

Institute for Language, Logic and Information

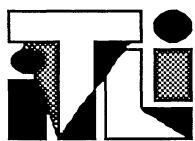
A RELATIONAL FORMULATION OF THE THEORY OF TYPES

Reinhard Muskens

ITLI Prepublication Series 86-04



University of Amsterdam



Instituut voor Taal, Logica en Informatie

Faculteit der Wiskunde en Informatica, Roetersstraat 15, 1018WB Amsterdam, or
Faculteit der Wijsbegeerte, Grimburgwal 10, 1012GA Amsterdam

A RELATIONAL FORMULATION OF THE THEORY OF TYPES

Reinhard Muskens
Department of Philosophy
University of Amsterdam

Received October 1986



Institute for Language, Logic and Information

Faculty of Mathematics and Computer Science, Roetersstraat 15, 1018WB Amsterdam, or
Faculty of Philosophy, Grimburgwal 10, 1012GA Amsterdam

1 Introduction

1.1 Relational Type Theory

In Montague semantics it is common procedure to specify a translation function taking the expressions of some fragment of natural language to logical expressions. If all is done well, the translated phrases and their translations show the same logical behaviour. Their truth conditions should match, for example, and the relation of logical consequence on the translations should mirror the relation of entailment that is imposed on the natural language fragment by natural logic.

The logic that is usually taken as the range of values of this translation function, is Montague's **IL** (Intensional Logic), defined in Montague [1973] and extensively described in Gallin [1975]. Being an intensional extension of Church's [1940] beautiful formulation of the simple theory of types, it can be embedded in a two-sorted version, **TY₂**, of this theory, as was shown by Gallin.

Historically, Church's formulation of type theory was much influenced by his formulations of the lambda calculus, which is a theory of functions. The 1940 article defining the logic is mainly of a syntactical character, but in the first section a brief suggestion is made concerning the intended interpretation of the system. This interpretation is to be *functional*. While in earlier and less precise formulations of type theory (see Russell [1908], Carnap [1929]) classes and relations played an important and more independent rôle, these now seem to have to be equated with their characteristic functions. Multi-argument relations are identified in this way with multi-argument functions, which in their turn, following Schönfinkel [1924], are equated with functions in one argument whose values are functions again.

Now these moves seem innocent enough. Technically it is clearly equivalent to consider relations directly or to explain them recursively with the help of Schönfinkel's Trick. But, although equivalent, this identification is - I claim - not very felicitous. Relations are 'moved up' recursively in the set-theoretical hierarchy and this complication makes it extremely difficult to formulate the usual model-theoretical notions for the logic. In fact, in almost all cases where an interesting notion *is* defined, this is done by the use of a recursion that reverses the effect of Schönfinkel's Trick.¹

¹See e.g. the definition of *persistence* in Gallin [1975, §4] (and compare it with the one in §9).

This kind of problems made S. Orey define his higher-order predicate calculus in 1959 (see also Gallin [1975], Van Benthem & Doets [1983]). Avoiding the Trick, he formulated type theory in such a way that model-theoretic concepts as, for example, *substructure* or *end extension* (of a general model) have simple and natural formulations. Types, in his system, are of a *relational*, not of a functional, character as they are in Church's, and the objects in his domains are either individuals or relations.

DEFINITION 1. The set of *types* is to be the smallest set such that:

- i. e and s are types,
- ii. if $\alpha_1, \dots, \alpha_n$ are types ($n \geq 0$), then $\langle \alpha_1 \dots \alpha_n \rangle$ is a type.

We shall equate $\langle \rangle$ with \emptyset or, equivalently, with 0. The types e and s we call *basic*, all other types *relational*.

DEFINITION 2. A *standard Orey frame* is a family of sets

$\{D_\alpha \mid \alpha \text{ is a type}\}$ such that $D_e \neq \emptyset$, $D_s \neq \emptyset$ and $D_{\langle \alpha_1 \dots \alpha_n \rangle} = P(D_{\alpha_1} \times \dots \times D_{\alpha_n})$.

(The cartesian product of the empty sequence of sets is to be equated with $\{\langle \rangle\}$. So $D_{\langle \rangle} = P(\{\emptyset\}) = \{0, 1\}$, the set of *truth values*.)

Orey's relational *models* can now be defined in the usual way adding an interpretation function to the frames just given (see also section 3 below). The use of these relational models instead of the standard functional ones is not only advantageous from a model-theoretic point of view, but has also much to be recommended from the perspective of applications of type theory in Montague semantics. I shall give four arguments in support of this.

The first argument is that, although the standard logic has in a sense to be explained below 'more' types than relational type theory has, these extra types are in fact seldom used. Almost all proposed translations of natural language expressions have (functional) types that correspond closely to the relational types defined above. In order to put this more accurately, I shall first give the familiar definition of Church's types and then define the subclass of them that is in fact - I claim - most popular.

DEFINITION 3. The set of *Church types* is to be the smallest set such that:

- i. e, s and t are Church types,
- ii. if α and β are Church types, then $(\alpha\beta)$ is a Church type.

DEFINITION 4. Define the function Σ (Σ is for Schönfinkel) taking types to Church types by the following double recursion:

- I $\Sigma(e) = e, \Sigma(s) = s$
 II i. $\Sigma(<>) = t$
 ii. $\Sigma(<\alpha_1 \dots \alpha_n>) = (\Sigma(\alpha_1)\Sigma(<\alpha_2 \dots \alpha_n>))$ if $n \geq 1$.

So, for example, $\Sigma(<e>) = (et)$, $\Sigma(<<e>>) = ((et)t)$, $\Sigma(<ee>) = (e(et))$ and $\Sigma(<<se><se>>) = ((s(et))(s(et))t)$. On the other hand, Church types like (se) or (ee) are not values of Σ . If α is the type of some relation then $\Sigma(\alpha)$ is the Church type of the unary function that codes this relation in functional type theory.

Ever since Bennett [1974] removed individual concepts from the standard formulation of Montague Grammar, the vast majority of types that have proposedly been assigned to linguistic categories have been values of Σ . If the semantics of a natural language is described with the help of a functional type theory, then linguistic expressions tend to get semantic values having values of Σ as their types. This seems to be an important fact about semantics, but it is a fact that is not reflected in the overall organization of current Montague Grammar. It would be so reflected if we could somehow trade the usual type theory for a relational one and assign relational types α to linguistic categories instead of their functional counterparts $\Sigma(\alpha)$. Since arguments of this function tend to be more simple than the corresponding values, this would give a slight simplification of the theory too.²

The second argument concerns the complexity of the objects that are used in functional semantics as compared with the complexity of their relational counterparts. In the functional theory, elements from Orey frames are coded as elements from Church frames:

DEFINITION 5. A *Church frame* is a family of sets $\{D_\alpha \mid \alpha \text{ is a Church type}\}$ such that $D_e \neq \emptyset$, $D_s \neq \emptyset$, $D_t = \{0,1\}$ and $D_{(\alpha\beta)}$ is the set of functions from D_α to D_β .

²The choice of a particular logic should of course not preclude certain analyses of natural language. It should, for example, not be made *impossible* by our logic to use individual concepts (type (se) objects, functions from worlds to entities). However, since all functions are relations, there is no problem. Those who think that individual concepts are useful (see Janssen [1984]) may keep them as $<se>$ -type relations (relations between worlds and entities). Expressions like **the pope** can then be treated as individual concepts. Note that **the pope** can't be a function since there have been times that there was no pope and once, during the Avignon period, there were three.

Let us take a closer look at the function S that codes multi-argument relations as unary functions. Its definition is somewhat less simple than might be thought:

DEFINITION 6. Let $\{D_\alpha \mid \alpha \text{ is a type}\}$ be a standard Orey frame and $\{D'_\alpha \mid \alpha \text{ is a Church type}\}$ the Church frame such that $D_e = D'_e$ and $D_s = D'_s$. For each type α , define a bijection $S_\alpha: D_\alpha \rightarrow D_{\Sigma(\alpha)}$ by the following double recursion:

- I $S_e(d) = d$, if $d \in D_e$; $S_s(d) = d$, if $d \in D_s$;
- II i. $S_{\langle \rangle}(d) = d$, if $d \in D_{\langle \rangle}$;
- ii. If $n > 0$, $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$ and $R \in D_\alpha$, then $S_\alpha(R)$ is the function F of type $(\Sigma(\alpha_1)\Sigma(\langle \alpha_2 \dots \alpha_n \rangle))$ such that for each $f \in D'_{\Sigma(\alpha_1)}$ $F(f) = S_{\langle \alpha_2, \dots, \alpha_n \rangle}(\{ \langle d_2, \dots, d_n \rangle \mid \langle d_1, d_2, \dots, d_n \rangle \in R \})$, where $d_1 = S_{\alpha_1}^{-1}(f)$.

It is routine to prove that this is well-defined. Define the function S to be $\bigcup_\alpha S_\alpha$.

