FREGE'S THEORY OF FUNCTIONS
IN APPLICATION TO
LINGUISTIC STRUCTURES

Antoni Romuald Diller

Submitted in accordance with the requirements

for the degree of Doctor of Philosophy

The University of Leeds

The Department of Philosophy

December 1986

Abstract

Introduction One of Frege's greatest contributions to logic was to extend the meaning of the word 'function' further than previous writers. I explore some of the ways in which his theory of functions can be applied to natural language.

Chapter 1 Frege's notion of an unsaturated expression is best understood in terms of - what Geach calls - a linguistic function. I investigate the consequences of such an interpretation.

Chapter 2 Geach's exegesis contains a lacuna, since there exist pathological linguistic functions which are not unsaturated expressions. Implicit in Frege's writings is a criterion for excluding these.

Chapter 3 It appears as if Frege's account of unsaturatedness is brought into question by the paradox of the concept horse. Both Geach and Dummett present solutions to this, but there are further difficulties in their solutions. I show that a sensible theory can be constructed in which expressions like 'the concept horse' are treated as genuine singular terms.

Chapter 4 The nexus of ideas involved in Dummett's distinction between simple and complex predicates is a distortion of some of Frege's most important insights. Dummett's views need to be rebuffed because they diminish the importance of the notion of a function in Frege's thought and in logic generally.

Chapter 5 Extending and generalising Frege's distinction between functions of different levels and between functions with different types of arguments results in the notion of a categorial grammar. But the syntactic coherence of all unified expressions cannot be justified by ideas derived from Frege. Geach and Potts extend the rules acceptable within a categorial grammar, but their rules are an unsatisfactory solution to these problems.

Chapter 6 In order to prepare the way for a better solution the system of combinatory logic - and the theory of functionality in particular - is presented. This is done gently, because of the prejudice against combinatory logic. It is hoped that others will be encouraged to investigate the combinatory grammar that I introduce.

Contents

Preface

In this thesis I look at Frege's notion of an unsaturated expression and the use to which he puts it in giving the formation rules of his formalised language. I also look at some of the ways in which these ideas can be applied to natural language and I come to the conclusion that natural language does not have the same multiplicity of unsaturated expressions as does Frege's formalised language. There are three main reasons for this: (a) It is possible to form phrases like 'the concept horse' in natural language - but it is impossible to do so in Frege's formalised language - and the best way to deal with them there is to think of them as singular terms. (b) Frege's formalised language contains variables, whereas natural language does not. (c) Many kinds of combination problems occur in natural language, whereas none occur in Frege's formalised language.

All this will be fully explained in the body of the thesis. Here, I just want to say something about why I say so little about Wittgenstein. I am well aware that there are many points of contact between what I say in this thesis and Wittgenstein's views. For example, in Chapter 3 I discuss the issues surrounding the fact that in natural language it is possible to form expressions like 'the concept horse'. There are connections

between Frege's views about this and Wittgenstein's views about showing and saying and also his views about formal concepts. I decided, however, against trying to explore those connections here for two main reasons. In the first place, I felt that if I were to attempt to do justice to Wittgenstein here it would distort the main thrust of my argument. Such discussions would have to be substantial and they would appear to be digressions. And, furthermore, including them would have made this thesis too long. In the second place, when I started studying Frege seriously I had the intention of undertaking a thorough investigation of Wittgenstein's philosophy when I finished my thesis on Frege. I felt that discussing Frege's influence on Wittgenstein and Wittgenstein's elaboration of Frege's views would fit better in an account of Wittgenstein's philosophy. We now, however, live in a climate in which academic philosophy is in decline and I think it unlikely that I will ever have the opportunity to study Wittgenstein's philosophy in the way I once had hoped to do.

A Note on Textual References Full details of all the books and articles that I mention in this thesis are given in the Bibliography. In the text I refer to a particular work by citing its author or authors and also its title. In the case of books written by Frege, I use an abbreviated form of the German title, namely, Begriffsschrift, Grundlagen and Grundgesetze. Letters are referred to by author, recipient and date. The translations of Frege's works that I use are noted in the Bibliography. In the case of the Begriffsschrift I have always used Bynum's

translation. In his critical study of Baker and Hacker's Frege, Dummett protests against their 'practice of referring to Frege's works solely by citing page numbers of English translations' ("An Unsuccessful Dig", p.377, footnote 1). Because of this I have decided to refer to Frege's works by citing the page numbers of the German originals. In the case of books, I often refer to a particular Section just by giving its number, rather than also including explicit page numbers. In the case of the Preface to Begriffsschrift I just to refer to it as that. It is fairly short, so no one should have any difficulty locating my quotations.

There are a number of differences between the first and second editions of Dummett's Frege. All of the passages that I either quote or refer to, however, are the same in both editions.

Acknowledgements I would like to thank my supervisors - Professor Peter Geach and Roger White - for numerous valuable discussions about Frege. I am also grateful to Peter Long for his encouragement and for many criticisms of earlier versions of my ideas. Philip Hoy read through a draft of the Introduction and most of Chapter 1 and made many helpful suggestions.

