

LACUS FORUM XXXV

Language and Linguistics in North America 1608–2008: Diversity and Convergence



© 2009 The Linguistic Association of Canada and the United States (LACUS). The content of this article is from *LACUS Forum* 35 (published 2009). This article and others from this volume may be found on the Internet at *http://www.lacus.org/volumes/35*.

YOUR RIGHTS

This electronic copy is provided free of charge with no implied warranty. It is made available to you under the terms of the Creative Commons Attribution-NonCommercial license version 3.0

(http://creativecommons.org/licenses/by-nc/3.o/)

Under this license you are free:

- to Share to copy, distribute and transmit the work
- **to Remix** to adapt the work

Under the following conditions:

- Attribution You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- Noncommercial You may not use this work for commercial purposes.

With the understanding that:

- **Waiver** Any of the above conditions can be waived if you get permission from the copyright holder.
- Other Rights In no way are any of the following rights affected by the license:
 - Your fair dealing or fair use rights;
 - The author's moral rights;
 - Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

Notice: For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to the web page cited above.

For inquiries concerning commercial use of this work, please visit *http://www.lacus.org/volumes/republication*

Cover: The front cover of this document is licensed under the Creative Commons Attribution-No Derivative Works 3.0 license (*http://creativecommons.org/licenses/by-nd/3.0/*) and may not be altered in any fashion. The LACUS "lakes" logo and Université Laval logo on the cover are trademarks of LACUS and Université Laval respectively. The Université Laval logo is used here with permission from the trademark holder. No license for use of these trademarks outside of redistribution of this exact file is granted. These trademarks may not be included in any adaptation of this work.

THE LIAR OR THE MAD DOG(S)? ANOTHER LACUS LOOK AT THE LIAR PARADOX

Rennie Gonsalves Brooklyn College

RECENTLY PAUL SAKA (2007A) HAS CLAIMED that his Attitudinal Semantics resolves the Liar paradox by suggesting that (a), "this sentence is false," should really be understood as (b), "X thinks that this sentence is false," where "X" is the speaker . Saka's suggestion is that while (b) leads to the conclusion that X has contradictory beliefs, this is fine because the possession of contradictory beliefs is, after all, ubiquitous. This paper will first present an informal truth value analysis of the Liar, and then go on to discuss Saka's proposal. Then we will look at an alternative approach to the Liar suggested by Jerrold J. Katz (2004). We will argue that Katz's approach not only takes full account of what actually happens when we try to assign a truth value to the Liar sentence, but also preserves both the consistency and the effabilism, or expressive completeness, of natural language.

Anyone who has gone through the process of learning to write programs in a computer programming language will be familiar with the commonplace error of writing code that sends the computer processor into an infinite loop. This can happen, for example, when a program contains a series of statements that need to be repeated several times until a condition is met, as might happen in what is referred to by programmers as a "do loop." Here is an example of a very simple "do loop" in C that works fine:

```
Program A;
x = 0;
do {
    x = x + 1;
} while (x<9);</pre>
```

What this means in simple words is as follows: begin by setting the value of x to zero; then increase the value of x by one so long as the condition (x is less than 9) is true; when the condition is false, stop. Of course this little loop will stop when the value of x is equal to 9, because it is at that point that the condition becomes false. But consider the following do loop:

```
Program B;
x = 1;
do {
    x = x + 1;
} while (x>0);
```

This loop will never terminate because the condition (x is greater than zero) will always be true.

