Draft of September 2022—please do not cite without permission.

A New Curry Paradox

T. Parent (Nazarbayev University)
nontology@gmail.com

1. *Introduction*

One version of Curry's (1942) paradox is as follows. Define an "informal proof" of *p* as a series of English sentences that express a valid argument to the conclusion that *p*. Then, the observation is that we can informally prove the following sentence:

(C) If (C) is informally provable, then 0=1.

The informal provability of (C), along with (C), implies that 0=1. (Such is a version of the "validity Curry" or v-Curry paradox; for details, see Whittle 2004, Shapiro 2011, and Beall & Murzi 2013.) Classical logicians have responded by restricting semantic terms like 'informal proof' so that it is not defined on sentences of its own language. Thus, (C) is ruled as a non-well-formed formula, which thwarts any argument for its truth or falsity.

Nonetheless, it is shown below that a variant on the paradox still results in formal arithmetic, by adapting an argumentative technique from Kripke (forthcoming).[1] Since the technique seems legitimate, this new Curry paradox could conceivably have drastic consequences.[2] However, the aim here is only to present the paradox; I leave it to others to decide what the best response is.

[1]All references to Kripke will be to his (forthcoming).

[2] E.g., the new paradox could be used as evidence against Church's thesis. However, Church's Thesis must not be given up easily: It has tremendous utility in the field and is confirmed by the striking convergence between Turing computable functions, lambda-computable functions, and recursive functions. (See Church 1936, Turing 1937, and Shepherdson & Sturgis 1963. A good summary of these results is found in chapters 12 and 13 of Kleene 1952.)

2. *An Informal Argument*

It is worth approaching the issue informally first, so to better grasp the situation within formal arithmetic. The informal version still employs an unrestricted semantic notion of an "informal proof;" hence, it is not entirely akin to the later argument (which will use only a formal notion of proof). But since the reasoning is similar in some key respects, it will be useful to consider the informal argument as preparatory.[3]

Consider, then, the following statement:

(Mu) Haskell is an informal proof that (Mu) entails '0=1'.[4]

The name 'Haskell' in (Mu) has a denotation, but for the moment, we shall hold off on specifying what it is. For it is important that, regardless of what 'Haskell' denotes, we can give an informal proof that (Mu) entails '0=1'.

Thus, assume provisionally that (Mu) is true. Then, given what (Mu) says, it follows that Haskell is an informal proof that (Mu) entails '0=1'. But the existence of such a proof means that 0=1 follows from our provisional assumption. So given the truth of (Mu), the truth of '0=1' follows, as desired.

Note that this argument does not make any assumptions about what 'Haskell' denotes. It does not even assume that it denotes a proof. (Mu) *claims* that Haskell is a certain type of informal proof, but the truth of (Mu) was not assumed either (except provisionally) when arguing that (Mu) entails '0=1'.

---

[3] A similar informal proof version of the Curry paradox is also discussed in Priest (1979; 2006). But unlike in Priest, the aim here is not to suggest the informal proof demonstrates something significant about formal arithmetic. Rather, the informal proof is merely preparatory for a version which focused exclusively on *formal* proof. And it is only the later version that is claimed to show something important for metamathematics.

[4] A version of the pathology which is more clearly related to (C), above, would be:

(Mu*) If Haskell* is an informal proof of (Mu*), then 0=1.

However, (Mu) is a bit simpler and parallels more closely the version in formal arithmetic, presented below.

However, we may now reveal that 'Haskell' names the forgoing informal proof that (Mu) entails '0=1'. Then, Haskell is an informal proof that (Mu) entails '0=1'—which is precisely what (Mu) claims. So (Mu) is true, even though (Mu) was also shown false. Contradiction.

Again, the classical logician is well-served here by restricting semantic notions like "informal proof" and "entailment." But it turns out that such notions are not necessary to re-create the Mu-paradox, as we shall now illustrate.

3. *Exposition of P_1*

The following remarks shall concern a restricted version of Gödel's system P which is exclusively first-order.[5] Call this system P_1. The terms are defined inductively as follows:

(i) '0' is a term.

(ii) If τ is a term, then so is τ'. ('0' followed by zero or more occurrences of ' ′ ' are the numerals. Generally, let $\underline{n}$ be the numeral for n.) [6]

(iii) For any n, $v_{\underline{n}}$ is a term.

(iv) If $0<m<4$ and $\tau_1, \ldots, \tau_m$ are terms, then $f^{m}_{\underline{n}}\tau_1\ldots\tau_m$ is a term. (We need consider only 1- to 4-ary functions in what follows; this will make the coding of the language easier.)

(v) Nothing else is a term.

---

[5] All references to Gödel will be to his (1931).

[6] In what follows, I often elide the distinction between use and mention. Quine's (1951) corner quotes could be used to distinguish an expression ⌜τ′⌝ from what it represents. (Some other notation for Gödel numbers would then be required.) But to avoid clutter, I instead rely on context to disambiguate.

For convenience, however, Arabic numerals are used and I often revert to using 'x', 'y', etc., as variables. Also, a term $f^{m}_{\underline{n}}\tau_1 \ldots \tau_m$ is usually written as $f_{\underline{n}}(\tau_1, \ldots, \tau_m)$.

