Propositions, Truth and Paradox
PHIL 428/628, Week 2: Jan. 26 2016

SOAMES ON TARSKI

1. Tarskian Definitions of Truth

Tarski believed that the Liar paradox shows that natural languages such as English (or Polish or German!) are 'inconsistent'. Thus, his aim was *not* to give an account of truth in the sense of giving an account of the English word 'true', or the concept that this word expresses, or the property it refers to.

His aim was rather to show how to define a 'consistent' notion (a notion of 'truth', in some sense!) that could do much of the work that we might have hoped that the English word 'true' can do.

In particular, starting with an *object language* L, he showed how to define a notion of truth in a *metalanguage* M. M is here assumed to contain L as a 'subpart', and also to contain the means to quantify over expressions and sets of expressions of L, as well as things like sets of n-tuples of the objects L is about (i.e., objects in the domain that L's quantifiers range over).

(So L and M can presumably be thought of as standard 'interpreted languages' of the sort one studies in logic. I.e., something like a formal language, plus an 'interpretation' or 'model'; that is, a domain of quantification together with an interpretation function assigning denotations of the appropriate type to L's non-logical vocabulary. However, in some ways it is perhaps rather anachronistic to think of things that way: because this way of thinking came after Tarski; and I think to at least some extent is due to the work on truth we are discussing; thus, what Tarski did was a lot less obvious than it might appear when viewed from this ahistorical perspective!)

Since we are aiming to replace an inconsistent notion, what is the criterion for success? (It is not capturing the meaning of the notion to be replaced, of course (!), or even being coextensive with it.)

Tarski thought that it was necessary and sufficient that every instance of 'schema T' be assertible (and true) in M:

Schema T. X is T in L iff P.

Here T is our putative truth predicate for L. An instance of this schema results from replacing 'X' with a name of a sentence of L (i.e., a quote-name), and replacing 'P' with either that sentence, or with a paraphrase of it. That is, Tarski claimed that T is a truth predicate for L iff every instance of schema T is itself assertible and true.

How then to give a definition that yields a predicate that meets this success condition?

Option 1. If L had only finitely many sentences S_1, …, S_n then one could simply give a definition with a clause for each sentence:

- For any sentence x of L, x is true in L iff either
  s = 'S_1' and S_1, or
  ⋮
  s = 'S_n' and S_n.

But of course typically we will be interested in languages that have infinitely many sentences (e.g., are closed under negation, so if A is a sentence, then so is ¬A). Thus, this method is of rather limited application.

Option 2. 'Quantify into quotes'. I.e., give a definition along the following lines:

- For all S, 'S' is true in L iff S.

However, here the variable S is not like a variable in 'name position' (as in, e.g., ∀x x = x). Rather, it is in 'sentence position' (to get a well-formed sentence one has to replace 'S' not by a name but by a sentence; since no sentence can end 'iff Obama' e.g.!). Thus, one must have some way of making sense of such quantification.

Another issue with this option is that it might seem that one simply cannot legitimately quantify into quotational contexts. And thus that the only way to understand the above is as having instances not such as:

'snow is white' is true in L iff snow is white

but rather:

'S' is true in L iff snow is white.

—And this is not of course what we want! (Since it is not an instance of schema T.)

(Cf. there is no legitimate reading of 'for all x, Felix loves x' on which 'Felicity loves city' in an instance. —Sorry, that was the best I could do!)

The more substantive issue, however, was that Tarski thought that if we allow the sort of quantification into quotes that this option requires then this will give rise to paradox.

The worry concerns sentences such as the following:

(1) ∀S[(1) = 'S' → ¬S].

Getting a contradiction from consideration of (1):

- Suppose ∀S[(1) = 'S' → ¬S].
- Then: (1) = '∀S[(1) = 'S' → ¬S]' → ¬∀S[(1) = 'S' → ¬S] (substituting (1) for 'S').
- Then: ¬∀S[(1) = 'S' → ¬S] (since the antecedent of last line is true).
- Contradiction!
- So: ¬∀S[(1) = 'S' → ¬S].
- Then: ∃S[(1) = 'S' ∧ S] (logic).
- Then: ∀S[(1) = 'S' → ¬S] (since this is the only sentence identical to (1)).
- Contradiction again!

This would seem to vindicate Tarski. But Soames apparently very reasonably makes the point that everything would be OK as long as we understood sentential quantification in essentially the same sort of hierarchical way that Tarski is proposing for truth.

That is, we would use a metalanguage M that contains sentential quantification over the sentences of L. But we would not allow these sentential quantifiers to themselves range over sentences that contain them.

Thus, although we could form a sentence along the lines of (1) using this sort of 'hierarchical' sentential quantification, it could not legitimately be substituted for 'S'. The first step in the above derivation would be blocked; and (1) would be vacuously and unparadoxically true.

Option 3. (AKA the preferred option!) Let's consider the simpler case where every object that the quantifiers of L range over is denoted by some closed term (i.e., some term without free variables).

Step A. One first defines denotation for closed terms of L, and application for predicate letters of L. As follows:

- In the case of an individual constant one just has a clause saying what the term denotes.
  E.g., saying '0' denotes 0, or 'Obama' denotes Obama, etc.
- For other closed terms, one has a clause saying what the denotation of the term is terms of the function symbol and the simpler terms that the term is constructed out of.
  E.g., saying the denotation of 't + s' is the sum of the denotation of 't' and the denotation of 's'.
- For predicate letters one just has a clause for each predicate letter.
  E.g., 'Loves' applies to a pair of objects $\langle o,p \rangle$ iff o loves p.

