom will be the same as before, except that the “*I am*” fragment of its self-referencing statement will reflect these changes in Group-2 in the obvious manner.)

Theorem 5.2 *Let D again denote either semantic tableaux or Tab-1 deduction, and β again denote a set of Π_1^* axioms. Then $\text{IS}_D^\#(\beta)$ will be consistent whenever all β 's axioms hold true under the standard model. (In other words, $\text{IS}_D^\#(\beta)$ will satisfy an analog of Theorem 4.2's consistency preservation property for $\text{IS}_D(A)$.)*

Theorem 5.2's proof is almost identical to [44]'s proof of Theorem 4.2. Its proof is too lengthy to repeat here. Instead §9 will briefly summarize its proof. This abbreviated discussion should be sufficient to explain the gist behind the proof's core formalism, without delving into [44]'s full details.

Our next definition will enable us to formalize the main application of Theorem 5.2 that will be considered here. It will essentially explain how **finite-sized** self-justifying logics can provide an **infinite amount** of “kernelized” Π_1^* styled information.

Definition 5.3 Let $\text{Test}_i(t, x)$ denote any Δ_0^* formula, and $\ulcorner \Psi \urcorner$ denote Ψ 's Gödel number. Then $\text{Test}_i(t, x)$ will be called a **Kernelized Formula** iff Peano Arithmetic can prove every Π_1^* sentence Ψ satisfies (6)'s identity:

$$\Psi \iff \forall x \text{ Test}_i(\ulcorner \Psi \urcorner, x) \quad (6)$$

There are infinitely many Δ_0^* predicates $\text{Test}_1(t, x)$, $\text{Test}_2(t, x)$, $\text{Test}_3(t, x)$... satisfying this kernelized condition (one of which is illustrated by Example 5.4). An enumerated list of all the available kernels is called a **Kernel-List**.

Example 5.4 The set of true Σ_1^* sentences is r.e. This implies there exists a Δ_0^* formula, called say $\text{Probe}(g, x)$, such that g is the Gödel number of a Σ_1^* statement that holds true in the Standard Model if and only if (7) is true:

$$\exists x \quad \text{Probe}(g, x) \wedge x \geq g \quad (7)$$

Now, let $\text{Pair}(t, g)$ denote a Δ_0^* formula that specifies t is the Gödel number of a Π_1^* statement and g is the Σ_1^* formula which is its negation. Then our notation implies that t is a true Π_1^* statement if and only if (8) holds true:

$$\forall x \quad \neg [\exists g \leq x \quad \text{Pair}(t, g) \wedge \text{Probe}(g, x)] \quad (8)$$

Thus if $\text{Test}_0(t, x)$ denotes the Δ_0^* formula of $\neg [\exists g \leq x \quad \text{Pair}(t, g) \wedge \text{Probe}(g, x)]$, it is one example of what Definition 5.3 would call a “Kernelized Formula”.

Definition 5.5 Let us recall Definition 5.3 defined **Kernel-List** to be an enumeration of all the kernelized formulae $\text{Test}_1(t, x)$, $\text{Test}_2(t, x)$, $\text{Test}_3(t, x)$ Assuming $\text{Test}_i(t, x)$ is the i -th element in this list and Ψ is an arbitrary Π_1^* sentence, the **i-th Kernel Image** of Ψ will be defined as the following Π_1^* sentence:

$$\forall x \quad \text{Test}_i(\lceil \Psi \rceil, x) \quad (9)$$

Example 5.6 The Definitions 5.3 and 5.5 suggest that there is a subtle relationship between a sentence Ψ and its i -th kernel image. This is because Definition 5.3 indicates that Peano Arithmetic can prove the invariant (6), indicating that Ψ is equivalent to its i -th kernel image. However, a weak axiom system can be plausibly uncertain about whether this equivalence does formally hold. This invariant is duplicated below:

$$\Psi \iff \forall x \quad \text{Test}_i(\lceil \Psi \rceil, x) \quad (10)$$

Thus if a weak axiom system proves statement (9) (rather than Ψ), it will not be able to equate these two results (unless it is able to verify (10)'s identity). This problem will apply to Theorem 5.7's formalism. However, Theorem 5.7 will still remain of much interest because §6 will illustrate a methodology that can overcome many of Theorem 5.7's limitations.

Theorem 5.7 Let A denote any system, whose axioms hold true in arithmetic's standard model, and i denote the index of any of Definition 5.3's kernelized formulae $\text{Test}_i(t, x)$. Then it is possible to construct a finite-sized collection of Π_1^* sentences, called say $\beta_{A,i}$, where $IS_D^\#(\beta_{A,i})$ satisfies the following invariant:

If Ψ is one of the Π_1^* theorems of A then $IS_D^\#(\beta_{A,i})$ can prove (9)'s statement (e.g. it will prove the “the i -th kernelized image” of Ψ).

Proof Sketch: Our justification of Theorem 5.7 will use the following notation:

1. $\text{Check}(t)$ will denote a Δ_0^* formula that produces a Boolean value of “True” when t represents the Gödel number of a Π_1^* sentence.
2. $\text{HilbPrf}_A(t, q)$ will denote a Δ_0^* formula that indicates q is a Hilbert-style proof of the theorem t from axiom system A .
3. For any kernelized $\text{Test}_i(t, x)$ formula, GlobSim_i will denote (11)’s Π_1^* sentence. (It will be called A ’s i -th “**Global Simulation Sentence**”.)

$$\forall t \ \forall q \ \forall x \ \{ \ [\ \text{HilbPrf}_A(t, q) \ \wedge \ \text{Check}(t) \] \implies \text{Test}_i(t, x) \ \} \quad (11)$$

In this notation, Theorem 5.7 shall be satisfied by any version of the axiom system $\text{IS}_D^\#(\beta)$, whose Group-2 schema β is a finite sized consistent set of Π_1^* sentences that has (11) as an axiom. (This includes the minimal sized such system, denoted as $\beta_{A,i}$, that has only (11) as an axiom.) This is because if Ψ is any Π_1^* theorem of A whose proof is denoted as \bar{p} , then both the Δ_0^* predicates of $\text{HilbPrf}_A(\ulcorner\Psi\urcorner, \bar{p})$ and $\text{Check}(\ulcorner\Psi\urcorner)$ will hold true. Moreover, $\text{IS}_D^\#$ ’s Group-1 axiom subgroup was defined so that it can automatically prove all Δ_0^* sentences that are true. Hence, $\text{IS}_D^\#(\beta_{A,i})$ will prove these two statements and then automatically corroborate (via axiom (11)) the further statement of:

$$\forall x \ \text{Test}_i(\ulcorner\Psi\urcorner, x) \quad (12)$$

Thus for each of the infinite number of Π_1^* theorems that A proves, the above defined formalism will prove a matching statement that corresponds to its i -th kernelized image. \square

