Intensionality from Self-Reference
T. Parent (Nazarbayev University)
nontology@gmail.com

1. Introduction

Yablo argues that paradox can be generated without self-referential expressions. But he denies that “self-reference suffices for paradox” (1993: 251). This is obvious if he means that not all cases of self-reference are paradoxical (e.g., ‘This sentence is a sentence.’). Yet since he justifies the claim by citing Tarski and Gödel, Yablo likely meant something stronger. An alternate reading is: Self-reference does not suffice for any pathological sentence, at least in a semantically open language – that is to say, in a language without semantic terms (‘true’, ‘refers, ‘satisfies’, etc.) which are defined on expressions of that self-same language.\(^1\) This is a standard view; however, in what follows it is shown to be strictly incorrect. Yet we shall also see that, due to the nature of the case, we are not forced into any radical revisions of classical logic.

Caveat: It is mostly left implicit in the literature that self-reference is unconstrained in a semantically open language. As far as I know, Gödel and Tarski never affirm this overtly; Yablo’s remark is the closest I can find. More often, self-referential devices seem to enjoy a “presumption of innocence,” as suggested by Tarski’s and Gödel’s work. (Consider also

\(^1\) The term ‘semantically open language’ is from Tarski (1944). Tarski restricted himself to such languages so to bypass Liar-like paradoxes, yet this is no longer popular due to alternate approaches by Kripke (1972) and Gupta & Belnap (1993) (although Kripke’s is patently nonclassical given that it rejects the law of excluded middle). Still, it is widely assumed that if a language is semantically open, unrestricted self-reference alone is insufficient for contradiction. That is what I question in this paper.
Henkin’s 1949 completeness proof for first-order logic.) At any rate, I expect that working logicians will recognize the presumption of innocence as orthodox; however, I shall suggest that it requires qualification.

2. An Argument for Absurdity

It is well known that substitution of co-referring terms within quotation marks is invalid. Suppose that the names or constants of the object language are ‘a₀’, ‘a₁’, ‘a₂’,… And suppose that ‘a₀’ and ‘a₁’ are interpreted as co-referring, whence it is true that:

\[ a₀ = a₁ \]

Then, given the truism that ‘a₁’ = ‘a₁’, substitution inside quotation would allow us to deduce the absurdity that: ‘a₀’ = ‘a₁’. So the maneuver is not permitted in classical logic.

What amounts to much the same, a substitutable variable inside quotation marks is sufficient to express an ill-defined function.² Thus, consider the expression:

‘x’

This would normally mention the symbol used as a variable, viz., the (italicized) 24th letter of the alphabet. But suppose instead that the variable inside the quotes is substitutable, so that it is functionally equivalent to a “blank” that can be filled in by any term. Then, the expression expresses an ill-defined function \( q(x) = ‘x’ \), one which yields different outputs for \( a₀ \) depending on which term for \( a₀ \) replaces the variable:

\[ q(a₀) = ‘a₀’ \]
\[ q(a₁) = ‘a₁’ \]

---

² This connection between substitution into quotes and an ill-defined function was observed already by, e.g., Tarski (1933/1983: 161).
To preclude this, a quoted variable is always regarded as a name rather than a functional expression.\(^3\)

Notwithstanding, my claim is that an analogous ill-defined function can be re-created by exploiting the possibilities for self-reference, even in a semantically open language. Suppose here that we stipulate the notion of a \textit{reflection} as follows. Where ‘\(n\)’ is a variable for a natural number with numeral \(n\):

\[ (*) \text{ The reflection of } x = y \text{ iff there is an } n \text{ such that } x = a_n \text{ and } y = \left\langle a_n \right\rangle. \]

Notation: Per Quine (1951), an expression with corner quotes denotes the concatenation of symbols enclosed in the corners, after the replacement of any metavariables. Thus, as concerns \(\left\langle a_n \right\rangle\), if ‘\(n\)’ is replaced with ‘0’, the corner-quoted expression then denotes the concatenation of the symbol ‘a’, followed by a subscripted ‘0’.\(^4\) I.e., it denotes ‘\(a_0\)’.

The definition at (*) institutes a type of self-reference – for when ‘\(x\)’ is replaced with a constant, (*) defines ‘the reflection of \(x\)’ to denote whichever constant is appended to that very descriptor. That is, when ‘\(x\)’ is replaced with a constant, the descriptor effectively means “The constant hereby concatenated with ‘the reflection of \(x\)’.”\(^5\) If self-reference is unrestricted in a semantically open language, such a descriptor seems thus far admissible.