Obviously, the function S tends to rather dramatically increase complexity. For example, an object of type $\langle \langle se \rangle \langle se \rangle \rangle$ (arguably the kind of object that can be taken to be the extension of a natural language determiner), which is a two-place relation taking relations between indices and entities in both its argument places, is coded as a function taking functions from indices to functions from entities to truth values to functions taking functions from indices to functions from entities to truth values to truth values.

Now, if there would be any need to do so, we could gladly accept these intricacies, since in a sense the functions S_α are *isomorphisms*: for all relations R (of any type) $\langle d_1, \dots, d_n \rangle \in R$ iff $S(R)(S(d_1)) \dots (S(d_n)) = 1$, as can easily be verified. But I think that this doubly recursive encoding is just a needless complication. If we want Montague Grammar to look a little less like a Goldberg machine (the comparison is taken from Barwise & Cooper [1981]), we may as well skip it.

My third argument pro a relational logic also has to do with elegance of formalization and simplification of the theory. In view of the fact that natural language **and**, **or** and **not** can be used with expressions of almost all linguistic categories, type domains should have a Boolean structure. This has been argued for by a variety of authors, beginning with Von Stechow [1974] (see also Keenan & Faltz [1978]). Obviously, Orey's relational models have a Boolean structure on all their (non-basic) domains, since these are power sets. So we can give a very simple rule for the interpretation of natural language conjunction, disjunction and negation:

they are to be treated as \cap , \cup and $-$ (complementation within a typed domain) respectively (I shall propose a slight emendation on the negation rule below). Entailment between expressions of the same category is to be treated as inclusion. This does not differ much from the usual treatment of both entailment and the expressions just mentioned in the literature (see Gazdar [1980] for generalized coordination, Groenendijk & Stokhof [1984] for entailment). The point is rather that the relevant Boolean operations are not as easily available in a functional type theory as they are here. They can only be obtained by using a pointwise recursive definition reversing the effect of Schönfinkels Trick.³ In a relational type theory such unnecessary complications can be avoided. If coding relations as unary functions makes very simple things more or less complex, it may very well make really complex things unintelligible. Therefore I think we should turn to a relational formulation of type theory in Montague Semantics.

A few paragraphs back I have contrasted the model-theoretic view upon type theory with the perspective from the point of view of its applicability in semantics. But this contrast may not be absolute. Elementary relations between models may play some part in the semantics of natural language expressions. In Muskens [1983] it was argued that the recipe for evaluating a degree adjective-noun construction like **small elephant** is: evaluate **small** (a recipe for *that* evaluation was given, but need not concern us here) on the substructure of your model that has the set of elephants as its e-domain. Similar evaluation rules, using submodels, can be given for other linguistic constructions (see Muskens [in preparation]). So my third argument runs as follows: If one tries to formalize the evaluation of natural language with the help of model theory, it may well be that elementary relations between models (like the relation of *substructure*) can fruitfully play a part in that formalization. Therefore, other things being equal, one should choose a logic that allows easy definitions of such relations. Defining a type theory for Montague Grammar, one should choose a relational rather than a functional formulation.

Are other things equal? Although there are, as I have just argued, good reasons to prefer Orey's relational *models* over Church's functional ones, there are equally good reasons to prefer Church's *syntax* over Orey's when it comes to choosing a logic for our purposes. In fact the latter logic, as it was defined in Orey [1959], lacks the operations of *application* and *abstraction*, which are absolutely crucial for the Montague semanticist. So at this point it may seem that we can either have an applicable logic with a

³But this cannot be done for *all* non-basic type domains, only for the so-called *conjoinable* ones; see Partee & Rooth [1983] for a definition and notice that a Church type is a value of Σ iff all its subtypes are either basic or conjoinable.

complex model theory or a logic with a more simple model theory which is inapplicable.

But the dilemma is only apparent. We can have our cake and eat it by taking the syntax of standard type theory and evaluate it on relational models. Let us consider application. Suppose that A and B are terms of types $\langle\beta\alpha_1\dots\alpha_n\rangle$ and β respectively. Then the value of the term AB (of type $\langle\alpha_1\dots\alpha_n\rangle$) in some model M (under an assignment a) is given by the following rule:

$$(1) \|AB\|^{M,a} = \{\langle d_1, \dots, d_n \rangle \mid \langle \|B\|^{M,a}, d_1, \dots, d_n \rangle \in \|A\|^{M,a}\}$$

Let, for example, the domain of M be some set of people and let *love* be a constant of type $\langle ee \rangle$ that is to be interpreted as the love relation among them ($\{\langle d_1, d_2 \rangle \mid d_2 \text{ loves } d_1\}$). Let j and m be constants of type e , interpreted on M by John and Mary respectively. Then $\|\textit{love}j\|^{M,a}$ will be equal to $\{d_2 \mid d_2 \text{ loves John}\}$, the set of persons loving John, while $\|\textit{love}jm\|^{M,a}$ equals $\{\langle \rangle \mid \text{Mary loves John}\}$, which is equal to the value 1 just in case Mary indeed loves John.

Suppose now that A is a term of some type $\langle\alpha_1\dots\alpha_n\rangle$ and that x is a variable of type β . Then we can define the value of the term λxA (of type $\langle\beta\alpha_1\dots\alpha_n\rangle$) in M under a as follows:

$$(2) \|\lambda xA\|^{M,a} = \{\langle d, d_1, \dots, d_n \rangle \mid d \in D_\beta \text{ and } \langle d_1, \dots, d_n \rangle \in \|A\|^{M,a[d/x]}\}$$

For example the term $\lambda x_e \lambda y_e \exists z_e (x < z \wedge y < z)$ will receive the relation 'having a joint successor' on the D_e domain as its interpretation in any model, as the reader can easily verify.

1.2 Definite and indefinite description operators

In the preceding pages I have sketched how type theory can be interpreted in a relational way. It would be easy now to fill in the details of this sketch and obtain a completely defined relational semantics. The crucial clauses in the Tarski truth definition would be (1) and (2) of course, and the resulting system would look a lot like Gallin's TY_2 , although its model theory would be much simpler.

Note, however, the following little asymmetry: while in the standard type theory it is possible to obtain terms of a basic (e or s) type by application, this is not so in relational type theory; the results of clauses (1) and (2) are always relations. In the functional theory the result of applying a (say)

(ee)-type function to an e-type argument gives a value of type e, but in the relational formulation the same function, seen as an <ee>-type relation now, applied to the same argument, gives an <e>-type singleton as a result. To get the original value we need a *description operator*.

Since a description operator is generally useful, we may add it to the logic⁴ and define $\iota x_\alpha(\varphi)$ to be a term of type α if φ is a formula (a term of type <>) and x_α a variable of type α and demand that at least:

(3a) $\|\iota x_\alpha(\varphi)\|^{M,a} =$ the unique $d \in D_\alpha$ such that $\|\varphi\|^{M,a[d/x]} = 1$, if there is such an object $d \in D_\alpha$

What to do if there is no such unique d ? This is a classical problem of course and it has been discussed extensively in the literature from Frege onwards (see Scott [1967], Renardel [1984] for short expositions of the main points of view). If α is a *relational* type, a type of the form $\langle \alpha_1 \dots \alpha_n \rangle$ that is, then there are two obvious candidates for the value of $\|\iota x_\alpha(\varphi)\|^{M,a}$ in case there is no unique d such that d satisfies φ : we can either let it be the empty set or we can take it to be the cartesian product $D_{\alpha_1} \times \dots \times D_{\alpha_n}$. Following Frege, we shall take the first option. If, on the other hand, α is *basic*, that is if $\alpha=e$ or $\alpha=s$, we must proceed in some other way.

To this end we shall follow Scott [1967] in distinguishing between the proper objects of some basic type and an *improper* one, designed to be the 'non-referent' of non-referring expressions. The proper objects are just those entities that you have always allowed in your domain D_e or just those worlds you allow in D_s . To those we now add an improper one. Since we can - up to isomorphism - take any set to play the part of this object, we might as well choose \emptyset again for uniformity's sake. So from now on we shall assume that $\emptyset \in D_e$ and $\emptyset \in D_s$ and we demand that:

(3b) $\|\iota x_\alpha(\varphi)\|^{M,a} = \emptyset$, if there is no unique $d \in D_\alpha$ such that $\|\varphi\|^{M,a[d/x]} = 1$

The sets $D_e \setminus \{\emptyset\}$ and $D_s \setminus \{\emptyset\}$ of entities and worlds proper, we shall denote by E_e and E_s respectively and we inductively define $E_{\langle \alpha_1 \dots \alpha_n \rangle}$ to be $P(E_{\alpha_1} \times \dots \times E_{\alpha_n})$. Note that $D_\alpha \supseteq E_\alpha$ for each type α .

⁴ It should be emphasized however, that the addition of description operators is not essential. Relational type theories can be formulated without them. See §3.