I am grateful to Birmingham University, Mr Alfred Spencer of Ralph Martindale & Co. Ltd., and my parents for financial support during the time I was developing the ideas reported here.

List of Notations

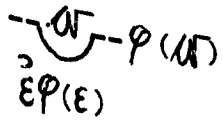
All frequently occuring notations are included here, except for familiar symbols like '+' for addition. Also not included are localised ad.hoc. notations which only occur a very small number of times. Some symbols are "overloaded" in the sense that they have several uses, but no confusion should arise as different uses of the same symbol never occur in the same context.

Fregean Symbols

ξ, ζ	Argument-place holders for complete expressions.
φ, ψ	Argument-place holders for predicates and functional signs.
Δ, Γ	Arbitrary complete expressions.
Φ, Ψ	Arbitrary predicates or functional signs.
Ω	Arbitrary second-level expression.
$\mathfrak{A}$	Bound variable in the second-level quantifier.
α, ε	Bound variables in the abstraction operator.
$\mathfrak{f}$	Bound function variable.
$\vdash$	Assertion sign.
$\text{----}\xi$	Horizontal.
$\text{-}\neg\text{-}\xi$	Negation sign.
$\backslash\xi$	Substitute for the definite article.

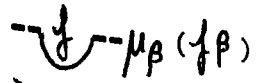
Material implication sign.

Sign for equality.

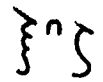


Second-level universal quantifier.

Second-level abstraction operator.



Third-level universal quantifier.



Sign for "set membership" or application.

Other Logical and Mathematical Symbols and Abbreviations

-	Negation sign.
v	Sign for disjunction.
&	Sign for conjunction.
=>	Material implication sign.
x, y, z, ...	Bound individual variables.
f, g, ...	Bound "predicate" variables.
F, G, ...	Arbitrary predicates.
(Ax) φ(x)	Second-level universal quantifier.
(Ex) φ(x)	Second-level existential quantifier.
(E! | F) φ(x)	Restricted unique second-level existential quantifier.
(Af) μ_β (fβ)	Third-level universal quantifier.
(Ef) μ_β (fβ)	Third-level existential quantifier.
$\lambda\varepsilon.\varphi(\varepsilon)$	Symbol for lambda or functional abstraction.
$\{\varepsilon \mid \varphi(\varepsilon)\}$	Symbol for set abstraction.
$\in$	Symbol for set membership.
f: D --> R	f is a function with domain D and range R.

f: d |--> r	The value of the function f for the particular argument d is r.
*	Cartesian product.
(d, r)	Ordered pair of d and r.
iff	Abbreviation for 'if and only if'.
|-	Abbreviation for 'therefore'.
mp	Abbreviation for 'modus ponens'.

Categorial Grammar Notations

N	The category of singular terms.
P	The category of propositions.
E	The category of all complete expressions (as in Frege's Begriffsschrift where singular terms and propositions are not distinguished).
J	The type of objects.
H	The type of truth-values.
x, y, z, ...	Arbitrary categories or types.
()	Empty list.
(x, y, z, ...)	The list consisting of x, y, z, ...
-->	Contraction.
S, T, U, ...	Arbitrary structures.
1, 2, 3, ...	Category and type name constructors.
BAS	The set of basic category names.
MAT	The set of all category names.
CAT	The set of all category names (using the constructors '1', '2', etc., rather that '-->' and '*').
FUN	The set of fundamental types names.
ONT	The set of all type names.

Combinatory Logic Notations

$\underset{\sim}{S}$, $\underset{\sim}{K}$, $\underset{\sim}{I}$, $\underset{\sim}{B}$, $\underset{\sim}{C}$, $\underset{\sim}{W}$, $\underset{\sim}{\Phi}$, $\underset{\sim}{\Psi}$, $\underset{\sim}{Y}$, up.	Combinators.
a, b, c, ...	Arbitrary obs.
X, Y, Z, ...	
$(\underset{\sim}{\xi}\,\underset{\sim}{\zeta})$	Functional application.
Q, R, ...	Arbitrary relations between obs.
F	The functionality primitive.
1	Constructor for names of functional characters and valency names.
x, y, z, ...	Arbitrary functional characters or valencies.
==>	(a) One-step reduction. (b) Constructor for names of functional characters and valency names.
-->	Reduction.
=	Conversion.
L, M, ...	Arbitrary lexicons.
.	Composite product.
X	Combinatory powers.
ALG	The bracket abstraction algorithm.
[b]	Bracket abstraction with respect to b.
ATM	The set of atomic ob names.
CMB	The set of all ob names.

Miscellaneous Symbols

@	Concatenation with the insertion of a space.
&	Concatenation without the insertion of a space.
⌜⌝, ⟦⟧	Quasi-quotation.
ΔX	The disquotation of X.
R, X, Y, ...	Metalinguistic variables.
A, B, ...	Metalinguistic constants.
V	Value assignment.
___ = ...	___ is defined as ... .
T	The True.
F	The False.