6 L-Fold Generalizations of Theorem 5.7

Theorem 5.7 is of interest because every axiom system A will have its formalism $\text{IS}_D^\#(\beta_{A,i})$ prove the i -th kernelized image of every Π_1^* theorem that A proves. This fact is helpful because (6)’s invariance holds for all Π_1^* sentences. Moreover, our “U-Grounded” Π_1^* sentences capture all Conventional Arithmetic’s *crucial* Π_1 information because they can view multiplication as a 3-way Δ_0^* predicate $\text{Mult}(x, y, z)$ via (13)’s encoding of this predicate.

$$[(x = 0 \vee y = 0) \Rightarrow z = 0] \ \wedge \ [(x \neq 0 \wedge y \neq 0) \Rightarrow (\frac{z}{x} = y \ \wedge \ \frac{z-1}{x} < y)] \quad (13)$$

One difficulty with $\text{IS}_D^\#(\beta)$ and $\text{IS}_D^\#(\beta_{A,i})$ was mentioned by Example 5.6. It was that while Peano Arithmetic can corroborate (6)’s invariance for every Π_1^* sentence Ψ , these latter systems cannot also do so.

While there will probably never be a perfect method for fully resolving this challenge, there is a pragmatic engineering-style solution that is often available. This is essentially because our

proof of Theorem 5.7 employed a formalism β that used essentially only one axiom sentence (e.g. (11)'s Π_1^* declaration).

Since the $IS_D^\#(\beta)$ formalism was intended for use by any finite-sized system β , it is clearly possible to include any finite number of formally true Π_1^* sentences in β . Thus for some fixed constant L , one can easily let β include L copies of (11)'s axiom framework for a finite number of different $Test_1, Test_2 \dots Test_L$ predicates, each of which satisfy Definition 5.3's criteria for being kernelized formulae. In this case, $IS_D^\#(\beta)$ will formally map each initial Π_1^* theorem Ψ of some axiom system A onto L resulting different Π_1^* theorems of the form (9).

Remark 6.1 Our basic conjecture is, essentially, that a goodly number of issues, concerning logic-based engineering applications called say E , may have convenient solutions via self-justifying logics, that follow the preceding outlined L-fold strategy. Thus, we are suggesting that if β is a large-but-finite set of axioms, that consists of L copies of (11)'s axiom framework for different $Test_1 \dots Test_L$ predicates, then some future engineering applications E may possibly have their needs met by an $IS_D^\#(\beta)$ formalisms, when a software engineer meticulously chooses an appropriately constructed finite-sized β .

Remark 6.2 The preceding was not meant to overlook that the Second Incompleteness Theorem is a robust result, applying to all logics of sufficient strength. Our suggestion, however, is that computers are becoming so powerful, in both speed and memory size as the 21st century is progressing, that there will likely emerge engineering-style applications E that will benefit from $IS_D^\#(\beta)$'s self-referencing formalisms when a *large-but-finite-sized* β is delicately chosen. Moreover, it is of interest to speculate whether such computers can partially imitate a human being's approximate instinctive conjectures about his own consistency (that, as common colloquially held conjectures, seem to serve as *essential prerequisites* for humans to gain their motivation to cogitate).

Sections 7-10 will examine the preceding issues in further detail. Also, Section 9 will offer an intuitive summary of the techniques that [44] used to prove Theorem 4.2, so that the reader can understand [44]'s gist without reading the full details of [44]'s formal proof.

7 Comparing Type-M and Type-A Formalisms

Let us recall axioms (1)-(3) indicated Type-A systems differ from Type-M formalisms by treating Multiplication as a 3-way relation (rather than as a total function). For the sake of accurately characterizing what our systems can and cannot do, we have described our results as being fringe-like exceptions to the Second Incompleteness Theorem, from the perspective of an Utopian view of Mathematics, while perhaps being more significant results from an engineering-style perspective

of knowledge. Our goal in this section will be to amplify upon this perspective by taking a closer look at Type-A and Type-M formalisms.

Let us assume that $x_0 = 2 = y_0$ and that x_1, x_2, x_3, \dots and y_1, y_2, y_3, \dots are defined by the recurrence rules of:

$$x_{i+1} = x_i + x_i \quad \text{AND} \quad y_{i+1} = y_i * y_i \quad (14)$$

The sequences x_0, x_1, x_2, \dots and y_0, y_1, y_2, \dots will thus represent the growth rates associated with the addition and multiplication primitives, lying in the statements (2) and (3)'s “**Type-A**” and “**Type-M**” axioms.

Since $x_0 = 2 = y_0$, the rule (14) implies $y_n = 2^{2^n}$ and $x_n = 2^{n+1}$. The y_0, y_1, y_2, \dots sequence will, thus, grow much more quickly than the x_0, x_1, x_2, \dots sequence (since y_n 's binary encoding will have an $\text{Log}(y_n) = 2^n$ length while x_n 's binary encoding will have a shorter length of size $\text{Log}(x_n) = n + 1$).

Our prior papers noted that the difference between these growth rates was the reason that [39, 41, 47] showed all natural Type-M systems, recognizing integer-multiplication as a total function, were unable to recognize their tableaux-styled consistency — while [38, 40, 44] showed some Type-A systems could simultaneously prove all Peano Arithmetic Π_1^* theorems and corroborate their own tableaux consistency. Their gist was that a Gödel-like diagonalization argument, which causes an axiom system to become inconsistent as soon as it proves a theorem affirming its own tableaux consistency, stems, ultimately, from the exponential growth in the series y_0, y_1, y_2, \dots .

This growth will, thus, facilitate an intense cascading amount of self-referencing, using the identity $\text{Log}(y_n) \cong 2^n$, that will, ultimately, invoke the force of Gödel's seminal diagonalization machinery. It will thus raise the following enticing question:

* * * How natural are exponentially growing sequences, such as y_0, y_1, y_2, \dots , whose n -th member needs 2^n bits for its encoding, when such lengths are greater than the number of atoms in the universe when merely $n > 100$? Is the use of such a sequence for corroborating the Second Incompleteness Effect resting upon an almost artificial construct (with an inherently dizzying growth rate)?

We will not attempt to derive a Yes-or-No answer to Question * * * because we think that such a direct response is too simplistic. Our point is that both a positive and negative reply to * * * are useful in different respects. This because the theoretical existence of a sequence integers of y_0, y_1, y_2, \dots , whose binary encodings are doubling in length, is tempting from the perspective of an Utopian view of mathematics, while awkward from an engineering styled perspective. We therefore ask: “*Why not be tolerant of both perspectives?*”

One virtue of this tolerance is it ushers in a greater understanding for the statements * and ** that Gödel and Hilbert made during 1926 and 1931. This is because the Incompleteness Theorem

demonstrates no formalism can display an understanding of its own consistency in an idealized Utopian sense. On the other hand, §6 suggested these two remarks by Gödel and Hilbert might receive more sympathetic interpretations, if one sought to explore such questions from a less ambitious almost engineering-style perspective.

