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 create inconsistency (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* inconsistency, 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 to radically revise our views of classical logic.[2]

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 nonclassical given that it does not uphold all instances of $p \vee \sim p$). Still, it is widely assumed that *if* a language is semantically open, unrestricted self-reference alone is insufficient for absurdity. That is what I question in this paper.

[2] The term 'classical logic' is a bit of a misnomer. As stated in the preface to the first edition of Priest (2008), it was not really the logic that was prominent in the classical era of Greece and Rome. It is thus perhaps better to call it "standard logic," as the logic that is typically taught in introductory symbolic logic courses. Yet since it has become a convention to refer to it as "classical logic" nonetheless, I shall follow this practice.

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. This is because at least one type of self-referential device is sufficient for an *intentional context*.

2. An Argument for Absurdity

It is well known that substitution of co-referring terms within quotation marks is invalid. Suppose that 'a$_0$' and 'a$_1$' are co-referring, whence it is true that:

$$a_0 = a_1$$

Then, given the truism that 'a$_0$' = 'a$_0$', substitution inside quotation would allow us to deduce the absurdity that: 'a$_0$' = 'a$_1$'. Such a maneuver is therefore not permitted.

What is much the same, a substitutable variable inside quotation marks expresses an ill-defined function.[3] 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 $i(x) = 'x'$, one which yields different outputs for a$_0$ depending on which term for a$_0$ replaces the variable:

$$i(a_0) = 'a_0'$$

$$i(a_1) = 'a_1'$$

---

[3] 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, nothing more.[4]

Notwithstanding, an analogous ill-defined function can be re-created by exploiting the possibilities for self-reference, even in a semantically open language. Suppose we define a symbol '*' so that when it precedes a constant, it denotes the very constant that it precedes. If self-reference could be given free rein, such a thing ought to be permissible and could be defined precisely as follows:

(D1) $*a_n = \ulcorner a_{\underline{n}} \urcorner$

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 $\ulcorner a_{\underline{n}} \urcorner$, if '$\underline{n}$' is replaced with '0', the corner-quoted expression then denotes the concatenation of the symbol 'a', followed by a subscripted '0'.[5] I.e., it denotes 'a_0'.

Observe, then, that (D1) introduces an expression that is ill-defined on a_0 in exactly the same way as $i(x)$. After all, (D1) implies that:

(1) $*a_0 = {}$ 'a_0'

(2) $*a_1 = {}$ 'a_1'

So like i from before, the *-operation varies between outputting 'a_0' and 'a_1' on input a_0, depending on which name for the input used.

Concurrently, such an operation must be excluded from a classical formal system, assuming such a system includes a rule that permits co-referring terms to be intersubstitutable. After all, such a rule would allow us to derive from (1) and (2):

---

[4] This should not suggest a problem with the function that maps an expression onto its quotation. If α is an expression, $q(\alpha) = \ulcorner `\alpha' \urcorner$ remains perfectly well defined; consider that $q(`a_0') = {}$ ''a_0'' and $q(`a_1') = {}$ ''a_1''. (The outputs here are different of course, but so are the inputs.)

[5] 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.

(3) $*a_1 = $ 'a_0'

Yet (3) is false. This is even clearer in noting that, by the transitivity of identity, (2) and (3)

imply that 'a_0' = 'a_1'. (And note that the substitution needed to derive (3) is not a substitution

inside quotation marks.)