Named Categorial Grammar Rules

(S1) S --> T, T --> U |- S --> U.

(S2) S --> T |- (S, U) --> (T, U).

(S3) S --> T |- (U, S) --> (U, T).

(R1) (1xy, y) --> x.

(R2) (w, x) --> y |- (w, 1xz) --> 1yz.

(R3) (1uv, w) --> 1xy |- (1u1vz, w) --> 1x1yz.

(D1) (1xy, 1yz) --> 1xz.

(D2) (11xz1yz, 1yz) --> 1xz.

(G1) (2xyy, y, y) --> x.

(G2) (u, v, v) --> x |- (u, 1vy, 1vy) --> 1xy.

Named Combinatory Logic Rules

(R) a R a.

(S) If a R b, then b R a.

(T) If a R b and b R c, then a R c.

(M) If a R b, then (c a) R (c b).

(N) If a R b, then (a c) R (b c).

(Fe) If X: 1xy and Y: y, then (X Y): x.

(Fi) If L, X: x ⊢ Y :y, and if X: x does not occur in L, then L ⊢ [X] Y :1yx.

Introduction

Orientation

In this thesis I look at Frege's views about the structure of formalised language and how they can be applied to natural language. Those views are presented most explicitly in Grundgesetze, where they are used to generate a second-order language, but Frege's formation rules are very different from those that a present-day logician would use. They make essential use of what he calls unsaturated expressions and many people find these mysterious. I argue that unsaturated expressions are linguistic functions, that is to say, functions whose values and arguments are either linguistic expressions or other linguistic functions. Understanding them in this way removes much of their mystery.

The main proponent of such an interpretation is Geach and yet there is a lacuna in his account. Whereas all unsaturated expressions are linguistic functions the converse is not true. I dub these awkward linguistic functions pathological and show

how a line of demarcation can be drawn between them and unsaturated expressions.

In order to further clarify the notion of unsaturatedness it is necessary to examine the language we use to talk about unsaturated expressions and entities. When Frege did this he found himself, for example, defending the view that the concept horse is not a concept but an object. His discussion is couched almost entirely in ontological terms, yet similar paradoxical sounding statements can be made about unsaturated expressions. Some people have thought that such statements are highly objectionable and they have sought to show that Frege was mistaken in holding them to be true. Dummett and Geach, for example, try to do this by arguing that expressions like 'the concept horse' are really predicative. I show, however, that their attempt to show that Frege was wrong leads to results which are far worse philosophically than his and I outline how a workable theory can be constructed in which such expressions are genuine singular terms. One of the consequences of this is to show that although - as Frege argued - every language must have some unsaturated expressions in it, there are no unsaturated expressions that every language must have.

Having prepared the ground in this way I am able to turn to my central concern, which is how Frege's insights into the workings of a formalised language can be applied to natural language. Both Dummett and Geach have tried to do this. Although their results differ greatly, they both assume that natural language contains at least the same multiplicity of unsaturated

expressions as Frege's Begriffsschrift.<1> This is the obvious way in which to apply Frege's views to natural language, but one of the conclusions of the present work is that it leads to unacceptable consequences in both of their accounts. This is because neither of them can account in a satisfactory way for combination problems. A simple example of such a problem is the use in natural language of conjunction to combine two predicates, as in the sentence 'Bucephalus is a horse and neighs'. This is problematical because conjunction is an operator which makes a sentence out of two sentences. Dummett solves this kind of problem by distinguishing between two types of non-grammatical analysis of sentences - which I call the logical and the semantic - and by including ideas typical of combinatory logic in his account of logical analysis. I show, however, that his arguments for distinguishing between these two types of analysis are faulty and so he has not succeeded in accounting for all the combination problems. The use of combinatory ideas is, however, to be applauded.

Applying Frege's insights to natural language leads Geach to develop a categorial grammar (which Potts has further elaborated). In this the various combination problems are solved by adding a number of extra grammatical rules. I show that these violate the basic assumptions of that grammar and so this attempt

---

<1> Throughout this thesis I refer to Frege's formalised language as the Begriffsschrift. There are differences between the Begriffsschrift of Begriffsschrift and that of Grundgesetze. No confusion should arise, however, because apart from talking about it in this Introduction I hardly ever mention the earlier work.

to extend Frege's ideas also fails.

This thesis ends with my own account of how Frege's ideas should be applied to natural language. This involves rejecting the assumption that natural language contains at least the same multiplicity of unsaturated expressions as Frege's Begriffsschrift. Instead, I take seriously a position which has much in common with Frege's view that expressions like 'the concept horse' are genuine singular terms. This position is formalised in combinatory logic which is expressed in a language that contains no variable-binding operators and in which there is only a single unsaturated expression. In such a logic it is possible to solve all of the combination problems in a satisfactory way. And as combination problems do not arise in languages with variable-binding operators, all of Frege's insights into such languages are retained.