Our main thesis is supported by a theorem from [46]. It indicated that tableaux variations of self-justifying systems have no difficulty in recognizing that an infinitized generalization of a computer's floating point multiplication (with rounding) is a total function. The latter differs from integer-multiplication, by not having its output become double the length of its input when a number is multiplied by itself. Thus, the intuitive reason [46]'s multiplication-with-rounding operation is compatible with self-justification is because it avoids the inexorable exponential growth under rule (14)'s sequence $y_0, y_1, y_2 \dots$.

Also, Theorem 7.1 indicates self-justifying logics can view double-precision integer multiplication similarly as a total function. In particular for any arbitrary pair of integers (a, b) , let us employ a notation convention where:

1. **Size(a,b)** denotes the maximum of $\lceil 1 + \log_2 a \rceil$ and $\lceil 1 + \log_2 b \rceil$.
2. The quantities **Left**(a, b) and **Right**(a, b) represent the multiplicative product of the integers a and b , insofar as Right(a, b) represents the rightmost bits of this product of length Size(a, b), and Left(a, b) encodes the remaining bits to the left of Right(a, b) (whose length will also be bounded by Size(a, b)).

Within this context, Theorem 7.1 indicates self-justifying logics self-justification are able to view double-precision integer-multiplication as a total function.

Theorem 7.1 *Let us assume the A in $IS_D(A)$ and β in $IS_D^\#(\beta)$ are axiom systems all of whose Π_1^* theorems are true statements under the standard model of the natural numbers. Then if D corresponds to either semantic tableaux or Tab-1 deduction, it is possible to formalize systems $A^* \supseteq A$ and $\beta^* \supseteq \beta$ such that $IS_D(A^*)$ and $IS_D^\#(\beta^*)$ are self-justifying extensions of respectively $IS_D(A)$ and $IS_D^\#(\beta)$ which can recognize each of the double-multiplicative precision operations of $Size(a, b)$, $Left(a, b)$ and $Right(a, b)$ as total functions.*

Proof Sketch; The justification of Theorem 7.1 is similar to [46]'s analysis of Floating Point Multiplication (with rounding). Our proof of Theorem 7.1 will therefore be quite abbreviated.

The first point is that it is straightforward to develop three Δ_0^* formulae, called $\theta_1(a, b, y)$, $\theta_2(a, b, y)$ and $\theta_3(a, b, y)$, that are the graphs of the functions $Size(a, b)$, $Left(a, b)$ and $Right(a, b)$. It is also easy to construct a finite set of Π_1^* sentences, holding true in the Standard Model, called γ , that know how to correctly interpret these three Δ_0^* formulae, insofar as γ knows:

1. For each a and b , there exists no more than one integer y that satisfies each of our three $\theta_j(a, b, y)$ formulae.
2. For each a and b , our three $\theta_j(a, b, y)$ formulae correctly simulate the graphs of the respective functions of $\text{Size}(a, b)$, $\text{Left}(a, b)$ and $\text{Right}(a, b)$.

Since our U-Grounding language contains the built-in function primitives of “Maximum” and “Double(x)”, the Group-1 component of IS_D and $\text{IS}_D^\#$ can easily verify that the operation $F(a, b)$, defined below is a total function:

$$F(a, b) = \text{Double}(\text{Double}(\text{Double}(\text{Max}(a, b)))) \quad (15)$$

This implies, in turn, that there exists a Π_1^* sentence, called γ^* , that will enable our formalism to verify that each of $\text{Size}(a, b)$, $\text{Left}(a, b)$ and $\text{Right}(a, b)$ are total functions (simply because their output values are less than $F(a, b)$ ’s output).

The main point is that the hypothesis of Theorem 7.1 indicated that all the axioms of A and β did hold true under the Standard Model, and the preceding paragraph showed the same was true for all the axioms in γ and γ^* . Hence all the axioms in $A^* = A + \gamma + \gamma^*$ and $\beta^* = \beta + \gamma + \gamma^*$ also hold true in the Standard Model. By Theorems 4.2 and 5.2, this implies that $\text{IS}_D(A)$ and $\text{IS}_D^\#(\beta)$ are self-justifying formalism satisfying Theorem 7.1’s claims. \square

Remark 7.2 One subtle aspect is that our positive results, involving [46]’s floating point multiplication primitive and Theorem 7.1’s analogous double precision multiplication operation, *should not be confused* with a quite different exploration of integer multiplication in the context of our analysis of Herbrand consistency in [48]. The latter took advantage of the fact that our deployed Herbrand-styled proofs in [48] were exponentially longer than their tableaux counterparts (thus allowing [48] to formalize a limited use of multiplication). This was because [48]’s deductive methods were exponentially less efficient at an inherent level. Thus [48]’s result, while of theoretical interest, is basically irrelevant to the core engineering environments, which constitutes the main focus of Theorems 4.2–7.1.

Remark 7.2’s contrast between [48]’s results and Theorem 7.1 is, once again, connected to the distinction between the engineering and mathematical viewpoints about the main intentions of theorem-proving. Theorem 7.1 is helpful from an engineering perspective because most pragmatic applications of integer multiplication are analogous to either computerized double-precision multiplication or its quadruple-precision or hexagonal generalizations.

Theorem 7.1 (and its quadruple-precision and hexagonal generalizations) indicate these operations are compatible with a formalism recognizing its own semantic tableaux consistency.

8 A Different Type of Evidence Supporting Our Thesis

Let us recall Pudlák and Solovay [29, 34] observed that essentially all Type-S systems, containing merely statement (1)'s axiom that successor is a total function, cannot verify their own consistency under Hilbert deduction. (See also related work by Buss-Ignjatovic [5], Hájek and Švejdar [35], as well as [40]'s Appendix A.)

It turns out that [43] generalized these results to show that Equation (2)'s Type-A systems are unable to verify their own consistency under the Tab–2 deduction system (defined in §3). At the same time, the IS_D and $\text{IS}_D^\#$ frameworks, from Sections 4 and 5, can verify their own consistency under Tab–1 deduction. Our goal in this section will be to illustrate how the tight contrast between these positive and negative results is analogous to the differing growth rates of the sequences x_0, x_1, x_2, \dots and y_0, y_1, y_2, \dots from rule (14).