At any rate, since ill-defined functions are to be excluded, a classical system must not

allow such a thing as (D1). But contra the usual attitude, this means that some kinds of self-

referential expressions must be absent in classical logic, even if the language is a semantically

open.[6]

3. Discussion

Further elaboration is in order. First, note that the underline in (D1) expressing the

numeral function is not equivalent to quotation; witness that $\underline{0} = \underline{0^2}$ whereas '0' $\neq$ '0^2'. However,

'$\underline{n}$' may appear to be a semantic expression, for $\underline{n}$ is the numeral that *denotes* the number n. If so,

then the language may not be semantically open. Yet the issue ultimately depends on neither '$\underline{n}$',

nor the underline, nor quotes, nor corner quotes. It is enough to have a (well-defined) function f

such that $f(n)$ outputs the nth constant – that is, $f(0) = $ 'a_0', $f(1) = $ 'a_1', and so on.[7] Then, define

the symbol '*' by the following:

(D2) $*a_n = f(n)$

Given that $a_0 = a_1$, the absurdity at (3) would be derivable in a similar fashion.[8]

---

[6] It might be asked what exactly counts as a "self-referential expression." I cannot resolve the matter here, but I assume that an expression like '$*a_0$' is one species of the kind. For it denotes the very constant that forms part of the expression.

[7] One could also avoid quotation marks here by instead using descriptions of the constants, such as 'the first letter of the Latin alphabet subscripted by the first Arabic numeral.'

[8] My thanks to Alexander Pruss for raising the concern about '$\underline{n}$' and suggesting a function like f in response.

It might be thought that the problem owes to (D1) or (D2) using a variable in 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 variables in subscript position. E.g., when introducing infinitely many constants into the language, strictly speaking one must invoke clauses such as:

(i) For all n, $\ulcorner a_{\underline{n}} \urcorner$ is a constant.

(ii) For all n, $\ulcorner a_{\underline{n}} \urcorner$ denotes a_n.

Variables in subscript position occur 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 '*' expresses an intensional notion, for its denotation depends on what term for its input is being used. This is noteworthy insofar as it evidences 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.[9] More important, however, is that the definition at (D2) clearly uses only formal, nonsemantic terms. This suggests that we can prove an absurdity in a semantically open language simply by exploiting self-reference. Or from another angle, unchecked self-reference apparently enables intensional contexts, even if the language lacks the usual intensional elements.

One may object that the preceding shows merely that definitions like (D2) are not legitimate. But that is exactly the point: Such definitions must be excluded. Still, the definition is problematic simply because '*' 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

---

[9] On such cases of intensionality, see Quine (1960, sections 30ff.).

implications. 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 expression '*i*' 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. Intensionality 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 an intensional context.

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 with abandon. 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.

The reasoning at issue is the following. Suppose that $g(\varepsilon)$ is a function that takes in an expression ε and outputs its Gödel number. Then, define the symbol '$' as follows:

(D3) $\$a_n = g(\ulcorner a_{\underline{n}} \urcorner)$

This defines '$' in such a way that it denotes the Gödel number for the nth constant, i.e., the code for the very constant that follows '$'. Thus:

1. $\$a_0 = g(`a_0\text{'})$ [By definition of '$']

2. $\$a_1 = g(`a_1\text{'})$ [By definition of '$']

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

3. $g(`a_0\text{'}) \neq g(`a_1\text{'})$ [By definition of 'g']

But consider again that:

4. $a_0 = a_1$ [By definition of the constants]

Then, if '$' expressed a legitimate operation, it would follow:

5. $\$a_0 = \$a_1$

However, the earlier observations entail:

6. $\$a_0 \neq \$a_1$ [From 1–3]

Yet lines 5 and 6 contradict.

Thus, '$' does not express a well-defined operation, for it does not permit the substitution of co-referring constants. The above definition of '$' is therefore precluded in classical logic. One might have thought, however, that it is acceptable to define an expression that denotes whichever Gödel number you please. But as our definition shows, '$' cannot be rigged to denote the Gödel number for the very constant occurring after '$'. Otherwise, for some $n \neq m$, it can turn out that $a_n = a_m$, even though $\$a_{\underline{n}} \neq \$a_{\underline{m}}$.

