

# The Tarski T-Schema is a Tautology (Literally)\*

Edward N. Zalta<sup>†</sup>

Center for the Study of Language and Information  
Stanford University  
zalta@stanford.edu

## Abstract

The Tarski T-Schema has a propositional version. If we use  $\varphi$  as a metavariable for formulas and use terms of the form *that- $\varphi$*  to denote propositions, then the propositional version of the T-Schema is: *that- $\varphi$*  is true if and only if  $\varphi$ . For example, *that Cameron is Prime Minister* is true if and only if Cameron is Prime Minister. If *that- $\varphi$*  is represented formally as  $[\lambda\varphi]$ , then the T-Schema can be represented as the 0-place case of  $\lambda$ -Conversion. If we interpret  $[\lambda\dots]$  as a *truth-functional* context, then using traditional logical techniques, one can prove that the propositional version of the T-Schema is a tautology, literally. Given how well-accepted these logical techniques are, we conclude that the T-Schema, in at least one of its forms, is a not just a logical truth but a tautology at that.

In this article, I show that the propositional version of the Tarski T-Schema (Tarski 1933, 1944) is a tautology in the literal sense of the term and then make a few observations about this result. By saying that a schema is a tautology, I mean that all of its instances are tautologies, as this latter concept has been defined in contemporary logic.

Take any standard language for the second-order predicate calculus, modified only so as to include complex  $\lambda$ -expressions. Interpret the  $\lambda$ -expressions *relationally* rather than functionally. The best-known principle governing such  $\lambda$ -expressions is:

\*This article appeared in *Analysis*, 74/1 (2014): 5–11.

<sup>†</sup>I'd like to thank Hannes Leitgeb, Christopher Menzel, Uri Nodelman, and Paul Oppenheimer for encouragement and comments.

**$\lambda$ -Conversion:**  $[\lambda y_1 \dots y_n \varphi]x_1 \dots x_n \equiv \varphi_{y_1, \dots, y_n}^{x_1, \dots, x_n}$ , provided  $x_1, \dots, x_n$  are substitutable, respectively, for  $y_1, \dots, y_n$  in  $\varphi$

This asserts:

Objects  $x_1, \dots, x_n$  exemplify *being a  $y_1, \dots, y_n$  such that  $\varphi$*  if and only if  $x_1, \dots, x_n$  are such that  $\varphi$

In this reading, we've italicized the nominalized predicate, which denotes an  $n$ -place relation.  $\lambda$ -Conversion has 0-place instances as well:

**Propositional Tarski T-Schema:**  $[\lambda \varphi] \equiv \varphi$

To see why we call this schema the Propositional Tarski T-Schema, note several things about the 0-place case of  $\lambda$ -Conversion:

- The expression  $[\lambda \varphi]$  is a 0-place relation term that denotes a 0-place relation (i.e., a proposition), just as the expression  $[\lambda xy \varphi]$  is a 2-place relation term that denotes a 2-place relation.
- The expression  $[\lambda \varphi]$ , however, is also a *formula*. That is why it can stand on the left side of the biconditional sign  $\equiv$ . The simultaneous definition of *term* and *formula* classifies  $[\lambda \varphi]$  as both.<sup>1</sup>
- Since the  $\lambda$  binds no variables in  $[\lambda \varphi]$ , the locution we used for reading the relation term in  $\lambda$ -Conversion, i.e., “being objects  $v_1, \dots, v_n$  such that”, reduces to the locution “that”. So we read the term  $[\lambda \varphi]$  as the proposition-denoting noun phrase *that- $\varphi$* .
- Since *truth* is the 0-place case of *exemplification*, we read the *formula*  $[\lambda \varphi]$  (e.g., when it occurs in the 0-place case of  $\lambda$ -Conversion)

<sup>1</sup> The definition goes as follows:

- Base clause for terms: Every simple term is a term. (Individual constants and variables are individual terms and  $n$ -place relation constants and variables are  $n$ -place relation terms, for  $n \geq 0$ .)
- Base clause for formulas: Where  $\Pi^n$  is any  $n$ -place relation term ( $n \geq 0$ ), and  $\kappa_1, \dots, \kappa_n$  are any individual terms,  $\Pi^n \kappa_1 \dots \kappa_n$  is a formula.
- Recursive clause for formulas: ... (Insert here the usual recursive clauses defining  $\neg\varphi$ ,  $\varphi \rightarrow \psi$ ,  $\varphi \equiv \psi$ ,  $\forall\alpha\varphi$ , etc.)
- Recursive clause for terms: Where  $\varphi$  is any formula and  $v_1, \dots, v_n$  are any individual variables ( $n \geq 0$ ),  $[\lambda v_1 \dots v_n \varphi]$  is an  $n$ -place relation term.