Thus, for each type α , we now have *two* domains of quantification (and lambda-abstraction and description): D_α and E_α . The E_α are domains that either consist of individuals, or of relations over those, or over relations over those relations, etcetera, etcetera. The D_α are somewhat larger, in order to allow for improper objects in the basic domains. I shall call D_α the *outer domain* of type α and E_α the *inner domain* of this type.⁵ Since we have only enlarged the inner domain in order to evaluate non-referring expressions, it seems reasonable to demand that in applications of the logic to natural language semantics all expressions except the non-referring ones should get their values in some inner domain; non-referring expressions may take their values outside the E_α 's. If we adopt this rule we must slightly revise the general rule for the interpretation of natural language negation. Since, for example, both the verb phrases **is bald** and **is not bald** are referring expressions (and therefore should take their denotations in some E_α), and since the second expression is the negation of the first, we can no longer treat negation as complementation with respect to some D_α . Instead, negation should be treated as complementation with respect to the relevant inner domain. If we do this, Scott's treatment of the iota-operator makes it possible to give e-type translations to natural language descriptions and have them behave in a Russellian way. Consider the famous sentence:

(4) **The present king of France is bald,**

which may be formalized by:

(5) $\text{bald}_1x(\text{king}x)$

(where both *bald* and *king* are type $\langle e \rangle$ constants).

In a model M where there is no unique king of France, such that $\|\text{king}\|^M$ is not a singleton, the interpretation of **the present king of France**, $\|\iota x(\text{king}x)\|^M$, will be equal to \emptyset . Since, as was assumed, $E_e \supseteq \|\text{bald}\|^M$, but $\emptyset \notin E_e$, rule (1) will ensure that $\|\text{bald}_1x(\text{king}x)\|^M = 0$, so the sentence is false in M . Of course this implies that its direct negation

(6) **It is not the case that the present king of France is bald**

is true in M . On the other hand the sentence

⁵ In this introduction we shall restrict our attention to abstraction and description operators over the outer domains, but in section 2 we shall define operators over the inner domains from those. The inner operators are best suited to the formalization of natural language, but the outer operators are useful in setting up the logical system and we shall take an outer operator as a primitive.

(7) **The present king of France is not bald,**

containing a verb phrase negation, will come out false in M. Again, since the interpretation of the verb phrase **is not bald** is a subset of E_e (the complement of $\|bald\|^M$ in E_e) and $\|\iota x(kingx)\|^M = \emptyset \notin E_e$, rule (1) ensures (7)'s falsity in M.

In recent times there has been a tendency to view all terms allowing discourse anaphora (proper nouns, definite descriptions and even indefinites) as being of type e. The following quotation is from Partee [1986] (I have changed both the example numbering and the type notation):

Evidence for type e. The claim that proper names are basically of type e and only derivatively of type ((et)t) hardly needs defense, and there is almost as much tradition (though more controversy) behind the treatment of singular definite descriptions as entity-denoting expressions. However, there seemed to be no harm and considerable gain in uniformity in following Montague's practice of treating these NP's always as ((et)t), until attention was turned to the relation between formal semantics and discourse anaphora by the work of Kamp [1981] and Heim [1982]. As illustrated in examples (8) and (9), not only proper names and definites license discourse anaphora, but indefinites as well; other more clearly "quantificational" NP's do not.

(8) John / the man / a man walked in. He looked tired.

(9) Every man / no man / more than one man walked in. *He looked tired.

The generalization seems to be that while any singular NP can bind a singular pronoun in its (c-command or f-command) domain, only an e-type NP can license a singular discourse pronoun. The analysis of indefinites is particularly crucial to the need for invoking type e in the generalization, since if it were only proper names and definite descriptions which licensed discourse anaphora, one could couch the generalization in terms of the retrievability of a unique individual from the standard Montagovian generalized quantifier interpretation (an ultrafilter in those cases).

Thus far, we have sketched a logic in which it is possible to treat both proper names and definite descriptions as e-type expressions. It still is not possible to treat indefinite descriptions in this way. If one wishes to do this, the most obvious idea that comes to one's mind is to add not only a definite description operator to the logic, but also an *indefinite* description operator. Can this be done?

An interesting thing is that it already *has* been done. In Church [1940] the author takes 'selection operators' (somewhat misleadingly denoted by iotas) as some of his logical primitives. These operators are constants of types $((\alpha t)\alpha)$ (in our notation), so intuitively they take sets of objects of type α to objects of type α . Church then proposes two alternative axiom

schemes that should govern the behaviour of these iotas. The first gives a set of *axioms of descriptions* : the iotas should assign to singletons their unique elements. This is of course still in line with the usual interpretation of the iota symbol. But the latter remark holds not true for the second, stronger, axiom scheme that Church proposes. This scheme gives *axioms of choice* : the iotas should pick out some element from *every* non-empty set, which makes the symbol into an indefinite description operator. Henkin, in his famous article in which the generalized completeness of Church's system is proved (Henkin [1950]), gives a (very sketchy) semantics for the selection operator that seems to be close in spirit to the semantics that we shall give to our indefinite description operator in section 2 below.

Another proof-theoretical treatment of an indefinite description operator was given by Hilbert & Bernays in their classical *Grundlagen der Mathematik* (Hilbert & Bernays [1939]), to which Church acknowledges a debt. It often happens in mathematical texts that when a statement of the form

(10) There are x such that φ

is derived, the author continues with a statement like

(11) Now let a be an arbitrary x such that φ ...

It is easy to reason away such talk about arbitrary objects by translating the whole mathematical argument in question into standard predicate logic. But Hilbert & Bernays do not take such a course. Instead, they take the arbitrary φ seriously, give it a name, $\epsilon x(\varphi)$, treat this as a term, and give axioms ruling its proof theory (first-order equivalent to Church's axiom of choice for $\alpha=e$). The ordinary quantifiers can then be defined using ϵ -terms and ordinary quantification theory can be derived from their ϵ -calculus.

What is the appropriate semantics for Hilbert's ϵ -symbol? Hilbert & Bernays themselves give none, since they are only interested in proof-theoretical investigations, but a semantics is given in Asser [1957] (see also Leisenring [1969]). Asser uses *choice functions*, choosing an element from every non-empty subset of the domain. The value of the term $\epsilon x(\varphi)$ in some model M is then a choice from the set of objects that satisfy φ in M . This seems a good way to interpret the indefinite description operator.

Again the classical problem arises: what if the set of φ 's is empty? Asser considers two possibilities to solve this problem. Either one can let choice

functions assign some arbitrary element of the domain to the empty set or one can leave them undefined for that set. As Leisenring correctly remarks, the first option gives a nice semantics for Hilbert's ε -symbol, but the second one is better suited to the interpretation of the η -symbol, another indefinite description operator that Hilbert & Bernays consider shortly.⁶

It is this η -symbol that we shall take as a logical primitive in this paper, and we shall let it be interpreted in a manner that resembles Asser's second way. Thus, the value of a term $\eta x(\varphi)$ in a model M will be an arbitrary x such that φ (given by the choice function on M) if there are φ 's in M and it will be \emptyset if there are none. The usual variable-binding operators (to wit the lambda-operator, the iota-operator and the quantifiers, both the 'inner' and the 'outer' versions) as well as the epsilon-operator can then be defined using η and the propositional connectives.

The next section of this paper will be devoted to a presentation of the technical details of all this. In the third section, for its own interest and for the benefit of those readers that may have some misgivings about basing the logic on an indefinite description operator, it will be shown that our system is conservative over the higher order predicate calculus.

⁶Hilbert's η -symbol has entered the linguistic literature on a modest scale via Reichenbach [1947].

2 The system TT_2

In this section I shall present a formal development of the logical system TT_2 , a two-sorted relational type theory with an indefinite description operator.

2.1 Syntax and semantics

Symbols come in four kinds. First, for each type α , we shall assume the existence of a set of *constants* of type α . There are two special constants, denoted by \perp and \rightarrow , of types $\langle \rangle$ and $\langle \langle \rangle \langle \rangle \rangle$ respectively, called *logical constants*. They will get a fixed interpretation. All other constants are called *non-logical*. The set of non-logical constants forms the *language* L . Second, for each type α , there are a denumerably infinite set of *free variables* of type α and, third, there are a countable infinity of *bound variables* of type α .⁷ I shall sometimes, but not always, indicate the type of a constant or a free or bound variable by a subscript. Fourth, there are four improper symbols, denoted by $(,)$, η and $=$. It is clear that the four sets of symbols should be disjoint. If σ and σ' are strings of symbols and s is a symbol, then $[\sigma'/s]\sigma$ denotes the string of symbols obtained from σ by replacing every occurrence of s in σ by σ' .⁸

DEFINITION 7. We define, for each α , the set of *terms* of that type by the following inductive definition:

- i. Every constant or free variable of some type α is a term of that type.
- ii. If A is a term of type $\langle \beta \alpha_1 \dots \alpha_n \rangle$ and B is a term of type β , then (AB) is a term of type $\langle \alpha_1 \dots \alpha_n \rangle$.
- iii. If A and B are terms of equal type, then $A=B$ is a term of type $\langle \rangle$ (a *formula*).
- iv. If ϕ is a formula, u_α a free variable of type α and x_α a free variable of that type and u does not occur in any substring of ϕ of the form $\eta x(\sigma)$, where an equal number of left and right brackets occur in σ , then $\eta x([x/u]\phi)$ is a term of type α .