\(^3\) This should not suggest a problem with the function that maps an expression onto its quotation. If \(\alpha\) is an expression, \(f(\alpha) = \left\langle \alpha \right\rangle\) remains perfectly well defined; consider that \(f(\langle a_0 \rangle) = \langle a_0 \rangle\) and \(f(\langle a_1 \rangle) = \langle a_1 \rangle\). (The outputs here are different of course, but so are the inputs.)

\(^4\) It may be desirable here and with similar sentences to use the notation of Boolos (1995), so to disambiguate which quotation marks are paired together.

\(^5\) When ‘\(x\)’ is replaced with a saturated function symbol, however, ‘the reflection of \(x\)’ does not denote the very term appended to the descriptor. It rather denotes a co-referring constant. But such cases will not be relevant here.
However, (*) introduces a function that is ill-defined on input \( a_0 \) in exactly the same way as \( q(x) \). After all, (*) implies that:

1. The reflection of \( a_0 = 'a_0' \);
2. The reflection of \( a_1 = 'a_1' \).

So like \( q(x) \), the reflection “function” varies between outputting \( 'a_0' \) and \( 'a_1' \) on input \( a_0 \), depending on which name for the input used. Concurrently, whereas the definition at (*) rules that (1) is true, (1) is provably equivalent to the following absurdity:

3. The reflection of \( a_1 = 'a_0' \).

The absurdity of (3) is even clearer in noting that, by the transitivity of identity, (2) and (3) imply that \( 'a_0' = 'a_1' \). (N.B., the substitution needed to show the equivalence between (1) and (3) is not a substitution inside quotation marks.)

The upshot is that since ill-defined are to be excluded, the language must somehow be regulated to prevent the introduction of such a thing. So contra the usual attitude, self-reference cannot be unrestricted in a semantically open language.

3. Discussion

Further elaboration is in order. First, note that the underline in (*) expressing the numeral function is not equivalent to quotation; witness that \( 0 = 0+0 \) whereas \( '0' \neq '0+0' \). However, \( 'n' \) may appear to be a semantic expression, for \( n \) is the numeral that denotes the number \( n \). If so, then the language may not be semantically open. However, the pathology ultimately depends on neither \( 'n' \), nor the underline, nor quotes, nor corner quotes. It is enough to have a (well-defined) function \( h \) such that:
\[ h(0) = \text{the first lowercase letter of the alphabet with subscript nought;} \]
\[ h(1) = \text{the first lowercase letter of the alphabet with the successor numeral to nought as its subscript.} \]

On all other inputs, assume \( h \) is undefined. We then can make do with a partial definition of a reflection:

\[ (***) \text{The reflection of } x = y \text{ if there is an } n \text{ such that } x = a_n \text{ and } y = h(n). \]

Assuming \( a_0 = a_1 \), the absurdity at (3) is derivable in a similar fashion.\(^6\)

It is natural to think that the problem owes to (*) and (**) quantifying into the subscript position. But while this seems to be sufficient for the problem, I do not believe it is necessary (see the Appendix). Besides, classical logicians otherwise have need of quantification into subscript position. E.g., when introducing infinitely many constants into the language, strictly speaking one must invoke clauses such as:

- For all \( n \), \( [a_n] \) is a constant.
- For all \( n \), \( [a_n] \) denotes \( a_n \).

Quantification into subscript position occurs elsewhere too, as in some presentations of the anti-diagonal function in Cantorian arguments. (This is not to suggest a problem with such presentations; the point instead is that our argument requires only standard formal devices.)

It can be said that the reflection of \( x \) is an intensional notion, for the matter depends on what term for \( x \) is being used. This is noteworthy insofar as it is a new kind of intensionality; it does not owe to a propositional attitude verb, an idiom like ‘so called’, attribute abstraction, a modal operator, or (to repeat) substitution inside quotes.\(^7\) More important, however, is that the

---

\(^6\) My thanks to Alexander Pruss for raising the concern about ‘\( n \)’ and suggesting a function like \( h \) in response.

\(^7\) On such cases of intensionality, see Quine (1960, sections 30ff.).
definition at (***) clearly uses only formal, nonsemantic terms. This reveals that we can prove an absurdity in a semantically open language simply by exploiting self-reference. Or from another angle, unchecked self-reference enables intensional contexts, even if the language lacks the usual intensional elements.