So when  $n = 0$ , the recursive clause for terms yields that  $[\lambda \varphi]$  is a 0-place relation term, and then the base clause for formulas yields that  $[\lambda \varphi]$  is a formula.

as “*that- $\varphi$  is true*” (cf. the reading above of the  $n$ -place case of  $\lambda$ -Conversion).

Given these facts, our 0-place instance of  $\lambda$ -Conversion,  $[\lambda \varphi] \equiv \varphi$ , asserts: *that- $\varphi$  is true iff  $\varphi$* . So, where  $P$  stands for the property *being Prime Minister* and  $c$  for Cameron, the formula  $[\lambda Pc] \equiv Pc$  would be read: *that Cameron exemplifies being Prime Minister is true if and only if Cameron exemplifies being Prime Minister.*<sup>2</sup> Or, using ordinary language: *that Cameron is Prime Minister is true if and only if Cameron is Prime Minister.* This explains why we labeled the schema for the 0-place case of  $\lambda$ -Conversion as the Propositional Tarski T-Schema. Henceforth, I simply call this the T-Schema.

Now one could define a semantic interpretation for the above language and formally show that the T-Schema is logically true (i.e., valid). The expression  $[\lambda \varphi]$  would denote a proposition that is assigned, as its extension, the truth value **The True** whenever the truth conditions of  $\varphi$  obtain. That would validate the T-Schema. As any first-year student of logical metatheory soon discovers, however, to say that the T-Schema is logically true is not to say that it is a tautology. In the predicate calculus, there are valid formulas that are not tautologies. I now present an argument concluding that the instances of the T-Schema are not just valid, but in fact are tautologies. To do this, let me put aside the traditional semantic interpretations and look at how logicians identify *tautologies* for the languages of predicate or modal logic.

Take a classic text, say Enderton 1972. If we ignore, for the moment, the formulas of the form  $[\lambda \varphi]$ , we can adapt Enderton’s definition of *tautology* to our second-order language. Enderton begins by defining the *prime* formulas (106). Intuitively, the prime formulas are treated as the atomic units, akin to sentence letters in a propositional logic.<sup>3</sup> They will be assigned truth values, and then those assignments will be extended to all the formulas of the language. So, adapting his definition to our language, we may say:

$\varphi$  is a *prime* formula if and only if either (a)  $\varphi$  is an atomic formula of the form  $\Pi^n \kappa_1 \dots \kappa_n$  (where  $\Pi^n$  is any relation term,  $n \geq 0$ , and

$\kappa_1, \dots, \kappa_n$  are any object terms) or (b)  $\varphi$  is a quantified formula of the form  $\forall \alpha \psi$  (where  $\alpha$  is any variable).

It is important to remember that since we are temporarily ignoring the 0-place  $\lambda$ -expressions, clause (a) yields that the simple 0-place relation constants and variables are the only prime formulas of the form  $\Pi^0$ . (For reasons that will become apparent below, we do not want 0-place  $\lambda$ -expressions to count as prime formulas.) Then we define:

$\varphi$  is a *non-prime* formula if and only if  $\varphi$  has the form  $\neg\psi$  or  $\psi \rightarrow \chi$ .

Enderton next defines a *truth-functional valuation* for any set of prime formulas  $\Sigma$ :<sup>4</sup>

A *truth-functional valuation* for a set of prime formulas  $\Sigma$  is any function  $v : \Sigma \rightarrow \{\mathbf{T}, \mathbf{F}\}$ .

Then where  $\bar{\Sigma}$  is the set of formulas that can be generated from the prime formulas in  $\Sigma$  using the operations of  $\neg$  and  $\rightarrow$ , Enderton next defines:<sup>5</sup>

For each truth-functional valuation  $v$  for  $\Sigma$ , an *extension*  $\bar{v}$  of  $v$  is any function with domain  $\bar{\Sigma}$  and range  $\{\mathbf{T}, \mathbf{F}\}$  that obeys the following conditions for every formula  $\varphi$  in  $\bar{\Sigma}$ :

- if  $\varphi \in \Sigma$ , then  $\bar{v}(\varphi) = v(\varphi)$
- if  $\varphi = \neg\psi$ , then  $\bar{v}(\varphi) = \begin{cases} \mathbf{T}, & \text{if } \bar{v}(\psi) = \mathbf{F} \\ \mathbf{F}, & \text{otherwise} \end{cases}$
- if  $\varphi = (\psi \rightarrow \chi)$ , then  $\bar{v}(\varphi) = \begin{cases} \mathbf{F}, & \text{if } \bar{v}(\psi) = \mathbf{T} \text{ and } \bar{v}(\chi) = \mathbf{F} \\ \mathbf{T}, & \text{otherwise} \end{cases}$

<sup>4</sup>Actually, Enderton doesn’t do this ‘next’. Rather, when he defines the tautologies of first-order logic, he makes use of definitions constructed much earlier in the book, in the section on propositional logic. So the definition of ‘valuation’ appears on p. 30, where he calls them ‘truth assignments’.