A term A of type α may be denoted by A_α . I shall suppress round brackets wherever this does not lead to confusion (under the understanding that association is to the left). Terms of the form $\eta x([x/u]\phi)$ will be called η -

⁷This distinction between free and bound variables makes it easy to avoid variable clashes in cases of substitution, but it is not an essential feature of the theory.

⁸To avoid any confusion: the $[\sigma'/s]\sigma$ notation is a way to *refer* to strings. The square brackets are not part of any string.

terms. We shall often write $\eta x(\varphi(x))$, or even $\eta x\varphi(x)$, for $\eta x([x/u]\varphi)$. A term is *closed* if it contains no free variables. A closed formula is called a *sentence*; a set of sentences is a *theory*.

Now, let us turn to the semantics of the logic. We shall give a standard interpretation as well as a generalized one (see Henkin [1950]).

DEFINITION 8. A *frame* is a family of sets $\{D_\alpha \mid \alpha \text{ is a type}\}$ such that $P(D_{\alpha_1} \times \cdots \times D_{\alpha_n}) \supseteq D_\alpha$ for each type $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$ and $\emptyset \in D_\alpha$ for each α . A frame is *standard* if $D_\alpha = P(D_{\alpha_1} \times \cdots \times D_{\alpha_n})$ for each $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$.

DEFINITION 9. Let $F = \{D_\alpha\}_\alpha$ be a frame. An *interpretation* for F is a function I having the set of constants as its domain, such that $I(c) \in D_\alpha$ for each constant c of type α , and such that $I(\perp) = 0$ and $I(\rightarrow) = \{\langle 0,0 \rangle, \langle 1,1 \rangle, \langle 0,1 \rangle\}$. An *assignment* to F is a function a , taking free variables as its arguments, such that $a(u) \in D_\alpha$ if u is a free variable of type α . If a is an assignment, then $a[d/u]$ is to be the assignment a' such that $a'(v) = a(v)$ if $v \neq u$ and $a'(u) = d$.

In order to be able to interpret η -terms we need choice functions:

DEFINITION 10. A *choice function* for a set D is a function $G: P(D) \rightarrow D$ such that:

- i. $G(X) \in X$, if $D \supseteq X$ and $X \neq \emptyset$,
- ii. $G(\emptyset) = \emptyset$.

Let $F = \{D_\alpha\}_\alpha$ be a frame. A *set of choice functions* for F is a set $\{H_\alpha\}_\alpha$ such that each H_α is a choice function for D_α .

DEFINITION 11. A *weak general model* is a triple $\langle F, I, H \rangle$ such that F is a frame, I is an interpretation for F , and H is a set of choice functions for F . A weak general model is a (*standard*) *model* if its frame is standard.

A note on notation: I shall follow the convention that a weak general model M , its frame F , its interpretation I , its set of choice functions H , and all the elements of both F and H will be denoted by metalinguistic variables that carry the same superscripts.

We are now able to give a Tarski truth definition (or, more adequately expressed: a Tarski value definition) for the logic:

DEFINITION 12. The *value* $\|A\|^{M,a}$ of a term A on a weak general model M under an assignment a is defined by induction on the complexity of terms in the following way:

- i. $\|c\|^{M,a} = I(c)$ if c is a constant
 $\|u\|^{M,a} = a(u)$ if u is a free variable
- ii. $\|AB\|^{M,a} = \{ \langle d_1, \dots, d_n \rangle \mid \langle \|B\|^{M,a}, d_1, \dots, d_n \rangle \in \|A\|^{M,a} \}$
- iii. $\|A=B\|^{M,a} = 1$ iff $\|A\|^{M,a} = \|B\|^{M,a}$
- iv. $\|\eta x_\alpha([x/u]\varphi)\|^{M,a} = H_\alpha(\{d \in D_\alpha \mid \|\varphi\|^{M,a[d/u]} = 1\})$

It would have been misleading to speak of 'the value of a term *in* a weak general model' since, in general, there is no guarantee that the value of a term A_α on M will be an element of D_α . This does not effect the correctness of the definition, however. In standard models as well as in general models (to be defined below) each term A_α will find its interpretation in D_α (and we may speak of the value of a term *in* a (general) model).

We say that a formula φ is *true* in a weak general model M under an assignment a , or, alternatively, that M *satisfies* φ under a , or, to use still another phrase, that M is a weak general model *of* φ under a , if $\|\varphi\|^{M,a} = 1$. As is usual, $\|A\|^{M,a}$ depends only on the values that a assigns to the free variables actually occurring in A . So if A is a closed term, we may write $\|A\|^M$ instead of $\|A\|^{M,a}$.

The usual logical operators may be obtained by means of definition now. The following definition needs no comment:

DEFINITION 13. Let φ and ψ be formulae.

$\neg\varphi$	abbreviates	$\varphi \rightarrow \perp$
$\varphi \vee \psi$	abbreviates	$\neg\varphi \rightarrow \psi$
$\varphi \wedge \psi$	abbreviates	$\neg(\varphi \rightarrow \neg\psi)$
$\varphi \leftrightarrow \psi$	abbreviates	$(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$

We can define the (outer) quantifiers with the help of the η -operator in essentially the same way as Hilbert & Bernays defined quantification in their ε -calculus with the help of the ε -symbol:

DEFINITION 14. Let φ be a formula

$\exists x_\alpha([x/u]\varphi)$	abbreviates	$[\eta x_\alpha([x/u]\varphi)/u]\varphi$
$\forall x_\alpha([x/u]\varphi)$	abbreviates	$[\eta x_\alpha(\neg[x/u]\varphi)/u]\varphi$

We shall often write $\exists x(\varphi(x))$ or $\exists x\varphi(x)$ instead of $\exists x([x/u]\varphi)$ and $\forall x(\varphi(x))$ or $\forall x\varphi(x)$ instead of $\forall x([x/u]\varphi)$ and follow analogous conventions with respect to other variable-binding operators to be defined below.

The outer quantifiers get the interpretations we want them to have:

LEMMA 1. For any weak general model M :

$\|\exists x_{\alpha}([x/u]\varphi)\|^{M,a} = 1$ iff there is a $d \in D_{\alpha}$ such that $\|\varphi\|^{M,a[d/u]} = 1$

$\|\forall x_{\alpha}([x/u]\varphi)\|^{M,a} = 1$ iff for all $d \in D_{\alpha}$ $\|\varphi\|^{M,a[d/u]} = 1$

PROOF. This follows easily from definition 14 and the following theorem.

THEOREM 1. (Substitution Theorem). Let M be a weak general model, a an assignment for M , A a term and B a term of the same type as the free variable u , then: $\|[[B/u]A]\|^{M,a} = \|A\|^{M,a[d/u]}$, where $d = \|B\|^{M,a}$.

PROOF. This is proved by an induction on the complexity of the term A .

Turning to the definite description operator now, we define:

DEFINITION 15. $\iota x_{\alpha}([x/u]\varphi)$ abbreviates $\eta x_{\alpha} \forall y_{\alpha}(x=y \leftrightarrow [y/u]\varphi)$, where y is the first bound variable of type α in some fixed ordering that is not equal to x .

Again, it is easy to see that the abbreviation thus defined has its intended interpretation:

LEMMA 2. For any weak general model M :

$\|\iota x_{\alpha}([x/u]\varphi)\|^{M,a} = d$, if d is the unique element of D_{α} such that

$\|\varphi\|^{M,a[d/u]} = 1$,

$= \emptyset$, if there is no such unique object.

Abstraction is defined in the following manner:

DEFINITION 16. Let A be a term of type $\langle \alpha_1 \dots \alpha_n \rangle$, x a bound variable of type β and u a free variable of that type, then $\lambda x([x/u]A)$ abbreviates $\eta R \forall x(Rx = [x/u]A)$, where R is the first variable of type $\langle \beta \alpha_1 \dots \alpha_n \rangle$ that does not occur in A .

This time there is no guarantee that an expression $\lambda x A(x)$ will get its intended interpretation on a weak general model. The reason for this is that the required relation may simply not be present in the relevant domain (in that case $\lambda x A(x)$ will get the value \emptyset). We may want to restrict

our attention to those weak general models wherein all lambda-terms *do* get their intended interpretations:

DEFINITION 17. Any sentence of the form $\forall y_1 \dots y_n \exists R \forall x (Rx = [x/u]A)$, where A , x , u and R are as in definition 16, is called a *comprehension axiom*. A *general model* is a weak general model that satisfies all comprehension axioms.