The well-formed formulae (wff) are defined thusly:

(vi) If τ_1 and τ_2 are terms, then $\tau_1 = \tau_2$ is a wff.

(vii) If ϕ is a wff, then so is $\sim\phi$.

(viii) If ϕ and ψ are wff, then so is $(\phi \supset \psi)$.

(ix) If $v_{\underline{n}}$ is a variable and ϕ is a wff, then $\forall v_{\underline{n}} (\phi)$ is a wff.

(x) Nothing else is a wff.

Assume that '$\sim$', '$\supset$', and '$\forall$' have their standard interpretations. As an added convenience, wff with '&', '$\vee$', or '$\equiv$' will replace the usual equivalents that use only '$\sim$' and '$\supset$'; relatedly, $\exists x (\phi)$ will be used as shorthand for $\sim\forall x (\sim\phi)$. Further, I will omit the parentheses when there is no danger of confusion.

The system has the following logical axioms:

(L1) $\phi \supset (\psi \supset \phi)$

(L2) $(\phi \supset (\psi \supset \rho)) \supset ((\phi \supset \psi) \supset (\phi \supset \rho))$

(L3) $(\sim\psi \supset \sim\phi) \supset (\phi \supset \psi)$

Also, where $\phi[v_{\underline{n}}/\tau]$ is the result of replacing $v_{\underline{n}}$ in ϕ with τ:

(L4) $\forall v_{\underline{n}} \phi \supset \phi[v_{\underline{n}}/\tau]$ – if no variable in τ is bound at a place where $v_{\underline{n}}$ was free.

(L5) $(\forall v_{\underline{n}} (\sim\phi \supset \psi) \supset (\sim\phi \supset \forall v_{\underline{n}} \psi))$ – if $v_{\underline{n}}$ is not free in ϕ.

Each of (L1)–(L5) is in fact an axiom scheme in which ϕ, ψ, ρ are arbitrary wff. As such, the schemes generate infinitely many axioms of their respective forms.

We shall assume the system also has the standard axiomatic analysis of '=', as per the Law of Universal Identity and the Indiscernability of Identicals:

(U$_=$) $x=x$

(I$_=$) $\tau_1=\tau_2 \supset (\phi[x/\tau_1] \supset \phi[x/\tau_2])$

We also include the following three axioms from Q (Robinson's 1950 arithmetic):

(Q1) $\sim 0=x'$

(Q2) $(x'=y' \supset x=y)$

(Q3) $(\sim x=0 \supset \exists y\,(x=y'))$

Further arithmetical axioms (also from Q) are used to define function symbols that express addition and multiplication. For ease of exposition later, suppose that P_1 uses 'f_{46}' and 'f_{47}' to express these functions, respectively (although for readability, I shall normally use '+' and '·' in place of these). The defining axioms are then:

f_{46}: $+(x, 0)=x$

$+(x, y')=+(x, y)'$

f_{47}: $\cdot(x, 0)=0$

$\cdot(x, y')=+(\cdot(x, y), x)$

As for rules of inference, P_1 features modus ponens and a version of universal introduction:

(MP) From $(\phi \supset \psi)$ and ϕ, ψ is derivable.

(∀I) From ϕ, $\forall x\,(\phi)$ is derivable.

To allow for brevity later, we also assume that P_1 has the following rule of "Curry conditional introduction:"

(CCI) From $\sim\phi$, $(\phi \supset 0=1)$ is derivable.

Later, I refer to $(\phi \supset 0=1)$ as *the Curry conditional for* ϕ.

A finite sequence Σ of wff counts as a *derivation* of ϕ in P_1 from a (possibly empty) set of wff Γ iff: The first members of Σ are the members of Γ (if any), the last member of Σ is ϕ, and any member of Σ is either a member of Γ, an axiom, or is derivable in Σ via some inference rule in the system. When Γ is empty, we say that the derivation is a *proof* of ϕ in P_1. N.B., the system is sound and complete with respect to first-order validities.

4. *Constructing* $\boldsymbol{P_1'}$*: Two Constants*

We shall now work our way toward two conservative extensions of P_1; call them P_1' and P_1''. Starting with P_1', we begin by adding two constants 'b' and 'c' to the language. To interpret these, we shall refer to the following two-membered sequence of wff, dubbed *the Curry sequence*. (Note that 'f_{45}' is yet to be interpreted.)

1. $\sim f_{45}(\mathrm{b}, \mathrm{c})=0$
2. $f_{45}(\mathrm{b}, \mathrm{c})=0 \supset 0=1$

N.B., in the system P_1', the Curry sequence will not qualify as a proof, since the first member is not an axiom. Regardless, suppose that the sequence has code i_1. Then, we can let the constant 'b' be interpreted by the following axiom in P_1':

(A1) $\mathrm{b}=\underline{i_1}$

Whereas, if the second wff in the Curry sequence has code i_2, then the constant 'c' shall be interpreted by the following:

(A2) $\mathrm{c}=\underline{i_2}$

These axioms secure that the Curry sequence contains a constant denoting the code of that very sequence, and another constant denoting the code of its second member. The constants thus exemplify what Kripke calls "direct" self-reference.