During our discussion $G_i(v)$ will denote the scalar-multiplication operation that maps an integer v onto $2^{2^i} \cdot v$. Also, Υ_i will denote the statement, in the U-Grounding language, that declares that G_i is a total function. Our paper [43] proved that Υ_i has a Π_2^* encoding. It also implied that G_i satisfied:

$$G_{i+1}(v) = G_i(G_i(v)) \quad (16)$$

It was noted in [43] that this identity implies one can construct an axiom system β , comprised of solely Π_1^* sentences, where a semantic tableaux proof can establish Υ_{i+1} from $\beta + \Upsilon_i$ in a constant number of steps. This implies, in turn, that a Tab–2 proof from β will require no more than $O(n)$ steps to prove Υ_n (when it uses the obvious n-step process to confirm in chronological order $\Upsilon_1, \Upsilon_2, \dots, \Upsilon_n$.)

These observations are significant because $G_n(1) = 2^{2^n}$. Thus, [43] established that a Tab–2 proof from β can verify in only $O(n)$ steps that this quite large integer exists.

This example is helpful because it illustrates the difference between the growth speeds under Tab–1 and Tab–2 deduction, is analogous to the differing growth rates of the sequences x_0, x_1, x_2, \dots and y_0, y_1, y_2, \dots from rule (14). Hence once again, a faster growth-rate will usher in the Second Incompleteness Theorem's power (e.g. see [43]).

This analogy suggests that the Second Incompleteness Theorem has different implications from the perspectives of Utopian and engineering theories about the intended applications of mathematics. Thus, a Utopian may possibly be comfortable with a perspective, that contemplates sequences y_0, y_1, y_2, \dots with elements growing in length at an exponential speed, but many engineers may be suspicious of such growths.

A hard-core engineer, in contrast, might surmise that the inability of self-justifying formalisms to be compatible with Tab–2 deduction is not as disturbing as it might initially appear to be. This is because Tab–2 differs from Tab–1 deduction by producing exponential growths that are

so sharp that their material realization has no analog in the everyday mechanical reality that is the focus of an engineer’s interest.

Our personal preference is for a perspective lying half-way between that of an Utopian mathematician and a hard-nosed engineer. Its dualistic approach suggests some form of diluted partial agreement with Hilbert’s goals in ** (in a context where the broad significance of the Second Incompleteness Theorem is obviously undeniable).

9 Outline of Theorem 5.2’s Proof and Its Implications

The prior two sections of this article offered an intuitive explanation about why our self-justifying axiom systems needed omit the assumption that multiplication is a total function and could verify their consistency only under semantic tableaux and Tab–1 deduction.

We already noted Theorem 5.2’s observation that $\text{IS}_D^\#$ is consistency-preserving has essentially an analogous proof as [44]’s demonstration that IS_D is consistency-preserving. It is not our intention to repeat such a proof here.

Instead, our goal will be to provide a brief overview of the techniques that [44] had used. This overview will be sufficient for a reader to appreciate the underlying intuition.

More precisely, two different types of proofs of Theorem 4.2 had appeared in our 2002 conference paper [42] and subsequent journal paper [44]. The latter was more appropriate for an archival journal because its self-justification result applied to both semantic tableaux deduction and its Tab–1 generalization. The more compressed conference paper [42] proved the analog of Theorem 4.2 only for tableaux deduction (using a technique that was somewhat shorter than [44]’s more elaborate result). Our summary of Theorem 4.2’s proof, here, will focus on the semantic tableaux deduction methodology so it can apply to either of [42] or [44]’s methods.

Both of [42, 44] justified Theorem 4.2 by means of proofs by contradiction. Thus if Theorem 4.2 was false, they noted a pair of proofs for a Π_1^* sentence and its negation would exist from $\text{IS}_D(A)$.

Let us call these two proofs P and Q . Then [42, 44] both showed (using different constructions) that one could construct from (P, Q) two other proofs (p, q) of another Π_1^* sentence and its negation such that:

$$\text{Max}(p, q) < \text{Max}(P, Q) \quad (17)$$

The inequality in (17) is significant because it will enable our proofs-by-contradiction to establish the non-existence of an ordered pair (P, Q) violating Theorem 4.2’s assumption. This is because (17) would otherwise violate the Principle of Induction by showing there exists no such minimal ordered pair (P, Q) eschewing Theorem 4.2’s formalism.

The exact details of these proofs by contradictions are too lengthy to fully summarize here. For the case where D in Theorem 4.2 is the semantic tableaux deduction method, they used the fact that if (P, Q) was the ordered pair with minimal $\text{Max}(P, Q)$ value violating Theorem 4.2's hypothesis, then one could isolate two particular root-to-leaf paths in the tableaux proofs P and Q that would enable us to construct an additional pair (p, q) that violated Theorem 4.2 and satisfied (17)'s inequality.

This construction of (p, q) from (P, Q) utilized the fact that Theorem 4.2's axiom system $\text{IS}_D(\alpha)$ recognized addition but not multiplication as a total function. Otherwise, Theorem 4.2's delicate proof-by-contradiction would collapse entirely (as a result of the exponentially faster growth properties of multiplication that was formalized by the series y_1, y_2, y_3, \dots under Line (14)'s recurrence relationship).

These observations reinforce the theme of §7 about the contrast between the slower growing series x_1, x_2, x_3, \dots and its exponentially faster counterpart y_1, y_2, y_3, \dots under Line (14)'s recurrence relationship. These two series defined the growth rates produced by the addition and multiplication function symbols as, respectively, $x_n = 2^{n+1}$ and $y_n = 2^{2^n}$. They thus illustrated how multiplication's faster growth rate leads to such a dizzying exponential speed-up, that makes one at least partially sympathetic to a hard-nosed engineer's skepticism about its implications.

Thus if one were to preclude such a dizzying growth rate then a partial justification of a diluted version of Hilbert's consistency program would arise, in the context of systems possessing *weak but well defined* knowledges of their own consistency. On the other hand, if the conventional assumption that multiplication is a total function is presumed, then the traditional interpretation of the Second Incompleteness Theorem will prevail.

10 Related Reflection Principles

An added point is that there are many types of self-justifying systems available, with some better suited for engineering environments than others.

For instance, our initial 1993 paper [38] employed a Group-3 “*I am consistent*” axiom that was much weaker than the current specimen. The distinction was that [38]'s self-consistency declaration excluded merely the existence of a semantic tableaux proof of $0 = 1$ from itself, while the sentence (5) is more elaborate because it excludes the existence of simultaneous proofs of a Π_1^* theorem and its negation.