Combinatory logic is not invoked as a deus ex machina just to solve the combination problems. Dummett, Geach and Potts all make use - with various degrees of awareness that they are doing so - of combinatory ideas. What I show is that it is essential to do so in order to solve these problems. Thus, the main conclusion of the present work is that if you begin with the idea that the way in which the expressions of a language combine should be understood by analogy with the way in which a function combines with its argument to yield its value, then the way in which this idea works out in the case of a formalised language with variables is different from how it works out in the case of a natural language. In the case of a formalised language - such

as Frege's Begriffsschrift - it results in a system which makes use of a multiplicity of unsaturated expressions, whereas - in the natural language case - it results in a system which uses only a single unsaturated expression.

Background

Frege was a logicist. He believed that arithmetical notions could be defined in terms of purely logical ones and that the laws of arithmetic could be derived from logical laws.<2> The main goal of his academic life was to try and carry out this logicist programme.<3>

The first step he took in this programme was 'to reduce the concept of ordering-in-a-sequence to the notion of logical ordering' (Begriffsschrift, Preface). The details of the construction he used need not concern us here. What is important for the present study, however, is the formalised language that he felt himself compelled to devise in order to carry out his derivations. By his own account he first tried to write proofs

---

<2> As Dummett points out on p.6 of The Interpretation of Frege's Philosophy Frege considered arithmetic to consist of the theories of cardinal and real numbers.

<3> I thus agree with Baker and Hacker when they write that 'Frege's avowed primary goal was to substantiate the logicist thesis that arithmetic is part of pure logic. Everything else he did was peripheral' (Frege, p.8). There are, however, a lot of valuable insights in those peripheral aspects of his work.

in natural language, in which - no doubt - he included familiar arithmetical notations (ibid.). But, he found that so formulated the expressions he needed to manipulate became so unwieldy and imprecise for his purposes that it was not at all clear whether in an attempted proof of an arithmetical proposition from logical ones some non-logical assumptions had been used.<4> Thus, his desire to carry out the logicist programme led him to the notion of a gapless proof and to a language in which such proofs could be constructed. He says of the Begriffsschrift:

> its chief purpose should be to test in the most reliable manner the validity of a chain of reasoning and expose each presupposition which tends to creep in unnoticed, so that its source can be investigated. For this reason, I have omitted the expression of everything which is without importance for the chain of inference. (Ibid.)

The formalised language that is familiar to all readers of Begriffsschrift was not Frege's first attempt at constructing a Begriffsschrift:

> In my first draft of a formula language, I was misled by the example of [ordinary] language into forming judgements by combining subject and predicate. I soon became convinced, however, that this was an obstacle to my special goal and led only to useless prolixity. (Begriffsschrift, Section 3.)

The special goal mentioned here must be that of devising a

<4> Here and throughout this thesis I use the word 'proposition' in the same way as Geach does, namely, 'for a sentence serving, as grammarians say, to express a complete thought, to say what is or is not so, rather than for the thought so expressed' ("Names and Identity", p.139).

formalised language in which the way in which propositions are represented clearly reveals their inferential properties and in which those aspects of a proposition that are irrelevant to its inferential properties are not depicted. Thinking of propositions as composed of a subject and predicate makes it difficult to distinguish between singular terms and quantifiers (as Frege himself mentions in Section 9) and it is probable that he realised the inadequacies of the first draft when he came to deal with quantifiers, and especially with propositions involving more than one sign of generality.<5>

Although he had been misled by it, at this stage of his career he was not as critical of natural language as he later became. He does not see natural language and the Begriffsschrift as being rivals. They are languages that are suited to different purposes. Natural language is a general-purpose language, whereas the Begriffsschrift reveals the inferential properties of the propositions expressed in it. Frege compares the relationship of natural language to the Begriffsschrift with that

---

<5> In the quotation, Frege actually talks of forming judgments - and not propositions - by combining subject and predicate. When he wrote Begriffsschrift he was not as careful about distinguishing between the use and the mention of a sign as he later became. So, it is not always clear whether he is using the words 'subject' and 'predicate' as though they applied to linguistic items or to certain non-linguistic entities. In Section 3, for example, he writes: 'If one says, "The subject is the concept with which the judgement is concerned.", this applies also to the object.' Here, 'subject' must be understood as dealing with something non-linguistic. My comments on Frege in the text are justified by the way in which he discusses the examples he introduces in Section 9 and by the ways in which he introduces the universal quantifier in Section 11. (I always use the terms 'subject' and 'predicate' to refer to linguistic items.)

of the eye to a microscope. (Begriffsschrift, Preface.)

It should be mentioned that Frege regarded the Begriffsschrift as a language. For him it was something to be used. He looked forward to a time when - by adding extra signs relating to different disciplines - the Begriffsschrift would be a unifying framework for existing formalised languages (and he specifically mentions those of arithmetic, geometry and chemistry in this connection) and a springboard for the formalisation of other disciplines (such as pure kinematics, mechanics and physics, ibid.). This is an important point because later workers in the field of mathematical logic have treated formalised languages as objects of study. If they are used at all, it is by beginning-students and in order to prove formally a few results that are necessary for the informal proof of metamathematical results.

