29

Truth and Disquotation

I. INTRODUCTION

The reader is familiar, I shall assume, with Tarski's method of defining truth.[1] Let me nevertheless sum up, for later reference, some of its significant traits. The goal is a truth predicate, 'is true', that is defined for the general context 'x is true', but defined in such a way as to fulfill the following schema whenever applied to the quotation of an actual sentence:

(1) '. . . . . ' is true ≡ . . . . . .

Tarski shows how to achieve this for any formalized language whose logical form is the classical logic of quantification. Most of the construction is an inductive definition of satisfaction, conceived as a relation of sequences to open or closed sentences. This inductive definition consists of a finite lot of sentences couched in a metalanguage that is an extension of the object language. The metalanguage contains, in addition to the object language, a name of each sign of the object language; also a functor whereby to refer to arbitrary complex expressions of the object language as functions of their components. Also it contains apparatus for referring to sequences of objects and recovering the occupants of their successive places. Also it contains the

Talks with Donald Davidson prompted my renewed preoccupation with Tarski's theory of truth and did much to set the lines of the present study. In 1970 I submitted the paper as an advance contribution to the symposium that was held in honor of Tarski at Berkeley in June 1971. It is reprinted with permission of the American Mathematical Society from *Proceedings of Symposia in Pure Mathematics*, vol. 25, pp. 373–384, © 1974. Portions are omitted in favor of cross-references.

[1]For a quick account see Chap. 3 of my *Philosophy of Logic*.

definiendum itself, the two-place predicate of satisfaction; for the definition, being inductive, does not in general make for elimination of its definiendum in the presence of variable arguments. Or, as Tarski observes, we can render the satisfaction predicate eliminable by furnishing the metalanguage with as much of the machinery of set theory as is needed for turning the inductive definition into a direct one.

Tarski's project was directed upon deductive systems, and he required that each instance of his schema (1) be deducible from the truth definition. But I shall be concerned with interpreted languages without regard to axiomatization, so I require only that the truth definition make the instances of the schema (1) come out true.[2]

The sequences to which Tarski's definition appeals are infinite sequences. There is a way of making do with finite sequences, but still it appeals to those sequences in their infinite generality, without limit of length. Moreover, one among the clauses of Tarski's inductive definition stands forth as more devious and complex than the rest; namely, the clause that copes with quantification. Both of these complications in Tarski's definition are occasioned by the apparatus, in the object language, of quantification and variables. One's thoughts consequently turn to other styles of logic, in which the work of quantifiers and variables is managed by constants. Will such systems open a shorter avenue to the truth predicate? I propose to explore this question in connection with Schönfinkel's combinatory logic,[3] which is a language of full set-theoretic strength, and in connection also with what I call predicate-functor logic, which is no stronger than the ordinary logic of quantification and identity. For Schönfinkel's language we get something that may be called *disquotation*—and in a stronger sense of the word than what Tarski's schema (1) above requires. We shall observe, further, why a general inductive definition of disquotation for arbitrary notations does not go through similarly. Finally, reverting to quantification of

[2]I am indebted to John Wallace for prompting this caveat.

[3]See §§II–IV of the preceding essay. But I now depart from the notation there used for functional application, namely the inverted comma, in favor of the form of notation '$f(x)$'. I do so to avoid confusion with quotation marks, and also so as not to have to add rules for the use of parentheses in grouping. These considerations matter now because of a need to be explicit about metalogical maneuvers.

a sort, we shall consider what happens to the truth definition when quantification is construed in terms of substitution rather than objective reference.

II. TRUTH FOR SCHÖNFINKEL'S LANGUAGE

The inductive truth definition for this object language will be couched in a metalanguage that contains the object language and contains in addition only the definiens ('$\triangle$', below) and a notation for naming the expressions of the object language. The names of the letters 'S', 'C', and 'ι' will be formed, as just here, by direct quotation. The form of expression '$\mathrm{ap}(x, y)$' will be used in the metalanguage to refer to the complex name that is formed by functional application from the respective names x and y in the object language. In other words, $\mathrm{ap}(x, y)$ consists of x followed by y in parentheses. Thus

$$\mathrm{ap}(\text{`}S\text{'}, \text{`}C\text{'}) = \text{`}S(C)\text{'}.$$

Tarski defined the truth of a closed sentence as a special or limiting case of satisfaction. A comparable indirectness is called for here, but now the intermediate concept to be inductively defined is that of the *designatum* rather than satisfaction. I shall write '$\triangle(x)$' for the designatum of x, the thing named by the formula x. The aim of the definition is to assure that every equation of the form

