Re: Exception to the rule? (Tarski´s T−scheme)

Source: http://sci.tech−archive.net/Archive/sci.logic/2004−06/1950.html

---

From: Andrew Boucher (*Helene.Boucher_at_wanadoo.fr*)
Date: 06/25/04

```
Date: 25 Jun 2004 14:58:11 −0700
```

"Jeffrey Ketland" <ketland@ketland.fsnet.co.uk> wrote in message
news:<cbhaqgrcd1@newsg3.svr.pol.co.uk>...
> Andrew Boucher wrote in message ...
> >"Jeffrey Ketland" <ketland@ketland.fsnet.co.uk> wrote in message
> news:<cb9pniqjp1@news5.svr.pol.co.uk>...
> >> Paul Holbach wrote in message
> >> <881c8779.0406211820.43c78402@posting.google.com>...
> >> >Let´s consider Tarski´s famous T−scheme:
> >> >
> >> >True("p") <−> p
> >> >
> >> >Now what about the statement "Nothing exists"?
> >> >
> >> >True("Nothing exists") <−> Nothing exists
> >> >
> >> >Truth is a property of statements, and if nothing exists, there aren´t
> >> >any statements either. The point is that nonexistent statements are
> >> >neither true nor false, and so it is not the case that "Nothing
> >> >exists" is true iff nothing exists.
> >>
> >> The T−scheme implies the existence of at least two things. In particular,
> a
> >> syntactic item A must be distinct from its negation ~A. (For each item,
> the
> >> T−scheme implies "~A is true if and only if A is not true", so "A = ~A"
> >> would be inconsistent with the T−scheme.)
> >> If one considers a model with at least two objects−−−and preferably one
> >> where all syntactic items are elements of the domain−−−then the relevant
> >> restricted T−scheme can be made *true* (including the instance using
> >> "~Ex(x=x)").
> >>
> >> But this merely tells us that the T−scheme itself implies that something
> >> exists. This is no surprise, since its instances refer to syntactical
> items.
> >
> >Just to clarify (?) this point.
> >

> >*The T−schema is like a universal generalization: you*
> >*can substitute any proposition you want in for p, but you still have to*
> >*have one to substitute.*
>
> *The T−scheme is not a universal generalization.*

Agreed. It's not. It's (as I said) *like* a universal generalization. If nothing existed, you wouldn't be able to substitute anything in for x in the universally generalized (x)Ax. Similarly, if no sentence existed, then you couldn't apply the T−scheme.

> *It is a scheme, whose*
> *instances are obtained by substituting formulas and quotation names of*
> *formulas. By definition, for a given language L (containing the symbol Tr*
> *and quotation names of all L sentences), it is the set of all formulas*
> *Tr("A") <−> A, where A is a sentence of L and "A" is its quotation name.*

And this set would be empty unless you have a sentence of L, i.e. unless you already know there exists one sentence. As I said, you have to get the existence of a sentence from elsewhere. It doesn't come from the T−schema. The most you can say is that the truth schema implies that, if there is one sentence, there are two things. And even that doesn't seem to be enough for your argument to go through. You need to know that, if there is one sentence, then this sentence also has a negation. It's not the T−schema which gives you that.

> *I.e.,*
>
> *T−Scheme for L = {Tr("A") <−> A : A is a sentence of L}*
>
> >*It therefore does not imply the existence of any*
> >*p, nor for that matter the existence of any two p.*
>
> *It implies the existence of at least two sentences. E.g., that the sentence*
> *"0=0" is distinct from "~0=0".*

It implies that a sentence is distinct from its negation. It does not imply that "0=0" or "not 0 = 0" exist. This you have to get from elsewhere, e.g. the rules of syntax of the language.

>
> >*The existence of p (and*
> >*the existence of "not p" given the existence of p) must come elsewhere,*
> >*from e.g. assumptions about what propositions can be formulated.*
>
> *There is no "p" in the T−scheme.*

Call it A, if you prefer. Give me a break! Are you saying no one has used the letter "p" in formulating the T−scheme, that everyone uses the letter "A" so get with it? Or is the fact that I put "not p" in

quotes confusing you? Should I have used (not p) instead? Anyway, to
use the letter A (which is obviously the only letter to use), we get:
The existence of A (and the existence of (not A) given the existence
of A) must come [from] elsewhere, from e.g. assumptions what sentences
can be formulated in the language.

>That is a (common) use/mention confusion.

I was using proposition as you are using sentences. (Proposition as
in "propositional calculus".) I'm sorry that I have confused you. My
apologies.

> The T−scheme doesn't have anything to do with propositions. Neither Tarski
> nor Quine believed in propositions. It refers to sentences, syntactical
> items.

I understand that.

> Tarski and Quine are extremely clear on this point.

Good!

>
> >In brief, the content of the T−schema implies only that a proposition
> >cannot be identical to its negation.
>
> The T−scheme doesn't even talk about propositions. It talks about
> syntactical items: sentences.

Okay then, change it to: In brief, the content of the T−schema
implies only that a sentence cannot be identical to its negation.
It's only if you go the Cartesian route that you can say that the
T−schema implies the existence of a (one) sentence, e.g. concluding
that, because you used a sentence to describe the T−schema, at least
one sentence exists.

> You seem to be worried about whether quotation names of expressions are
> names or not.

Nope.

>In classical logic, quotation names are names, and one can
> apply existential generalization.

>
> Do you agree that
>
> (i) "Snow" contains four letters
> implies
> (ii) There is something which contains four letters

Yes. But, unless I'm missing something, that doesn't seem to be
relevant to what I'm saying.

>
> *In ordinary logic, this is valid, since (i) contains a quotation name. (A*
> *quotation name denotes the sentence quoted.)*
> *Tarski understood "Snow" to be the concatenation of the letter "S", then the*
> *letter "n", then the letter "o", then the letter "w". I.e., in syntax,*
>
> *"Snow" is identical to "S" ^ ("n" ^ ("o" ^ "w")).*
>
> *So, for Tarski, (i) is equivalent to*
>
> *(i)* The concatenation "S" ^ ("n" ^ ("o" ^ "w")) contains four letters*
>
> *Goedel showed how to model concatenations as finite sequences of*
> *expressions, and how to represent finite sequences as numbers. E.g., "Snow"*
> *is the finite sequence ("S", "n", "o", "w").*
>
> *−−− Jeff*