Now consider the following sentence:

(1)
$$[_{S}[_{NP} \text{This sentence}]_{i}[_{VP} \text{ is false.}]]_{i}$$

Sentence 1 is a version of the so called Liar sentence, with the additional annotation of identical subscripts to plainly indicate the co-reference relation between the subject NP and the S-node of the sentence itself.¹ Looked at in one way this sentence is paradoxical because if it is taken to be false, then it must be true, and if it is taken to be true, then it must be false. It thus seems to violate the law of non-contradiction in logic, which can be written as follows:

(2) \neg (P ^ \neg P); (it is not the case that P and not P, or P can never be both true and false at the same time in the same circumstances)

The following parallel sentence, however, results in no such paradox:

(3) $[_{S}[_{NP} \text{This sentence}]_{i}[_{VP} \text{ is false.}]]_{k}$

The reason why sentence 3 is not paradoxical, of course, is that the subscripts are no longer identical: the NP is subscripted with the letter i, while the S-node of the sentence is subscripted with the letter k. We can imagine the very natural occurrence of sentence 3 in a text, for example, where the NP subject here refers to the immediately preceding sentence of a paragraph, a sentence appropriately subscripted with the letter i, with the result that sentence 3 simply makes the claim that that other sentence is false. The following example illustrates this situation.

(4) a. $[_{S}[_{NP}John]_{i}[_{VP}is hungry.]]_{j}$ b. $[_{S}[_{NP}This sentence]_{i}[_{VP}is false.]]_{k}$

The use of indexing to indicate anaphoric reference goes back at least to Chomsky's famous article,
 "On Binding," (Chomsky 1981). Here is an example from a more current introductory text in syntax that closely parallels the way I have used indexing (Carnie 2007:142) :

- ii) *Heidi, believes Martha's description of herself.
- iii) Heidi, believes Martha,'s description of herself.

The only possibly unusual feature of my use of indexing is that the expression "this sentence," an NP, is co-indexed with the sentence itself, which is arguably of a different syntactic category. But this is only the result of the meta-linguistic nature of the relevant referential relationship in the Liar sentence.

What this sequence of two sentences says, ultimately, is that John is not hungry, which could turn out to be true or false, but certainly not both. So it would seem that the liar sentence depends crucially on the co-reference relation between the NP subject and the S for its status as a paradox.

Let's imagine that we are interested in writing a very simple program for the computer that will figure out the truth values of sentences like the ones in 1 and 4 above in a particular situation. We will do this informally so that we do not run into trouble with real-world C programmers. Before we start we will outline a few preliminary conventions. We will assume that the semantic value of predicates, like the VPs in our sentences, are sets of individuals, and we will use notation as in the following:

- (5) $S = \{a, b, c\}$; S is the set whose members are a, b, and c.
- (6) $a \in S$; a is a member of the set S.

We will assume that the semantic value of NPs are individuals; here we have only three: John, and the truth values true (indicated with the numeral 1) and false (indicated with the numeral 0). Let's imagine that the VP "is hungry" has the semantic value, in a particular situation v, of the set whose members are John, Mary and Harry; we will indicate this as follows:

(7)
$$[[is hungry]]^v = \{ John, Mary, Harry \}$$

Here the double square brackets on the left of the equal sign indicate the semantic value of their contents, and the superscript v denotes the situation v. We will also assume the following:

- (8) [[John]]^v = John; the NP "John" denotes the individual whose name is John.
- (9) $[[\text{This sentence}]]_x^v$ is the semantic value of the co-indexed constituent, the constituent whose index is the same as x.
- (10) $[[is false]]^v = \{o\};$ the predicate "is false" denotes the set whose only member is 0.
- (11) $[[is true]]^v = \{1\};$ the predicate "is true" denotes the set whose only member is 1.
- (12) $[[[_{S}NP VP]]]^{v} = I \text{ iff } [[NP]]^{v} \in [[VP]]^{v} \text{ and } o \text{ otherwise; a subject-predicate sentence is true if and only if the semantic value of its NP subject is a member of the set denoted by the semantic value of its VP predicate.$

Let's imagine that our program for finding the truth values for our sentence in the situation v is as follows:

```
Evaluate (S<sup>v</sup>);
[[S]]<sup>v</sup> = unknown;
do {
    a. find the semantic value of the NP in S;
    b. find the semantic value of the VP in S;
    c. if the semantic value of the NP is a member of the
```

```
set denoted by the semantic value of the VP, then
[[S]] = 1, else [[S]] =0;
} while ([[S]] = unknown);
```