All standard models are of course general models. Note that *lambda-conversion* is just another form of the comprehension axioms under our abbreviatory definitions: by definition any general model satisfies all sentences of the form $\forall y_1 \dots y_n \forall z (\lambda x ([x/u]A)z = [z/u]A)$.

It is not difficult to see that the following lemma holds:

LEMMA 3. Let M be a general model, a an assignment to M , A a term of type $\langle \alpha_1 \dots \alpha_n \rangle$ and x a variable of type β , then:

$$\|\lambda x ([x/u]A)\|^{M,a} = \{ \langle d, d_1, \dots, d_n \rangle \mid d \in D_\beta \text{ and } \langle d_1, \dots, d_n \rangle \in \|A\|^{M,a[d/u]} \}.$$

So general models conform to requirement (2) of the introduction.

The last outer operator that we consider is Hilbert's ϵ -symbol. The following definition is given in Hilbert & Bernays [1939]:

DEFINITION 18. The string $\epsilon x_\alpha ([x/u]\varphi)$ abbreviates

$\eta x_\alpha (\exists y_\alpha [y/u]\varphi \rightarrow [x/u]\varphi)$, where y is the first bound variable of type α in some fixed ordering that is not equal to x .

This gives a semantics for the ϵ -symbol that is closely analogous to that in Hermes [1965]:

LEMMA 4. For any weak general model M :

$$\begin{aligned} \|\epsilon x_\alpha ([x/u]\varphi)\|^{M,a} &= H_\alpha(\{d \in D_\alpha \mid \|\varphi\|^{M,a[d/u]} = 1\}), \text{ if there is some } d \in D_\alpha \\ &\quad \text{such that } \|\varphi\|^{M,a[d/u]} = 1. \\ &= H_\alpha(D_\alpha), \text{ else.} \end{aligned}$$

As I have written in the introduction, the outer operators are fundamental in setting up the logical system, but we need inner operators too. We have introduced an object into our models that we do not wish to consider as a 'real' object, rather we use it as a formal device to be able to treat non-referring descriptions. And since in applications we wish to quantify, abstract, etcetera only over objects that we consider to be proper, we need

to relativize the outer operators to the domains E_α , defined in the introduction. To this end we define:

DEFINITION 19. For each α the term E^α of type $\langle\alpha\rangle$ is defined by the following induction on α :

- i. E^e abbreviates $\lambda x_e(\neg x = \eta y_e \perp)$
 E^s abbreviates $\lambda x_s(\neg x = \eta y_s \perp)$
- ii. E^α abbreviates
 $\lambda R_\alpha \forall x_{\alpha_1} \dots x_{\alpha_n} (R x_{\alpha_1} \dots x_{\alpha_n} \rightarrow (E^{\alpha_1} x_{\alpha_1} \wedge \dots \wedge E^{\alpha_n} x_{\alpha_n}))$ if
 $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$.

It is clear that in general models M : $\|E^\alpha\|^M = E_\alpha$.

DEFINITION 20.

- | | | |
|-----------------------------------|-------------|---|
| $\eta^{E x_\alpha}([x/u]\varphi)$ | abbreviates | $\eta x_\alpha(E^\alpha x \wedge [x/u]\varphi)$ |
| $A_\alpha \equiv B_\alpha$ | abbreviates | $A = B \wedge E^\alpha A$ |

LEMMA 5. Let M be a general model, a an assignment to M .

$$\|\eta^{E x_\alpha}([x/u]\varphi)\|^M, a = H_\alpha(\{d \in E_\alpha \mid \|\varphi\|^M, a[d/u] = 1\})$$

$$\|A_\alpha \equiv B_\alpha\|^M, a \text{ iff } \|A\|^M, a = \|B\|^M, a \text{ and } \|A\|^M, a \in E_\alpha.$$

The other inner operators are now defined with the help of these and the predicates E^α :

DEFINITION 21.

- | | | |
|-------------------------------|-------------|--|
| $\exists^{E x}([x/u]\varphi)$ | abbreviates | $E^\alpha \eta^{E x}([x/u]\varphi)$ |
| $\forall^{E x}([x/u]\varphi)$ | abbreviates | $\neg \exists^{E x} \neg ([x/u]\varphi)$ |
| $\iota^{E x}([x/u]\varphi)$ | abbreviates | $\eta^{E x} \forall^{E y} (x \equiv y \leftrightarrow [x/u]\varphi)$ |
| $\lambda^{E x}([x/u]A)$ | abbreviates | $\eta^{E R} \forall^{E x} (R x \equiv [x/u]A)$ |

It is easy to check that these operators have the semantics we want them to have:

LEMMA 6. For any general model M :

$$\|\exists^{E x_\alpha}([x/u]\varphi)\|^M, a = 1 \text{ iff there is a } d \in E_\alpha \text{ such that } \|\varphi\|^M, a[d/u] = 1;$$

$$\|\forall^{E x_\alpha}([x/u]\varphi)\|^M, a = 1 \text{ iff for all } d \in E_\alpha \|\varphi\|^M, a[d/u] = 1;$$

$$\begin{aligned}
\| \iota^{E_{X_\alpha}}([x/u]\varphi) \|^{M,a} &= d, \text{ if } d \text{ is the unique element of } E_\alpha \text{ such that} \\
&\quad \| \varphi \|^{M,a[d/u]} = 1, \\
&= \emptyset, \text{ if there is no such unique object;} \\
\| \lambda^{E_{X_\alpha}}([x/u]A) \|^{M,a} &= \{ \langle d, d_1, \dots, d_n \rangle \mid d \in E_\alpha \text{ and } \langle d_1, \dots, d_n \rangle \in \| A \|^{M,a[d/u]} \}.
\end{aligned}$$

Relational type theory enables us to generalize the notion of entailment somewhat. Not only formulae can entail another formula, but any set of terms of a relational type can entail some term of that type:

DEFINITION 22. Let $\Gamma \cup \{A\}$ be a set of terms of some type $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$. Γ entails A (Γ *g-entails* A , Γ *wg-entails* A), $\Gamma \models A$ ($\Gamma \models_g A$, $\Gamma \models_{wg} A$), if $\| A \|^{M,a} \supseteq \bigcap_{B \in \Gamma} \| B \|^{M,a}$ for all models (general models, weak general models) M and assignments a to M .

In natural language too, expressions of many categories may entail one another (see Groenendijk & Stokhof [1984]). It is obvious that definition 20 is indeed a generalization of the usual notion of entailment:

LEMMA 7. Let $\Gamma \cup \{\varphi\}$ be a set of formulae. $\Gamma \models \varphi$ ($\Gamma \models_g \varphi$, $\Gamma \models_{wg} \varphi$) iff for each model (general model, weak general model) M and assignment a to M holds that if M satisfies all $\psi \in \Gamma$ under a , then M satisfies φ under a .

DEFINITION 23. We say that two terms A and B are (*g-,wg-*)*equivalent* if both A (*g-,wg-*) entails B and B (*g-,wg-*) entails A .

2.2 Proof theory and completeness

I shall now give a standard Henkin proof to the effect that the notions \models_g and \models_{wg} , defined in the preceding section, are recursively axiomatizable. Of course, \models is not axiomatizable by Gödel's Theorem and the fact that the natural number system is categorically definable in \mathbf{TT}_2 with the standard semantics.

DEFINITION 24. All formulae of one of the following forms are *axioms* :

Propositional axioms:

AS1 $\varphi \rightarrow (\psi \rightarrow \varphi)$

AS2 $(\varphi \rightarrow (\psi \rightarrow \chi)) \rightarrow ((\varphi \rightarrow \psi) \rightarrow (\varphi \rightarrow \chi))$

AS3 $((\varphi \rightarrow \perp) \rightarrow \perp) \rightarrow \varphi$

Eta axioms:

AS4 $\neg \eta_{X_\alpha}(\perp) B_1 \dots B_n$, where $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$ and each B_i is of type α_i .

AS5 $\varphi(A_\alpha) \rightarrow \varphi(\eta_{X_\alpha} \varphi(x))$

AS6 $\forall x_\alpha (\varphi(x) \leftrightarrow \psi(x)) \rightarrow \eta_{Y_\alpha} \varphi(y) = \eta_{Z_\alpha} \psi(z)$

Extensionality:

AS7 $\forall R_\alpha \forall R'_\alpha (\forall x_{\alpha_1} \dots x_{\alpha_n} (R x_{\alpha_1} \dots x_{\alpha_n} \leftrightarrow R' x_{\alpha_1} \dots x_{\alpha_n}) \rightarrow R = R')$, where $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$

Identity axioms:

AS8 $A = A$

AS9 $A = B \rightarrow (\varphi(A) \rightarrow \varphi(B))$ (Leibniz' Law).

