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

Source: http://sci.tech−archive.net/Archive/sci.logic/2004−06/1923.html

---

From: Jeffrey Ketland (*ketland_at_ketland.fsnet.co.uk*)
Date: 06/25/04

```
Date: Fri, 25 Jun 2004 14:50:22 +0100
```

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. 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.
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".

>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. That is a (common) use/mention confusion.
The T−scheme doesn't have anything to do with propositions. Neither Tarski
nor Quine believed in propositions. It refers to sentences, syntactical
items. Tarski and Quine are extremely clear on this point.

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

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