The most important fact about the Begriffsschrift for the present study, however, is that the model that Frege used for the formation of propositions in it was that of the application of a function - as understood in its mathematical sense - to its argument or arguments in order to yield a value. Closely related to this is the fact that Frege also explained the way in which the extra-linguistic correlates of the constituents of a proposition combined together by analogy with the way in which a function combined with its argument or arguments to yield its value.<6> Frege was very aware of the innovative nature of this.

<6> I do not think that it is possible to say which - if either - of these had the priority in Frege's thought at this time, or even if they were clearly separated at all times in his thinking, because he tended to mix up use and mention in the early part of his career.

In the Preface to Begriffsschrift he justifies the novel features of the Begriffsschrift by saying that he was 'driven by a necessity inherent in the subject matter itself' and he adds:

> These deviations from the traditional find their justification in the fact that logic up to now has always confined itself too closely to language and grammar. In particular, I believe that the replacement of the concepts of subject and predicate by argument and function will prove itself in the long run.<7>

This innovative use of function theory profoundly influenced all of Frege's philosophy. Dummett is correct in saying that Frege cannot be properly understood unless you place him in his epoch and one of the things that Dummett means by this is that one needs to be aware of what was known and what was not known when Frege was writing (The Interpretation of Frege's Philosophy, p.xvi). Although I agree with Dummett on this point, I think he is wrong to single out Frege's discovery of the notation of quantifiers and variables as being the discovery which dominated all his subsequent thought (Frege, p.8). Frege's discovery of this notation was just one instance of his more general extension of the notion of a mathematical function. We must not read the usual modern understanding of quantifiers into Frege's writings.

<7> The mention of 'language' and 'grammar' here strongly suggests that 'subject' and 'predicate' are to be understood as referring to linguistic items and, therefore, that 'function' and 'argument' should also be so understood. The next sentence, however, is: 'It is easy to see how regarding a content as a function of an argument leads to the formation of concepts.' Here 'function' must refer to something non-linguistic. This shows that linguistic and extra-linguistic concerns were not always clearly separated in Frege's thought at this time.

For Frege a quantifier was a specific example of a functional sign and what it referred to was a particular type of function. Thus, in the case of quantification over objects, the quantifier involved refers to a function which maps a concept to a truth-value. Quantification is not usually understood like this in modern treatments of first-order logic (but it is understood in a very similar way in illative combinatory logic).<8>

There is quite a lot of textual evidence in Frege's writings (from various stages in his career) to justify the claim that what really dominated his thinking was the way in which he extended the meaning of the word 'function'. In addition to the remarks from the Preface to Begriffsschrift that I have just quoted I would just like to mention two other passages in his work.

In his paper "Function and Concept" Frege discusses how the meaning of the word 'function' has been stretched as mathematics has progressed and he identifies two directions in which this has happened: (a) Mathematicians have added such operations as taking the limit of a series and functions that can only be expressed in ordinary language to the usual mathematical functions of, for example, addition and exponentiation. (b) Complex numbers have been added to the class of allowable arguments and values of functions. He then says: 'In both directions I go still further' (p.12) and he spells this out by mentioning how he has extended both the class of signs that can

---

<8> I thus agree with Baker and Hacker when they write: 'There is no doubt that the key to Frege's achievement lay in his extension of function theory to logic' (Frege, p.15).

be used to make up functional signs and the class of entities that can be both the values and arguments of functions. For example, to the first class he added the signs '>', '<' and '=' (amongst others) and to the second class he added the truth-values and human beings (as well as other entities). It is important to note that although Frege does extend the notion of a function in these ways, he has a distinct reluctance to make use of functions that can only be expressed in ordinary language.

The final passage that I want to mention in order to establish the importance of function theory on Frege's thought occurs in the posthumously published fragment "What may I regard as the Result of my work?" In that fragment, he answers the question which is its title as follows:

> It is almost all tied up with the concept-script. a concept construed as a function. a relation as a function of two arguments. the extension of a concept or class is not the primary thing for me. unsaturatedness both in the case of concepts and functions. the true nature of concept and function recognized. (P.200. The original punctuation has been retained.)

(Although he goes on to say that he probably should have first mentioned the judgment-stroke and the dissociation of assertoric force from the predicate, this quotation shows that his first thoughts in response to the question concern functions.) These passages also show that Frege applied function theory both to language and to the extra-linguistic correlates of expressions. His understanding of the latter changed over the years, because he did not always distinguish between sense and reference.