DEFINITION 25. A *proof* for a formula $\varphi = \varphi_n$ is a sequence $\varphi_0, \dots, \varphi_n$ of formulae such that each φ_k ($k \leq n$) is either an axiom or follows from two formulae earlier in the sequence by the rule of *modus ponens* ($\varphi, \varphi \rightarrow \psi / \psi$). A formula φ is *provable*, $\vdash \varphi$, if there is a proof for it. The formula φ is said to be *derivable* from a set of formulae Γ , $\Gamma \vdash \varphi$, if there are $\varphi_0, \dots, \varphi_n \in \Gamma$ such that $\vdash (\varphi_0 \wedge \dots \wedge \varphi_n) \rightarrow \varphi$. A set of formulae is *consistent* if \perp is not derivable from it.

LEMMA 8.

i. $\vdash \forall x \varphi(x) \rightarrow \varphi(B)$

ii. If $\vdash \varphi \rightarrow \psi(u)$ and the free variable u does not occur in φ , then $\vdash \varphi \rightarrow \forall x \psi(x)$.

PROOF. Part i. of the lemma follows easily from AS6 and the propositional axioms AS1-AS3. To prove part ii., suppose that χ_0, \dots, χ_n is a proof for $\varphi \rightarrow \psi(u)$; it is not difficult to verify that $[\eta x \neg \varphi(x)/u] \chi_0, \dots, [\eta x \neg \varphi(x)/u] \chi_n$ is then a proof for $\varphi \rightarrow \forall x \psi(x)$.

From lemma 8 it follows that the usual quantification theorems are provable for \exists and \forall .

THEOREM 2. Let T be a theory and φ a formula, then $T \vdash \varphi \Rightarrow T \models_{wg} \varphi$

PROOF. By a straightforward induction on the length of proofs.

THEOREM 3. Let T be a theory and φ a formula, then $T \models_{wg} \varphi \Rightarrow T \vdash \varphi$

PROOF. This is proved in the usual way, with the help of the Consistency Theorem below.

COROLLARY (Generalized Completeness Theorem). Let T be a theory and φ a formula, then $T \models_g \varphi \Leftrightarrow T \cup \text{COMP} \vdash \varphi$, where COMP is the set of all comprehension axioms.

THEOREM 4 (Consistency Theorem). If a theory is consistent, then it has a weak general model.

PROOF. Let T be a consistent theory. We construct a maximal consistent set of sentences $\Gamma \supseteq T$ having the so-called Henkin property. To this end

let A_0, \dots, A_n, \dots be some enumeration of all terms (of all types). For each $n \in \omega$ define a set of formulae Γ_n by: $\Gamma_0 = T$ and $\Gamma_{n+1} = \Gamma_n \cup \{A_n = u\}$, where u is the first free variable in our enumeration which has the same type as A_n has and which does neither occur in A_n nor in any of the formulae in Γ_n . Clearly, all Γ_n are consistent and hence $\bigcup_n \Gamma_n$ is consistent.

Next, we expand $\bigcup_n \Gamma_n$ to a maximal consistent set by the Lindenbaum construction. Let $\varphi_0, \dots, \varphi_n, \dots$ be an enumeration of all formulae. Let $\Gamma_0' = \bigcup_n \Gamma_n$ and let $\Gamma_{n+1}' = \Gamma_n' \cup \{\varphi_n\}$ if $\Gamma_n' \cup \{\varphi_n\}$ is consistent, else $\Gamma_{n+1}' = \Gamma_n'$. The union Γ of all Γ_n' is consistent; moreover, it is *maximal* (for each φ either $\varphi \in \Gamma$ or $\Gamma \cup \{\varphi\} \vdash \perp$) and - by the construction in the previous paragraph, the properties of maximal consistent sets of formulae and Leibniz' Law - it has the *Henkin property* : if $\varphi(u) \in \Gamma$ for all free variables u of some type α , then (since $\eta x \neg ([x/u]\varphi) = u' \in \Gamma$ for some u') $\forall x \varphi(x) \in \Gamma$.

Now define an equivalence relation \sim between terms: $A \sim B$ iff $A = B \in \Gamma$. For each term A , let $[A]$ be the equivalence class $\{B \mid A \sim B\}$. For each type α we define a function Φ_α having the set $\{[A] \mid A \text{ is a term of type } \alpha\}$ as its domain. If $\alpha = e$ or $\alpha = s$, then let $\Phi_\alpha([\eta x_\alpha \perp]) = \emptyset$ and let $\Phi_\alpha([A_\alpha]) = [A]$ if $[A] \neq [\eta x_\alpha \perp]$. If $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$ let

$\Phi_\alpha([A_\alpha]) = \{ \langle \Phi_{\alpha_1}([B_{\alpha_1}]), \dots, \Phi_{\alpha_n}([B_{\alpha_n}]) \rangle \mid AB_{\alpha_1} \dots B_{\alpha_n} \in \Gamma \}$. This is well-defined by Leibniz' Law and the maximal consistency of Γ .

The functions Φ_α are injections. This is obvious in case $\alpha = e$ or $\alpha = s$, so let $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$. Suppose that $\Phi_\alpha([A]) = \Phi_\alpha([A'])$. Let u_1, \dots, u_n be free variables of types $\alpha_1, \dots, \alpha_n$ respectively, then $Au_1 \dots u_n \in \Gamma$ iff $A'u_1 \dots u_n \in \Gamma$. From this it follows that $Au_1 \dots u_n \leftrightarrow A'u_1 \dots u_n \in \Gamma$. By the Henkin property: $\forall x_1 \dots x_n (Ax_1 \dots x_n \leftrightarrow A'x_1 \dots x_n) \in \Gamma$, so, using Extensionality and the maximal consistency of Γ , we see that $A = A' \in \Gamma$ and $[A] = [A']$.

For each type α define: $D_\alpha = \{ \Phi_\alpha([A]) \mid A \text{ is a term of type } \alpha \}$. From the definition of the functions Φ_α and the fact that AS4 is an axiom scheme it follows that $F = \{D_\alpha\}_\alpha$ is a frame. Define $I(c_\alpha) = \Phi_\alpha([c_\alpha])$ and define for each α and each D such that $D_\alpha \supseteq D$:

$$\begin{aligned} H_\alpha(D) &= \Phi_\alpha([\eta x_\alpha \varphi(x)]), \text{ if } D = \{ \Phi_\alpha([u]) \mid \varphi(u) \in \Gamma \}; \\ &= \Phi_\alpha([u]), \text{ where } u \text{ is the first free variable such that} \\ &\quad \Phi_\alpha([u]) \in D, \text{ else.} \end{aligned}$$

The functions H_α are well-defined. First, note that $\emptyset = \{\Phi_\alpha([u]) \mid \perp \in \Gamma\}$, so the second clause is all right. Next, suppose that for all free variables u_α : $\varphi(u) \in \Gamma$ iff $\psi(u) \in \Gamma$. Then $\forall x(\varphi(x) \leftrightarrow \psi(x)) \in \Gamma$ and by AS6: $\eta x \varphi(x) = \eta y \psi(y) \in \Gamma$, from which it follows that $[\eta x \varphi(x)] = [\eta y \psi(y)]$.

The functions H_α are choice functions for the sets D_α . Clearly $H_\alpha(\emptyset) = \Phi_\alpha([\eta x_\alpha \perp])$. Suppose that $D \neq \emptyset$. If the second clause of H_α 's definition obtains, then obviously $H_\alpha(D) \in D$; if, on the other hand, $D = \{\Phi_\alpha([u]) \mid \varphi(u) \in \Gamma\}$ for some φ , then $\varphi(u) \in \Gamma$ for some u and, by AS5 and the properties of Γ , $\varphi(\eta x \varphi(x)) \in \Gamma$. Since $u' = \eta x \varphi(x) \in \Gamma$ for some u' , both $\varphi(u') \in \Gamma$ and $[u'] = [\eta x \varphi(x)]$. Hence $\Phi_\alpha([\eta x \varphi(x)]) \in D$.

Now, let M be the model $\langle F, I, H \rangle$ and let the assignment a be defined by $a(u_\alpha) = \Phi_\alpha([u])$. We prove by induction on term complexity that $\|A\|^{M,a} = \Phi_\alpha([A])$ for all terms A of type α , hence that $\|\varphi\|^{M,a} = 1$ iff $\varphi \in \Gamma$, for all formulae φ and hence that M is a weak general model of T :

- i. $\|c\|^{M,a} = I(c) = \Phi([c])$ if c is a constant;
 $\|u\|^{M,a} = a(u) = \Phi([u])$ if u is a free variable;
- ii. $\|AB\|^{M,a} = \{ \langle d_1, \dots, d_n \rangle \mid \langle \|B\|^{M,a}, d_1, \dots, d_n \rangle \in \|A\|^{M,a} \} =$
 $\{ \langle d_1, \dots, d_n \rangle \mid \langle \Phi([B]), d_1, \dots, d_n \rangle \in \Phi([A]) \} = \Phi([AB]);$
- iii. $\|A=B\|^{M,a} = 1 \Leftrightarrow \|A\|^{M,a} = \|B\|^{M,a} \Leftrightarrow \Phi([A]) = \Phi([B]) \Leftrightarrow [A] = [B]$
 $\Leftrightarrow A=B \in \Gamma \Leftrightarrow \Phi([A=B]) = 1;$
- iv. $\|\eta x_\alpha([x/u]\varphi)\|^{M,a} = H_\alpha(\{d \in D_\alpha \mid \|\varphi\|^{M,a[d/u]} = 1\}) =$
 $H_\alpha(\{\Phi_\alpha([u]) \mid \|\varphi\|^{M,a} = 1\}) = H_\alpha(\{\Phi_\alpha([u]) \mid \varphi \in \Gamma\}) = \Phi_\alpha([\eta x([x/u]\varphi)]).$

3 Conservativity over the higher order predicate calculus

In this section I shall compare \mathbf{TT}_2 with the higher-order predicate calculus as it was defined in Orey [1959] (see also Gallin [1975]). It will be proved that \mathbf{TT}_2 is a conservative extension of this logic.

