THE NAME RELATION AND THE LOGICAL ANTINOMIES

K. REACH

1. Introduction. Every system of signs that is defined in logical syntax may be called a *formal language*. It need not always be a language in the ordinary sense of the word. The rules of chess, e.g., can be expressed in the terminology of the syntax, but it would not occur to anybody to call chess a language.

A language in the ordinary sense has a meaning. That is to say that certain words in it are names of things or states or properties or relationships. In general it can be said that in any language having meaning all words subject to the rule of types are names. (Even sentences might be called names, i.e. names of facts, but we will not go so far.)

A syntax of a language L is a language with meaning, for the words of a definite type (in this paper of the type of individuals) contained in it are names of expressions of L.

If to a language we add its syntax, we get a language containing the syntax of one of its parts. If we formulate the syntax of a language L, in so far as its means allow, in the language L itself, we get a language containing a part of its syntax. We call such languages *autosyntactic*. It is well known that languages containing the whole of their own syntax do not exist.

Autosyntactic languages contain names of their own expressions. This being so it would be inhuman, so to speak, if predicates expressing the relation between expressions and their names were not also to be found in such languages. We call this relation a *semantical name relation*, or briefly, a *name relation*.

It is the aim of this paper to develop on quite broad lines the theory of this important and interesting though hitherto neglected relation.

2. The syntax of "word." Before discussing the relation between an object and its name, it is necessary to say a few words about the syntactic character of the word "word." It can be introduced into a language in various ways:

(a) The place x at the time t has the physical state F and F is a word. Symbolically:

$$F(x, t), \quad \text{Word } 1(F).$$

(b) The word b is a thing consisting of some coloring matter, and having a certain shape. Symbolically:

$$\text{Word } 2(b).$$

For most syntactic and semantical investigations it is advantageous to speak of word-figures or word-shapes rather than of words. By word-figures we are to understand a class of words which are regarded as "equal" to each other. The

Received June 16, 1938.

https://doi.org/10.2307/2267594 Published online by Cambridge University Press

unlimited reproducibility of words is of course a characteristic peculiarity of language. Word-figures constitute abstraction classes in relation to a very generously conceived equality relation, *word-equality*.[1] But we must distinguish between word-equality among word-states and word-equality among word-things:

(c) If F is a class of equal word-states, we can say:

$$F \supset \text{Word } 1, \quad \text{Word-figure } 1(F).$$

(d) If F is a class of equal word-things, we can say:

$$F \supset \text{Word } 2, \quad \text{Word-figure } 2(F).$$

3. The name relation. Let us now consider words which are names. Suppose "Peter" is the name of a certain person. If we wish to express this relationship, we must also give a name to the name "Peter," for we cannot speak of what is nameless. (Permanent descriptions, such as "William's father," can also be names.) Suppose that there is a space-time point (x, t) at which the name "Peter" occurs. We will call this state of (x, t) "Pet." Thus the sentence "Word 1(Pet)" is valid and we can say:

Name 1(Pet, Peter).

It is very important to note that this sentence does not state that "Pet" is the name of "Peter," but that "Peter" is the name of the person Peter. In any other case this warning would be superfluous; when I say that Peter is taller than Paul, I do not intend to express a relation between the words "Peter" and "Paul," but between the two persons.

Let us suppose the word "Peter" is a thing; we designate it by the individual constant "Pt" and Word 2(Pt). Then the following sentence is valid:

Name 2(Pt, Peter).

If we wish to avoid an infinity of "equal" names for one and the same thing, we must take word-figures as names. Let Petfig be the abtraction class containing Pet; then we can write:

Name 1Fg(Petfig, Peter).

Let Ptfig be the abstraction class containing Pt; then we can write:

Name 2Fg(Ptfig, Peter).

The object named can itself be a word. The name relation is then a relation between words, and as we have introduced four word predicates, Word 1, Word 2, Word-figure 1, Word-figure 2, which are all typologically different, we obtain sixteen types of name relations. By the signs, "1," "2," "1Fg," "2Fg," which we add in pairs as suffixes to the word "Name," we can easily distinguish these relations. "Name 1 2Fg," e.g., has as domain[2] a sub-class of the class Word 1, as converse domain[2] the class Word-figure 2.

---

[1] On the concept "abstraction class," cf. Rudolf Carnap, ***Abriss der Logistik*** 20 b.

[2] Cf. Whitehead and Russell, ***Principia mathematica*** *33.

We call, e.g., everything with the shape ";" a semicolon, just as we call everything with such and such properties a lion. "Semicolon" is thus a name of a word-figure. Let us call the figure "Semicolon" itself "Secol." We can then say: "Name 1Fg 1Fg(Secol, Semicolon)." Carnap calls this relation *syntaktische Zuordnung* (SZ).[3] By the employment of SZ we can solve many important and interesting problems.[3] But there are other problems connected with the name relation which cannot be approached by SZ.