$$\triangle(\text{`}\ldots\ldots\text{'}) = \ldots\ldots, \tag{2}$$

with any one name in the blanks, come out true. This is achieved by an inductive definition which, in a preliminary rendering, runs simply thus:

$$\triangle(\text{`}S\text{'}) = S, \quad \triangle(\text{`}C\text{'}) = C, \quad \triangle(\text{`}\iota\text{'}) = \iota, \tag{3}$$

$$\triangle(\mathrm{ap}(x, y)) = \triangle(x)(\triangle(y)). \tag{4}$$

This already defines truth. For, truth is the special case of $\triangle$ where the argument is a sentence. Where x is a sentence, '$\triangle(x)$' amounts to 'x is true'. This is evident from the schemata (1) and (2) when we keep in mind that sentences now are names, names of T and $\mathsf{\bot}$.

I called (3)–(4) a preliminary rendering because it is couched in an excessive idiom. We see these further symbols that are foreign to the announced metalanguage: '$=$', 'x', 'y', and the comma of 'x, y'. I shall dispose of the comma first. Schönfinkel explained two-place functions in Frege's way: he explained a two-place function f as the one-place function which, applied to anything x, yields as value $f(x)$ a one-place function which, applied to anything y, yields as value the desired $f(x, y)$. In short, $f(x, y)$ is $f(x)(y)$. We may render '$\mathrm{ap}(x, y)$' accordingly. (4) becomes

$$\triangle(\mathrm{ap}(x)(y)) = \triangle(x)(\triangle(y)). \tag{5}$$

The variables 'x' and 'y' will next be disposed of. It is known that Schönfinkel's S and C are *combinatorially complete*, in this sense: any desired permutation and regrouping of any of the terms in a formula beyond the first, and any fusing of recurrences, can be achieved by applying to the first term some function compounded purely of S and C. In particular we can express, purely in terms of S, C, and functional application, two functions Φ and Ψ that have the following effects:

$$\Phi(\triangle)(\mathrm{ap})(x)(y) = \triangle(\mathrm{ap}(x)(y)).$$

$$\Psi(\triangle)(x)(y) = \triangle(x)(\triangle(y)).$$

Accordingly (5) becomes

$$\Phi(\triangle)(\mathrm{ap})(x)(y) = \Psi(\triangle)(x)(y).$$

But to affirm this for all x and y is simply to identify the functions thus:

$$\Phi(\triangle)(\mathrm{ap}) = \Psi(\triangle).$$

Finally, we know how to translate '$=$' into terms of ι; '$x = y$' becomes '$\iota(x)(y)$'. So the above equation and those in (3) go over into full Schöfinkel style thus:

$$\iota(\triangle(\text{`}S\text{'}))(S),\ \iota(\triangle(\text{`}C\text{'}))(C),\ \iota(\triangle(\text{`}\iota\text{'}))(\iota),\ \iota(\Phi(\triangle)(\mathrm{ap}))(\Psi(\triangle)). \tag{6}$$

For readers conversant with Schönfinkel it would be a routine exercise to expand 'Φ' and 'Ψ' appropriately into terms of S and C. So now the four clauses of the inductive definition of designatum are couched completely in the signs of the object language plus their quotations and 'ap' (and of course the definiendum itself). The version (3)–(4), however, is the one to turn to whenever perspicuity is in point.

III. CONTRAST WITH TARSKI'S CONSTRUCTION

A certain similarity was noted between the role of designatum in the above definition of truth and the role of satisfaction in Tarski's. Both are defined inductively, as a means of defining truth. In both constructions, truth falls out at the end as a special case. But there are notable differences. One difference is that truth is a special case more directly of designatum than of satisfaction.

A more conspicuous contrast between the two constructions is seen in Tarski's dependence on an apparatus of sequences, and in the complexity of his recursion clause regarding quantifiers (see §I above). The striking simplicity of the present inductive definition of designatum and truth for combinatory logic is due to the freedom of this object language from variables and quantifiers.