<sup>5</sup>Note the difference between Enderton 1972 and Chellas 1980. Chellas defines truth-functional valuations as functions that assign truth values to *all* the prime (i.e., propositionally atomic) formulas of the language. Hence, the extensions of such valuations assign truth values to every formula of the language. Enderton’s method of defining valuations relative to arbitrary sets of prime formulas has the virtue that it plays nicely into the construction of an effective procedure for determining whether an arbitrary formula  $\varphi$  is a tautology. A formula  $\varphi$  has a finite number  $n$  of prime subformulas, and so there will be only  $2^n$  valuations to check when implementing the effective procedure. An effective procedure based on Chellas’ method, however, requires that one partition all the valuations into a finite number of equivalence classes, each class containing all the valuations that agree on the prime formulas of  $\varphi$ .

<sup>2</sup>Here I’ve italicized the nominalization that constitutes the subject term of the left condition.

<sup>3</sup>Chellas 1980 (8–9) similarly introduces the notion of *propositionally atomic* formulas to define the notion of tautology in modal logic.

In other words, an extension  $\bar{v}$  of  $v$  must agree with  $v$  on the formulas in  $\Sigma$ , and when  $\varphi$  is a formula built up from the formulas in  $\Sigma$  using the truth-functional connectives  $\neg$  and  $\rightarrow$ ,  $\bar{v}$  assigns to  $\varphi$  a truth value in just the way one assigns truth values to such complex formulas in a truth table. Note that no matter whether you define the biconditional in the usual way, or take it as primitive and allow  $\bar{\Sigma}$  to be the set of formulas that can be generated from the prime formulas in  $\Sigma$  using the operations of  $\neg$ ,  $\rightarrow$  and  $\equiv$ , the following clause governs the behavior of  $\bar{v}$ :

$$\bullet \text{ if } \varphi = (\psi \equiv \chi), \text{ then } \bar{v}(\varphi) = \begin{cases} \mathbf{T}, & \text{if } \bar{v}(\psi) = \bar{v}(\chi) \\ \mathbf{F}, & \text{otherwise} \end{cases}$$

(And so on for  $\&$  and  $\vee$ .) Note that any formula  $\varphi$  can be regenerated from prime formulas using the connectives  $\neg$  and  $\rightarrow$  (given how the other connectives can be defined in terms of these two). For example, if  $\varphi = Pa \rightarrow (\neg\forall xQx \rightarrow Pa)$ , then the prime formulas in  $\varphi$  are  $Pa$  and  $\forall xQx$ . So we can regenerate  $\varphi$  by applying  $\neg$  to  $\forall xQx$  to obtain  $\neg\forall xQx$ , applying  $\rightarrow$  to the latter and  $Pa$  to obtain  $\neg\forall xQx \rightarrow Pa$ , and applying  $\rightarrow$  to  $Pa$  and  $\neg\forall xQx \rightarrow Pa$  to obtain  $\varphi$ . Thus, if  $v$  is a truth-functional valuation for  $\Sigma$ , where  $\Sigma$  is the set of prime formulas in  $\varphi$ , then  $\varphi$  is a member of  $\bar{\Sigma}$  and so will be assigned a truth value by  $\bar{v}$ .

Enderton then defines:  $\varphi$  is a *tautology* if and only if for every valuation  $v$  of the prime formulas in  $\varphi$ ,  $\bar{v}(\varphi) = \mathbf{T}$ .<sup>6</sup> Sure enough, one can prove that classical tautologies satisfy this definition. The reader should try it on predicate calculus formulas such as  $\forall xPx \rightarrow \forall xPx$ ,  $\forall xPx \vee \neg\forall xPx$ , etc.

Now, I claim, the proper way to extend this classical technique to our second-order language with 0-place  $\lambda$ -expressions, is to treat the context  $[\lambda \dots]$  as a truth-functional connective, in the same way that we treat  $\neg$ ,  $\rightarrow$ ,  $\equiv$ , etc., as truth-functional connectives. To make this happen, we redefine:

$\varphi$  is a *prime* formula iff  $\varphi$  is either (a) an atomic formula of the form  $\Pi^n \kappa_1 \dots \kappa_n$  ( $n \geq 1$ ), or (b) a simple 0-place relation term (i.e., a 0-place relation constant or variable), or (c) a formula of the form  $\forall \alpha \psi$ .