In order to see the thing at a glance, we shall represent the relation between the expressions of a language and their names by means of a table. It consists of two rows. In the upper row there is to each word-figure 1 F of our language a representative X_F. Beneath each X_F there is a word Y_F which represents the name-figure of F. If the language contains, as far as is possible, its own syntax, all expressions of the lower row also occur in the upper one. (By the way, instead of "table" one should say "museum," for a table contains an arrangement of names, whereas a museum places things and names together.) We call this table a *name table*. We give here that part of it which we need for this paper:

;	Semicolon	Secol	A	a	aa	x	xx	y	p	Nm	B	b	c
Semicolon	Secol	Sco	a	aa	aaa	xx	xxx	yy	pp	nom	b	bb	cc

The SZ expresses the relation between neighbouring figures above the line, which are not separated by double vertical lines, by joining the corresponding names under the line with the words "is the name of."

4. Defects of the SZ. The purpose of the sentence "Secol is the name of Semicolon" is to give information about the meaning of Secol (i.e., "Semicolon"). Does this sentence serve its purpose? Suppose somebody asks "What is the meaning of Secol?" and he receives the answer "Secol is the name of Semicolon." If the answer is to convey anything to the questioner, it must be understood; i.e., the questioner must know what Sco and Secol stand for in the sentence. That he knows the former is shown by the *form* of his question, but the *meaning* of his question is that he does not know the latter. Hence the answer is incomprehensible to the questioner.

This could be objected to on the ground that it is not the business of a language to explain the meaning of its own symbols, and that whoever makes use of a language must understand its words. The reply to this is that a language has to perform the tasks that are asked of it and, further, if it is not the business of a language to explain the meaning of its symbols then the introduction of the SZ would be useless.

But there are also further defects of the SZ. Suppose we wish to express that, if a sentence pp is true, then also p and vice versa. We will call this sentence "Tr." With the means so far at our disposal this sentence cannot be formulated. If b is a constant sentence, we can write "True$(b) \equiv B$," or, making

[3] Rudolf Carnap, ***Logische Syntax der Sprache***. (Abbreviation: ***Synt.***)

https://doi.org/10.2307/2267594 Published online by Cambridge University Press

use of the SZ, "Name 1Fg 1Fg(*bb*, *b*) $\supset$ (True(*b*) $\equiv B$)." The universal sentence Tr then runs "Name 1Fg 1Fg(x, y) $\supset$ (True(y) $\equiv p$)." The placing of "Name 1Fg 1Fg(x, y)" before "True(y) $\equiv p$" is an attempt to express that the SZ applies between the constant signs which can be substituted for *yy* and *pp*; and this must be expressed, for the sentence "True(y) $\equiv p$" alone would not say what has to be said. The attempt, however, fails; the connexion between *yy* and *pp* is not to be seen from "Name 1Fg 1Fg(x, y)," as *pp* does not occur in it. Besides the formulation is contradictory, for "$q . \sim q$" follows from it.

The sentence Tr, however, can be correctly formulated within the limits of semantics, as Tarski has shown.[4]

5. A new name relation and its syntax. In what follows it will be shown that all these difficulties can be avoided in a simple and natural manner by enriching the language with expressions of the kind " ; has the name Semicolon," that is by expressions where, instead of the names, *the representatives of the word-figures themselves appear*.[5] Instead of this we can say that Semicolon and Secol function in this sentence as their own names. Using the terminology of *Synt*, we may therefore call these predicates *autonymous*. With our new name relation the sentence Tr can be quite simply formulated: If p has the name x, True(x) $\equiv p$.

But sentences such as "*a* has the name *c*" have an important peculiarity. If "$a=b$" is valid, *we cannot derive "b has the name c"*; for "$a=b$" means that *aa* and *bb* are two different names of one and the same object, and it cannot be maintained that two different names (even of one object) have the same name. Since, as we have just seen, sentences such as "*a* has the name *b*" are not subject to the second axiom if identity, $x=y \supset (F(x) \supset F(y))$, the introduction of our new name relation requires a fundamental alteration of the syntax of our language.

The principal features of this new syntax will now be developed.

The atomic sentences of our language are to take the form,

$$\text{"}F[x,y,\cdots](u,v,\cdots)\text{,"}$$

in which one, at least, of the two bracket expressions must occur. If the first bracket expression does not occur, we get the usual form of atomic sentences "$F(u,v,\cdots)$." We call the argument places in the first bracket expression *intensional* and those in the second *extensional*. Expressions (predicates) with intensional argument places may be called "intensional expressions (predicates)." Signs in intensional argument places are not subject to the second axiom of identity.

---

[4] Alfred Tarski, *Der Wahrheitsbegriff in den formalisierten Sprachen*, ***Studia philosophica***, vol. 1 (1935). In this paper Tarski also gives a definition for the relation "x designates a" which is identical with Carnap's SZ.

[5] The author formulated this idea in 1932. The same idea is to be found in a lecture by Helmer in the ***Actes du Congrès International de Philosophie Scientifique***, Paris 1936, part VII. Unfortunately Helmer applies this idea in a way which the present writer cannot approve of.

If the sentence S, "x is an F," of everyday language means that F is a property of the object x designated by xx, the second axiom of identity is of course valid, and we may express S by "$F(x)$" and call the place of xx extensional. But if the sentence S means that F is a property of another object y determined by xx and different from x, the second axiom of identity is in general not applicable and we express S by "$F[x]$," and call the place of xx intentional. The object y need not be the figure xx itself; thus it can happen that in an intensional place we find a non-autonymous expression.