Expressions that are built up in the metalanguage by applying 'ap' to quoted letters may in a broad sense be called *quotations*. They are, in Tarski's phrase, structural-descriptive equivalents of quotations. E.g.,

$$\text{ap}(\text{ap}(\text{`}\iota\text{'}, \text{`}S\text{'}), \text{`}S\text{'}) = \text{`}\iota(S)(S)\text{'}.$$

(For perspicuity I revert to the style of 'ap(x, y)' now that (6) is finished.) Now an interesting third point of contrast between the designatum approach to truth and Tarski's approach is that the designatum approach renders truth as a direct *disquotation*. That is, if you attribute truth to a sentence by attaching the truth predicate (or '$\triangle$') to a quotation of the sentence, and then you eliminate '$\triangle$' step by step according to the inductive definition (6) (or, more intuitively, (3)–(4)), you come out with the very sentence that had been quoted and not just some equivalent.

Thus take the above example. We want to say that '$\iota(S)(S)$' is true. That is,

$$\triangle(\text{ap}(\text{ap}(\text{`}\iota\text{'}, \text{`}S\text{'}), \text{`}S\text{'})). \tag{7}$$

Now let us unwind this according to the definition (3)–(4). By (4) we reduce (7) successively thus:

$$\triangle(\text{ap}(\text{`}\iota\text{'}, \text{`}S\text{'}))(\triangle(\text{`}S\text{'})), \qquad \triangle(\text{`}\iota\text{'})(\triangle(\text{`}S\text{'}))(\triangle(\text{`}S\text{'})),$$

and this reduces by (3) to:

$$\iota(S)(S).$$

If we work not with the inductive definition (3)–(4) but with its equivalent (6) in Schönfinkel style, the unwinding of (7) will depend in part upon laws governing the Φ and Ψ that were used in (6). Those laws would come down ultimately to the logical laws governing S and C. So, when I say that the inductive definition (6) renders truth as direct disquotation, I do not deny the need of logical transformations in the unwinding. My point is rather this: as soon as the unwinding has done its job of eliminating the last metalinguistic sign so that only a formula of the object language remains, that formula will be literally the formula that was quoted in the first place and not just an equivalent.

This disquotational property is stronger than what is called for by Tarski's classical schema (1), and it is not preserved under Tarski's inductive definition of satisfaction. What we usually come out with under Tarski's truth definition is not literally the sentence to whose quotation the truth predicate had been attached, but another sentence that is equivalent to it under the logical laws of quantification and identity.

The direct disquotational character that we have observed in our truth construction holds, of course, for our designatum construction generally. It matters none whether the formula whose quotation is appended to '$\triangle$' is a sentence or is a name of a function; the effect is just to disquote the quoted formula, whatever it is. Take, say, '$S(C(\iota)(S))$'; that is,

$$\text{ap}(\text{`}S\text{'}, \text{ap}(\text{ap}(\text{`}C\text{'}, \text{`}\iota\text{'}), \text{`}S\text{'})).$$

Apply $\triangle$:

$$\triangle(\text{ap}(\text{`}S\text{'}, \text{ap}(\text{ap}(\text{`}C\text{'}, \text{`}\iota\text{'}), \text{`}S\text{'}))). \tag{8}$$

Proceeding then to unwind by (3) and (4), we recover precisely the formula originally quoted:

$$S(C(\iota)(S)). \tag{9}$$

This is obvious from (3) and (4). But it is not obvious in the way one imagines who might say, "Of course, that is what 'designatum' means: the very formula that was named." That would be a confusion. Any version of designatum would be worthy of the name as long as it fulfilled the schema (2); and it could fulfill (2) even though its definition were to unwind (8) not into (9)

but into the different expression '$S(\iota)$'. For it happens that $S(\iota)$ and $S(C(\iota)(S))$ are the same object, the same function, since $C(x)(y) = x$. This thing $S(\iota)$ *is* the designatum of

$$\text{ap}(\text{`}S\text{'}, \text{ap}(\text{ap}(\text{`}C\text{'}, \text{`}\iota\text{'}), \text{`}S\text{'})),$$

that is, of '$S(C(\iota)(S))$', just as genuinely as is $S(C(\iota)(S))$; for $S(\iota)$ is $S(C(\iota)(S))$. But the special point about the definition (3)–(4) of designatum is that it unwinds (8) directly and literally into (9) rather than into '$S(\iota)$', and that it unwinds '$\triangle$(ap('S', 'ι'))' directly and literally into '$S(\iota)$' rather than into (9).

