Draft of February 2019—please do not cite without permission.

The Arithmetization of Syntax and the New Paradoxes of Self-Reference[1]

T. Parent

In Parent (ms.), I argued that if self-reference is unconstrained in a language L, then L is not classical. Specifically, it will include well-formed sentences that are both true and false. That is so, even if L is "semantically open" (using Tarski's 1944 idiom), that is, even if L is free of semantic terms like 'true' and 'denotes' defined on expressions of L.

In conversation, Tim Button worried that this may have dire consequences for Peano Arithmetic (PA), or any other axiomatizations extending Robinson Arithmetic (Q). After all, the method of Gödel numbering allows for something functionally like self-reference. So if unrestricted self-reference enables wff that are both true and false, as the new paradoxes suggest, then Gödel numbering could conceivably be used to demonstrate that the language of arithmetic is non-classical—and in particular, that there are truths in the language which are also false.

I do not believe that the new paradoxes show any such thing. They indeed show that something is awry, but they need not show that the problem lies in the object language. Rather, the problem may well lie in the metalanguage; in particular, it may be that the arithmetization of the object language within the metalanguage enables the paradoxes. If this is correct, then it is not arithmetic itself which is unsound, but rather, any metatheory where Gödel numbering has no restrictions on its use, including Gödel's (1931) metatheory.

---

[1] I owe thanks to friends at Notre Dame for excellent feedback on this draft: Rachael Alvir, Tim Bays, Dan Turetsky, and especially Matteo Bianchetti.

1. Preparatory work

The task shall be to present a variant of a paradox from the earlier paper (the "Laputan" paradox), and here, it is to be formulated exclusively in the language of arithmetic; call it "$\mathcal{L}$." (Since this paper is somewhat more technical, readers may be better served by reviewing the earlier paper first.)

Expressions of language $\mathcal{L}$ shall be understood as follows. First, its terms are defined recursively as follows:

1. $\underline{0}$ is a term.
2. If τ is a term, then so is $\ulcorner \tau' \urcorner$. (Terms introduced by clauses 1 and 2 are the numerals.)
3. If $\underline{n}$ is a numeral, then $\ulcorner v_n \urcorner$ is a term. (Terms introduced by clause 3 are the variables.)
4. If τ_1 and τ_2 are terms, then so are $\ulcorner \tau_1 + \tau_2 \urcorner$ and $\ulcorner \tau_1 \cdot \tau_2 \urcorner$.
5. For any $k > 0$, if $\langle \tau_1,\ldots,\tau_k \rangle$ is any sequence of terms, and $\underline{n}$ and $\underline{m}$ are numerals, then $\ulcorner f^n_m (\tau_1,\ldots,\tau_k) \urcorner$ is a term.
6. Nothing else is a term.

Assume that terms formed by clauses 1, 2, and 4 have their standard interpretation, where $\underline{0}$ is the numeral denoting 0, ' $'$ ' expresses the successor-function, '+' expresses the addition-function, and ' $\cdot$ ' expresses the multiplication-function. Variables from clause 3 have their denotation relative to a variable-assignment—understood as a selection of a sequence, such that the *n*th member is to be the denotation of the *n*th variable. Finally, a saturated function-symbol $\ulcorner f^n_m (\tau_1,\ldots,\tau_k) \urcorner$ denotes *m*, where *m* = the output of the expressed function, given the sequenced denotations of $\langle \tau_1,\ldots,\tau_k \rangle$ as input.

The well-formed formulae (wffs) can then be defined by recursion thusly:

1. If $\langle \tau_1, \tau_2 \rangle$ is any pair whose members are terms, then $\ulcorner =(\tau_1 , \tau_2) \urcorner$ is a wff.

2. For any $k > 0$, if $\langle \tau_1,\ldots,\tau_k \rangle$ is any sequence of terms, and $\underline{n}$ and $\underline{m}$ are numerals, then $\ulcorner F_m^n(\tau_1,\ldots,\tau_k)\urcorner$ is a wff.

3. If Φ is a wff, then so is $\sim\Phi$.

4. If Φ and Ψ are wffs, then so is $\ulcorner \Phi \wedge \Psi \urcorner$.

5. If $\underline{n}$ is a numeral, and Φ is a wff with exactly $\ulcorner v_n \urcorner$ free, then $\ulcorner \forall v_n \Phi \urcorner$ is a wff. (A variable $\ulcorner v_n \urcorner$ is free in Φ iff Φ has $\ulcorner v_n \urcorner$ as a part but not $\ulcorner \forall v_n \urcorner$.)