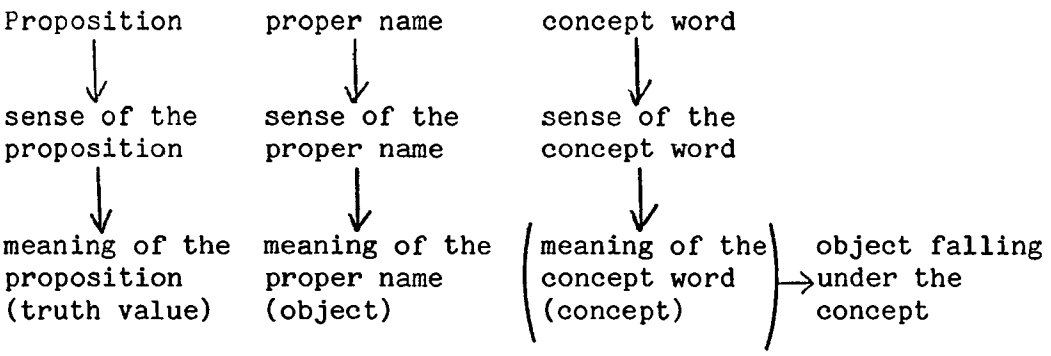
We should be careful to distinguish our present-day understanding of functions from that which was current when Frege was writing. The extension of the meaning of the word 'function' that Frege reports in "Function and Concept" is - to us nowadays - uncontroversial and even commonplace, but at that time it was a very considerable advance. As I am not a historian of mathematics, I am dependent for my information in this area on secondary sources.<9> It appears that it was only towards the latter part of the nineteenth century that our modern notion of a function was being forged. And here I am not thinking about the debate whether functions are primarily rules or a particular sort of set of ordered pairs. What I am thinking of is the recognition of higher-order functions and functions not expressable as analytical formulas as first-class citizens in the realm of functions.<10> We take it for granted that differentiation, for example, makes a real-valued function which has real numbers as arguments out of the same type of function, but such an understanding was a long time coming. And Frege's work helped to create our modern notion of a function.

<9> Such as Kline's Mathematical Thought from Ancient to Modern Times and Monna's "The Concept of a Function in the 19th and 20th Centuries". Baker and Hacker survey some of this mathematical background in the Section entitled "The Race for the Mathematicization of Logic" of their book Frege (pp.11-16) and so does Church in An Introduction to Mathematical Logic, pp.22-23.

<10> Treating a function as a first-class citizen involves such things as letting it be the argument or value of some other function and, possibly, allowing quantification over that type of function. There is still a reluctance amongst some mathematicians to admit higher-order functions to the status of first-class citizens.

A Hermeneutical Principle

In Frege's mature philosophy there is a parallelism between his ontology and his account of the logical structure of language; it also holds between these two and his views about the senses that various linguistic expressions possess. The parallelism that exists between these three realms - the linguistic, that of sense and that of reference - comes out very clearly in a letter that Frege wrote to Husserl (24 May 1891) in which he includes the following schema:



(The purpose of this schema is to show Husserl that he leaves out the notion of a concept - construed as the referent of a predicate or concept-word - from his account of the functioning of language. It should be noted that 'meaning' here is the translation of 'Bedeutung'.)

What I want to convey by using the word 'parallelism' in this context is that Frege held that for each category of expressions that he dealt with there were properties that were true of those expressions, true of their senses and true also of their

referents. For example, in the case of proper names one such property is that of completeness. For him, proper names were complete expressions and objects - the referents of proper names - were complete entities. Furthermore, the senses of proper names were also complete.

To establish that there was such a parallelism in Frege's mature thought would require a thorough investigation of his writings (just quoting the schema from his letter to Husserl is not sufficient proof!), but all I need for the purposes of this thesis is to show that Frege viewed functional signs, their senses and their referents in remarkably similar ways in respect of their essential characters. From this we can derive an important hermeneutical principle to apply to Frege's writings and a useful argumentative weapon to use against certain expositions of his philosophy. The principle in question is that it would be wrong to give an exegesis of Frege's account of either functional signs or their senses or their referents that could not - with suitable changes - be turned into an interpretation of his views on the other two parallel notions. If someone does offer an interpretation of one of these parallel notions that cannot - with suitable changes - be turned into an exegesis of the other two, then such an account must be mistaken.

One way in which I use this principle is as follows. I say very little in this thesis about Frege's views about sense. I am here concerned with functions and especially with the ways in which they can be used to give an account of linguistic structures. Sometimes, however, Frege says something about the

essence of functions as these occur in the realm of sense and does not explicitly state the same things about functions on the linguistic and ontological levels. For example, much of what he says about negation in "Negation" applies explicitly only to the function in the realm of sense which makes thoughts out of thoughts. In particular, the remark that double negation can be construed as the amalgamation of two such functions, made on p.156, is made in regard of what happens in the realm of sense. I have felt justified to assume that he also thought that the linguistic negation operator could combine with itself to form a function that turned a proposition into its double negative.

Having stated the principle of parallel interpretation and having given an example of its use it only remains for me to establish its correctness. Arguing against the view of Marshall and Grossmann that Frege did not apply the word 'Sinn' to incomplete expressions and that when he applied the word 'Bedeutung' to them this has to be understood in the ordinary English sense of 'meaning', Dummett conclusively shows that Frege did distinguish between sense and reference in the case of incomplete expressions.<11> Not only did he do this, but he also thought that the senses and referents of incomplete expressions were themselves in need of supplementation. In the case of the referents of unsaturated expressions he explicitly writes:

<11> Frege, pp.204ff. See also the following articles: Marshall "Frege's Theory of Functions and Objects", Grossmann "Frege's Ontology" and Dummett "Frege on Functions: A Reply" and "Note: Frege on Functions".