Thus far, P_1' is a conservative extension of P_1 in the model-theoretic sense. Kripke explains as follows, where S and S' can be read as P_1 and P_1', respectively:[7]

> Pretty clearly S' is a conservative extension of S. It simply extends S by adding…[new] constants with specific numerical values—new names for particular numbers. Every proof in S' becomes a proof in S if each constant is replaced by the corresponding numeral. (pp. 3-4)

However, we shall conservatively extend the system further. One minor extension is that we include the following as an additional axiom:

(A3) ~+(b, c)=0

Assume, moreover, that this is not coded by its standard Gödel number but rather by 21. Although the wff is obviously derivable in P_1' independently of (A3), including it as an axiom in P_1' will facilitate the introduction of P_1'' later.

5. *Constructing* $\mathbf{P_1'}$*: P.R. Functions*

The next order of business is to formulate an axiom which defines '$f_{45}(x, y)=0$' as a *primitive recursive proof predicate for* P_1'—whereby ~$f_{45}(\underline{n}, \underline{m})=0$ is true in the standard model exactly when n codes a proof in P_1' of the formula coded by m. Some readers may be willing to grant that such a definitional axiom for 'f_{45}' can be given—in which case, they may skip the intricacies of this section and move on to the next. For the remaining readers, the plan is to formulate such an axiom by adapting Gödel's construction of his proof predicate $B(x, y)$ (p. 186, #45 in Gödel's list of p.r. functions and relations).

[7] Kripke does not specify his system S in sufficient detail to say whether it is akin to P_1. But it is natural to assume he means his remarks to apply to such cases.

Most significantly, we will need to modify the definition of Gödel's "axiom predicate" Ax(x) so that it refers to the codes of axioms for P_1' rather than for P. To this end, we will disjoin to its definition three formulae denoting the codes of (A1), (A2), and (A3)—but crucially, we will also need to have the axiom for Ax(x) denote the code of *the axiom for Ax(x) itself.*

This reveals a contrast between Gödel's system P and the present system P_1'. Gödel's definitions of Ax(x), $B(x, y)$, etc., were offered not as *axioms* of P; rather, he used the definitions to show that certain relations were p.r. and thus strongly represented among the *theorems* of P.[8] However, while it is important that such relations are p.r., our purposes also demand that *the specific predicate occurring in the Curry sequence* should be a proof predicate. We thus require the right sort of definitional axiom for 'f_{45}' as part of P_1'.

We will make the axiom for Ax(x) "indirectly" refer to itself by exploiting Gödel's p.r. function #31, viz., the "substitution function." Roughly, this is the ternary p.r. function outputting the code of a formula which results after uniformly replacing the variable at position n (in some formula ψ) with some expression α. However, if '*sub*' is used in formulating our axioms, then we will need to include an axiom defining '*sub*' as well. And that in turn requires as axioms several other Gödelean definitions, which I shall now present. (Caveat: I skip certain details when they require more effort than seems warranted. The omitted details may always be recovered from pp. 182-186 of Gödel.)

---

[8] This is not always made clear in discussions Gödel; e.g. van Heijenoort (1967, p. 303) describes system P as a version of primitive recursive arithmetic (cf. Skolem 1923, Hilbert & Bernays 1934, Curry 1941). In PRA, each function symbol has an axiom that defines it as expressing one of the p.r. functions. But if Gödel had counted his definitions of p.r. relations as an *axioms* of P, then at least one definition should have made reference *to its own code* (in order for his proof predicate to recognize it as an axiom). Gödel's versions of the definitions do no such thing—yet for his purposes, they don't need to. *Our* purposes will require this however. So in this respect, P_1' *can* be seen as a truncated version of PRA. Hedge: Unlike PRA, P_1' contains unbound quantifiers. Though in fact, they are unnecessary for the arguments below; indeed, it is crucial that unbound quantifiers do *not* occur in our definitions for the p.r. functions.

Some preliminaries: I first assume that 'f_0' is axiomatically defined to express exponentiation along the following lines:

f_0: $\hat{}(x, 0)=1$

$\hat{}(x, y')=\cdot(\hat{}(x, y), x)$

Even so, a term of the form $\hat{}(\tau_1, \tau_2)$ is henceforth written as $\tau_1^{\tau_2}$. Also, as further notational conveniences:

- $\tau_1+\tau_2$ and $\tau_1\cdot\tau_2$, replace $+(\tau_1, \tau_2,)$ and $\cdot(\tau_1, \tau_2)$, respectively;
- $\tau_1<\tau_2$ is shorthand for $(\tau_1=0) \vee (\tau_1=1) \vee (\tau_1=2) \vee \ldots \vee (\tau_1= \underline{\tau_2-1})$;
- $\tau_1\leq\tau_2$ is shorthand for $(\tau_1<\tau_2) \vee (\tau_1=\tau_2)$;
- $\forall x\leq\tau\ (\phi)$ and $\exists x\leq\tau(\phi)$ are shorthand for $\forall x(x\leq\tau \supset \phi)$ and $\exists x(x\leq\tau\ \&\ \phi)$, respectively. Relatedly, $\forall x,y\leq\tau\ (\phi)$ and $\exists x,y\leq\tau\ (\phi)$ are shorthand for $\forall x\leq\tau(\phi)\ \&\ \forall y\leq\tau(\phi)$ and $\exists x\leq\tau(\phi)\ \&\ \exists y\leq\tau(\phi)$, respectively—and in like manner for three or more variables.