Let us now say a few words about the postulate of extensionality. The sentence "a is a P" of everyday language ascribes the property P, either to an object a designated by aa, or to another object b determined by aa. In the latter case, however, we can say that b is a P and therefore a predicate "P'" can be found having the same meaning as "P" and differing from "P" only typologically, so that $P[a] \equiv P'(b)$. From this it follows that the postulate of extensionality is fulfilled in every language without variables.

Now we shall determine the syntactic character of our new name relation. We have ascertained that in "a has the name b" aa and bb are autonymous. If aa and cc are two names of one object, i.e., $a=c$, cc need not have the name bb as well, and hence the second axiom of identity is not applicable and the place of aa is intensional. If, however, "$b=c$" is valid (this means that bb is a name of the same object as cc), we can derive the sentence "a has the name c." Thus the second axiom of identity is applicable, and the place of bb is extensional. By this the syntactic character of our new name relation is determined, and if we employ the sign nom ("Nm") as a translation of "name" we can express the sentence "a has the name b" by "Nm[a](b)."

We have seen that autonymous expressions can occur in both intensional and extensional places, and that they are not logically different from other expressions. In other words, autonymity is not a syntactic property, but a matter of interpretation ("Deutung," cf. *Synt*).

All predicates of our language are subject to the rule of types, for otherwise a Russell antinomy could easily be obtained.

Our language has a class of fundamental predicates "Nm," "Nm0," "Nm 1," etc., having the meaning of "name" and differing only typologically. The forms of the corresponding atomic sentences are: "Nm[p](x)," "Nm 0[y](x)," "Nm 1[F](x)," etc. (Translated as "p has the name x" etc.)

Now we see that the sentence "Name 1Fg 1Fg(Secol, Semicolon)" is equipollent with "Nm[;](Semicolon)" and "Name 1Fg 1Fg(Sco, Secol)" with "Nm 1[Semicolon](Secol)." "Nm[p](x) $\supset$ (True(x)$\equiv p$)" is equipollent with the sentence Tr. The answer to the question "What is a Secol?" runs. "Nm[;](Semicolon)."

In the intensional place in nom is that expression, the name of which is in the extensional place. The converse domain of Nm is, however, not the class of all names, but only the class of all names of expressions. "Peter," for example, cannot stand in the extensional place in nom, for, if it did, the person Peter would have to be placed in the intensional one.

But a sentence such as "Nm 0[x](xx)" is ambiguous. It can mean that xxx

is the name of xx (i.e. "x"), but it can also express that xxx is the name of all individuals. In order to take this difference into account, we shall, if the first interpretation is correct, put an accent over xx; that is to say, we shall write "Nm $0[\acute{x}](xx)$." In the other case "Nm $0[x](xx)$" will have the same meaning as "(x) Nm $0[x](xx)$" and the transformative rules will lead, e.g., to "Nm $0[a](xx)$." In the first case we shall call xx *accent-bound.* Only signs in expressions in intensional places may be accent-bound. If a variable is free,[6] it is not accent-bound.

In "$P[a]$" P is not a property of the sign aa, but of the figure represented by it. Otherwise not even the sentence "$P[a] \supset P[a]$" would be valid.

6. Formative rules of our language. All sentences of our language are formed by use of atomic sentences, the negation symbol, the connection symbols and the quantifiers; the method is well known.

The rule for explicit definitions is the usual one, but has the following supplement:

Suppose x is an atomic sentence, y a sentential function, the equivalence between x and y an explicit definition, z the defined predicate in x, and u a free variable in y; if u occurs as a free variable only in extensional places of y, u occurs in an extensional place of x; if u appears as a free variable in an intensional place of y, it stands in x in an intensional place. (Rule DR 1.)

As to other kinds of definitions, we shall here only admit one special form of conditioned definitions which we shall call *semantical definitions.*

Rule DR 2. A definition S is called *semantical*, if it has the form:

$$\text{"(Nm } 0[x_1](y_1) \,.\, \text{Nm } 0[x_2](y_2) \,.\, \cdots \,.\, \text{Nm } 0[x_n](y_n)) \supset (A \equiv B)\text{"}$$

(for all types), where b is in the place of a definiens and a in the place of an atomic sentential function with a "new" predicate introduced by S and fulfilling the following conditions: its arguments are variables different from one another, and among its arguments are all the second arguments of the nom relations before the implication sign and all the free variables of the definiens not occurring in the nom relations. Whether the variables in the definiendum (in the place of a) stand in extensional or intensional places is to be decided as in rule DR 1.

In explicit definitions the definiendum must contain all the free variables of the definiens, otherwise contradictory definitions would be possible. This need not, however, be required in the case of semantical definitions. Suppose we have the definition,

$$(\text{Nm } 0[x](y) \,.\, \text{Nm } 0[u](v)) \supset (P[x](y, v, z) \equiv A(x, y, u, v, z)).$$

In order to determine whether a contradiction is possible we deduce from the definition the following sentences:

$$(\text{Nm } 0[c](a) \,.\, \text{Nm } 0[b](f) \,.\, A(c, a, b, f, g)) \supset P[c](a, f, g),$$

$$(\text{Nm } 0[e](a) \,.\, \text{Nm } 0[d](f) \,.\, \sim A(e, a, d, f, g)) \supset \sim P[e](a, f, g).$$

[6] "Real variable" in the terminology of ***Principia mathematica.***

For a contradiction to arise there must be constants "*a*," "*b*," "*d*," "*e*," "*f*," "*g*" such that the sentences before the implication signs are valid, which is, however, incompatible with the lemma Dd 1.1 mentioned below. (Instead of nom we could take any one-many[7] binary—i.e., two-termed—predicate.)