Ideally, one would like to develop self-justifying systems S that can corroborate the validity of (18)'s reflection principle for all sentences Φ .

$$\forall p \ [\text{Prf}_S^D(\Gamma\Phi^\neg, p) \Rightarrow \Phi] \quad (18)$$

Löb's Theorem establishes, however, that all systems S , containing Peano Arithmetic's strength,

are able to prove (18)'s invariant *only in the degenerate case* where they do prove Φ itself. Also, the Theorem 7.2 from [40] showed essentially all axiom systems, *weaker* than Peano Arithmetic, are unable to prove (18) for all Π_1^* sentences Φ simultaneously. Thus, Theorem 10.1 will be near optimal:

Theorem 10.1 *Let us recall that the difference between Theorem 4.2's axiom system $IS_D(A)$ and Theorem 5.7's formalism $IS_D^\#(\beta_{A,i})$ was that the latter replaced $IS_D(A)$'s infinite-sized Group-2 axiom schema with $IS_D^\#(\beta_{A,i})$'s compact 1-sentence axiom (11), so that the latter system could at least verify (19)'s kernelized statement for each Π_1^* theorem that A proved.*

$$\forall x \ Test_i(\Gamma \Psi \sqcap, x) \quad (19)$$

Let likewise $IS_\#^\lambda(\beta_{A,i})$ denote the modification of [40]'s $IS^\lambda(A)$ self-justifying system that replaces the latter's Group-2 schema with (11)'s more compact single-sentence axiom declaration (and then has its Group-3 "I am consistent" axiom statement reflect this change, once again). Then in a context where "semtab" is an abbreviation for semantic tableaux deduction, the formalism $IS_\#^\lambda(\beta_{A,i})$ will be able to:

1. Verify that semantic tableaux deduction supports the following analog of (18)'s self-reflection principle under $IS_\#^\lambda(\beta_{A,i})$ for any Δ_0^* and Σ_1^* sentences Φ :

$$\forall p [Prf_{IS_\#^\lambda(\beta_{A,i})}^{\text{semtab}}(\Gamma \Phi \sqcap, p) \Rightarrow \Phi] \quad (20)$$

2. Verify (21)'s more general "**root-diluted**" reflection principle for $IS_\#^\lambda(\beta_{A,i})$ whenever θ is Σ_1^* and Φ is a Π_2^* sentence of the form " $\forall u_1 \dots \forall u_n \ \theta(u_1 \dots u_n)$ ".

$$\forall p [Prf_{IS_\#^\lambda(\beta_{A,i})}^{\text{semtab}}(\Gamma \Phi \sqcap, p) \implies \forall x \ \forall u_1 < \sqrt{x} \ \dots \ \forall u_n < \sqrt{x} \ \theta(u_1 \dots u_n)] \quad (21)$$

As is suggested by the similarity between the definitions of $IS^\lambda(A)$ and $IS_\#^\lambda(\beta_{A,i})$, the proof of Theorem 10.1 is essentially identical to [40]'s analysis of $IS^\lambda(A)$. For the sake of brevity, we will not repeat the relevant proof here.

Analogous to our other results, Theorem 10.1 reinforces our theme about how exceptions to the Second Incompleteness Theorem may appear to be *quite minor* from the perspective of an Utopian view of mathematics, while being significant from an engineering standpoint. In Theorem 10.1's particular case, this is because:

- A.** The ability of Theorem 10.1's system to support (20)'s self-reflection principle under tableaux proofs for any Δ_0^* and Σ_1^* sentence, as well as to support (21)'s root reflection principle for Π_2^* sentences, is clearly significant.
- B.** The incompleteness result of [40]'s Theorem 7.2 imposes, however, sharp limitations upon Item A's generality (in that it cannot be extended to fully all Π_1^* sentences, *in an undiluted sense*).

Thus, the tight fit between A and B is reminiscent of other slender borderlines, that separated generalizations and boundary-case exceptions for the Incompleteness Theorem, explored earlier. Once again, the Second Incompleteness Theorem is seen as robust, from an idealized Utopian perspective on mathematics, while permitting caveats from engineering styled perspectives.

This dualistic viewpoint allows one to nicely share *partial (and not full)* agreement with Hilbert's main aspirations in **, while also appreciating the stunning achievement of the Second Incompleteness Theorem.

11 Concluding Remarks

At a purely technical level, this article has reached beyond our prior papers in several respects, including §5's demonstration that any initial system A can have a kernelized image of its Π_1^* knowledge duplicated by $\text{IS}_D^\#(\beta_{A,i})$'s **strictly finite sized** self-justifying system, as well as Section 6's and Remark 6.2's quite pragmatic L-fold generalizations of Theorem 5.7.

These perspectives help resolve the mystery that has enshrouded the Second Incompleteness Theorem and the statements * and ** of Gödel and Hilbert. This is because we have *meticulously separated* the goals of a pristine theoretical study of mathematical logic from those of a *finite-sized* axiomatic subset of mathematics, intended for modeling mostly an engineering environment.

There is no question that Gödel's Second Theorem is ideally robust, relative to a purely pristine approach to mathematics. On the other hand, we suspect Hilbert was *half-way correct* by speculating in ** about humans possessing a knowledge about their own consistency, *in at least some weak and tender sense*, as essentially a prerequisite for *psychologically motivating* their cogitations. Thus in a context where the limitations of axiom systems, that fail to recognize multiplication as a total function, are manifestly obvious, it is legitimate to inquire whether some future specialized 21st century computers might find some *partial-albeit-and-not-full* redeeming value in formalisms having *weak-style* knowledges of their Tab-1 consistency, as well as possessing a knowledge of Peano Arithmetic's Π_1^* theorems.

Sections 5-10 were, thus, intended to provide a unified broad-scale interpretation of our diverse earlier results that had appeared in [38]-[48]. In a context where the Incompleteness Theorem is sufficiently ubiquitous to preclude Hilbert's aspirations in ** from ever being fully realized, they show how some *fragmentary portion* of Hilbert's conjectures can be corroborated by *judiciously weakened* logics, using a formalism, that is *much less* than ideally robust, *although not fully immaterial*.

Such partial evasions of the Second Incompleteness Effect are certainly not broad-scale, but they do corroborate a fragment of what Gödel and Hilbert had sought as their desired goals, expressed in the statements * and **.

Acknowledgments: I thank Bradley Armour-Garb and Seth Chaiken for many useful suggestions about how to improve the presentation of our results. This research was partially supported by NSF Grant CCR 0956495.

