Tarski's definitions of truth

A Tarskian definition of truth concerns the sentences of a language L and is formulated in a language L′. We call L the *object-language,* and L′ the *metalanguage*. L may be part of L′; that is, it may be that among the sentences of L′ are all the sentences of L. In Tarski's own work, the object-language is always a symbolic, artificial language, and the metalanguage also involves many symbols. In what follows, our metalanguage will be English, enriched (whenever necessary) with technical means.

A *truth-predicate* for a language L is a predicate that applies to the true sentences of L and to nothing else. A Tarskian definition of truth for L (more precisely, a Tarskian definition of truth for L formulated in English) is a definition of a truth-predicate **T** for L which, for every sentence of L, allows us to prove an equivalence of the form

S is **T** iff (if and only if) p

where the letter S has been replaced by a term referring to the sentence in question (e.g., it has been replaced by the sentence written in quotation marks) while the letter p has been replaced by a translation of that sentence into English (possibly, by that same sentence, if L is part of English).

Tarski showed how we can construct Tarskian definitions of truth for various symbolic languages. Let's look at two examples of my own. The first is unlike Tarski's own examples, but it is simpler because it concerns an object-language without quantifiers. The second is quite like Tarski's examples (in his monograph on truth) except that his object-language concerns sets and not people.

Let L be a language that has two names, 'b' ('Bob', for a particular Bob) and 'm' ('Mary'), and two predicates, 'I' ('intelligent') and 'C' ('courageous'). The basic sentences of L are the following four: 'Ib' ('Bob is intelligent'), 'Im', 'Cb' and 'Cm'. L also has the negation symbol '¬' and the conjunction symbol '∧'. If **S** is any sentence of L, ¬**S** is also a sentence of L. If **S** and **S**′ are any sentences of L, (**S** ∧ **S**′) is also a sentence of L. L has no other sentences.

We want to define a truth-predicate for the language L. Our predicate will be the (new) word 'alg' and the definition will be as follows:

- the sentence 'Ib' is alg iff Bob is intelligent
- the sentence 'Im' is alg iff Mary is intelligent
- the sentence 'Cb' is alg iff Bob is courageous
- the sentence 'Cm' is alg iff Mary is courageous
- for each sentence **S** of L, ¬**S** is alg iff **S** is not alg
- for each sentence **S** of L and each sentence **S**′ of L, (**S** ∧ **S**′) is alg iff **S** is alg and **S**′ is alg
- anything that is not a sentence of L is not alg.

In its first four parts, the definition explains when we call a basic sentence of L ‘alg’. In its next two parts, it explains when we call a compound sentence of L ‘alg’, and explains it using ‘alg’ as applied to shorter sentences of L. Because it has that form, the definition is characterized as *recursive*. (To be precise, in its next two parts, the definition explains when we call ‘alg’ a sentence of L in which the symbol of negation and the symbol of conjunction occur a total of n times, where n > 0, and explains it using ‘alg’ as it applies to sentences in which the symbol of negation and the symbol of conjunction occur fewer times.)

The definition allows us, for each sentence of L, to prove an equivalence of the form

S is alg iff p

where the letter S has been replaced by the sentence in question in quotation marks, while the letter p has been replaced, without quotation marks, by a translation of that sentence into English. For example, it allows us to prove that

‘¬Ib’ is alg iff Bob is not intelligent,

as well as that

‘(Cm ∧ Cb)’ is alg iff Mary is courageous and Bob is courageous.

The word ‘alg’, as we have defined it, is indeed a truth-predicate for L. For, as those equivalences jointly show, any sentence of L is alg iff it is true.

Now, let N be a language that lacks names, has two predicates, which are ‘C’ and the two-place predicate ‘E’ (‘loves’), and possesses the symbols '¬' and ‘∧’, as well as the universal quantifier ‘∀’ and the infinitely many variables ‘x_1’, ‘x_2’, ... The sentences of N constitute a subset of its formulas. Each basic formula of N either consists of ‘C’ followed by a variable or consists of ‘E’ followed by two variables (or one written twice); every such string of symbols is a basic formula of N. If **A** is any formula of N, ¬**A** is also a formula of its. If **A** and **A**′ are any formulas of N, (**A** ∧ **A**′) is also a formula of its. And if **A** is any formula of N while **v** is any variable of N, (∀**v**)**A** is also a formula of N. N has no other formula. The sentences are precisely those formulas in which no variable has any free occurrence. The set over which the variables range will be the set H of people. Thus the sentence ‘$(\forall x_1)Cx_1$’ tells us, about the members of H, that they are all courageous.