We can now, if we wish, establish the following *semantical definition of the predicate "True"*:

$$\mathrm{Nm}[p](x) \supset (\mathrm{True}(x) \equiv p).$$

It is entirely uninteresting (and interesting that it is uninteresting) what the truth of x means if $\sim\mathrm{Nm}[p](x)$.

7. Transformative rules of our language. Now we lay down the axioms of our language. These are:

The axioms of the sentential calculus (propositional calculus);
The first axiom of identity, $x=x$, for all types;
The axioms of the functional calculus, concerning the quantifiers;
The multiplicative axiom;
Russell's axiom of infinity (if needed);
The axiom of extensionality, $(x)(F(x) \equiv G(x)) \supset (F=G)$;
The second axiom of identity, $x=y \supset (p \supset p^x_y)$, for all types, applicable only if *xx* does not occur as a free variable in an intensional place of *pp*. ("p^x_y" results from *pp* by substitution of *yy* for *xx*.)
Axioms concerning the nom predicate (*Axioms of semantics*),

A 1.1. $(y)\,(\exists x)\,\mathrm{Nm}\ 0[y](x)$,

A 1.2. $(F)\,(\exists x)\,\mathrm{Nm}\ 1[F](x)$,

A 1.3. $(p)\,(\exists x)\,\mathrm{Nm}[p](x)$,

and so forth for every kind of expression. (This infinity of axioms can be replaced by a single syntactical rule.) Since *xx* is autonymous in "Nm $0[y](x)$," A 1.1 does not affirm the existence of "*y*," but of the name *xx* itself! It seems to be quite natural to require that every expression in an autosyntactic language is to have its name in it, but later we shall see that the axioms A 1.2 and A 1.3 are untenable.

A 2. $(\exists z)(\mathrm{Nm}\ 0[z](x)\ .\ \mathrm{Nm}\ 0[z](y)) \supset x=y$ (for all types).

This axiom means that, if two expressions *aa* and *bb* are names of one figure, the sentence "$a=b$" is valid. The sentence,

$$\text{“}x=y \equiv (\exists z)(\mathrm{Nm}\ 0[z](x)\ .\ \mathrm{Nm}\ 0[z](y)),\text{”}$$

is false and must not be used as a definition of the identity. For "$x=y$" can also be true, if *xx* and *yy* designate an object other than a word-figure and in such cases there is no "*z*." Since "*z*" is autonymous, "$(\exists z)\mathrm{Nm}\,0[z](x)$" does not affirm the existence of an object designated by "*z*," but the existence of the expression "*z*" (not "ẑ") itself.

[7] Cf. ***Principia mathematica*** *70–*72.

https://doi.org/10.2307/2267594 Published online by Cambridge University Press

Our language has to be autosyntactic; hence it must contain some fundamental syntactic concepts such as "constant," "comma," etc., and axioms such as, "If a type contains variables, it also contains constants," etc. We quote some concepts which can be defined in our language:

Dm(x) . . . . x is demonstrable,

Ng(y, x) . . . . . . y is the negation of x,

Dc(x) . . . . x is decidable.

The definition of "Dc(x)" is:

$$\mathrm{Dc}(x) \equiv (\mathrm{Dm}(x) \vee (\exists y)\ (\mathrm{Ng}(y, x)\ .\ \mathrm{Dm}(y))).$$

Now we lay down two of the extremely interesting axioms connecting the nom predicate with syntactic predicates:

A 3. $(\mathrm{Nm}[p](x)\ .\ \mathrm{Ng}(y, x)) \supset \mathrm{Nm}[\sim p](y)$.

A 4. $\mathrm{Nm}[p](x) \supset (\mathrm{Dm}(x) \supset p)$.

The last axiom is of the greatest importance. From A 3 and A 4 we can easily deduce the equally important sentence:

4.1. $\mathrm{Nm}[p](x) \supset ((\mathrm{Dc}(x)\ .\ \sim\mathrm{Dm}(x)) \supset \sim p)$.

A 4 and 4.1 signify: From the demonstrability of a sentence S, S itself follows, but from S its demonstrability follows only if S is decidable. In symbols:

4.2. $(\mathrm{Nm}[p](x)\ .\ \mathrm{Dc}(x)) \supset (\mathrm{Dm}(x) \equiv p)$.

As rules of deduction we adopt the ordinary well-known rules, and in addition a new semantical deduction rule:

First we give two auxiliary definitions:

We call two expressions *cognate* if we can change one into the other by replacing certain signs in them by others of the same type ("$F(a\supset$" is, e.g., cognate with "$P(x\equiv$").

If in the sentence S, "$(\exists z)(\mathrm{Nm}[A](z)\ .\ \mathrm{Nm}[B](z))$," we put for a an expression x and for b an expression y, we get the sentence equ(x, y).