IV. DISQUOTATION IN THE GENERAL CASE

Thus '$\triangle$' emerges as a disquotation operator. To this trait we are indebted for two advantages that the designatum approach has been seen to have over the satisfaction approach: the avoidance of sequence theory, and the avoidance of the complex recursion condition regarding quantification. To what extent are these benefits tied to the Schönfinkel style of language? Can we define disquotation for languages more generally?

At first it seems so. Consider an arbitrary language. Suppose its signs are the Greek letters. To define a general disquotation operator 'disq' for this language inductively, we begin with twenty-four definitions explaining the notations:

$$\text{disq alpha}, \text{disq beta}, \ldots, \text{disq omega}$$

respectively as:

$$\alpha, \beta, \ldots, \omega,$$

and then, using Tarski's arch symbol of concatenation, we provide for recursion by explaining 'disq $x^\frown y$' in general as 'disq x disq y'.

Disquotation for Schönfinkel's formulas took the form of a designatum function $\triangle$, since all his formulas are names—whether of functions or of truth values. I have had now to write 'disq' instead of '$\triangle$', because the expressions of our unspecified Greek-letter language are not known to be names. For the same reason, I am unable to render the inductive definition of 'disq' by the equations:

$$\text{disq alpha} = \alpha, \ldots, \text{disq omega} = \omega,$$

$$\text{disq } x^\frown y = \text{disq } x \text{ disq } y$$

since, failing some special convention, it is incoherent to put '=' between expressions other than singular terms. For the same reason we can write no analogue of the schema (1) or (2) for disquotation generally; '=' is not available, nor '≡'.

Note that 'alpha', 'beta', 'alpha^beta', etc., are indeed names or singular terms in good standing in the metalanguage; and 'disq' is an operator on such terms. But it is an odd one, yielding in general neither a term nor a sentence as output. Some of the expressions 'α', 'β', '$\alpha\beta$', etc., that are named by 'alpha', 'beta', 'alpha^beta', etc., can be gibberish.

We begin to see the obstacle to such a general inductive definition of disquotation. We may see the obstacle more clearly if, to begin with, we consider how far we can proceed unobstructed. For definitions in the narrow sense, the genuinely eliminative definitions, any medium of presentation is of course welcome as long as it gives an effective procedure for transforming the defined sign or its contexts into previous notation. Thus the account of 'disq' three paragraphs back satisfactorily defines 'disq' in application to all singular cases, however long; all constant spellings. For it gives an effective procedure for eliminating 'disq' from any such context. It explains 'disq alpha^kappa^rho^omicron^nu' as '$\alpha\kappa\rho o\nu$', and no matter whether this string of signs is a name or a sentence or an incoherency in the imagined language.

But that account of 'disq' is no good as an inductive definition of 'disq x' for variable 'x'. For consider what is wanted of such a definition. One way in which its clauses may be used is as axioms governing 'disq' but not eliminating it. For such axiomatic use the clauses must have the explicit form of sentences in the metalanguage. Or perhaps the inductive definition is destined to be turned into a direct and eliminative definition of 'disq x', by recourse to a sufficiently strong set theory. But a direct definition so obtained incorporates those clauses as component sentences. Thus for either purpose the inductive definition would have to use 'disq' in some grammatical position in sentences. But in the general case this is impossible. In general 'disq' attaches to a singular term uniformly enough, but the trouble is that the grammatical category of the resulting compound may be any or none, depending as it does on the reference of that singular term.

V. PREDICATE FUNCTORS

Schönfinkel's language lent itself to an inductively well-defined and generally applicable disquotation operator '△'. One conspicuous trait of that language is the absence of variables. Another trait, which contributed to the success of the definition, is that each formula of Schönfinkel's language is a name of something. One is disinclined, however, to rest with Schönfinkel's language. One is put off by its excessive power. It is adequate for general set theory, and is heir to all the problems raised by the antinomies of set theory.

Tarski's inductive definition of satisfaction, and therewith of truth, is geared to any system couched in the framework of the classical logic of quantification; it is not reserved to strong object languages. So a natural next thought is to see how disquotation might fare in what I call *predicate-functor* logic.[4] For this is no stronger than the classical logic of quantification and identity; indeed it is intertranslatable with that. And at the same time it resembles Schönfinkel's logic in dispensing with variables. Its way of dispensing with them even resembles Schönfinkel's, up to a point.