6. Nothing else is a wff.

Assume that '=', '~', '∧', and quantifier-expressions $\ulcorner \forall v_n \urcorner$ have their standard interpretation, and that an *n*-place predicate has a set of *n*-tuples as its extension. Per usual, a sentence is a wff with no free variables; let us also stipulate that a *designator* is any term which has no variable as a (proper or improper) part. Also, n.b., Arabic numerals are used for subscripts and superscripts, so to reduce clutter; although '1' as a subscript or superscript will often just be omitted.

Let us now define Gödel numbers for expressions of the language. The coding scheme for the basic symbols of $\mathcal{L}$ is as follows (where $n > 0$ and $m > 0$):

Symbol:	0	′	(	)	,	~	∧	∀	=	+	·	$\ulcorner v_n \urcorner$	$\ulcorner F_m^n \urcorner$	$\ulcorner f_m^n \urcorner$
Code:	3	5	7	9	11	13	15	17	19	21	23	$2{\cdot}5^n$	$2^2{\cdot}3^n{\cdot}5^m$	$2^3{\cdot}3^n{\cdot}5^m$

The compound expressions have a unique Gödel number determined in the usual way, exploiting Gaussian prime decomposition: The codes of the *n* basic parts are first assigned (in order) as exponents to the first *n* members of the sequence of primes $\langle 2, 3, 5, 7, \ldots \rangle$. The compound expression's Gödel number is then the product of the exponentiated primes. Thus, the sentence '=(0, 0)' will have a Gödel number equal to $2^{19} \cdot 3^7 \cdot 5^3 \cdot 7^{11} \cdot 11^3 \cdot 13^9$. (Further measures are needed to code proofs as sequences of sentences, but this is unnecessary for the remarks below.)

Another preliminary is necessary. Given a sentence of the form $\ulcorner \forall v_3 \forall v_2 \forall v\ \Phi(v, v_2, v_3) \urcorner$, let C be a "coordinated" substitution instance (or for short, a "CSI") of the sentence iff:

$$C = \ulcorner \forall v\ \Phi(v, \underline{n}, \delta,) \urcorner$$

In this, δ is a designator with Gödel number n whose numeral $\underline{n}$ has replaced 'v_2'. (The numeral in question we shall call the *Gödel-numeral* of δ.)

Thus, a CSI of a sentence $\ulcorner \forall v_3 \forall v_2 \forall v\ \Phi(v, v_2, v_3) \urcorner$ will replace the second variable with the Gödel-numeral for δ, and replace the third variable with δ itself. Since it is decidable whether $\underline{n}$ is the Gödel-numeral for δ, and since this type of sentence is otherwise defined by its form, it is decidable whether a sentence is of this type. (This can be justified via Church's thesis).

2. Paradox with Gödel numbers

Consider now the following sentence of $\mathcal{L}$:[2]

$$(\dagger)\ \ \forall v_3 \forall v_2 \forall v\, (=(f(v), v_2) \supset (\mathrm{F}(v_3) \equiv (=(f(v), 0''') \wedge =(0, v_3))))$$

We have yet to define the predicate 'F(v_3)' and the function-symbol '$f(v)$'—regardless, we can still decide whether a sentence is a CSI of (†). For instance, the following sentence is such a CSI:

$$(1)\ \ \forall v\, (=(f(v), 0''') \supset (\mathrm{F}(0) \equiv (=(f(v), 0''') \wedge =(0, 0))))$$

The reason is that 'v_2' in (†) is replaced with the Gödel-numeral for the designator that replaces 'v_3'. In particular, 'v_2' is replaced by $\underline{0'''}$, and 'v_3' is replaced by $\underline{0}$.

[2] For concision's sake, '⊃' and '≡' are here used as if they are part of the object language, even though they were not mentioned in specifying $\mathcal{L}$. But if preferred, one could revise formulae of the form $\ulcorner\Phi \supset \Psi\urcorner$ to $\ulcorner {\sim}(\Phi \wedge {\sim}\Psi)\urcorner$, and revise formulae of the form $\ulcorner\Phi \equiv \Psi\urcorner$ to $\ulcorner{\sim}(\Phi \wedge {\sim}\Psi) \wedge {\sim}(\Psi \wedge {\sim}\Phi)\urcorner$.

Let us next define '$f(v)$' as denoting the function which maps the CSI of (†) coded by v onto the Gödel code of its final designator. "The final designator" is the designator replacing 'v_3', i.e., the designator replacing the final variable in (†).[3] Thus, where $g(\Phi)$ takes in an expression Φ and outputs its Gödel code (and where $g^{-1}(v)$ is its inverse function), the suggestion is to define '$f(v)$' as follows:

$$f(v) = \begin{cases} n & \text{if } g^{-1}(v) = \text{the CSI of (†) whose final designator is } \delta, \text{ where } g(\delta) = n. \\ \uparrow & \text{otherwise.} \end{cases}$$

Again, one can roughly think of $f(v)$ as mapping the CSI of (†) coded by v onto the Gödel code for the CSI's final designator. (If v does not code such a CSI, then the function is undefined, although the undefined cases will have no bearing on the paradox below.)