> The peculiarity of functional signs, which we here called 'unsaturatedness', naturally has something answering to it in the functions themselves. They too may be called 'unsaturated' and in this way we mark them out as fundamentally different from numbers ("What is a Function?", p.665).<12>

Similarly, in the case of the senses of functional signs Frege writes:

> If we call the parts of the sentence that show gaps unsaturated and the other parts complete, then we can think of a sentence as arising from saturating an unsaturated part with a complete part... To the unsaturated part of the sentence there corresponds an unsaturated part of the thought and to the complete part of the sentence a complete part of the thought, and we can also speak here of saturating the unsaturated part of the thought with a complete part... Each of the sentence-parts
>
> "1 is greater than 2 " and "1^2 is greater than 2"
>
> can also be seen as put together out of the proper name "1" and an unsaturated part. The corresponding holds for the related thoughts.<13>

And again, in "On Concept and Object" he writes that:

> the sense of the phrase 'the number 2' does not hold together with that of the expression 'the concept <u>prime number</u>' without a link. We apply such a link in the sentence 'the number 2 falls under the concept <u>prime number</u>'; it is contained in the words 'falls under' which need to be completed in two ways -

---

<12> See also "On Schoenflies: <u>Die logischen Paradoxien der Mengenlehre</u>", pp.191-192.

<13> "A Brief Survey of My Logical Doctrines", pp.217-218. In the first sentence of this quotation the translation actually has '... saturating a saturated part with a complete part ...' This is an incorrect rendering of the German and I have altered it in the text.

> by a subject and an accusative; and only because their sense is thus 'unsaturated' are they capable of serving as a link. (P.205.)

These passage clearly show that Frege used the same terminology of functional signs, their senses and their referents. He applies, for example, the expressions 'unsaturated' and 'incomplete' to the functional entities on all three levels. Moreover, they show not only that, but also that Frege conceived of the combination of an unsaturated item with a saturated one on each level in the same terms. He would not have done this if he had radically different ideas about the modes of combination involved on each level.

The principle of parallel interpretation only applies to what is true of a functional sign (or its sense or its referent) in virtue of its being unsaturated or incomplete. There are many things that are true of a functional sign that are not true of its referent, say. For example, linguistic negation is an operator which makes a proposition out of a proposition. Clearly, the referent of linguistic negation does not do this as well. For Frege it maps truth-values into truth-values. But if we can establish that when predicated of ontological negation the word 'unsaturated' is used to indicate that it is a <u>function</u>, then we can conclude by means of the principle of parallel interpretation that linguistic negation is also a function. This principle is quite useful and I need it at several places in what follows.

Chapter 1: Unsaturated Expressions

Introducing Functions

In this Chapter I examine Frege's characteristic method of introducing functions and functional signs. I begin by looking in detail at the first few pages of his paper "Function and Concept". The progression of ideas is similar in "What is a Function?" and in Section 1 of Grundgesetze, entitled "The Function is Unsaturated", but the account given in "Function and Concept" is fuller.

Frege takes as his starting point the definition of a function that some mathematicians of his day would have given. They defined a function of x to be a mathematical expression containing the letter 'x'.<1> Thus, such a mathematician would have held that the mathematical expression '$2.x^3 + x$', say, was a numerical function of x and, similarly, that the expression '$2.2^3 + 2$' was a function of 2. ("Function and Concept", p.2.)

<1> To begin with - following Frege - I will restrict my discussion to functions of one argument.

Frege easily shows why such an account is radically mistaken. It completely fails to distinguish between a linguistic expression and that for which such an expression stands. This difference is obvious if we consider the fact that there are properties which can truly be ascribed to numbers but which cannot truly - or even meaningfully - be ascribed to numerical expressions. For example, every number has the property that it is equal to the product of itself and the number one, but this is not true of any of the numerals. Nowadays, this distinction is known as that between _using_ an expression and _mentioning_ it. For example, in the sentence '2 is a prime number' the numeral '2' is used, but in '"2" is a numeral' it is mentioned. Frege, however, knows the distinction as that between the _form_ and the _content_ of an expression ("Function and Concept", pp.2-3).

Accepting this criticism someone might modify the original account of what a function is by saying that it is not the mathematical expression that is a numerical function, but rather that it is what the mathematical expression '$2.x^3 + x$' stands for that is a function. This is unsatisfactory because it embodies a mistaken understanding of the role of such letters as 'x' in mathematical discourse.<2> Frege points out that an expression like '$2.x^3 + x$' is just used to indefinitely indicate or indefinitely signify a number and in this respect the use of the

---

<2> Frege objected to calling letters like 'x' variables in order to avoid the possibility of thinking that there were such things as variable numbers. Quite clearly there are no such things as variable numbers, but having made this point I will use the word 'variable' to talk about such linguistic expressions as the letter 'x'.

expression '$2.x^3 + x$' is essentially the same as that of the simple variable 'x'. It is inappropriate to ask what a variable or an expression containing a variable stands for, since such expressions have a different semantic role from that of expressions which do refer to some entity. Geach points out that Frege distinguished between at least three different semantic roles that an expression could have:

> a formula beginning with the assertion sign behauptet (asserts), a proper name bedeutet, a variable deutet etwas unbestimmt an - indefinitely indicates something. ("Saying and Showing in Frege and Wittgenstein", p.59.)