In \mathbf{TT}_2 terms are evaluated with the help of choice functions. There is a wide class of terms, however, whose evaluations do not depend on choice functions:

DEFINITION 26. A term A of any type is called *choice independent* if for any two general models M and M' such that $F=F'$ and $I=I'$ and any assignment a for M : $\|A\|^{M,a}=\|A\|^{M',a}$.

This semantical notion can be characterized syntactically:

DEFINITION 27. A term A is called an *{application, λ , ι , =}-term* if it is built up in the usual way with the help of non-logical constants, free variables, application, abstraction, definite description and identity. A formula that is an *{application, λ , ι , =}-term* is called an *{application, λ , ι , =}-formula*.

So, in *{application, λ , ι , =}-terms*, *etas* may occur, but only in certain contexts.

THEOREM 5. A formula is choice independent iff it is *g-equivalent* to a *{application, λ , ι , =}-formula*.

PROOF. That any formula that is equivalent to an *{application, λ , ι , =}-formula* is choice independent is trivial. Let ϕ be a choice independent formula. For each type α and each term G of type $\langle\langle\alpha\rangle\alpha\rangle$ let ' G is a choice function' be an abbreviation of the conjunction of the following three formulae:

1. $\forall Y_{\langle\alpha\rangle}\exists x_{\alpha}\forall z_{\alpha}((GY)z\leftrightarrow x=z)$
2. $\forall Y_{\langle\alpha\rangle}(\exists x_{\alpha}Yx \rightarrow \iota x((GY)x)$
3. $\forall Y_{\langle\alpha\rangle}(\neg\exists x_{\alpha}Yx \rightarrow \iota x((GY)x)=\iota x(\perp))$

Clearly, if M and a satisfy ' $G_{\langle\langle\alpha\rangle\alpha\rangle}$ is a choice function' then $\|G\|^{M,a}$ is indeed a choice function on D_{α} . Now, for each type α such that some η -term $\eta x_{\alpha}\psi(x)$ occurs in ϕ , let $F_{\langle\langle\alpha\rangle\alpha\rangle}$ be a bound variable of type $\langle\langle\alpha\rangle\alpha\rangle$ and let F_1, \dots, F_n be the bound variables that can be obtained in this way. Define ϕ' to be the formula

$\exists F_1 \dots \exists F_n (F_1 \text{ is a choice function} \wedge \dots \wedge F_n \text{ is a choice function} \wedge \varphi')$, where φ' is obtained from φ by replacing each η -term $\eta x_\alpha \psi(x)$ by $\iota z_\alpha (F \lambda x (\psi(x)) z)$, where $F = F_{\langle \alpha \rangle}$. The formula φ' is not an $\{\text{application}, \lambda, \iota, =\}$ -formula itself, but it is g-equivalent to one, since the propositional connectives and the quantifiers can be defined with the help of abstraction, application and identity alone (see Henkin [1963]). On the other hand, using the choice independency of φ , it can easily be shown that φ and φ' are g-equivalent. So φ is g-equivalent to an $\{\text{application}, \lambda, \iota, =\}$ -formula.

COROLLARY. A term is choice independent iff it is g-equivalent to a $\{\text{application}, \lambda, \iota, =\}$ -term.

To facilitate comparison with the higher order predicate calculus, we may describe the latter as a subsystem of our logic:

DEFINITION 28. An *O-term* (Orey-term) is either a non-logical constant or a free variable. An *atomic formula* is either a formula of the form $AB_1 \dots B_n$, where A is an O-term of type $\langle \alpha_1 \dots \alpha_n \rangle$ and each B_i is an O-term of type α_i , or it is a formula of the form $A=B$, where both A and B are O-terms, or it is \perp . The set of *O-formulae* is the smallest set such that any atomic formula is an O-formula and if φ and ψ are O-formulae, then both $\varphi \rightarrow \psi$ and $\forall x_\alpha ([x/u]\varphi)$ are O-formulae.

$\{\text{application}, \lambda, \iota, =\}$ -formulae can almost be reduced to O-formulae. The reason that the reduction cannot be complete is that in all weak general models the terms $\iota x_e(\perp)$ and $\iota x_s(\perp)$ denote the improper object \emptyset , while in some weak general models there may be no constants that name this object:

THEOREM 6. Let φ be a $\{\text{application}, \lambda, \iota, =\}$ -formula in a language L ; let $*_e$ and $*_s$ be two non-logical constants, of types e and s respectively, that are not elements of L . There is an O-formula φ' in the language $L \cup \{*_e, *_s\}$ such that for each weak general model M such that $I(*_e) = I(*_s) = \emptyset$ and each assignment a for M : $\|\varphi\|^{M,a} = \|\varphi'\|^{M,a}$.

PROOF. The proof combines proofs in Gallin [1975] and Scott [1968]. For each $\{\text{application}, \lambda, \iota, =\}$ -term A_α in the language L and each free variable u_α not occurring in A , define the O-formula $\text{Eq}(A, u)$ (A equals u) in the language $L \cup \{*_e, *_s\}$ by induction on the term A :

- i. $\text{Eq}(u',u) := u'=u$, if u' is a free variable;
 $\text{Eq}(c,u) := c=u$, if c is a non-logical constant;
- ii. $\text{Eq}(BC,u) :=$
 $\exists R \exists y (\text{Eq}(B,R) \wedge \text{Eq}(C,y) \wedge \forall x_{\alpha_1} \dots x_{\alpha_n} (u x_{\alpha_1} \dots x_{\alpha_n} \leftrightarrow R y x_{\alpha_1} \dots x_{\alpha_n}))$,
 if B is a term of type $\langle \beta_{\alpha_1} \dots \alpha_n \rangle$ and C a term of type β ;
- iii. $\text{Eq}(\lambda x_{\beta} ([x/u']B), u) :=$
 $\forall x_{\beta} \exists R (\text{Eq}([x/u']B, R) \wedge \forall y_{\alpha_1} \dots y_{\alpha_n} (u x y_{\alpha_1} \dots y_{\alpha_n} \leftrightarrow R y_{\alpha_1} \dots y_{\alpha_n}))$, if
 B is a term of type $\langle \alpha_1 \dots \alpha_n \rangle$;
- iv. $\text{Eq}(\iota x_{\alpha} ([x/u']\phi), u_{\alpha}) :=$
 $\forall x (\text{Eq}([x/u']\phi, \perp \rightarrow \perp) \leftrightarrow x=u) \vee (\neg \exists y \forall x (\text{Eq}([x/u']\phi, \perp \rightarrow \perp) \leftrightarrow x=u) \wedge \xi)$,
 where $\xi := \neg \exists x_{\alpha_1} \dots x_{\alpha_n} (u x_{\alpha_1} \dots x_{\alpha_n})$ if $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$ and $\xi := u = *_{\alpha}$ if
 $\alpha = e$ or $\alpha = s$;
- v. $\text{Eq}(A=B, u_{\langle \rangle}) := \exists xy (\text{Eq}(A, x) \wedge \text{Eq}(B, y) \wedge u \leftrightarrow (x=y))$.

Define ϕ' to be $\text{Eq}(\phi, \perp \rightarrow \perp)$ and the theorem is proved.

We now give a quick sketch of Orey's logic, so that we can compare it to ours.

DEFINITION 29. An *Orey frame* is a family of sets $\{D_{\alpha} \mid \alpha \text{ is a type}\}$ such that $D_e \neq \emptyset$, $D_s \neq \emptyset$ and $P(D_{\alpha_1} \times \dots \times D_{\alpha_n}) \supseteq D_{\langle \alpha_1 \dots \alpha_n \rangle}$. A *general model in Orey's sense* is a tuple $\langle F, I \rangle$ such that F is an Orey frame and I is an interpretation function for F . A *(standard) model in Orey's sense* is a general model in Orey's sense such that its Orey frame is standard (see definition 2).