References

- [1] Z. Adamowicz. Herbrand consistency and bounded arithmetic. *Fundamenta Mathematicae*, 171(3):279–292, 2002.
- [2] Z. Adamowicz and P. Zbierski. On Herbrand consistency in weak theories. *Archive for Mathematical Logic*, 40(6):399–413, 2001.
- [3] A. Bezboruah and J. C. Shepherdson. Gödel’s second incompleteness theorem for Q. *Jour. Symb. Logic*, 41(2):503–512, 1976.
- [4] S. R. Buss. *Bounded Arithmetic*. Studies in Proof Theory, Lecture Notes 3, published by Bibliopolis, 1986. (Revised version of Ph. D. Thesis.).
- [5] S. R. Buss and A. Ignjatovic. Unprovability of consistency statements in fragments of bounded arithmetic. *Annals Pure and Applied Logic*, 74(3):221–244, 1995.
- [6] J. W. Dawson. *Logical Dilemmas the life and work of Kurt Gödel*. AKPeters, 1997.
- [7] S. Feferman. Arithmetization of metamathematics in a general setting. *Fundamenta Mathematicae*, 49:35–92, 1960.
- [8] M. Fitting. *First Order Logic and Automated Theorem Proving*. Springer, 1996.
- [9] H. M. Friedman. On the consistency, completeness and correctness problems. Technical report, Ohio State Univ, 1979. See Pudlák [30]’s summary of this result.
- [10] H. M. Friedman. Translatability and relative consistency. Technical report, Ohio State Univ, 1979. See Pudlák [30]’s summary of this result.
- [11] K. Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Math. Phys.*, 38:173–198, 1931.
- [12] K. Gödel. The present situation in the foundations of mathematics. In S. Feferman, J. W. Dawson, W. Goldfarb, C. Parsons, and R. Solovay, editors, *Collected Works Volume III: Unpublished Essays and Lectures*, pages 45–53. Oxford University Press, 2004. Our quotes from this 1933 lecture come from its page 52.
- [13] R. Goldstein. *Incompleteness The Proof and Paradox of Kurt Gödel*. Norton, 2005.
- [14] P. Hájek. Mathematical fuzzy logic and natural numbers. *Fundamenta Mathematicae*, 81:155–163, 2007.
- [15] P. Hájek. Towards metamathematics of weak arithmetics over fuzzy. *Logic Journal of the IPL*, 19(3):467–475, 2011.
- [16] P. Hájek and P. Pudlák. *Metamathematics of First Order Arithmetic*. Springer Verlag, 1991.
- [17] D. Hilbert. Über das Unendliche. *Mathematische Annalen*, 95:161–191, 1926.
- [18] D. Hilbert and P. Bernays. *Grundlagen der Mathematik (Vol II)*. Springer, 1939.
- [19] R. G. Jeroslow. Consistency statements in formal theories. *Fundamenta Mathematicae*, 72:17–40, 1971.
- [20] S. C. Kleene. On notation for ordinal numbers. *Jour. Symb. Logic*, 3(1):150–156, 1938.
- [21] L. A. Kolodziejczyk. On the Herbrand notion of consistency for finitely axiomatizable fragments of bounded arithmetic theories. *Jour. Symb. Logic*, 71(2):624–638, 2006.
- [22] J. Krajíček. *Bounded Propositional Logic and Complexity Theory*. Cambridge University Press, 1995.
- [23] G. Kreisel and G. Takeuti. Formally self-referential propositions in cut-free classical analysis and related systems. *Dissertationes Mathematicae*, 118:1–50, 1974.
- [24] M. H. Löb. A solution of a problem of Leon Henkin. *Jour. Symb. Logic*, 20(2):115–118, 1955.

- [25] E. Mendelson. *Introduction to Mathematical Logic*. Chapman Hall, 2010.
- [26] E. Nelson. *Predicative Arithmetic*. Math Notes, Princeton Univ Press, 1986.
- [27] R. Parikh. Existence and feasibility in arithmetic. *Jour. Symb. Logic*, 36(3):494–508, 1971.
- [28] C. H. Parsons. On n -quantifier induction. *Jour. Symb. Logic*, 37(3):466–482, 1972.
- [29] P. Pudlák. Cuts, consistency statements and interpretations. *Jour. Symb. Logic*, 50(2):423–442, 1985.
- [30] P. Pudlák. On the lengths of proofs of consistency. In *Collegium Logicum: Annals of the Kurt Gödel*, volume 2, pages 65–86. Springer-Wien, 1996.
- [31] H. A. Rogers. *Theory of Recursive Functions and Effective Compatibility*. McGraw Hill, 1967.
- [32] S. Salehi. Herbrand consistency of some arithmetical theories. *Jour. Symb. Logic*, 77(3):807–827, 2012.
- [33] R. M. Smullyan. *First Order Logic*. Diver, 1995.
- [34] R. M. Solovay. Telephone conversation in 1994 describing Solovay’s generalization of one of Pudlák’s theorems [29], using some methods of Nelson and Wilkie-Paris [26, 37]. (The Appendix A of [40] offers a 4-page summary of this conversation.).
- [35] V. Svejdar. An interpretation of Robinson arithmetic in its Grzegorczyk’s weaker variant. *Fundamenta Mathematicae*, 81:347–354, 2007.
- [36] A. Visser. Faith and falsity. *Annals Pure and Applied Logic*, 131(1):103–131, 2005.
- [37] A. J. Wilkie and J. B. Paris. On the scheme of induction for bounded arithmetic. *Annals Pure and Applied Logic*, 35:261–302, 1987.
- [38] D. E. Willard. Self-verifying axiom systems. *Springer LNCS*, 713:325–336, 1993. Proceedings of Third Kurt Gödel Symposium.
- [39] D. E. Willard. The semantic tableaux version of the second incompleteness theorem extends almost to Robinson’s arithmetic Q. *Springer LNCS*, 1847:415–430, 2000. Proceedings of “Tableaux-2000” Conference.
- [40] D. E. Willard. Self-verifying systems, the incompleteness theorem and the tangibility reflection principle. *Jour. Symb. Logic*, 66(2):536–596, 2001.
- [41] D. E. Willard. How to extend the semantic tableaux and cut-free versions of the second incompleteness theorem almost to Robinson’s arithmetic Q. *Jour. Symb. Logic*, 67(1):465–496, 2002.
- [42] D. E. Willard. Some new exceptions for the semantic tableaux version of the second incompleteness theorem. *Springer LNCS*, 2381:281–297, 2002. (“Tableaux-2002” Conference Proceedings).
- [43] D. E. Willard. A version of the second incompleteness theorem for axiom systems that recognize addition but not multiplication as a total function. In V. Hendricks, F. Neuhaus, S. A. Pederson, U. Scheffler, and H. Wansing, editors, *First Order Logic Revisited*, pages 337–368, Berlin, 2004. Logos Verlag.
- [44] D. E. Willard. An exploration of the partial respects in which an axiom system recognizing solely addition as a total function can verify its own consistency. *Jour. Symb. Logic*, 70(4):1171–1209, 2005.
- [45] D. E. Willard. A generalization of the second incompleteness theorem and some exceptions to it. *Annals Pure and Applied Logic*, 141(3):472–496, 2006.
- [46] D. E. Willard. On the available partial respects in which an axiomatization for real valued arithmetic can recognize its consistency. *Jour. Symb. Logic*, 71(4):1189–1199, 2006.
- [47] D. E. Willard. Passive induction and a solution to a Paris-Wilkie open question. *Annals Pure and Applied Logic*, 146(2):124–149, 2007.
- [48] D. E. Willard. Some specially formulated axiomizations for $\text{I}\Sigma_0$ manage to evade the Herbrandized version of the second incompleteness theorem. *Information and Computation*, 207(10):1078–1093, 2009.
- [49] D. E. Willard. On the broader epistemological significance of self-justifying axiom systems. *Springer LNCS*, 8652:221–236, 2014. (an earlier more abbreviated version of this current article that had appeared in the Proceedings of 21st Wollic Conference).
- [50] C. Wrathall. Rudimentary predicates and relative computation. *Siam Journal on Computing*, 7:194–209, 1978.
- [51] P. Yourgrau. *A World Without Time: The Forgotten Legacy of Gödel and Einstein*. Basic Books, 2005. See page 58 for the passages we have quoted.