Step B. One defines truth for L in terms of denotation and application.

- One has clauses for atomic sentences.
  E.g., saying that an atomic sentence 'Loves(t,s)' iff 'Loves' applies to the pair of objects $\langle o,p \rangle$ denoted by 't' and 's' (resp.).
- One then has clauses for compound sentences.
  E.g., a sentence $\neg A$ is true iff A is not true; and
  a sentence $\forall xB$ is true iff every sentence that results from replacing all free occurrences of x in B with some closed term is true.

It is then easy to show that from such a definition one can derive all instances of schema T—as desired. E.g.:

- 'Loves(Peter,Mary) $\wedge$ Loves(Mary,Peter)' is true iff 'Loves(Peter,Mary)' is true and 'Loves(Mary,Peter)' is true (clause for conjunctions).
- 'Loves(Peter,Mary)' is true iff Peter loves Mary (combining clauses for atomic sentences with clauses for 'Loves', 'Peter' and 'Mary' in definitions of application and denotation).

- Similarly: 'Loves(Mary,Peter)' is true iff Mary loves Peter.
- Combining gives: 'Loves(Peter,Mary) ∧ Loves(Mary,Peter)' is true iff Peter loves Mary and Mary loves Peter (an instance of schema T).

The more complicated case where some things L's quantifiers range over are not denoted by closed terms is similar. But denotation and truth must be defined not absolutely but relative to assignments to variables (i.e., relative to sequences of objects). One then says that a sentence is true in L iff it is true relative to some (or equivalently: all) assignments.

2. Evaluating the Approach

2.1. Tarskian Hierarchies

This approach blocks the Liar paradox by defining a truth predicate for L not in L itself, but in a metalanguage. Thus, rather than a language which contains its own truth predicate, one will have a hierarchy of languages and truth predicates:

L (our original object language)
L_1 (L plus T_0, our truth predicate for L)
L_2 (L_1 plus T_1 a truth predicate for L_2)
⋮

2.2. The Liar

This hierarchy will not be susceptible to the Liar paradox because none of the truth predicates can ever apply to themselves. Thus, an attempted Liar sentence $\neg T_n(c)$, where c denotes this sentence, will not be T_n (since it contains T_n); so it will be T_{n+1}; so we have $T_{n+1}(c) \wedge \neg T_n(c)$. But there is of course nothing contradictory about that!

2.3. Nixon and Dean

But can these Tarskian truth predicates do everything that one might hope? Soames considers the following case (taken from Kripke): Dean utters (1), and Nixon utters (2).

(1) Nothing Nixon says about Watergate is true.
(2) Nothing Dean says about Watergate is true.

Now, if Dean and Nixon don't say anything else about Watergate, then these utterances are somewhat 'weird', and it is not clear how we should evaluate them for truth or falsity. However, there are other cases where it seems that we can perfectly straightforwardly classify them as true or false.

Case 1. Both Dean and Nixon say something straightforwardly true about Watergate. (E.g., 'Watergate has caused some problems'.) Then it seems that we want to say that Dean and Nixon's utterances of (1) and (2) are simply false.

Case 2. Dean says something straightforwardly true about Watergate. Nixon says something straightforwardly false about Watergate. (E.g., 'Watergate will blow over'.) Neither says anything else about Watergate. Then it seems that Nixon's utterance of (2) is

straightforwardly false. *Then* it seems that Dean's utterance of (1) is straightforwardly true.

What does this example show? Well, it shows that sometimes pairs of utterances can each be about whether the other is true, but can nevertheless straightforwardly be classified as either true or false. We might thus hope that utterances with this general structure could be accommodated within Tarski's framework.

But there are problems with that. If Dean and Nixon are required to speak 'Tarskian' (i.e., use predicates in the Tarskian hierarchy in place of 'true'), then the closest we can get to utterances along the lines of (1) and (2) will be something like the following:

(1') Nothing Nixon says about Watergate is T_0.
(2') Nothing Dean says about Watergate is T_0.

In cases 1 and 2, we will get the correct truth-values. However, we will get them for the 'wrong' reasons, since neither utterance will depend on the truth-value of the other.

A different case where we don't even get the right truth-values is as follows. Suppose Dean utters (3) and Nixon utters (4).

(3) Something Nixon says about Watergate is true.
(4) Something Dean says about Watergate is true.

Suppose that Dean says nothing else about Watergate, but Nixon says something else that is straightforwardly true. Then it seems that both utterances are straightforwardly true.

But we cannot get this effect with Tarskian predicates—or at least not in any natural way. Since if they use the same Tarskian predicate, then Nixon's utterances will not come out as true.

This might prompt the thought: what if Dean and Nixon use a context sensitive term $T_?$ (say) that functions like a Tarskian truth predicate in any given context, but where *which* truth predicate it functions as is determined not by local features of the context, but by which sort of things the people you are talking about say? This is something we will consider in due course (when we discuss Burge)!

2.4. Quantifying into Subscript Position?

Of course, what we really want to say are things like: something Nixon says about Watergate is T_n for some n. Tarski does not show how such quantification is possible, however, and it is hard to see how it is possible within his hierarchy without getting into difficulty.

For consider a sentence:

$$\neg\exists n T_n(c)$$

where c denotes this sentence. Suppose this is a sentence of L_m (for some m). So we would have:

$$T_m(c) \leftrightarrow \neg\exists n T_n(c).$$

So $\neg T_m(c)$. But we also have: $\exists n T_n(c) \leftrightarrow T_m(c)$. So $\neg \exists n T_n(c)$ and thus $T_m(c)$—contradiction.