Below, I also write some definitions as if they concerned predicate expressions, in accord with the following notational rule:

(NR) A definition of the form:

$$F_n(\tau_1, \ldots, \tau_k) \equiv \phi$$

…is shorthand for:

$$(\phi \supset f_n(\tau_1, \ldots, \tau_k)=0)\ \&\ (\sim\phi \supset f_n(\tau_1, \ldots, \tau_k)=1)$$

For concreteness, let us also commit to a specific Gödel-numbering scheme. The basic symbols shall be coded as follows:

‘0’…1 ‘ ′ ’… 3 ‘∼’… 5 ‘⊃’… 7 ‘∀’… 9 ‘(’… 11 ‘)’… 13

‘=’…15 ‘b’… 17 ‘c’… 19 $v_{\underline{n}}$ … $2{\cdot}5^n$ $f^m_{\underline{n}}$ …$2^2{\cdot}3^n{\cdot}5^m$

In addition, if $\langle n_1, \ldots, n_k\rangle$ is a sequence of numbers, we code the sequence as $2^{n_1}{\cdot}3^{n_2}{\cdot}5^{n_3}{\cdot}\ldots{\cdot}p^{n_k}$, where p is the kth prime in ascending order. Thus, if a wff is composed of n basic symbols $s_1, \ldots, s_n$ (in that order), then where $\ulcorner s_k \urcorner$ is the code of the kth basic symbol, the wff is also associated with sequence $\langle \ulcorner s_1 \urcorner, \ldots, \ulcorner s_k \urcorner \rangle$. Let the code of that sequence function as the code for the wff itself.

Definitions #1–4 are exactly as in Gödel, though again, such definitions stand as axioms in P_1' for the symbols ‘f_1’–‘f_4’. Similarly, definitions #5–10 are unchanged in the essentials; these are given below in intuitive terms for the reader’s immediate reference.

f_5: $npr(n)$ = the nth prime number (in ascending order).

f_6: $mem(n, x)$ = the n^{th} member in the sequence coded by x.

f_7: $length(x)$ = the number of members in the sequence coded by x.

f_8: $x{*}y$ = the code of a sequence of numbers resulting from “concatenating” the sequence coded by x with the sequence coded by y (in that order).

f_9: $seq(x) = 2^x$

$seq(x)$ codes the sequence that consists only of the number x (for $x>0$)

f_{10}: $paren(x)=seq(11){*}x{*}seq(13)$

$paren(x)$ corresponds to the operation of “parethesizing” (11 and 13 are codes for the basic signs ‘(‘ and ‘)’).

The next two Gödelian definitions have been revised into something considerably simpler, owing to the fact that P_1' has first-order variables only:

f_{11}: $\mathrm{V}(n, x) \equiv x{=}2{\cdot}5^{n}$

x codes the nth variable.

f_{12}: $\mathrm{Var}(x) \equiv \exists n{\leq}x(\mathrm{V}(n, x))$

x codes a variable.

Definitions #13–17 are lifted directly from Gödel; however, they are also good to have in hand:

f_{13}: $neg(x){=}seq(5)*x$

$neg(x)$ codes the negation of the wff coded by x.

f_{14}: $imp(x, y){=}paren(x*seq(7)*y)$

$imp(x, y)$ codes the implication of the wff coded by y, from the wff coded by x.

f_{15}: $gen(x, y){=}seq(9)*seq(x)*paren(y)$

$gen(x, y)$ codes the generalization of the wff coded by y, by the variable coded by x.

f_{16}: $succ(x, 0){=}x$

$succ(x, n{+}1){=}\ succ(x, n)*seq(3)$

$succ(x, n)$ codes the symbol(s) coded by x appended by ‘ ′ ’ n times.

f_{17}: $num(n){=}succ(seq(1), n)$

$num(n)$ codes the numeral denoting n.

The next few definitions show significant adjustments from Gödel’s, due to P_1' being first-order and having terms for 1- to 4-ary functions:

f_{18}: $\mathrm{TSeq}(0, x) \equiv \exists m{\leq}x(x{=}seq(1) \vee x{=}seq(17) \vee x{=}seq(19) \vee x{=}seq(2{\cdot}5^{m}))$

TSeq(n+1, x) ≡ $\forall i<x \forall k<length(x)[mem(k, x)=i \supset$ (TSeq(0, i) ∨

$\exists w,v,u,t,m<x \exists p,q,r,s<k\ [mem(p, x)=w\ \&\ mem(q, x)=v\ \&\ mem(r, x)=u$

& $mem(s, x)=t$ & TSeq(n, w) & TSeq(n, v) & TSeq(n, u) & TSeq(n, t) &

$i=w \vee i=w*seq(3) \vee i=seq(2^2 \cdot 3^1 \cdot 5^m)*w \vee i=seq(2^2 \cdot 3^2 \cdot 5^m)*w*v \vee$

$i=seq(2^2 \cdot 3^3 \cdot 5^m)*w*v*u \vee i=seq(2^2 \cdot 3^4 \cdot 5^m)*w*v*u*t])]$

x codes what is known in the literature as a "term formation sequence" or "term construction sequence."