One may object that the preceding shows merely that definitions like (*) are not legitimate. But that is exactly the point: Such definitions must be excluded. Still, the definition is problematic simply because ‘The reflection of x’ fails to express a well-defined function, and ill-defined functions are already verboten. Agreed – this is why the argument does not have any drastic revisionary implications for classical logic. At the same time, special attention is called for when commonplace formal devices give rise to an ill-defined function. This is how one might regard the ill-defined ‘quoting function’ from section 2. Quotation seems unremarkable at first glance; it is just a convenient way to introduce metalinguistic terms. But substitution inside quotation makes for an ill-defined function, and so, a word of warning is required.

Similarly, self-reference has been “presumed innocent” by the working logician. But unrestricted self-reference suffices for an ill-defined function, even in a semantically open language – and so, an explicit advisory is apt. Put differently, the ban on ill-defined functions implies certain restrictions on self-reference; even though this seem not to be acknowledged.

4. A Pathology with Gödel Numbers

In conversation, Tim Button has worried that if unrestricted self-reference suffices for absurdity, then Peano Arithmetic might be shown unsound via the method of Gödel numbering. For Gödel numbering enables something functionally like self-reference, and unrestricted self-reference now appears sufficient for inconsistency.
A derivative argument is indeed readily available using Gödel numbers, as I shall explain. But the argument shows only that Gödel numbering in the metalanguage cannot be used without restriction. It hardly implies that the arithmetical object language is defective. (Nothing in Peano Arithmetic functions like self-reference prior to Gödel numbering.) So again, the general prohibition on ill-defined functions means that there is no revolutionary import here. Even so, it is worth remarking that the ban on ill-defined functions implies certain restraints on self-referential mechanisms, including Gödel numbering.

To illustrate: Suppose that \( g(\varepsilon) \) is a function that takes in an expression \( \varepsilon \) and outputs its Gödel number. Then, define the function symbol ‘\( r(x) \)’ as follows:

\[
\begin{align*}
  r(x) &= y \quad \text{if there is an } n \text{ such that } x = f_n(0) \text{ and } y = g(\lceil f_n(0) \rceil); \\
  &\uparrow \quad \text{otherwise.}
\end{align*}
\]

This defines ‘\( r(x) \)’ in such a way that if, for some \( n \), \( x = \text{the } 0^{\text{th}} \text{ output of the } n^{\text{th}} \text{ function} \), then \( r(x) = \text{the Gödel number for the concatenation of the } n^{\text{th}} \text{ function symbol with ‘0’} \). So in particular, if ‘\( x \)’ is replaced with an expression of the form \( \lceil f_n(0) \rceil \), \( r(x) \) then outputs the Gödel number for that self-same concatenation. In such a case, it outputs the Gödel number for the very sequence of symbols that replaces ‘\( x \)’. For example:

1. \( r(f_0(0)) = g('f_0(0)') \) \quad [By definition of ‘\( r(x) \)’]
2. \( r(f_1(0)) = g('f_1(0)') \) \quad [By definition of ‘\( r(x) \)’]

And since each expression has its own Gödel number, we know that:

\[8\] Since any recursive function is numeralwise expressible in Peano Arithmetic, it may seem that if contradiction is provable in the metalanguage, then contradiction is provable in the arithmetical object language. Yet the characteristic function for the set of reflection pairs (as defined here) is not a well-defined function, much less a recursive one. So again, formal arithmetic is in no immediate danger.
3. \( g('f_0(0)') \neq g('f_1(0)') \) \hspace{1cm} \text{[By definition of ‘} g(x) \text{’]}

Suppose now that ‘\( f_0(x) \)’ and ‘\( f_1(x) \)’ have been earlier defined in such a way that:

4. \( f_0(0) = f_1(0) \) \hspace{1cm} \text{[By definition of ‘} f_0(x) \text{’ and ‘} f_1(x) \text{’]}

Then, given the truism that \( r(f_0(0)) = r(f_0(0)) \), the indiscernability of identicals assures us:

5. \( r(f_0(0)) = r(f_1(0)) \) \hspace{1cm} \text{[From 4]}

However, the earlier observations entail:

6. \( r(f_0(0)) \neq r(f_1(0)) \) \hspace{1cm} \text{[From 1–3]}

Yet lines 5 and 6 contradict.