Now let's see what happens when we try to calculate the semantic values of sentence (4)b and sentence 1. In computing the value of sentence (4)b, our Evaluate program initially sets the semantic value of sentence (4)b to "unknown" and then proceeds to the "do loop"; it encounters statement a, and so tries to find the semantic value of the NP, [This sentence]. According to our rule 9 above, the semantic value of this NP in sentence (4)b is equal to the semantic value of the co-indexed constituent, which happens to be sentence (4)a. So sentence (4) a needs to be evaluated. As in the case of sentence (4) b, initially sentence (4)a is assigned the semantic value "unknown." But very quickly our evaluation routine finds that the semantic of the NP in (4)a is the individual John, that the semantic value of the VP in (4)a is the set {John, Mary, Harry}, and indeed that the semantic value of the NP in (4)a is a member of the set denoted by the semantic value of the VP in (4)a; therefore, following statement c, our evaluation routine sets the semantic value of sentence (4)a to 1, or true. Now, according to our rule 9, this gets to be the semantic value of the NP in (4)b. Proceeding then with our evaluation of (4)b, we find by statement b of Evaluate and rule 10 above that the semantic value of the VP "is false" is equal to the set whose only member is zero, $\{0\}$; then by statement c of Evaluate we find that 1 is not a member of the set $\{0\}$, so we do the "else" which tells us to set the value of the sentence, here sentence (4)b, to \circ or false. The next step in Evaluate is to check the condition; since [[S]] is 0 or false and so is no longer equal to "unknown," the condition of the "do loop" turns out to be false, and so we stop. Again, in our imaginary situation v, sentence (4) b also turns out to be false, consistent with our intuitions.

When we try to evaluate sentence 1 with our Evaluate routine, here is what happens. As before, we initially assign sentence 1 the value "unknown." We then proceed to the first statement in the "do loop" which tells us to find the value of the NP. Here our rule 9 tells us that the value of our NP here, [This sentence]_i is the value of sentence 1 itself, since this is the co-indexed constituent. So our next step is to evaluate sentence 1, again; and this happens again, and again, and again without end because we are in an infinite loop, just as in our little C program, Program B.²

Notice that the reason for the infinite loop in Program B is different from the reason for the infinite loop in the Liar sentence, sentence 1. Program B results in an infinite loop because, even though it gets tested again and again, the condition in the do loop is always true; it never becomes false so we never get to stop. But with our evaluation of the liar sentence, we never get to even test the condition; we are stuck at evaluating the co-referential

² Note that the looping occurs without ever evaluating the predicate "is false." This indicates, as noted by Saul Kripke (1975:693), that even if the predicate were "is true," the sentence would be just as problematic from a referential point of view, even though it would not give the same impression of being paradoxical.

constituents NP and S. This is very much like a dog chasing its own tail, spinning around in a circle. The difference of course is that sooner or later the dog gets tired, but not the Liar.

However, while the reasons for the infinite loops in these two procedures, Program B and the Liar sentence, might be different, the result is the same: processing continues indefinitely without a resolution. We never get a numerical value of x in Program B, and we never get a truth value for the Liar sentence. Because of the enormous complexity of many computer programs it would seem that there must be a lot of different ways to wind up with infinite loops in programs; but, by virtue of being infinite, it also seems as though the result of any infinite loop must always be the same: namely, that no value for whatever is sought in the program is ever arrived at.

Of course, we are not the first people to notice this infinite looping of the Liar. Here is my colleague at Brooklyn College, Emily Michael (1975:370),³ in her paper "Peirce's Paradoxical Solution to the Liar's Paradox," quoting one of the greatest American philosophers, Charles Sanders Peirce, on the topic:

A logically meaningful sentence will satisfy the laws of logic. Peirce argues that this logical law does not apply to SI [the Liar] because this symbol has no object. Logic, Peirce says, is concerned with assertoric propositions[1]... He says of assertoric propositions, "Propositions which assert always assert something of an object, which is the subject of the proposition."[2]... In the case of SI, however, the proposition "does itself state that it has no object. It talks of itself and only of itself and has no external relation whatever."[3]... That is, the subject of the proposition being the proposition itself, the predicate makes no assertion of an object to which the proposition refers. An assertoric proposition, then, makes reference to an external object, but this proposition "talks of itself and only of itself and has no external relation whatever." "Logical laws," however, "only hold good as conditions of a symbol having an object."[4]...