Before defining a truth-predicate for N, we should define what it means to say that a sequence of members of H *satisfies* a formula of N. The definition is again recursive, and the phrase ‘for every sequence s of members of H’ is understood at the beginning of each part except the last. It reads as follows:

- for each j, s satisfies the formula Cx_j iff the jth term of s is a courageous person
- for each j and each k, s satisfies the formula Ex_jx_k iff the jth term of s loves the kth term

- for each formula **A** of N, s satisfies the formula ¬**A** iff s does not satisfy **A**
- for any formulas **A** and **A**′ of N, s satisfies the formula (**A** ∧ **A**′) iff s satisfies **A** and s satisfies **A**′
- for each j and each formula **A** of N, s satisfies the formula $(\forall x_j)$**A** iff, for every sequence s′ of members of H that differs from s in at most the jth term, s′ satisfies **A**
- if either x is not a sequence of members of H or y is not a formula of N, x does not satisfy y.

In its first two clauses, the definition explains the word 'satisfies' as regards any sequence of members of H and a basic formula. In its next three clauses, the definition explains the word 'satisfies' as regards any sequence of members of H and a compound formula, and explains it using the word 'satisfies' as applied to sequences of members of H and shorter formulas.

We can now define a truth-predicate for N. Our predicate will be the word 'ald' (new again), and the definition is as follows: something is ald iff it is a sentence of N and is satisfied by all sequences of members of H. The definition allows us, for every sentence of N, to prove an equivalence of the form

S is ald iff p

where the letter S has been replaced by the sentence in question in quotation marks, while the letter p has been replaced, without quotation marks, by a translation of that sentence into English (or, rather, into English enriched with variables). For example, it allows us to prove that

'$(\forall x_1)Cx_1$' is ald iff, for every member x of H, x is a courageous person.

What is the proof? First of all, the definition of 'satisfies' and 'ald' is added to a formal theory that possesses syntactic axioms (that is, axioms concerning the syntax of the language N) and contains an elementary set theory. The syntactic axioms will allow us to prove that the formula '$(\forall x_1)Cx_1$' is a sentence of N. Thus the sentence '$(\forall x_1)Cx_1$' is ald iff it is satisfied by all sequences of members of H. Now, on the basis of the definition of satisfaction,

for every sequence s of members of H, s satisfies '$(\forall x_1)Cx_1$'

iff for every sequence s of members of H, for every sequence s′ of members of H that differs from s in at most the first term, s′ satisfies 'Cx_1'

iff for every sequence s of members of H, for every sequence s' of members of H that differs from s in at most the first term, the first term of s′ is a courageous person.

The elementary set theory will allow us to prove that for any sequence s of members of H [(for every sequence s' of members of H that differs from s in at most the first term, the first term of s' is a courageous person) iff, for every member x of H, x is a courageous person]. Hence for every sequence s of members of H, for every sequence s' of members of H that differs from s in at most the first term, the first term of s' is a courageous person

iff for every sequence s of members of H, for every member x of H, x is a courageous person.

Finally, in the sentence after the last ‘iff’, the phrase ‘for every sequence s of members of H’ is redundant and can be deleted.

Similarly, we can prove that

the sentence ‘$(\forall x_3)(Cx_3 \wedge Ex_3x_3)$’ is ald iff, for every member x of H, x is a courageous person and x loves x.

And we can also prove that

the sentence ‘$(\forall x_3)(Cx_3 \wedge \neg (\forall x_1)Ex_3x_1)$’ is ald iff for every member x of H (x is a courageous person and it is not the case that, for every member y of H, x loves y).

In all these proofs, the last step is to delete the phrase ‘for every sequence s of members of H’, which is now redundant. The predicate ‘ald’, as we have defined it, is indeed a truth-predicate for the language N.

Yannis Stephanou