The new syntactic rule runs:

Rule Dd 1. If x is cognate with y, z a variable in x, u a word in y corresponding to z in x, and S^z_u a sentence got by replacing the free variable z in S by u, then S^z_u is a consequence of S and equ(x, y).

From rule Dd 1 follows the lemma:

Dd 1.1. $(\exists z)(\mathrm{Nm}\ 0[x](z)\ .\ \mathrm{Nm}\ 0[y](z)) \supset (p \supset p^x_y)$.

This lemma shows that Dd 1 compensates for the restriction upon the second axiom of identity. The sentence equ(xx, yy) expresses a kind of super-identity between xx and yy; the two signs do not *designate* the same thing, but *are* the same thing.

There are many other most interesting axioms and rules of semantics which we shall not quote here because they are not needed in this paper.

8. The antinomies. In an autosyntactic language with nom predicates the danger of inconsistency is especially acute. We shall first consider what can be done to free a language of contradictions. In order to construct a contradiction, certain premisses are necessary. If the premisses are valid, the language is inconsistent and must be altered. It is reasonable to make only those alterations which are the least portentous and sweeping possible, and especially to take care to diminish the capacity of the language to express meaningful thoughts as little as possible. If one of the premisses leading to a contradiction is empirical, we shall cancel that rather than a logical one. If there are no empirical premisses, we are obliged to cancel a logical axiom or a formative rule. Russell, for instance, had the task of freeing his language from the antinomy arising from the definition of "impredicable." To derive this antinomy only the formative rules and the logical axioms are employed. Russell recognized that an alteration of the rules of formation, namely the introduction of types, was an alteration of the language that did not decrease its power of expression.

Let us now investigate the influence of the logical antinomies upon the consistency of our autosyntactic language.

9. The dilemma of the crocodile. A crocodile, having stolen a child, said to the mother: "Only if you guess whether I am going to give you back your child or not, shall you get it back." The mother said: "You will not give it back." The crocodile replied: "If you have guessed wrongly, you have not fulfilled my condition, if you have guessed rightly, your guess is true; thus in neither case do you get back your child." The mother answered: "If I have guessed rightly, I have fulfilled your condition, if I have guessed wrongly, my guess is false; thus in either case you must give back my child."

First we must lay down a definition for the sentence "x guesses correctly whether "p" is true or false." This is true only if x says "p" or "$\sim p$" but not both, and if x says "p" then p, and if x says "$\sim p$" then $\sim p$. Thus we obtain the semantical definition:

$$\text{K 1.}\quad \mathrm{Nm}[p](y) \supset (\mathrm{Guess}(x, y) \equiv ((\mathrm{Say}(x, y) \equiv \sim \mathrm{Say}(x, \mathrm{Ng}‘y)) \,.\, (\mathrm{Say}(x, y) \supset p) \,.\, (\mathrm{Say}(x, \mathrm{Ng}‘y) \supset \sim p)).$$

Let the sentence "The crocodile gives back the child to its mother" have the abbreviation "A," and "A" the name "a." Instead of "mother," we shall write "m." Then the sentence "$\mathrm{Nm}[A](a)$" is valid, and from K 1 follows:

$$\text{K 2.}\quad \mathrm{Guess}(m, a) \equiv ((\mathrm{Say}(m, a) \equiv \sim \mathrm{Say}(m, \mathrm{Ng}‘a)) \,.\, (\mathrm{Say}(m, a) \supset A) \,.\, (\mathrm{Say}(m, \mathrm{Ng}‘a) \supset \sim A)).$$

And hence after a little transformation:

K 3. $\sim$Guess$(m, a) \equiv ((\text{Say}(m, a) \equiv \text{Say}(m, \text{Ng}\text{`}a)) \vee ((\sim\text{Say}(m, a) \supset A) \,.\, (\sim\text{Say}(m, \text{Ng}\text{`}a) \supset \sim A)))$.

The crocodile's condition:

K 4. Guess $(m, a) \equiv A$.

The mother answers Ng'a, but not a. In symbols:

K 5. $\text{Say}(m, \text{Ng}\text{`}a) \,.\, \sim\text{Say}(m, a)$.

From K 5 and K 3 follows:

K 6. $\sim\text{Guess}(m, a) \supset ((\sim\text{Say}(m, a) \supset A) \,.\, (\sim\text{Say}(m, \text{Ng}\text{`}a) \supset \sim A))$.

If Nm[Guess(m, a)](g), the conclusion of the crocodile is this: Ng'a follows from Ng'g by K 4, and from g by K 5 and K 2.

The argument of the mother is this: a follows from g by K 4, and from Ng'g by K 5 and K 6.

The premisses of the antinomy are K 4, K 5, and the semantical definition K 1. All the other sentences are consequences. If we wish to retain the semantical definitions—and we do wish to—we have to seek the contradiction between K 4 and K 5 or in one of them. Indeed it is easy to show that the negation of K 5 can be derived from K 4 after replacing g by its definition in K 2. K 5 expresses an empirical fact. The crocodile's condition is an assertion which can be true or false. As it involves a false prediction, it is false. That is the whole mystery.