Basically, ‘\( r(x) \)’ expresses a well-defined function only if no two function expressions of the form \( f_n(0) \) co-refer. Hence, ‘\( r(x) \)’ does not express a well-defined function, for our language can be assumed to contain such co-referring expressions. The above definition of ‘\( r(x) \)’ must therefore be precluded. One might have thought, however, that it is acceptable to define a functional expression that denotes whichever Gödel number you please. But as our definition of ‘\( r(x) \)’ shows, if ‘\( x \)’ is replaced by any functional expression of the form \( f_n(0) \), ‘\( r(x) \)’ cannot then be rigged to denote the Gödel number for that very functional expression. Otherwise, for some \( m \neq n \), \( f_m(0) \) will co-refer with \( f_n(0) \), yet \( r(f_m(0)) \) will denote a different Gödel number than \( r(f_n(0)) \).

\[9\text{E.g., 4 is true when ‘} f_0(x) \text{’ expresses the constantly 0 function and ‘} f_1(x) \text{’ expresses the identity function. N.B., a similar argument can be made in relation to a functional expression ‘} \ast \text{’, defined as } \ast(x) = y \text{ if there is an } n \text{ such that } f(n) = x \text{ and } y = g(f(n)) \text{, where } f \text{ is any non-injective function. This is an argument which does not depend on the use of subscripts. However, } \ast \text{ is patently ill-defined, since for some } x, \ast(x) \text{ is defined partly by what “the” input } n \text{ is to a non-injective function } f \text{ when } f(n) = x. \text{ In contrast, the argument above involving } r \text{ does not require } f_n \text{ to be non-injective.}\]
5. Closing Remark

Thus, even Gödel numbering cannot be used without restriction. Again, one can assume that existing formal systems are free of the intensional cases. Even so, it is worth acknowledging how such self-referential devices require some restraint, especially since the matter is glossed over by the usual “presumption of innocence.”

Appendix

This appendix presents a different way of exploiting self-reference in a semantically open language to define an intensional function. Notably, this avenue does not require the use of numerals (much less numeric subscripts). Instead, it requires an approach to defining functions that is not universally employed, although it is standard in Tarskian semantics.

The relevant approach is one where a function is defined not directly but rather in virtue of defining a function symbol in the metalanguage. Thus, in a Tarskian semantics, the function symbols are defined among the base clauses, which are then used in connection with recursive clauses to interpret the wffs of the object language. For example, the Tarskian base clauses might include the following, where α is any metalinguistic term:

\[ f(\alpha) \] denotes ‘α’

Given any expression, \( f \) outputs the quotation of the expression; e.g., \( f('a_0') = 'a_0' \), \( f('a_1') = 'a_1' \), etc. (This is a well-defined function; it is not to be confused with the intensional ‘quoting function’ from section 2.) Such a clause in the metalanguage introduces a term in the object language. On some occasions, however, one might just simply introduce the function using ‘=’ in the object language directly, rather than the using the metalanguage ‘denotes’. Specifically, the
object language might contain an axiom such as the following, where ‘x’ is to be replaced by any metalinguistic term:

\[ f(x) = \text{the quotation of } x. \]

Having noted that, however, we assume the Tarskian “metalevel” approach.

The point, then, is that unrestricted self-reference allows the following expression to be defined. Where \( \tau \) is any singular term (metalinguistic or not):

\[ \text{‘} r(\tau) \text{’} \]

denotes \( \tau \)

Given a singular term \( \tau \), the concatenation of ‘\( r(\ ) \)’ with \( \tau \) is defined to denote \( \tau \). If self-reference were given free reign, such a definition would be thus far permissible. But suppose that \( a = b \).

Then, the definition implies:

\[ r(a) = \text{‘} a \text{’} \]

\[ r(b) = \text{‘} b \text{’} \]

Even though the input is the same, the output differs depending on which term for the input is used. Unrestrained self-reference is again sufficient by itself for an intensional function, and in this case, it is made possible without the use of numerals.

Acknowledgments

I thank Bradley Armour-Garb, John G. Bennett, Matteo Bianchetti, Ethan E. Brauer, Tim Button, David Chalmers, Morgan Davies, Anil Gupta, Thomas Hofweber, Cory Johnson, Yael Loewenstein, William Lycan, David Mccarty, Jay Newhard, Bryan Pickel, Alexander Pruss, Michael Resnik, David Ripley, and Keith Simmons for discussion on various versions of this material. I also thank an audience at the 2018 Central Division meeting of the American Philosophical Association.
References