The suggestion that Gödel numbering enables us to infer an absurdity can provoke strong reactions—but again, let me reassure the reader that the foregoing has no radical implications. Granted, since any recursive function is numeralwise expressible in Peano Arithmetic, it may seem that that if contradiction is provable using a Gödel numbering function, then contradiction is provable in the arithmetical object language. Yet the Gödel numbering operation at (D3) *is not a well-defined function*, much less a recursive one. So again, formal arithmetic is in no danger.

It remains true that self-reference—and the coding analog of self-reference—cannot be completely unrestrained. But this hardly means that every case of such a thing leads to pathology. (An analogy: The intensional expression '*i*' from section 2 does not show all use of quotation to be illegitimate.) The point is just that *if* Gödel codes are abused in a certain way, the proviso against ill-defined functions will be violated.

5. Closing Remark

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

Appendix

In conversation, Gabe Dupre emphasizes that self-reference is not needed for the kind of intensionality at issue. After all, instead of (D1) or (D2), the following could be used to define '*' and no self-reference would thereby be implicated:

(D4) $*a_n = n+57$

But while this is correct, the point remains that the self-referential operation defined at (D1) or (D2) is also sufficient for the pathology. This, again, is contrary to the usual presumption of innocence.

 Indeed, the intensionality also results from self-reference absent any variables in subscript position, as I shall now illustrate. But for this, we shall require an approach to definitions that is not always employed, although it is standard in Tarskian semantics.

The relevant approach is one where a functional expression is defined not by using '=' in the object language, as at (D1) – (D3) but rather by using the denotation-relation in the metalanguage, akin to how an infinitely many number of constants were introduced at (ii). In Tarskian semantics, the function symbols are defined among the metalanguage base clauses; these are then used in connection with recursive clauses to determine the truth-conditions of wff in the object language. Thus, the Tarskian base clauses might include the following, where α is any metalinguistic term:

$\ulcorner q(\alpha) \urcorner$ denotes $\ulcorner `\alpha' \urcorner$

Given any expression, this outputs the quotation of the expression. E.g., if 'a', 'b', 'c'… 't' are constants, then $q(`a') = ``a''$, $q(`b') = ``b''$, etc. (The function q is a well-defined function; it is not the intensional operation i from section 2.)

The key observation, then, is that unrestricted self-reference would allow the following to be introduced into the object language. Where τ is any singular term (metalinguistic or not):

(D5)　⌜*τ⌝ denotes τ

Given a singular term τ, the concatenation of '*', followed by τ, is interpreted to denote τ itself. Again, if self-reference were unconstrained, such a definition would be thus far permissible. But suppose that a = b. Then, (D5) implies:

*a = 'a'

*b = 'b'

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 expression, and in this case, it is made possible without a variable in subscript position.

References

Boolos, G. (1995). Quotational ambiguity. (In P. Leonardi and M. Santambrogio (Eds.), *On Quine*, (pp. 283-296). Cambridge: Cambridge University Press.)

Gupta, A. & Belnap, N. (1993). *The Revision Theory of Truth*. (Cambridge, MA: MIT Press.)

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

Kripke, S. (1972). Outline of a theory of truth. *Journal of Philosophy*, 72, 690-716.

Priest, G. (2008). *An Introduction to Nonclassical Logic: From Ifs to Is*, 2nd edition. (Cambridge: Cambridge University Press.)

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

____. (1960). *Word and Object*. (Cambridge, MA: MIT Press.)

Tarski, A. (1933/1983). Pojęcie prawdy w językach nauk dedukcyjnych. Towarszystwo Naukowe, Warsaw. Reprinted as: The concept of truth in formalized languages. (In J. H. Woodger & J. Corcoran (Eds.), *Logic, Semantics, and Metamathematics: Papers from 1923-1938*, 2nd edition (pp. 152-269). Oxford: Oxford University Press.)

____. (1944). The semantic conception of truth and the foundations of semantics. *Philosophy and Phenomenological Research*, 4, 341-376.

Yablo, S. (1993). Paradox without self-reference. *Analysis*, 53, 251-252.