<sup>6</sup>Actually, Enderton defines, for an arbitrary set  $\Gamma$  of formulas, the concept  $\Gamma$  *tautologically implies*  $\varphi$  (1972, 33). But the definition of tautological implication reduces to the definition of *tautology* just given as the special case where  $\Gamma$  is the null set.

$\varphi$  is *non-prime* if and only if  $\varphi$  has the form  $\neg\psi$ ,  $\psi \rightarrow \chi$ , or  $[\lambda \psi]$ .

In the above redefinition of *prime*, it is important not to collapse clauses (a) and (b) by allowing  $n = 0$  in clause (a), for that would make the definitions of *prime* and *non-prime* inconsistent: such a collapse would count any complex 0-place relation term  $[\lambda \psi]$  as a prime formula, since such terms are instances of the metavariable  $\Pi^0$  (recall footnote 1). The formula  $[\lambda \psi]$  is stipulated, along with  $\neg\psi$  and  $\psi \rightarrow \chi$ , to be non-prime.

Now if we let valuations  $v$  be defined as before, relative to a set of prime formulas  $\Sigma$ , we need to add only one clause to the definition of an extension  $\bar{v}$  of  $v$ :

$$\bullet \text{ if } \varphi = [\lambda \psi], \text{ then } \bar{v}(\varphi) = \begin{cases} \mathbf{T}, & \text{if } \bar{v}(\psi) = \mathbf{T} \\ \mathbf{F}, & \text{otherwise} \end{cases}$$

Clearly, this treats  $[\lambda \dots]$  as a truth functional connective: when  $\varphi$  is assigned the value  $\mathbf{T}$ ,  $[\lambda \varphi]$  is assigned the value  $\mathbf{T}$ , and when  $\varphi$  is assigned the value  $\mathbf{F}$ ,  $[\lambda \varphi]$  is assigned the value  $\mathbf{F}$ .

Now if we keep the *same* definition of tautology as the one introduced above, then (a) we get a new class of tautologies, and (b) the T-Schema becomes a tautology. To see (a), let  $\varphi$  be an arbitrary formula and consider  $[\lambda \varphi] \rightarrow \varphi$ , which asserts: if *that- $\varphi$*  is true, then  $\varphi$ . This formula, unlike  $\varphi \rightarrow \varphi$ , is *not* a traditional form for a tautology. To see that it is a tautology, pick any valuation  $v$  of the prime formulas in  $[\lambda \varphi] \rightarrow \varphi$  and consider its extension  $\bar{v}$ . No matter what prime formulas are in  $[\lambda \varphi] \rightarrow \varphi$ , we may reason as follows:

- if  $\bar{v}(\varphi) = \mathbf{T}$ , then  $\bar{v}([\lambda \varphi] \rightarrow \varphi) = \mathbf{T}$ , by the constraint on  $\bar{v}$  for conditionals
- if  $\bar{v}(\varphi) = \mathbf{F}$ , then  $\bar{v}([\lambda \varphi]) = \mathbf{F}$ , by the constraint on  $\bar{v}$  for  $[\lambda \dots]$ , and so  $\bar{v}([\lambda \varphi] \rightarrow \varphi) = \mathbf{T}$ , by the constraint on  $\bar{v}$  for conditionals

So no matter whether  $\bar{v}(\varphi)$  is  $\mathbf{T}$  or  $\mathbf{F}$ , the truth value assigned to  $[\lambda \varphi] \rightarrow \varphi$  is  $\mathbf{T}$ . Since our reasoning started from the assumption that  $v$  was arbitrary, our conclusion holds for any  $v$ , and so  $[\lambda \varphi] \rightarrow \varphi$  is a tautology. Since  $\varphi$  was arbitrarily chosen, the schema  $[\lambda \varphi] \rightarrow \varphi$  is a tautology.