10. Grelling's antinomy. Grelling calls a predicate "heterological," if it has not the property it designates. We therefore obtain the following semantical definition:

H 1. $\text{Nm } 1[F](x) \supset (\text{Het}(x) \equiv \sim F(x))$.

Putting "Het" for "F" we get

H 2. $\sim(\exists x)\ \text{Nm } 1[\text{Het}](x)$,

which is incompatible with the axiom A 1.2. If we wish to retain the definition rule DR 2—as we do—and the common axioms and deduction rules of logic, we are compelled to drop A 1.2.

At first sight it looks somewhat strange to cancel such an axiom, for it seems to be within our power to create names for every expression of our language and thus to make the axiom valid; we need, e.g., only stipulate that the name of every expression is the expression itself put between quotation marks. But that is a delusive proof, for if you invent a name for an expression in your language, you have not proved that this name can be a word of the language. If Nu is a language erected above the empty individual class, it comprises the predicate "null class," but no Name 2 1Fg of it.

Suppose we have an autosyntactic arithmetical language M containing the definition H 1. We choose as names for the signs of M certain numbers, called "sign numbers."[8] In M, therefore, the predicates "sign number" and "Nm⌈ Sign" are synonymous. H 1 shows that the arithmetical predicate "Het" has no sign number. In M a rule can be introduced according to which the sign number of a defined predicate is a function of the sign numbers of the other signs occurring in the definition.[9] Hence "sign number" has no sign number and nom has no name.

Let us consider some generalizations of Grelling's antinomy.

We can generalize the contradictory sentence,

H 3. $(\mathrm{Nm}\ 1[F](x) \supset (\mathrm{Het}(x) \equiv {\sim}F(x))) \,.\, (\exists x)\ \mathrm{Nm}\ 1[\mathrm{Het}](x)$,

in two ways. We can replace "Het" by any predicate and we can put for nom any intensional or extensional one-many predicate. Thus we get the formulae:

H 4′. $(\exists H, P)\ ((\exists x)\ P[H](x) \,.\, 1{\rightarrow}\mathrm{Cls}(P) \,.$
$(x, F)\ (P[F](x) \supset (H(x) \equiv {\sim}F(x))))$,

and

H 4″. $(\exists H, P)\ ((\exists x)\ P(H, x) \,.\, 1{\rightarrow}\mathrm{Cls}(P) \,.$
$(x, F)\ (P(F, x) \supset (H(x) \equiv {\sim}F(x))))$.

In consequence of $1{\rightarrow}\mathrm{Cls}(P)$ we can also write:

H 5. $(\exists H, P)\ ((\exists x)\ H = P\text{‘}x \,.\, (x)\ (H(x) \equiv {\sim}(P\text{‘}x)(x)))$.

From H 5 follows:

H 6. $(\exists H, P)\ ((\exists x)\ (y)\ (H(y) \equiv (P\text{‘}x)(y)) \,.\, (x)\ (H(x) \equiv {\sim}(P\text{‘}x)(x)))$.

Hence:

H 7. $(\exists H, P, x)\ ((H(x) \equiv (P\text{‘}x)(x)) \,.\, (H(x) \equiv {\sim}(P\text{‘}x)(x)))$.

Hence:

H 8. $(\exists P, x)\ ({\sim}(P\text{‘}x)(x) \equiv (P\text{‘}x)(x))$.

Thus H 5 is contradictory. It can be written in an abbreviated form:

H 9. $(\exists P, y)\ \hat{x}\{{\sim}(P\text{‘}x)(x)\} = P\text{‘}y$.

We infer from it that there is no descriptive function[10] P assigning to every predicate H an individual x such that "$H = P\text{‘}x$" is valid. Hence the sentence "${\sim}(\exists P)(H)(\exists x)\ H = P\text{‘}x$" is analytic.

Carnap gives in *Synt* the following definition for "Het":

H 10. $\mathrm{Het}(x) \equiv {\sim}\mathrm{Analytic}(\mathrm{subst}(x, 3, \mathrm{str}(x)))$.

The question, whether H 1 or H 10 is the right translation into the logistic language of Grelling's antinomy formulated in everyday language, is meaningless, for it can only be raised if a translation rule is given. In our case there are

---

[8] "Gliedzahlen," cf. *Synt*.

[9] Cf. *Synt*.

[10] *Principia mathematica* *30.

https://doi.org/10.2307/2267594 Published online by Cambridge University Press

two translation rules, at least, and thus we get at least two quite different formulations. The antinomies in everyday language show us that it is inconsistent. To translate them into an exact language means to formulate new antinomies in it. As the number of exact languages and of translation rules is unlimited, there is an infinity of formulations of every antinomy. Thus we see that there is no "definite" solution of an antinomy.

11. The antinomy of the liar. The present antinomy will be solved by the same method as the preceding one, which shows the efficiency of our procedure.

If we abbreviate "I am lying" by "L," "L" is a sentence affirming its own falsity. In symbols:

L 1. $\mathrm{Nm}[L](x) \supset (L \equiv \mathrm{False}(x))$.

We give a semantical definition of "False(x)":

L 2. $\mathrm{Nm}[p](x) \supset (\mathrm{False}(x) \equiv \sim p)$.