And here is Hans G. Herzberger (1970:150), explaining more exactly what's wrong with sentences like the Liar, and using a very apt expression (in quotes) from John Searle's 1959 dissertation:

All the cases so far have involved a sentence that belongs to its own ancestral domain. On a deeper analysis, however, it is not the cyclical character that renders them pathological; and it is a virtue of the grounding condition that it brings it out. Consider that all the cases so far suffer from "unconsummated reference."⁴

³ I have renumbered the author's footnotes for this passage. They are as follows: I. Ms 743, "Rules of Logic Logically Deduced," June 23, 1860. 2. *Ibid.* 3. Ms 340 [1864]. 4. *Ibid.* The manuscript (Ms) numbers here are to "The Charles S. Peirce Papers," at the Houghton Library of Harvard University, as provided in Robin 1967.

⁴ It seems that Searle's use of the word "unconsummated" here is intended to capture a sense similar to the sense of "consummation" in Hamlet's famous "to be, or not to be" soliloquy:

And finally, here is Gilbert Ryle (1951:68), in his paper, "Heterologicality," letting us know why he thinks the Liar never really makes a statement, and coming to a conclusion that is very similar to our own here:

If unpacked our pretended assertion would run "The current statement {namely that the current statement [namely that the current statement (namely that the current statement . . ." . The brackets are never closed: no verb is ever reached; no statement of which we can even ask whether it is true or false is ever adduced.

However, in his book *How to Think About Meaning*, Paul Saka will have none of this talk about an evaluation of the truth value of the liar sentence leading to an infinite loop without ever terminating in a truth value because of Peirce's lack of an object or Searle's "unconsummated reference" or Ryle's failure to get to the verb. Here is the closest Saka (2007b:221) ever comes to acknowledging the presence of such problems in the liar:

... in order to understand " λ is false", one arguably does need to identify the referent of " λ "; and in order to grasp the referent of " λ " one arguably does need to understand the statement " λ is false". Thus, in some cases self-reference might present a circle that is impossible to break into; such reference is [according to Kripke 1975:706] "ungrounded" ...

However, Saka very quickly goes on to argue that we should not make such a fuss about "self-reference" since "it is not clear that self-reference is necessary for paradox" (2007b:221). He uses the following example as illustrative of a paradox without self-reference:

- (13) Nixon states the following and nothing but the following: "What Jones says is true."
- (14) Jones states the following and nothing but the following: "What Nixon says is false."

Using our co-indexing notation, simplifying the syntax a bit, and abbreviating the example, we might write it as follows:

 $\begin{array}{ll} (15) & \left[{}_{S} \left[{}_{NP} \text{What Jones says} \right]_{k} \left[\text{is true} \right] \right]_{m} \\ (16) & \left[{}_{S} \left[{}_{NP} \text{What Nixon says} \right]_{m} \left[\text{is false} \right] \right]_{k} \end{array}$

To die: to sleep: No more; and by a sleep to say we end The heart-ache and the thousand natural shocks That flesh is heir to; 'tis a consummation Devoutly to be wished.

In the Liar, the consummation of the referential value of the expression "this sentence" is "devoutly to be wished." Alas, that wish remains unfulfilled.

It is clear that an evaluation of this example will lead our little Evaluate routine into just as endless an infinite loop as with the Liar sentence, because in order to come up with a truth value for 15 we have to first come up with a truth value for 16, since the NP in 15, $[_{NP}$ What Jones says]_k is co-indexed with the S-node in 16; but in order to come up with a truth value for 16 we have to first come up with a truth value for 15, since the NP in 16, $[_{NP}$ What Nixon says]_m, is co-indexed with the S-node in 15. Again, to find the truth value for sentence 15 we need to find the truth value for sentence 16, and to find the truth value for sentence 16 we need to find the truth value for sentence 15. So we have another case of tail-chasing; or rather, this time we have two dogs with each one simultaneously chasing the other's tail—a slightly larger circle, but a circle nonetheless, and more importantly a failure to refer and to ever get to the verb, as Ryle points out concerning the more traditional Liar.