Appendix A: Definition of a Semantic Tableaux Proof

The definition of a semantic tableaux proof, provided here, will be similar to analogous definitions used in say Fitting's or Smullyan's textbooks [8, 33].

During our discussion, a **Φ -Based Candidate Tree** for an axiom system α will be defined to be a tree structure whose root corresponds to the sentence $\neg\Phi$, rewritten in prenex normal form, and whose all other nodes are either axioms of α or deductions from higher nodes of the tree (using the Rules 1-6 defined below). More precisely, our six rules (below) have " $\mathcal{A} \mapsto \mathcal{B}$ " denote that \mathcal{B} is a valid deduction from \mathcal{A} . They specify when such a descendant node \mathcal{B} is allowed to appear below an ancestor \mathcal{A} in a candidate tree:

1. $\Upsilon \wedge \Gamma \mapsto \Upsilon$ and $\Upsilon \wedge \Gamma \mapsto \Gamma$.
2. $\neg\neg\Upsilon \mapsto \Upsilon$. Other rules for the " \neg " symbol include: $\neg(\Upsilon \vee \Gamma) \mapsto \neg\Upsilon \wedge \neg\Gamma$, $\neg(\Upsilon \Rightarrow \Gamma) \mapsto \Upsilon \wedge \neg\Gamma$, $\neg(\Upsilon \wedge \Gamma) \mapsto \neg\Upsilon \vee \neg\Gamma$, $\neg\exists v \Upsilon(v) \mapsto \forall v \neg\Upsilon(v)$ and $\neg\forall v \Upsilon(v) \mapsto \exists v \neg\Upsilon(v)$
3. A pair of sibling nodes Υ and Γ is allowed in a proof tree when their ancestor is $\Upsilon \vee \Gamma$.
4. A pair of sibling nodes $\neg\Upsilon$ and Γ is allowed in a proof tree when their ancestor is $\Upsilon \Rightarrow \Gamma$.
5. $\exists v \Upsilon(v) \mapsto \Upsilon(u)$ where u denotes a newly introduced "Parameter Symbol".
6. $\forall v \Upsilon(v) \mapsto \Upsilon(t)$ where t denotes a "Composite Term". These terms here are built out of combination of the U-Grounding Function symbols, the constant symbols representing "0" and "1" and the parameter symbols u_1, u_2, \dots, u_n , where each u_i was previously introduced by applying Rule 5 to an ancestor of the node storing " $\Upsilon(t)$ ".

Define a particular leaf-to-root branch in a candidate tree T to be **Closed** iff it contains both some sentence Υ and its negation $\neg\Upsilon$. A **Semantic Tableaux** proof of Φ will then be defined to be a candidate tree whose root stores the sentence $\neg\Phi$ (written in prenex normal form) and all of whose root-to-leaf branches are closed.

Our current article has, for simplicity, used the preceding definition for a semantic tableaux proof. Some of our prior articles used a minor modification of this definition where there were two additional deduction rules for "bounded quantifiers" of the form " $\exists v \leq t \ \Upsilon(v)$ " and " $\forall v \leq t \ \Upsilon(v)$ ". It is technically unnecessary to use special rules for such bounded quantifiers because these two expressions can be treated as being equivalent to (22) and (23), respectively.

$$\exists v \quad v \leq t \wedge \Upsilon(v) \tag{22}$$

$$\forall v \quad v \leq t \Rightarrow \Upsilon(v) \tag{23}$$

Thus, we technically do not need special Elimination Rules for bounded quantifiers of the form " $\exists v \leq t \ \Upsilon(v)$ " and " $\forall v \leq t \ \Upsilon(v)$ " because statement (22) allows the former to be eliminated by applying Rules 5 and 1, and likewise (23) can be processed via Rules 6 and 4.

Appendix B: Summary of Gödel Encoding Method

Every generalization and boundary-case exception for the Second Incompleteness Theorem does require deploying a Gödel encoding methodology (to make it well defined). Such an encoding scheme will be called **Optimally Linearly Compressed** if it requires:

- A. Only $O(1)$ bits to store each occurrence of any logical symbol appearing in a tableaux proof (except for the objects that Items 5 and 6 of Appendix A called the i -th “variable” and “parameter” symbols).
- B. No more than $O(1 + \text{Log}(i))$ bits to encode a proof’s i -th “variable” and “parameter” symbols. (This $O(1 + \text{Log}(i))$ magnitude is unavoidable because there is no finite limit to the number of different variable and parameter objects that may appear in one of Appendix A’s semantic tableaux proofs.)

All our published results about either generalizations or boundary-case exception for the Second Incompleteness Theorem have used such optimally compressed encodings.

In particular, our scheme for encoding a semantic tableaux proof will use the following 24 language symbols:

1. The standard connective symbols of \wedge , \vee , \neg , \rightarrow , \forall and \exists .
2. Two left and two right parenthesis symbols denoted as: $($, $)$, $(\underline{})$ and $(\underline{})$.
3. Two symbols to represent the special constants of “0” and “1”.
4. Eight function symbols for representing the eight formal U-grounding functions of Addition, Doubling, Subtraction, Division, Logarithm, etc.
5. The relation symbols of “ $=$ ” and “ \leq ”.
6. The symbol \hat{V} for designating the presence of a basic variable v in a logical sentence.
7. The symbol \hat{U} for designating the presence of a parameter constant u in a logical sentence (which is produced by Appendix A’s deduction rule 5 for eliminating existential quantifiers).

Define a byte to be an unit consisting of six bits. We may think of a proof as comprising either a sequence of bytes or being an equivalent integer written in base 64. Each of the 24 symbols (above) will be given some unique 6-bit code, ranging between 32 and 55. Our method for representing the presence of the i -th variable v_i will be to encode it as a string comprised of $\lceil \log_{32}(i+1) \rceil + 1$ bytes, where the first byte is the “ \hat{V} ” symbol and the remaining bytes encode i as a base-32 number. The same convention will be used to denote the presence of the i -th parameter u_i except its first byte will be the “ \hat{U} ” symbol.