By substitution we get,

L 3. $\mathrm{Nm}[L](x) \supset (\mathrm{False}(x) \equiv \sim L)$,

and together with L 1 we get,

L 4. $\sim(\exists x)\ \mathrm{Nm}[L](x)$,

which is in contradiction to A 1.3. Which of our premisses is responsible for the antinomy? We have assumed the existence of a sentence ascribing a property to itself. It would be useless to forbid such sentences, for the antinomy can also be derived—as Carnap has pointed out—if there is a sentence ascribing a property to another sentence and the latter similarly reciprocating. In such circumstances we do best if we cancel A 1.3, as we have done with A 1.2.

There are still other interesting formulations of the antinomy of the liar.

We can consider "L" as a sentence affirming its own undemonstrability. In symbols,

L 1′. $\mathrm{Nm}[L](x) \supset (L \equiv \sim\mathrm{Dm}(x))$.

From L 1 and 4.2 follows:

L 4′. $\sim(\exists x)(\mathrm{Nm}[L](x) \,.\, \mathrm{Dc}(x))$.

But in this case—following a method of Gödel[11]—we can construct the name, i.e. the sign number, of "L." Hence "$(\exists x)\ \mathrm{Nm}[L](x)$" is valid and we obtain

L 5′. $(\exists x)\sim\mathrm{Dc}(x)$,

which is in agreement with Gödel's result.[11]

A third interpretation: We consider "L" as a sentence affirming the demonstrability of its denial; in symbols,

L 1″. $(\mathrm{Nm}[L](x) \,.\, \mathrm{Ng}(y, x)) \supset (L \equiv \mathrm{Dc}(y))$.

[11] Kurt Gödel, *Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I*, ***Monatshefte für Mathematik und Physik,*** vol. 38 (1931).

From A 3 and 4.2 follows:

L 3″. $(\mathrm{Nm}[L](x) \,.\, \mathrm{Ng}(y, x) \,.\, \mathrm{Dc}(y)) \supset (\mathrm{Dm}(y) \equiv \sim L)$.

From L 1″ and L 3″ follows:

L 4″. $\sim(\exists x, y)(\mathrm{Nm}[L](x) \,.\, \mathrm{Ng}(y, x) \,.\, \mathrm{Dc}(y))$.

As "$(\exists x)\mathrm{Nm}[L](x)$" is valid we obtain

L 6″. $(\exists x)(y)(\mathrm{Ng}(y, x) \supset \sim\mathrm{Dc}(y))$,

and as we may suppose $(x)(\exists y)\mathrm{Ng}(y, x)$, we obtain

L 5″. $(\exists y)\sim\mathrm{Dc}(y)$,

the same result as in the former case.

From L 2 follows: $\mathrm{Nm}[p](x) \supset (\mathrm{False}(x) \equiv (\mathrm{Nm}[p](x) \supset \sim p))$. With the sentence "$(p)(\mathrm{Nm}[p](x) \supset \sim p)$" an interesting antinomy can be obtained[12] if we stipulate that the abbreviation "T" of "$(p)(\mathrm{Nm}[p](x) \supset \sim p)$" has the name x. We define:

T 1. $T \equiv (p)(\mathrm{Nm}[p](x) \supset \sim p)$.

From D 1.1 follows:

T 2. $(\mathrm{Nm}[p](x) \,.\, \mathrm{Nm}[q](x)) \supset (p \equiv q)$.

These are the premisses of the antinomy. The notations in parenthesis after the following sentences will indicate the sentences employed in deducing them.

T 3. $(\mathrm{Nm}[q](x) \supset (\sim q \supset (p)\ (\mathrm{Nm}[p](x) \supset \sim p))$ (T 2)

T 4. $\mathrm{Nm}[T](x) \supset (\sim T \supset T)$ (T 3, T 1)

T 5. $T \supset (\mathrm{Nm}[T](x) \supset \sim T)$ (T 1)

T 6. $\mathrm{Nm}[T](x) \supset (T \supset \sim T)$ (T 5)

T 7. $\mathrm{Nm}[T](x) \supset (T \equiv \sim T)$ (T 4, T 6)

Hence "T" has no name. (It looks like a contradiction, if we write " "T" " and affirm that "T" has no name, but there must be a distinction between the language we are considering and the language of this paper, the latter being the syntax of the former. The sign "T" has no name in the language considered, but may have a name in its syntax.)

Our solutions of the antinomies suggest the following semantical rule:

No expression synonymous with an expression containing the nom predicate has a name.

12. Gödel's theorem. In order to give an example of the application of the nom predicate less elementary than the former, we shall construct Gödel's non-decidable sentence in our language and prove its non-decidability. We shall furnish this proof without Gödel's assumption of the ω-consistency of the

[12] Tarski, loc. cit.

language, but we shall be compelled to employ the axiom A 4 where Gödel manages with the weaker proposition 4.1.[13]

First let us give two definitions:

G 1. $\mathrm{Sb}[x](y) \equiv (\exists F)\ (\mathrm{Nm}[F[\acute{z}]](x) \,.\, \mathrm{Nm}[F[x]](y)$.

Read: "x has the self-substitution y."

G 2. $G;F[y] \equiv (\exists z)\ (F(z) \,.\, G[y](z))$.