But that is not all; we can put as many dogs into the ring as we please. Notice that there is no reason for us to stop at just two sentences in a so called "loop liar"; we could have as many sentences as we wish. Consider the following n+1 sentences, where n is any positive integer:

(17) I. $[_{S}[_{NP} What Jones says]_{l+1}[is true]]_{m}$ 2. $[_{S}[_{NP} What Smith says]_{l+2}[is true]]_{l+1}$ 3. $[_{S}[_{NP} What Richardson says]_{l+3}[is true]]_{l+2}$ 4. $[_{S}[_{NP} What Roberts says]_{l+4}[is true]]_{l+3}$... n+1. $[_{S}[_{NP} What Edwards says]_{m}[is false]]_{l+n}$

This loop is just as "paradoxical" as the pair in (15) and (16). It is clear from this that the liar sentence is just the limiting case of the loop liar consisting of n+1 sentences, where n happens to be equal to zero, as in sentence (18).

(18) $\left[{}_{S} \left[{}_{NP} W hat I am saying now} \right]_{i} [is false] \right]_{i}$

So there is nothing very special about the loop liar sentences at all; reference is just as ungrounded as in the case of the single liar sentence. In fact, what the loop liar does for us is that by widening the loop a bit it helps to underline and make clear the endlessly looping reference situation that is characteristic of the liar, a situation that results in the "unconsummated reference" noticed by John Searle and others.

The failure of reference in the case of the Liar, a grammatical sentence which quite literally makes sense, underlines the need for a clear distinction between sense and reference. Jerrold J. Katz's autonomous sense theory makes just such a distinction, and in his last book, *Sense, Reference, and Philosophy*, Katz clearly lays out the solution his theory provides for the problems related to the Liar sentence and other similar cases. Katz refers to the situation created in philosophy and especially logic by the Liar sentence as the Epimenidean dilemma, after the sixth-century Greek philosopher Epimenides, whose line in a poem, "the Cretans, always liars," apparently came to be associated with the Liar paradox, perhaps by Bertrand Russell, because Epimenides was himself a Cretan. One of the problems posed by the Liar paradox is the problem of the consistency of language, since as we have seen, the Liar might be taken to lead to a violation of the law of non-contradiction. Katz puts the problem this way: "how can we consistently reason to the conclusion that the language on which our reasoning depends is inconsistent" (Katz 2004:95). The other side of the coin of consistency is the concept of effabilism, or expressive completeness, on which Katz quotes Alfred Tarski as saying the following: "it could be claimed that 'if we can speak meaningfully about anything at all, we can also speak about it in colloquial language" (*ibid*, Tarski 1956:164) Tarski thus seems to come down on the side of effabilism, and to sacrifice the notion that natural languages are consistent. Katz suggests that both Tarski and Gottlob Frege favored the establishment of a "logically perfect" artificial language in order to overcome the various inadequacies of natural languages (Frege 1952:70), such as that highlighted by the Liar sentence. Katz sees his autonomous sense theory as providing a way through these "two horns" of the dilemma. Unlike Frege's theory of sense (Frege 1952:56–78), by which sense determines reference, in Katz's theory sense mediates but does not determine reference. This allows Katz to distinguish the reference or extension of an expression from its sense or intension. Thus a sentence has both an extensional proposition and an intensional propostion, and either or both of these can be defective. In the case of the Liar sentence the intensional proposition is fine: the sentence makes perfect sense. However, the extensional proposition is defective. Here there is a case of "unconsummated reference," the verb never being reached, or as I prefer it, the mad dog constantly chasing his own tail. Katz's solution thus seems consistent with the diagnosis of Hans G. Herzberger (1960) that the subject NP in the liar sentence is "ungrounded." Herzberger is clear on the notion that the exact logical criterion for groundedness is in need of more precise specification, but for me the metaphor is sufficient to explain what's going on with the Liar. Moreover, Katz's solution seems to save both effabilism, the expressive power of language, and consistency, its logical soundness, and without indicting anyone.