To see (b), that the T-Schema is a tautology, note that the reasoning we just gave generalizes to  $[\lambda \varphi] \equiv \varphi$ , for an arbitrarily chosen  $\varphi$ . To see that this is a tautology, pick any valuation  $v$  of the prime formulas in  $[\lambda \varphi] \equiv \varphi$  and consider its extension  $\bar{v}$ . No matter what prime formulas are in  $[\lambda \varphi] \equiv \varphi$ , we may reason as follows:

- if  $\bar{v}(\varphi) = \mathbf{T}$ , then  $\bar{v}([\lambda \varphi]) = \mathbf{T}$ , by the constraint on  $\bar{v}$  for  $[\lambda \dots]$ , and so  $\bar{v}([\lambda \varphi] \equiv \varphi) = \mathbf{T}$ , by the constraint on  $\bar{v}$  for biconditionals
- if  $\bar{v}(\varphi) = \mathbf{F}$ , then  $\bar{v}([\lambda \varphi]) = \mathbf{F}$ , by the constraint on  $\bar{v}$  for  $[\lambda \dots]$ , and so  $\bar{v}([\lambda \varphi] \equiv \varphi) = \mathbf{F}$ , by the constraint on  $\bar{v}$  for biconditionals

So no matter whether  $\bar{v}(\varphi)$  is  $\mathbf{T}$  or  $\mathbf{F}$ , the truth value assigned to  $[\lambda \varphi] \equiv \varphi$  is  $\mathbf{T}$ . Since  $v$  was arbitrary, our conclusion holds for every  $v$  and so  $[\lambda \varphi] \equiv \varphi$  is a tautology. Since  $\varphi$  was arbitrarily chosen, the schema  $[\lambda \varphi] \equiv \varphi$  is a tautology.

So the T-Schema  $[\lambda \varphi] \equiv \varphi$  is literally a tautology. Since tautologies are valid (i.e., logical truths), but not all logical truths are tautologies, our conclusion shows that the T-Schema falls within an even narrower class of logical principles than the logical truths. Now in a famous passage, Frege said (1918, 61):

It is also worth noticing that the sentence ‘I smell the scent of violets’ has just the same content as the sentence ‘it is true that I smell the scent of violets’. So it seems, then, that nothing is added to the thought by my ascribing to it the property of truth.  
(1984 translation, 354)

Of course, Frege didn’t have the concept of *tautology* as we know it today, though it looks as though Frege might have said that if ‘*that- $\varphi$*  is true’ and ‘ $\varphi$ ’ have identical content, then ‘*that- $\varphi$*  is true iff  $\varphi$ ’ is a tautology. More recently, some have argued that the *sentential* version of the T-Schema is logically true (Priest 2007, 193), while others have argued that it is not (Cook 2012, 235–236).<sup>7</sup> To the best of my knowledge, however, no one has produced an argument to the conclusion that the Tarski T-Schema has a reading on which it is a *tautology*.

## Bibliography

- Chellas, B. 1980. *Modal Logic: An Introduction*. Cambridge: Cambridge University Press.
- Cook, R. 2012. The T-schema is not a logical truth. *Analysis* 72(2): 231–239.
- Enderton, H. 1972. *A Mathematical Introduction to Logic*. San Diego: Academic Press.
- Frege, G. 1918. Der Gedanke. Eine Logische Untersuchung. In *Beiträge zur Philosophie des deutschen Idealismus*, 1: 58–77. Translated as “Thoughts,” by P. Geach and R. Stoothoff, in *Gottlob Frege: Collected Papers on Mathematics, Logic, and Philosophy* (1984) ed. B. McGuinness, 351–372. Oxford: Blackwell.
- Priest, G. 2007. Paraconsistency and dialetheism. In *Handbook of the History of Logic* (Volume 8: The Nonmonotonic Turn in Logic), eds. Dov M. Gabbay and John Woods, 129–204. Amsterdam: North Holland.
- Tarski, A. 1933. “Pojęcie prawdy w językach nauk dedukcyjnych”, *Prace Towarzystwa Naukowego Warszawskiego, Wydział III Nauk Matematyczno-Fizycznych*, 34, Warsaw. Translated as “The concept of truth in formalized languages,” by J.H. Woodger, in *Logic, Semantics, Metamathematics*, 2nd edition (1983) ed. John Corcoran, 152–278. Indianapolis: Hackett.
- Tarski, A. 1944. The semantic conception of truth. *Philosophy and Phenomenological Research* 4(3): 341–376.

<sup>7</sup>The work in this article is immune to the argument in Cook 2012. His argument is based on a principle of Logical Substitutivity, which asserts the preservation of logical truth under substitution of non-logical constants. But our  $\lambda$ -operator for forming terms that denote propositions is a *logical* constant, given that it is interpreted as a truth-functional operator. The resulting terms of the form  $[\lambda \varphi]$  are therefore *not* non-logical singular terms but rather logical (0-place) general terms. So Cook’s principle of Logical Substitutivity doesn’t apply. Moreover, as noted, Cook’s argument concerns the sentential version of the T-Schema, which governs a truth-predicate of (names of) sentences, and doesn’t necessarily apply to the present version, which governs the truth of propositions.