Our notation has employed *two types* of parenthesis symbols because the first pair of parenthesis symbols will have their usual meaning in punctuating a mathematical sentence, whereas the

latter pair of symbols $\underline{()}$ will *separate* the individual sentences in a Semantic Tableaux proof tree. For example, consider a tree which stores 1) the sentence ψ_1 as its root, 2) the sentences ψ_2 and ψ_3 as the root's children, and 3) ψ_4 as the child of ψ_3 . There are several possible notation conventions for using the $\underline{()}$ symbols to encode a Semantic Proof tree. Our encoding convention will presume ψ_i is an “ancestor” of ψ_j if and only if the range beginning with the parenthesis to ψ_i 's immediate left and continuing to the matching right parenthesis includes ψ_j . The example of our 4-node proof tree is thus encoded as:

$$\underline{(\psi_1}\underline{\psi_2)}\underline{\psi_3}\underline{\psi_4)}\underline{)}$$
 (24)

The preceding paragraph summarized our method for encoding semantic tableaux proofs. Its generalization for the encoding of Tab-1 proofs is straightforward. Thus if p_1, p_2, \dots, p_n collectively constitute a list of semantic tableaux proofs then the natural concatenation of their byte strings will be the corresponding Tab-1 proof.

This “Optimally Linearly Compressed” encoding scheme is essential because all the core axiom systems, employed in this article, are Type-A formalisms, that recognize Addition but not Multiplication as a total function. If such formalisms were less than optimally compressed then our main theorems would lose relevance because the formalization of unnecessarily expansive encodings would be awkward in the context of the slow growth properties of Type-A formalisms. Thus, our results carry much greater significance when their encodings of a proof satisfy the maximal compression properties, defined in this appendix.

Appendix C: Formal Encoding of the Group-3 Axiom

Let us recall Appendix A reviewed the definition of a semantic tableaux and Tab-1 proof, and Appendix B formalized the encodings of such proofs. The goal of this appendix will be to summarize the methodology used to define Statmenent (5)'s Group-3 axiom in [44] .

Let $\text{UNION}(A)$ denote the union of $\text{IS}_D(A)$'s Group-Zero, Group-1 and Group-2 axioms. It will be useful to employ the following notation:

- i** $\text{Prf}_{\text{UNION}(A)}^D(t, p)$ will denote a formula designating that p is a proof of the theorem t from the axiom system $\text{UNION}(A)$ using the deduction method D .
- ii** $\text{ExPrf}_{\text{UNION}(A)}^D(h, t, p)$ will be a formula stating that p is a proof (using the deduction method D) of a theorem t from the union of the axiom system $\text{UNION}(A)$ with the added axiom sentence specified by the integer h .
- iii** $\text{Subst}(g, h)$ will denote Gödel's classic substitution formula — which yields TRUE when g is an encoding of a formula and h is an encoding of a sentence that replaces all occurrence of free variables in g with an encoded term \overline{g} (that designates g 's Gödel number.)

- iv** $\text{SubstPrf}_{\text{UNION}(A)}^D(g, t, p)$ will denote the natural hybridizations of the constructs from Items (ii) and (iii) which yields a Boolean value of TRUE exactly when there exists some integer h simultaneously satisfying *both* the conditions $\text{Subst}(g, h)$ and $\text{ExPrf}_{\text{UNION}(A)}^D(h, t, p)$.

Each of (i)–(iv) can be encoded as Δ_0^* formulae. Thus, Appendices C and D of [40] explained how the first three of these predicates can receive Δ_0^* encodings when one applies the theory of LinH functions [16, 22, 50]. Hence, (25) illustrates one possible Δ_0^* encoding for $\text{SubstPrf}_{\text{UNION}(A)}^D(g, t, p)$'s graph. (It is equivalent to the statement “ $\exists h [\text{Subst}(g, h) \wedge \text{ExPrf}_{\text{UNION}(A)}^D(h, t, p)]$ ”, but (25) is a Δ_0^* formula — *unlike* the quoted expression.)

$$\text{Prf}_{\text{UNION}(A)}^D(t, p) \vee \exists h \leq p [\text{Subst}(g, h) \wedge \text{ExPrf}_{\text{UNION}(A)}^D(h, t, p)] \quad (25)$$

Let us recall that $\text{Pair}(x, y)$ is a Δ_0^* sentence specifying that x and y are the encodings of a Π_1^* and Σ_1^* sentence, that are logical negations of each other. Using (25)'s Δ_0^* encoding for $\text{SubstPrf}_{\text{UNION}(A)}^D(g, t, p)$, we can now explain how statement (5)'s Group-3 Axiom can be formally encoded. Let $\Gamma(g)$ denote Equation (26)'s formula, n denote $\Gamma(g)$'s Gödel number and \overline{n} denote a term encoding n in the U-Grounding language. Then it will turn out that “ $\Gamma(\overline{n})$ ” will be a Π_1^* sentence that is equivalent to this Group-3 axiom.

$$\forall x \forall y \forall p \forall q \neg [\text{Pair}(x, y) \wedge \text{SubstPrf}_{\text{UNION}(A)}^D(g, x, p) \wedge \text{SubstPrf}_{\text{UNION}(A)}^D(g, y, q)] \quad (26)$$

More precisely, (27) formalizes the encoding of “ $\Gamma(\overline{n})$ ”.

$$\forall x \forall y \forall p \forall q \neg [\text{Pair}(x, y) \wedge \text{SubstPrf}_{\text{UNION}(A)}^D(\overline{n}, x, p) \wedge \text{SubstPrf}_{\text{UNION}(A)}^D(\overline{n}, y, q)] \quad (27)$$

Thus, if we view “ $\text{SubstPrf}_{\text{UNION}(A)}^D(\overline{n}, t, p)$ ” in (27) as our formal method of encoding the concept that was previously informally called “ $\text{Prf IS}_D(A)(t, p)$ ” by Statement (5), then (27) amounts to the formal encoding of (5)'s Group-3 “*I am consistent*” axiom declaration.

Reminder about the Significance of (27)'s Encoding : The preceding construction had showed merely that it is possible to encode Sentence (5)'s Group-3 “*I am consistent*” axiom declaration in a well-defined manner as a Π_1^* sentence. It does not answer the more subtle question about whether or not its “*I am consistent*” axiom declaration holds true under the Standard model. As we have noted before, most analogs of (27) produce false statements under the Standard Model because a conventional Gödel-like diagonalization argument will imply that most deduction methods D will produce axiom systems $\text{IS}_D(A)$ that are inconsistent.

The reason for our particular interest in (27)'s formal encoding is that Theorems 4.2 and 5.2 indicate that $\text{IS}_D(A)$ is consistent when D denotes either the semantic tableaux or Tab-1 deduction methodologies. Thus (27)'s Fixed-Point construction should be seen as a methodology that has limited applications, but which is also quite helpful (when it is feasible).