Which takes us back to Paul Saka's solution to the dilemma of the Liar. He seems to take the metaphor of the Liar literally and then indicts the speaker of the sentence as capable of having inconsistent beliefs. Here is how he lays out his theory (2007b:236) with respect to the Liar:

As I see it, truth-conditional T-schemas fail because they assume a sort of objectivism; they assume that the analysis of truth need not make essential and explicit reference to some subject who thinks about truth. They need to be rejected in favor of the following attitudinal T-schema:

 (T_{ψ}) S thinks " Φ is true" \equiv S thinks that P.

Instantiation by the original Liar yields:

(OK) S thinks " λ is true" \equiv S thinks λ is false.

If S thinks λ is true then S thinks λ is false; and if S thinks λ is false then S thinks

 λ is true. Either way, S is highly irrational. But this is not at all paradoxical because there is a difference between an inconsistent state of affairs and an inconsistent system of beliefs. The former, by its very nature, cannot obtain, yet the latter is not only possible but common, even ubiquitous.

Here λ is the Liar sentence. Notice that on our diagnosis here to believe or not to believe the Liar sentence is not to believe or to disbelieve anything of consequence since the Liar makes no claim about any object on Peirce's account, or never reaches the verb on Ryles's account. Actually, on a personal note, having grown up in the West Indies the image that I get from the prospect of believing or not believing the Liar sentence is the image of someone looking at the cloud of dust created on a hot tropical day in a dry dirt West Indian backyard by a mad dog desperately chasing his own tail. That person looking at the mad dog must wonder what's wrong with him. This is just like the computer programmer who tries to debug his flawed code that sent the computer processor into an infinite loop. He too is looking for some error. Maybe he made an error in his program, but he is not satisfied until he finds the mistake, corrects it, and gets the entire program working properly.

 \sim

REFERENCES

- CARNIE, ANDREW. 2007. Syntax: A generative introduction. Malden MA: Blackwell Publishing.
- CHOMSKY, NOAM. 1981. On binding. Linguistic inquiry 11:1-46.
- FREGE, GOTTLOB. 1952. *Translations from the philosophical writings of Gottlob Frege*, ed. Peter Geach & Max Black. Oxford: Basil Blackwell.
- HERZBERGER, HANS G. 1960. Paradoxes of grounding in semantics. *The journal of phi*losophy 47(6):145-67.
- KATZ, JERROLD J. 2004. *Sense reference and philosophy*. New York: Oxford University Press.
- KRIPKE, SAUL. 1975. Outline of a theory of truth. *The journal of philosophy* 72(19):690– 716.
- MICHAEL, EMILY. 1975. Peirce's paradoxical solution to Liar Paradox. *Notre Dame jour*nal of formal logic 16(3):369–74.
- PEIRCE, CHARLES SANDERS. 1967. "The Charles S. Peirce Papers" at the Houghton Library of Harvard University, catalogued by R. Robin in his Annotated catalogue of the papers of Charles S. Peirce. Amherst MA: University of Massachussetts Press.
- RYLE, GILBERT. 1951. Heterologicality. *Analysis* 11(3):61–68.
- SEARLE, JOHN. 1959. Problems arising in the theory of meaning out of the notions of sense and reference. Diss., Oxford University.

SAKA, PAUL. 2007a. Truth and beyond: Thinking about the Liar Paradox. Paper presented at LACUS Forum 34. (2007 Meeting Handbook, p. 46; published as Beyond truth. *LACUS forum* 34:211–20.)

------. 2007b. *How to think about meaning*. Dordrecht: Springer.

TARSKI, ALFRED. 1956. The concept of truth in formalized languages. In *Logic, semantics, metamathematics*, 152–278. Oxford: Clarendon Press.