In Schönfinkel's language the well-formed formulas were names. They named functions or, at the extreme, truth values. Hence the disquotation functor took the form there of the name of a designatum function. In the predicate-functor language, on the other hand, the well-formed formulas are n-place predicates or, at the extreme ($n = 0$), sentences. Here, consequently, disquotation will take the form of a functor 'sat' of satisfaction. It differs in category from Tarski's satisfaction, which is a relation. The functor 'sat' attaches to a name, or singular term, to form a predicate. In interesting cases, that singular term names a predicate.

EXAMPLE.

sat 'is human'.

The complex predicate thus formed may be read:

satisfies 'is human'

[4]See preceding essay, last four sections.

and is hence meant to be coextensive with the originally named predicate itself:

is human.

Since singular terms are foreign to predicate-functor logic, the functor 'sat' will require some adapting if it is to fit into a metalanguage that is a direct extension of this object language. In predicate-functor logic the role of singular terms is played by predicates. Instead of having a singular term that names an object x, we make do with a predicate that is true solely of x. Adapted accordingly, 'sat' becomes a predicate functor; where 'F^1' is a predicate that is true solely of a predicate 'G^n', 'sat F^1' becomes an n-place predicate coextensive with 'G^n'.

As George Myro has pointed out to me, however, trouble remains. We cannot determine the degree of 'sat F^1' without some prior knowledge as to what 'F^1' is true of. Consequently we cannot subject 'sat F^1' to predicate-functor logic; for it is evident from the explanation of the permutation functor, in particular, that the logical transformations can hinge upon the degree of a predicate.

Nor is the difficulty resolved by retreating to a metalanguage of ordinary quantificational form. Here 'sat' would revert to the status of a predicate-yielding functor upon singular terms; 'sat x' would become, for each predicate x, a predicate coextensive with x. Its degree would depend on that of x and would thus be indeterminate for variable 'x'. The ordinary logic of quantification has no place for a predicate symbol 'sat x' of variable degree.

Truth *can* be defined for a predicate-functor style of language in a predicate-functor style of metalanguage, or again in a quantificational style of metalanguage. This is assured by the fact that predicate-functor logic and the classical logic of quantification and identity are intertranslatable. We can use Tarski's truth definition in double translation: first we transform it to match the translation of the object language from quantificational style to predicate-functor style, and then we translate the resulting truth definition from its quantificational style of metalanguage into predicate-functor style. This is a routine exercise. A few corners can be cut, but not enough, so far as I have seen, to be interesting.

VI. SUBSTITUTIONAL QUANTIFICATION

Tarski's satisfaction relation has to do with objective reference, relating open sentences as it does to sequences of objects that are values of the variables. Disquotation as such is indifferent to objective reference; but in §IV our attempt at an inductive definition of disquotation broke down in the general case. An inductive definition of disquotation did go through nicely for Schönfinkel's combinatory logic, and this was because a suitable connective was available that was not available for the general case of disquotation. It was the connective '=' of identity (or its equivalent in terms of ι); and disquotation was the designatum function Δ, a matter of objective reference after all. Are objective reference and the definition of truth then inseparable?

For further light on this question let us try defining truth for a quantificational form of language whose quantifiers are explained in terms not of objective values of variables but of notational substitutions for variables.[5]

Again our metalanguage is to contain quotations of the signs of the object language and again the arch of concatenation and, as usual, the logic of quantification and identity (and hence singular description). In short, it is to contain what I have called protosyntax.[6] Its quantifiers may even be read substitutionally, since the relevant values of the variables are quotable expressions. Now it is well known that in protosyntax we can define notations to the following effect, if the object language follows usual and reasonable lines.

Sen x:	x is a sentence (of the object language).
Var x:	x is a variable (of the object language).
Term x:	x is a singular term (of the object language).
$\text{subst}_y^z x$:	the result of substituting z for y in x.
$\text{qfn}_y x$:	the universal quantification of x on the variable y.

Let us now look to what one thinks of as the hard part of a truth definition: the recursion dealing with quantifiers. This goes through lucidly in terms of the truth predicate itself.

$$(10) \qquad (x)(y)(\text{Sen } x \,.\, \text{Var } y \,.\supset. \; \text{True}(\text{qfn}_y x) \equiv (z)(\text{Term } z \supset \text{True}(\text{subst}_y^z x))).$$

[5]For more on substitutional quantification see Essays 16 and 27 above, "Reply to Professor Marcus" and "The variable."