f_{19}: Term(0, x) ≡ $\exists m \leq x(x=1 \vee x=17 \vee x=19 \vee$ Var(x))

Term(n+1, x) ≡ $\exists y<x$ (TSeq(n+1, y) & mem(length(y), y)=x)

x codes a term composed from n function symbols.

f_{20}: Efm(x) ≡ $\exists y,z,n,\leq x$ (TSeq(n, y) & TSeq(n, z) & $x=y*seq(15)*z$)

x codes an elementary formula. ('15' is the code for '=')

Definitions #21–31 are as in Gödel; these stand as axioms in P_1' which define 'f_{20}' – 'f_{31}', respectively. For later reference, we shall gloss four of these as follows:

f_{24}: Bound(v, n, x) ≡ The variable coded by v is bound at position n in the wff coded by x.

f_{25}: Free(v, n, x) ≡ The variable coded by v is free at position n in x.

f_{26}: Fr(v, x) ≡ The variable coded by v is free in x.

f_{31}: $sub(x, v, y)$=the code of the wff that results after uniformly replacing the variable coded by v in the formula coded by x with the expression coded by y.

The four definitions in Gödel's entry #32 are the same as well; but here, these can be seen as defining 'f_{32}', 'f_{33}', 'f_{40}', and 'f_{41}', respectively.[9] (Gödel's definition #33 will be omitted as inappropriate for a first-order system; similarly with definitions #40 and #41.)

The next few definitions have been adjusted given P_1' uses the Q-axioms instead of Peano's axioms, and given that our logical axioms are slightly different:

f_{34}: Qax(x) ≡ (x=$\underline{q_1}$ ∨ x=$\underline{q_2}$ ∨ x=$\underline{q_3}$)

x codes an arithmetical axiom of Q (q_n codes the nth Q-axiom). The codes for other Q-axioms are covered in the definition of Ax(x), below.

f_{35}: Lax1(x) ≡ ∃y,z≤x(Form(y) & Form(z) & x=*imp*(y, *imp*(z, y))

x is an axiom that has been obtained by inserting into the axiom scheme (L1). We define Lax2(x) and Lax3(x) analogously.

f_{36}: PropAx(x) ≡ Lax1(x) ∨ Lax2(x) ∨ Lax(x)

x codes an axiom that results from substituting into (L1), (L2), or (L3).

Due to the notational differences in P_1', definitions #37 – 39 are adjusted as follows:

f_{37}: Q(y, v, t) ≡ ∀n≤*length*(y)[(TSeq(t) & Free(v, n, y)) ⊃

∀m≤*length*(t) ∀w≤*sub*(y, v, t)[(Var(w) & *mem*(m, t)=w) ⊃

Free(w, n+m–1, *sub*(y, v, t))]]

If v codes a free variable in the wff coded by y, then all variables are free in term corresponding to t, within the wff coded by *sub*(y, v, t). (This is a requirement on the acceptability of an axiom generated by (L4).)

---

[9] At #32, Gödel defines *imp* whereas we defined this earlier at #14. In Gödel, #14 instead defines *dis*, the function that outputs the codes for disjunctive formulae. Above, I treat 'f_{32}' as expressing *dis*.

f_{38}: Lax4(x) ≡ ∃v,y,z≤x [Var(v)& Form(y) & Term(t) & Q(y, v, t) &

x=*imp*(*gen*(v, y), *sub*(y, v, t))

x codes an axiom obtained by substitution from the scheme (L4).

f_{39}: Lax5(x) ≡ ∃v,q,p≤x [Var(v) & Form(p) & ~Fr(v, p) & Form(q) &

x=*imp*(*gen*(v, *imp*(*neg*(p), q)), *imp*(*neg*(p), *gen*(v, q)))

x codes an axiom obtained by substitution on axiom scheme (L5).

[As earlier noted, definitions #40 and #41 are unneeded.]

Definition #42 is where we shall use the function *sub* to achieve a kind of self-referential axiom. First, consider first the following (non-axiomatic) formula wherein ‘v’ is free, and where ⌜$\underline{v}$⌝ is the numeral for the Gödel number of the variable ‘v’.

Ax(x) ≡ Qax(x) ∨ PropAx(x) ∨ Lax4(x) ∨ Lax5(x) ∨ $x=\underline{k_1}$ ∨ $x=\underline{k_2}$ ∨ x=21 ∨

$x=\underline{h_0}$ ∨ $x=\underline{h_1}$ ∨ $x=\underline{h_2}$ ∨ … ∨ $x=\underline{h_{41}}$ ∨ x=*sub*(v, ⌜$\underline{v}$⌝, *num*(v)) ∨

$x=\underline{h_{43}}$ ∨… ∨ $x=\underline{h_{47}}$

In this, k_1, k_2, and 21 are the codes for (A1), (A2), and (A3), respectively—whereas h_0, …, h_{41}, h_{43}, …, h_{47} are the codes for the axioms defining ‘f_0’… ‘f_{41}’, ‘f_{43}’, …, ‘f_{47}’, respectively. Suppose now that the above v-free formula has code j. Then, after replacing the two free occurrences of ‘v’ with j, let P_1' contain as an axiom the wff that results:

f_{42}: Ax(x) ≡ Qax(x) ∨ PropAx(x) ∨ Lax4(x) ∨ Lax5(x) ∨ $x=\underline{k_1}$ ∨ $x=\underline{k_2}$ ∨ n=21 ∨