Suppose now that the predicate 'F' is defined by (†). Then, (1) is an instance of the definition which makes explicit a condition on which 0 is F. Basically, it says that if we consider the CSI of (†) whose final designator is coded by 3, then 'F' is satisfied by 0 iff the final designator of that CSI is indeed coded by 3 and 0 = 0. Notice, then, that the right-hand side of that biconditional is true. Therefore, it indicates that 0 is F. (N.B., (1) itself is the CSI of (†) which has its final designator coded by 3. So 0 is defined as F with reference to features of (1) itself.)

But the paradox is that we can also show that 0 is not F. Suppose here that $f_2(v) = 0$, for any v (i.e., it is the constantly zero function). And observe that the designator '$f_2(0')$' (i.e., the function-symbol with $\underline{0'}$ as the instantiating constant) is coded by $2^{240} \cdot 3^7 \cdot 5^3 \cdot 7^5 \cdot 11^9$. For short,

[3] Since there are two occurrences of 'v_3' in (†), "the final designator" could be what replaces the second occurrence of 'v_3', or it could be the expression-*type* replacing both occurrences. Either precisification is fine for our purposes.

let us say that this is a number h with numeral h. Then, another CSI of (†) would be the following:

$$(2)\ \forall v\,(=(f(v), \mathrm{h}) \supset (\mathrm{F}(f_2(0')) \equiv (=(f(v), 0''') \wedge =(0, f_2(0')))))$$

This counts as a CSI of (†) given that 'v_2' in (†) is replaced by the Gödel-numeral for the designator that replaces 'v_3'.

Consider, then, (2) also provides a condition on which 0 is F, given that $f_2(0') = 0$. It indicates that, where v codes a CSI of (†) whose final designator is coded by h, 'F' is satisfied by 0 iff that final designator is coded by 3 and $0 = f_2(0')$. Now in the antecedent of (2), the formula '$(=(f(v), \mathrm{h})$' is satisfied when v is the code of (2) itself. After all, (2) is itself the CSI of (†) whose final designator has code h, given that its final designator is '$f_2(0')$'. Yet, contra the final clause of (2), it is false that its final designator is also coded by 3. Thus, (2) reveals that 0 is not F.

And so, the predicate as defined by (†) determines that 'F(0)' is both true and false.

3. Objections and replies

Objection 1: The first objection is that my symbol '$f(v)$' is ill-defined, for its definition refers to instances of (†), and such instances contain the very symbol being defined. Such circularity is thought to be dubious.

It is correct that the function-symbol defined is with reference to the sentence (†), and (†) indeed has the function-symbol as a part. However, when (†) is first identified, the function-symbol is thus far treated as uninterpreted. So it is not as if the function-symbol had to be interpreted before one could interpret the function-symbol. If that were so, that may be a dubious kind of circularity. Rather, the symbol just needs to exist, in order to define the symbol. (This is hardly unusual—one always needs the symbol to exist before one can define the symbol.)

It is a bit odd, however, that the function-symbol is defined with reference to a string that includes the function-symbol itself. Yet that just is a type of self-reference.

Accordingly, if one dislikes the self-reference in how the symbol defined, then this is already to accept the lesson of the paradox. We should indeed forbid certain kinds of self-reference in a classical setting. In relation to $\mathcal{L}$, this apparently means we must restrict Gödel-numbering in some way, since unrestricted Gödel-numbering is what enables defining a self-referential symbol like '$f(v)$'.

This point against self-reference I take to be a significant and novel result. After all, self-reference is ubiquitous in allegedly classical languages, and it has had free reign. One might think, for example, of Henkin's (1949) construction in proving completeness, where the denotation of each constant is reassigned to denote the constant itself. Or one might think of a language where 'x is a wff' is defined on sentences containing that very predicate. In this case, just like with '$f(v)$', the symbol is defined (in part) with reference to that very symbol.

Nothing here shows that a classical language cannot indulge in *any* self-reference. Nor, more specifically, has it been shown that a classical language cannot talk about which strings are wffs. Rather, the point is just that a classical language must handle self-reference carefully, in ways that have not been acknowledged.

<u>Objection 2</u>: The second objection is whether (†) is a legitimate means to define the predicate 'F(v_3)'. The issue concerns the fact that 'v' and 'v_2' are universally quantified, suggesting that the truth-conditions for 'F(0)' are less straightforward than what is indicated by (1) alone.