DEFINITION 30. Let M and M' be general models in Orey's sense. An *isomorphism* from M onto M' is a set of functions $\{h_{\alpha}\}_{\alpha}$ such that:

- i. Each h_{α} is a bijection from D_{α} onto D_{α}' .
- ii. If $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$, $R \in D_{\alpha}$ and $d_1 \in D_{\alpha_1}, \dots, d_n \in D_{\alpha_n}$ then $\langle d_1, \dots, d_n \rangle \in R$
 $\Leftrightarrow \langle h_{\alpha_1}(d_1), \dots, h_{\alpha_n}(d_n) \rangle \in h_{\alpha}(R)$
- iii. $I'(c_{\alpha}) = h_{\alpha}(I(c_{\alpha}))$ for each constant c_{α} .

M and M' are said to be *isomorphic* if there exists an isomorphism from M onto M' .

DEFINITION 31. Define the *value* of an O-term t in a (general) model in Orey's sense M under an assignment a , $\|t\|^{M,a}$, to be equal to $I(c)$ if $t=c$ for some constant c , and to $a(u)$ if $t=u$ for some free variable u . Define the satisfaction relation $M \models \phi[a]$ (the (general) model in Orey's sense M *satisfies* the O-formula ϕ *under* the assignment a) by the following inductive clauses:

- i. $M \models \perp[a] \Leftrightarrow 0=1$;
- ii. $M \models R_{\alpha} t_{\alpha_1} \dots t_{\alpha_n}[a] \Leftrightarrow \langle \|t_{\alpha_1}\|^{M,a}, \dots, \|t_{\alpha_n}\|^{M,a} \rangle \in \|R_{\alpha}\|^{M,a}$, if $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$;
- iii. $M \models t=t'[a] \Leftrightarrow \|t_{\alpha}\|^{M,a} = \|t'_{\alpha}\|^{M,a}$;
- iv. $M \models \phi \rightarrow \psi[a] \Leftrightarrow$ if $M \models \phi[a]$ then $M \models \psi[a]$;
- v. $M \models \forall x_{\alpha} \phi[a] \Leftrightarrow$ for each $d \in D_{\alpha}$: $M \models \phi[a[d/u]]$.

DEFINITION 32. A set of O-formulae Γ *O-entails* (*O,g-entails*) an O-formula ϕ , $\Gamma \models_O \phi$ ($\Gamma \models_{O,g} \phi$), if for each (general) model in Orey's sense M and assignment a for M such that $M \models \psi[a]$ for all $\psi \in \Gamma$ it also holds that $M \models \phi[a]$.

It is clear that if $\langle F, I, H \rangle$ is a (weak general) model then $\langle F, I \rangle$ is a (general) model in Orey's sense. On the other hand, if the general model in Orey's sense M' satisfies NULL, the set of all sentences of the form $\exists R_{\alpha} \neg \exists x_{\alpha_1} \dots x_{\alpha_n} (R x_{\alpha_1} \dots x_{\alpha_n})$, where $\alpha = \langle \alpha_1 \dots \alpha_n \rangle$, then M' is isomorphic to a general model in Orey's sense $\langle F, I \rangle$ that can be extended, using the axiom of choice, to a weak general model $\langle F, I, H \rangle$. Standard models in Orey's sense always satisfy NULL and can always be extended to our standard models. From this and the previous theorem it follows that we can embed the {application, λ , ι , $=$ }-part of the logic into the higher order predicate calculus (with extra constants $*_e$ and $*_s$):

THEOREM 7 (AC). Let $\Gamma \cup \{\phi\}$ be a set of {application, λ , ι , $=$ }-formulae. For each {application, λ , ι , $=$ }-formula ψ , let ψ' be as in theorem 6 and let $\Gamma' = \{\psi' \mid \psi \in \Gamma\}$; then:

- i. $\Gamma' \cup \text{NULL} \models_{O,g} \phi'$ iff $\Gamma \models_{wg} \phi$;
- ii. $\Gamma' \models_O \phi'$ iff $\Gamma \models \phi$.

The conservativity of \mathbf{TT}_2 over the higher order predicate logic now follows as a corollary:

COROLLARY. Let $\Gamma \cup \{\phi\}$ be a set of O-formulae, then:

- i. $\Gamma \cup \text{NULL} \models_{O,g} \phi$ iff $\Gamma \models_{wg} \phi$;
- ii. $\Gamma \models_O \phi$ iff $\Gamma \models \phi$.

Acknowledgements

I would like to thank Johan van Benthem and Dick de Jongh for their moral support and their critical comments on earlier versions of this paper. Martin Stokhof and Elias Thijssse also made very valuable remarks.

The research for this paper was supported by the Netherlands Organization for the Advancement of Pure Research (Z.W.O.).

References

- Asser, G.: 1957, Theorie der logischen Auswahlfunktionen, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik* 3, 30-68.
- Barwise, J. and Cooper R.: 1981, Generalized Quantifiers and Natural Language, *Linguistics and Philosophy* 4, 159-219.
- Bennett, M.: 1974, *Some Extensions of a Montague Fragment of English*, UCLA, Dissertation.
- Van Benthem, J.F.A.K. and Doets, K.: 1983, Higher-Order Logic, in Gabbay & Guentner [1983], pp. 275-329.
- Carnap, R.: 1929, *Abriss der Logistik*, Vienna.
- Church, A.: 1940, A Formulation of the Simple Theory of Types, *The Journal of Symbolic Logic* 5, 56-68.
- Gabbay, D. and Guentner, F. (eds.): 1983, *Handbook of Philosophical Logic*, Vol. 1, Dordrecht.
- Gallin, D.: 1975, *Intensional and Higher-Order Modal Logic*, Amsterdam-Oxford.
- Gazdar, G.: 1980, A Cross-Categorial Semantics for Coordination, *Linguistics and Philosophy* 3, 407-409.
- Groenendijk, J. and Stokhof, M.: *Studies on the Semantics of Questions and the Pragmatics of Answers*, Dissertation, University of Amsterdam.
- Heim, I.: 1982, *The Semantics of Definite and Indefinite Noun Phrases*, Dissertation, University of Massachusetts.
- Henkin, L.: 1950, Completeness in the Theory of Types, *The Journal of Symbolic Logic* 15, 81-91.
- Henkin, L.: 1963, A Theory of Propositional Types, *Fundamenta Mathematicae* 52, 323-344.
- Hermes, H.: 1965, *Eine Termlogik mit Auswahloperator*, Berlin-Heidelberg-New York.
- Hilbert, D. and Bernays, P.: 1939, *Grundlagen der Mathematik*, Vol. 2, Berlin.
- Hodges, W.: 1983, Elementary Predicate Logic, in Gabbay & Guentner [1983], pp. 1-131.
- Janssen, T.: 1984, Individual Concepts are Useful, in F. Landman and F. Veltman (eds.), *Varieties of Formal Semantics*, Dordrecht, 171-192.
- Kamp, H.: 1981, A Theory of Truth and Semantic Representation, reprinted in J. Groenendijk, T. Janssen and M. Stokhof (eds.), *Truth, Interpretation and Information*, Amsterdam, 1984, 277-322.
- Keenan, E. and Faltz, L.: 1978, *Logical Types for Natural Language*, UCLA Occasional Papers in Linguistics, 3.
- Leisenring, A.C.: 1969, *Mathematical Logic and Hilbert's ϵ -Symbol*, London.
- Montague, R.: 1973, The Proper Treatment of Quantification in Ordinary English, reprinted in Montague [1974], pp. 247-270.
- Montague, R.: 1974, *Formal Philosophy*, New Haven.
- Muskens, R.A.: 1983, *Over de semantiek van adjectieven*, unpublished manuscript, University of Amsterdam.
- Muskens, R.A.: in preparation, *Submodels and the Semantics of Degree Adjectives* (working title).
- Orey, S.: 1959, Model Theory for the Higher Order Predicate Calculus, *Transactions of the American Mathematical Society* 92, 72-84.
- Partee, B.: 1986, Noun Phrase Interpretation and Type-Shifting Principles,
- Partee, B. and Rooth, M.: 1983, Generalized Conjunction and Type Ambiguity, in R. Bauerle, C. Schwarze and A. von Stechow (eds.), *Meaning, Use and Interpretation of Language*, Berlin, 361-383.
- Reichenbach, H.: 1947, *Elements of Symbolic Logic*, New York.

- Renardel de Lavalette, G.: 1984, Descriptions in Mathematical Logic, *Studia Logica* **43**, 281-294.
- Russell, B.: 1908, Mathematical Logic as Based on the Theory of Types, *American Journal of Mathematics* **30**, 222-262.
- Schönfinkel, M.: 1924, Über die Bausteine der mathematischen Logik, *Mathematische Annalen* **92**, 305-316.
- Scott, D.: 1967, Existence and Description in Formal Logic, in R. Schoenman (ed.), *Bertrand Russell, Philosopher of the Century*, London, 660-696.
- Von Stechow, A.: 1974, ε - λ kontextfreie Sprachen: Ein Beitrag zu einer natürlichen formalen Semantik, *Linguistische Berichte* **34**, 1-33.