$x=\underline{h_0}$ ∨ $x=\underline{h_1}$ ∨ $x=\underline{h_2}$ ∨ … ∨ $x=\underline{h_{41}}$ ∨ x=*sub*(j, ⌜$\underline{v}$⌝, *num*(j)) ∨

$x=\underline{h_{43}}$ ∨… ∨ $x=\underline{h_{47}}$

It can be verified that *sub*(j, ⌜$\underline{v}$⌝, *num*(j)) denotes the code for that very axiom. So Ax(x) is satisfied iff x is the code of any axiom in P_1', including the axiom which defines Ax(x) itself.

Next, we need a slight adjustment to Gödel's definition #43, concerning the "immediate consequence" relation, since P_1' includes (CCI) as an inference rule:

f_{43}: $\quad \mathrm{IC}(x, y, z) \equiv y{=}imp(z, x) \vee \exists v{\leq}x\,(\mathrm{Var}(v)\ \&\ z{=}gen(v, y)) \vee$

$(x{=}imp(z, \ulcorner \underline{0{=}1} \urcorner\ \&\ \exists n{\leq}x\ n{=}neg(z))$

x codes a wff that is derivable from the wff coded by y or z, by one application of a P_1'-inference rule.

The last two definitions are as in Gödel:[10]

f_{44}: $\quad Bw(x) \equiv \forall 0{<}n{\leq}length(x)[Ax(mem(n, x)) \vee$

$\exists 0{<}p,q{<}n\ IC(mem(n, x), mem(p, x), mem(q, x))]\ \&\ 0{<}length(x)$

x codes a proof array for P_1'.

f_{45}: $\quad B(x, y) \equiv Bw(x)\ \& mem(length(x), x){=}y$

x codes a proof in P_1' of y.

[Gödel's definition #46 is omitted as it is unnecessary for our purposes.]

Despite all these details, it should be reasonably clear that P_1' remains a conservative extension of P_1. If a theorem of P_1' uses an expression of the form $f_{\underline{n}}(\tau_1, \ldots, \tau_m)$, the above definitions can be used to find an equivalent theorem in which $f_{\underline{n}}(\tau_1, \ldots, \tau_m)$ does not appear. This reveals that any theorem of P_1' is equivalent to some theorem of P_1, which vindicates that P_1' is model-theoretically conservative vis-à-vis P_1.

[10] $\forall 0{<}n{<}\tau\ (\phi)$ and $\exists 0{<}p,q{<}n(\phi)$ are shorthand, respectively, for $\forall n((0{<}n\ \&\ n{<}\tau) \supset \phi)$ and $\exists p(0{<}p\ \&\ p{<}n\ \&\ \phi)\ \&\ \exists q(0{<}q\ \&\ q{<}n\ \&\ \phi)$.

6. *From* $\boldsymbol{P}_1'$ *to* $\boldsymbol{P}_1''$

Having described P_1', we can now prove the following metatheorem before moving on to P_1''. Where 'B(b, c)' stands in for 'f_{45}(b, c)=0', the claim is:

(MT) ~B(b, c) is a theorem of P_1'.

Proof: Assume for *reductio* that B(b, c) is a theorem of P_1'. Then, since B is a proof predicate, the formula coded by c has a proof in P_1'. Yet the formula coded by c is B(b, c) ⊃ 0=1. And if that formula is a theorem along with B(b, c), then 0=1 is a theorem and the system is unsound. So by *reductio*, B(b, c) is not a theorem. However, since $B(x, y)$ expresses a p.r. relation, either B(b, c) or its negation is a theorem; after all, every p.r. formula is decidable by any consistent system containing the Q-axioms. So ~B(b, c) is a theorem, as desired.

Note that in the argument just given, it was not presumed that b really does code a proof of B(b, c) ⊃ 0=1. That might beg the question on whether (MT) holds. Of course, B(b, c) "says" that b codes a proof of B(b, c) ⊃ 0=1, but B(b, c) was not assumed either (except provisionally) in the preceding argument.

Since (MT) is true, this means we can add ~B(b, c) as an axiom and get a system that proves no more, no less than P_1'. Suppose we do this and remove the redundant axiom (A3). The result is the system P_1''. Clearly, P_1'' is a conservative extension of P_1', and hence, of P_1.

Also, let 21 code the new axiom ~B(b, c) rather than the wff at (A3). (The latter can now be coded by its standard Gödel number.) Crucially, this allows 'B' to function as a proof predicate *for the system* P_1''. That's because the only difference between P_1' and P_1'' is that the axiom (A3) has been replaced with ~B(b, c). At the same time, ~B(b, c) receives the code that (A3) had previously. So despite the difference between P_1' and P_1'', exactly the same numbers remain codes for axioms. Because of this, the proof predicate in P_1' becomes, within P_1'', the proof predicate for P_1''.

At the same time, it is crucial that *B* expresses precisely the same first-order, arithmetic relation in P_1' versus P_1''. For *B* has the same definition in the two systems. Thus, ~*B*(b, c) still expresses an arithmetic truth in P_1'', given that it is a theorem of P_1'. The difference in coding is only a difference in the metatheory for P_1' versus P_1''.