The best way to illustrate the concern is to suppose first (contrary to fact) that <u>0</u> is the only designator in $\mathcal{L}$ for 0—thus, compound designators for 0 like '$f_2(0')$' are assumed not to exist. (Imagine, if you like, that we are dealing with a fragment of $\mathcal{L}$.) Regardless, the truth-condition of 'F(0)' would not be determined by (1) alone. Consider, after all, the following instance of (†):

$$(3)\quad \forall v\,(=(f(v), \text{h}) \supset (\text{F}(0) \equiv (=(f(v), 0''') \wedge =(0, 0))))$$

Note well that (3) is *not* a CSI of (†). After all, $\underline{\text{h}}$ replaces 'v_2', and $\underline{\text{h}}$ is not the Gödel-numeral of the final designator in (3). Yet since (†) is fully general, (3) still gives us a truth-condition for 'F(0)'. Thus, (1) is not the only formula that determines the truth-value for 'F(0)'.

But a sentence may well have its truth-value determined by multiple formulae. Thus, for a given set A, the truth-value of 'A is an ordered pair' is determined by several definitions, including:

(Kuratowski) A is an ordered pair iff, for some a and b, $A = \{\{a\}, \{a, b\}\}$.

(K-reverse) A is an ordered pair iff, for some a and b, $A = \{\{b\}, \{a, b\}\}$.

There is no problem here, since the different conditions are equivalent.

Yet perhaps the problem with 'F(0)' is precisely that the different formulae determining its truth-value are non-equivalent. Indeed, notice that the embedded conjunction in (3) appears to be false. After all, when $f(v) = \text{h}$, then it cannot also be that $f(v) = 3$. Specifically, *in the case of (2)*, if the code for that CSI is assigned to 'v', then the antecedent of (3) will be true, but its embedded conjunction will be false. This reveals that (3) gives the opposite verdict of (1). Namely, (3) determines that 'F(0)' is false.

But in the end, this just seems to be an independent demonstration for how the predicate 'F(v_3)' is paradoxical.[4] And my claim all along has been that 'F(0)' is both true and false.

---

[4] The argument just given is, in fact, entirely parallel to Jay Newhard's argument for one of the new paradoxes of self-reference. See sections 3 and 5 of Parent (ms.) for more on Newhard's argument.

Furthermore, I *agree* that this means we must somehow exclude such a predicate from a classical language. However, no provisions against such a predicate have ever been given. It may be possible to correct for that, but the lesson here is that it needs correcting.

4. Closing remarks

The suggestion that Gödel's (1931) metatheory contains paradoxical sentences can cause dramatic reactions. But the most revolutionary conclusions do not follow. Thus, it does not follow that the Gödel-sentence specifically is both true and false. Nor does it follow that Gödel's reasoning about the Gödel-sentence is unsound. Mathematics may well remain incomplete, and for roughly the reasons that Gödel offered.[5] Besides, there is at least one alternate proof of incompleteness due to Kripke, reported by Putnam (2000).[6] Regardless, restrictions on Gödel-numbering seem necessary in relation to the language of mathematics, even though this has hitherto been unacknowledged.

---

[5] Granted, a paradox in Gödel's metatheory would mean that some kind of adjustment is needed, and this may confound his proof as originally formulated. On the other hand, it may not: Some restrictions on his metatheory would be harmless, e.g., if we insisted that the code for '(' and ')' had to be '9' and '7' (respectively) instead of '7' and '9'. But the question has not been settled here, and I hope to investigate it in future research.

[6] Notably, Kripke's proof turns on a number-theoretic statement which, according to Putnam, "is not at all 'self-referring'" (p. 55). Nonetheless, it utilizes Gödel-numbering without restrictions, and so may still be vulnerable to an objection based on the considerations offered here. This is also something I hope to explore in future work.

References

Gödel, K. (1931). Über formal unentscheidbare Sätze der *Principia Mathematica* und verwandter Systeme, I. *Monatshefte für Mathematik und Physik* 38: 173–98.

Henkin, L. (1949). The completeness of the first-order functional calculus. *Journal of Symbolic Logic* 14: 159–66.

Parent, T. (ms.). Paradox with just self-reference. Available at http://tparent.net/d.pdf.

Putnam, H. (2000). Nonstandard models and Kripke's proof of Gödel's theorem. *Notre Dame Journal of Formal Logic* 41(1): 53–58.

Tarski, A. (1944). The semantic conception of truth and the foundations of semantics. *Philosophy and Phenomenological Research* 4(3): 341–76.