Read: "y has a G from F."

First we shall prove three lemmas. We start with an assumption which is true if we suppose that our language has a rule for the construction of names of expressions.

G 3. $\mathrm{Nm}[F[\acute{z}]](x) \supset (\exists y)\mathrm{Sb}[x](y)$.

G 4. $(\mathrm{Nm}[F[\acute{z}]](x) \,.\, \sim\mathrm{Sb};K[x]) \supset ((\exists y)\mathrm{Sb}[x](y) \,.\, (z)(\mathrm{Sb}[x](z) \supset \sim K(z)))$. (G 3)[14]

G 5. $(\mathrm{Nm}[F[\acute{z}]](x) \,.\, \sim\mathrm{Sb};K[x]) \supset \mathrm{Sb};\sim K[x]$. (G 4)

G 6. $(\mathrm{Sb}[x](y) \,.\, \mathrm{Sb}[x](z)) \supset y = z$. (Dd 1)[15]

G 7. $(K(y) \,.\, \mathrm{Sb}[x](y)) \supset (\mathrm{Sb}[x](z) \supset K(z))$. (G 6)

G 8. $\mathrm{Sb};K[z] \supset \sim\mathrm{Sb};K[x]$. (G 7)

G 9. $\mathrm{Nm}[F[\acute{z}]](x) \supset (\mathrm{Sb};K[x] \equiv \sim\mathrm{Sb};\sim K[x])$. (G 5, G 8)

This is the first lemma.

G 10. $(\mathrm{Nm}[F[\acute{z}]](x) \,.\, \mathrm{Sb}[x](y)) \supset \mathrm{Nm}[F[x]](y)$. (Dd 1)

Let "A" be an abbreviation for "$\mathrm{Nm}[\mathrm{Sb};\sim\mathrm{Dm}[\acute{z}]](x)$."

G 11. $(A \,.\, \mathrm{Sb}[x](y) \,.\, \mathrm{Dm}(y)) \supset (\mathrm{Nm}[\mathrm{Sb};\sim\mathrm{Dm}[x]](y) \,.\, \mathrm{Dm}(y))$. (G 10)

G 12. $(A \,.\, \mathrm{Sb};\mathrm{Dm}[x]) \supset \mathrm{Sb};\sim\mathrm{Dm}[x]$. (G 11, A 4)

This is the second lemma.

G 13. $(A \,.\, \mathrm{Sb}[x](y) \,.\, \mathrm{Dc}(y) \,.\, \sim\mathrm{Dm}(y)) \supset (\mathrm{Nm}[\mathrm{Sb};\sim\mathrm{Dm}[x]]\ (y) \,.\, \mathrm{Dc}(y) \,.\, \sim\mathrm{Dm}(y))$. (G 10)

G 14. $(A \,.\, \mathrm{Sb};\mathrm{Dc}[x] \,.\, \mathrm{Sb};\sim\mathrm{Dm}[x]) \supset \sim\mathrm{Sb};\sim\mathrm{Dm}[x]$. (G 13, A 4.1, G 6)

G 15. $(A \,.\, \mathrm{Sb};\mathrm{Dc}[x]) \supset \sim\mathrm{Sb};\sim\mathrm{Dm}[x]$. (G 14)

---

[13] In his paper "Satz V."

[14] "$\sim\mathrm{Sb};K[x]$" means "$\sim(\mathrm{Sb};K[x])$."

[15] Proof

$$(\mathrm{Sb}[x](y).\mathrm{Sb}[x](z)) \equiv (\exists F, G)(\mathrm{Nm}[F[\acute{z}]](x).\mathrm{Nm}[G[\acute{z}]](x).\mathrm{Nm}[F[x]](y).\mathrm{Nm}[G[x]](z)).$$

"$F[\acute{z}]$" is cognate with "$G[\acute{z}]$" and thus we may, according to Dd 1, in the sentence after the equivalence sign put "F" for "G." By axiom A 2 we get G 6.

https://doi.org/10.2307/2267594 Published online by Cambridge University Press

This is the third lemma.

G 16. $A \supset$ Sb;$\sim$Dm[x]. (1st and 2nd lemma)

G 17. $A \supset \sim$Sb;Dc[x]. (G 16 and 3rd lemma)

G 18. $A \supset$ Sb;$\sim$Dc[x]. (G 17 and 1st lemma)

Having accepted a construction rule for the names of expressions, we may assume that the name of "Sb;$\sim$Dm[x]" exists. Hence "$(\exists x)A$" is valid and we get:

G 19. $(\exists x, y)$ (Nm[Sb;$\sim$Dm[$\acute{z}$]](x) . Sb[x](y) . $\sim$Dc(y)). (G 18)

Thus we have proved the existence of a non-decidable sentence. How does the sentence run?

G 20. $(\exists x, y)$ (Nm[Sb;$\sim$Dm[$\acute{z}$]](x) . Nm[Sb;$\sim$Dm[x]](y) . $\sim$Dc(y)).
(G 19, Dd 1)

Thus the non-decidable sentence is:

"Sb;$\sim$Dm["Sb;$\sim$Dm[$\acute{z}$]"]."

PRAGUE

https://doi.org/10.2307/2267594 Published online by Cambridge University Press