By the way, rather than switching which wff is coded by 21, the same effect could be achieved by instead switching the standard codes of '~*B*(b, c)' and the wff at (A3)(which, recall, are officially the wff '~f_{45}(b, c)=0' and '~f_{46}(b, c)=0', respectively). That is, we could swap their standard Gödel numbers in the metatheory for P_1'—and then put them back to normal when working in the metatheory for P_1''. That approach may have more appeal in some respects. But I didn't want to trigger the objection that this somehow has our metatheory for P_1' treating '~*B*(b, c)' as an axiom of P_1'. (The objection would misfire, but I didn't want to trigger it regardless.)

Such maneuvers with Gödel numbering may be unfamiliar, though Kripke also experiments with non-standard Gödel numberings. And the basic idea here should be uncontroversial enough: We can let a number like 21 code whatever expression we like, as long as it uniquely reserved as the code for that expression. Also, replacing (A3) with some other arithmetical truth as an axiom should not render the system unsound. Moreover, we saw that ~*B*(b, c) *is* an arithmetical truth, as per (MT).

7. *The Central Argument*

Recall now that the Curry sequence is:

1. $\sim B$(b, c)
2. B(b, c) $\supset$ 0=1

Consider that the Curry sequence qualifies as a proof of P_1''. The second member follows from the first by (CCI), and the first is an axiom of P_1''. Therefore, where i_1 is the code of the Curry sequence, and i_2 is the code of B(b, c) $\supset$ 0=1, we know that $B(\underline{i_1}, \underline{i_2})$ is a theorem of P_1'', given that B is a p.r. proof predicate for P_1''. Consider, moreover, that b=$\underline{i_1}$ and c=$\underline{i_2}$ are axioms of P_1''. So it follows that B(b, c) is a theorem. But again, $\sim B$(b, c) is an axiom. Thus, by present assumptions, P_1'' is inconsistent. And since P_1'' conservatively extends P_1, it follows that P_1 is inconsistent.[11]

Stepping back, where b= i_1, the sentence B(b, c) effectively means "i_1 codes a proof of my Curry conditional." Since i_1 indeed codes such a proof, the sentence in question is true. But at the same time, such a proof suffices for its falsity. So B(b, c) is inconsistent, and this is captured within P_1'' via the arithmetization of the proof relation.

8. *Some Remarks*

Kripke's technique of introducing constants makes our argument possible. Consider that on a standard Gödel numbering, it is not possible for a proof to contain its own Gödel numeral. (The code for $\underline{i_1}$ is greater than i_1—and the code for a formula containing $\underline{i_1}$ is even larger, whereas the code for a proof with such a formula is larger still. In which case, the

[11] Really, P_1'' is just a truncated version of primitive recursive arithmetic (PRA); cf. note 8. So the argument equally can be used to suggest that PRA is inconsistent. (One could go further and contend that P_1'' is merely an extension of Q. However, Q only has two function symbols '+' and '·'. Since the expressive resources are much greater in P_1'', I refrain from insisting that P_1'' is a version of Q.)

code for a proof containing $\underline{i_1}$ could not be equal to i_1.[12]) However, Kripke's technique allows a term for i_1 other than $\underline{i_1}$ to occur in the proof coded by i_1. Kripke's own application of the technique was just to create wff which could refer to their own codes. But the technique seems equally applicable to create a *proof* with a term for its own code.[13] And again, the technique seems quite legitimate; as Kripke explained in the block quote in section 4, the constants are just additional names for specific numbers to supplement the numerals.

One could make our argument a bit shorter by defining the Curry sequence as consisting of only ~*B*(b, c) and reinterpreting the constants so that its sole member effectively says "b codes a disproof of this very sentence." But I included Curry conditionals to emphasize a kinship with the v-Curry; like the v-Curry, the present paradox is enabled by a notion of "proof." Though unlike the v-Curry, it is paramount that ours is enabled by a formal rather than a semantic notion of proof. At the same time, a comparison with Tarski's (1933) version of the Liar paradox is apt. After all, "b codes a disproof of this very sentence" is strongly analogous to a Liar-sentence. The difference of course is that the notion of proof is used in lieu of truth. Yet since proof and truth coincide with p.r. formulae, it makes sense that a Liar-like paradox might result when considering proofs of such formulae.

9. *In Search of a Solution*

Unlike Tarski, however, we cannot solve our paradox by concluding that the proof predicate is not part of the object language. It is reasonable to suggest this of the truth

[12] I am grateful to Panu Raatikainen for discussion here.

[13] Instead of using the constants for direct self-reference to a wff, we could instead appeal to the Fixed-Point Lemma to create the usual sort of indirect self-reference. So it is only the direct self-reference to a *proof*, via the constant 'b', which seems essential to creating the paradox.

predicate since that predicate does not express a recursive operation. Yet *B*(*x*, *y*) indeed expresses a recursive operation. Tarski's way seems unavailable to us.