[6]*Mathematical Logic*, last chapter.

There is no talk here of satisfaction or of sequences or, indeed, of objective reference at all.[7] Even the metalinguistic quantifiers '(x)', '(y)', and '(z)' in (10) can be read substitutionally, as remarked.

In the inductive definitions hitherto considered, the work has come in the recursion clauses. The initial clauses were effortless; witness (3). But now the tables turn. The recursion clause (10) is straightforward. The recursion clauses for truth functions are transparent.

$$(11) \quad \text{True}(\text{neg } x) \equiv \sim\text{True } x, \quad \text{True}(\text{conj } xy) \equiv .\, \text{True } x \,.\, \text{True } y.$$

The initial clauses are now the serious part: the definition of truth for atomic sentences devoid of variables.

If substitutional quantification is not to resolve to mere finite conjunction without quantifiers, the supply of substitutible terms must be infinite or indefinite. Singular descriptions, moreover, are not in point; they are contextually definable as usual, and the definition uses quantification. They should be supposed eliminated by contextual definition at the start. But there will be, we may suppose, an infinite stock of constant terms, built from a finite lot of simple terms by iteration of a finite lot of grammatical constructions. The atomic sentences will consist each of a primitive predicate followed by one or more of these perhaps quite long terms as arguments. The sentences are atomic in the sense of containing no further sentences; they may contain complex terms.

EXAMPLE. '0' might be the sole simple term, and the grammatical construction might be application of the successor accent. Then the terms are the numerals '0', '0′', '0″', etc. The primitive predicates might be the triadic predicates 'Σ' and 'Π' of sum and product:

$$\Sigma xyz \equiv .\, x = y + z, \qquad \Pi xyz \equiv .\, x = yz.$$

Each atomic sentence consists of 'Σ' or 'Π' followed by three numerals. Truth functions and substitutional quantification complete the language—a redundant language of elementary number theory.

We can define truth directly for the atomic sentences of this language. Truth for them is decidable, and any decidable predicate of expressions is translatable into protosyntax. Since the re-

[7]This point is remarked by Parsons.

cursions (10) and (11) also can be rendered in that medium, we see that truth for this language of elementary number theory is inductively definable in protosyntax. Or, what is equivalent, it is inductively definable in elementary number theory via Gödel numbering. And indeed it was so defined, along just these lines, by Hilbert and Bernays.[8]

There is a striking contrast between this sort of truth definition and the others. No longer do we build up some preliminary and more general notion of satisfaction or designatum or disquotation, from which to draw truth as a special case. No longer does the disquotational pattern show itself at all, except slightly in the recursion clauses (11). No longer is there any appeal to sequences of objects, of course, nor to objective reference at all; for quantification here is reconstrued in terms of substitution of expressions that need not name anything.

And yet, curiously, the special subject matter of the object language plays a more distinctive role in this truth definition than in Tarski's. Tarski's definition could be fitted to any specific theory, of the classical quantificational pattern, by just filling in the predicates and names and functors specific to that theory. In defining truth for a theory built on substitutional quantification, on the other hand, the main job comes in the atomic sentences; and the lines that this job takes will vary utterly with the structure of the particular theory at hand. The lines that the job takes in our present example of number theory are those of the computation of sums and products.

In this particular example, the adequacy of protosyntax for an inductive definition of truth was assured by the direct protosyntactical definability of decidable predicates. But protosyntax is adequate likewise to an inductive definition of truth for theories in which the atomic sentences are not decidable. What is required, obviously, is just the direct protosyntactical definability of truth for the atomic sentences. This is a very liberal requirement. It does not even require that the atomic sentences admit of a complete proof procedure, let alone a decision procedure. The class of atomic truths can stand at any level of Kleene's arithmetical hierarchy; (10) and (11) will still round out the truth definition

[8]Vol. 2, pp. 334ff.

for atomic sentences into a full inductive truth definition, with protosyntax as metalanguage. If on the other hand a theory built on substitutional quantification has atomic sentences whose truth is stubborn to a hyperarithmetical degree, or is to be left open indefinitely for empirical determination, then there is in protosyntax no hope of an inductive truth definition for the theory. A truth definition in Tarski's style, not in protosyntax, could of course still be available.