As a different tack, an intuitionist may try to reject *B*(b, c) ∨ ~*B*(b, c), and propose that *B*(b, c) is undecidable. But to repeat, *B*(b, c) is an instance of a p.r. formula; hence, its undecidability in P′′ seems untenable. Besides, this view would be unavailable to classical logicians, for it is premised on the claim that *B*(b, c) violates bivalence.[14]

I myself am left with little idea on what the proper resolution of the paradox should be. My hope is that others might make progress on the matter.[15]

Works Cited

Beall, J.C. (2009). *Spandrels of Truth*. Oxford: Oxford University Press.
Beall, J.C. & J. Murzi. (2013). 'Two Flavors of Curry's Paradox,' *Journal of Philosophy* 110: 143–165.
Brower, L. E. J. (1912). 'Intuïtionisme en Formalisme,' *Inaugural address at the University of Amsterdam*. Translated by A. Dresden as 'Intuition and Formalism,' in P. Benacerraf & H. Putnam (eds.), *Philosophy of Mathematics: Selected Readings*, 2nd edition. Cambridge: Cambridge University Press, 1983, pp. 77-89.
Church, A. (1937). 'An Unsolvable Problem of Elementary Number Theory,' *American Journal of Mathematics* 58: 345–363.
Curry, H. (1941). 'A Formalization of Recursive Arithmetic,' *American Journal of Mathematics* 63: 263-282.
Curry, H. (1942). 'The Inconsistency of Certain Formal Logics,' *Journal of Symbolic Logic* 7: 115–117.
Gödel, K. (1931). 'Über Formal Unentscheidbare Sätze der *Principia Mathematica* und Verwandter Systeme I,' *Monatshefte für Mathematik Physik* 38: 173–198. Pagination is from Gödel, K. (1986). *Collected Works I. Publications 1929–1936*. S. Feferman et al. (eds.), Oxford: Oxford University Press, pp. 144–195.
Heijenoort, J. van (ed.). (1967). *From Frege to Gödel : A Source Book in Mathematical Logic, 1879–1931*. Cambridge, MA: Harvard University Press.

[14] Consider also that a dialetheist response seems unhelpful; cf. Priest (2006), Beall (2009). The dialetheist might accept and deny *B*(b, c) itself—yet this would leave untouched the argument showing that a contradiction is derivable inside P_1''.

[15] My thanks to Bill Gasarch, Bill Mitchell, Panu Raatikainen, Lionel Shapiro, Henry Towsner, Nic Tideman, Bruno Whittle, Noson Yanofsky, and Richard Zach for discussion of issues relevant to this paper. I also thank an audience at the 2022 meeting of the Australasian Association of Philosophy.

Heyting, A. (1930). 'Die formalen Regeln der intuitionistischen Logik,' *Sitzungsberichte der Preussischen Akademie von Wissenschaften. Physikalisch-mathematische Klasse*, pp. 42–56.

Hilbert, D. & Bernays, P. (1934). *Grundlagen der Mathematik*, *Vol. I*, Berlin: Springer.

Kleene, S. (1952). *Introduction to Metamathematics*. Amsterdam: North-Holland Publishing.

Kripke, S. (forthcoming.) 'Gödel's Theorem and Direct Self-Reference,' *Review of Symbolic Logic*.

Priest, G. (1979). 'The Logic of Paradox,' *Journal of Philosophical Logic* 8: 219–241

Priest, G. (2006). *In Contradiction: A Study of the Transconsistent*, 2nd edition. Oxford: Oxford University Press.

Quine, W. (1951). *Mathematical Logic*, revised edition. Cambridge, MA: Harvard University Press.

Robinson, R.M. (1950). 'An Essentially Undecidable Axiom System,' *Proceedings of the International Congress of Mathematics*: 729-730.

Shapiro, L. (2011). 'Deflating Logical Consequence,' *Philosophical Quarterly* 61: 320–342.

Shepherdson, J. & H.E. Sturgis. (1963). 'Computability of Recursive Functions,' *Journal of the Association of Computing Machinery* 10: 217–255.

Skolem, T. (1923). 'Begründung Der Elementaren Arithmetik Durch Die Rekurrierende Denkweise Ohne Anwendung Scheinbarer Veranderlichen Mit Unendlichem Ausdehnungsbereich', *Videnskapsselskapets Skrifter, I. Matematisk-Naturvidenskabelig Klasse*, 6: 1–38. Translated by S. Bauer-Mengelberg as 'The Foundations of Elementary Arithmetic Established by the Recursive Mode of Thought, without the Use of Apparent Variables Ranging over Infinite Domains.' In Heijenoort, J. van (ed.). (1967), pp. 302-333.

Tarski, A. (1933). 'Pojęcie prawdy w językach nauk dedukcyjnych,' *Prace Towarzystwa Naukowego Warszawskiego, Wydzial III Nauk Matematyczno-Fizycznych* 34, Warsaw. Expanded version translated by J.H. Woodger as 'The Concept of Truth in Formalized Languages,' in his *Logic, Semantics, Metamathematics*: *Papers from 1923 to 1938*, 2nd edition. J. Corcoran (ed.), Indianapolis: Hackett, pp. 152–278.

Turing, A. (1937). 'Computability and λ-Definability,' *Journal of Symbolic Logic*. 2: 153–163.

Whittle, B. (2004). 'Dialetheism, Logical Consequence and Hierarchy,' *Analysis* 64: 318–326.