It does not make sense to ask of a variable what it stands for. It is imprecise to say that variables do not stand for anything, because variables are fundamentally different from, say, vacuous definite descriptions, such as 'the first American Pope', which do not stand for anything at present, but might come to do so. A variable is not the sort of linguistic item that could have a reference.

As well as saying that $2.x^3 + x$ is a function of x mathematicians also say - as they did in Frege's day - that $2.4^3 + 4$, for example, is the value of the function $2.x^3 + 2$ for the argument 4 or simply that $2.4^3 + 4$ is a function of 4. So, we might attempt to characterise this function by saying that $2.4^3 + 4$ is its value for the argument 4, $2.1^3 + 1$ for the argument 1 and $2.2^3 + 2$ for the argument 2. But this way of speaking does not characterise a unique function. $2.4^3 + 4$ is just 132, $2.1^3 + 1$ is just 3 and $2.2^3 + 2$ is just 18; and there

are lots of different functions which for the arguments 4, 1 and 2, respectively, give the values 132, 3 and 18, respectively. One example of such a function is:

$$(1)\ x^4 - 5.x^3 + 14.x^2 - 7.x.$$

This gives the desired values for the arguments 4, 1 and 2, but for the argument 3, say, it does not give 57, that is to say, $2.3^3 + 3$, instead it returns 51. So, we cannot define a numerical function by giving a table or list of a finite number of argument-value pairs; and, quite clearly, we cannot display an infinite table or list.<3>

Although the mathematical expression '$2.x^3 + x$' just indicates a number indefinitely - just as the simple variable 'x' does - Frege says that it leads us to the correct conception of what functions are ("Function and Concept", p.6). He says that people

---

<3> There is nothing difficult about constructing functions such as that expressed in (1). Let f be an arbitrary function, then the function:

$$2.x^3 + x + (x - 4)(x - 1)(x - 2)f(x)$$

will agree with $2.x^3 + x$ for the arguments 4, 1 and 2, respectively, but - in general - will disagree for other arguments. The astute reader will no doubt have noticed that I have been forced to use imprecise language in making this point because I have still to present Frege's account of what functions are.

recognize the same function again in

$'2.1^3 + 1,'$

$'2.4^3 + 4,'$

$'2.5^3 + 5,'$

only with different arguments, viz. 1, 4, and 5. From this we may discern that it is the common element of these expressions that contains the essential peculiarity of a function; i.e. what is present in

$'2.x^3 + x'$

over and above the letter 'x'. We could write this somewhat as follows:

$'2.(\ )^3 + (\ ).'$

I am concerned to show that the argument does not belong with a function, but goes together with a function to make up a complete whole; for a function by itself must be called incomplete, in need of supplementation, or 'unsaturated.' (Ibid.)

On a superficial reading of this passage it looks as if Frege is confused between the use of the first three expressions that he displays and their mention, because he talks of people recognising the same (numerical) function in those three expressions only with different (numerical) arguments.<4> It is highly unlikely that Frege is confusing use and mention, because

---

<4> In the Begriffsschrift Frege does refer to certain expressions as being functions (for example, in Section 11), but in his later writings he reserves the word 'function' for non-linguistic items. So, the first occurrence of the word 'function' in this passage has to be construed as meaning a numerical function.

earlier in this paper he has criticised others for being so confused (pp.2-4) and because he goes on to say:

> Moreover, we now see how people are easily led to regard the form of the expression as what is essential to the function. We recognize the function in the expression by imagining the latter as split up, and the possibility of thus splitting it up is suggested by its structure.<5>

Frege is not confusing use and mention here. What he is doing is introducing a particular numerical function and one of the functional signs that refers to it together. One of the advantages of talking about mathematical expressions in this context, rather than the numbers they stand for, is that the expression '$2.1^3 + 1$' is different from the numeral '3', although both of them refer to the same number. Furthermore, the structure of the complex expression '$2.1^3 + 1$', in the context of similar complex expressions, suggests how it is to be split up.

We can bring out this idea more clearly by observing that the first three displayed expressions in the quotation on p.22 above are derived in a uniform way from the numerals '1', '4' and '5', respectively. Quite clearly there are a good many more similar complex expressions and it would be impossible to list them all. However, what it is possible to do is to give a recipe for constructing all such similar numerical designations. The recipe goes as follows: To construct a complex designation of a number,

---

<5> "Function and Concept", p.7. I quote these passages at length, because they well illustrate Frege's characteristic method of introducing functional signs and the functions they refer to. He uses this method a lot in Grundgesetze; see especially pp.6, 23, 36 and 37.

that belongs with those expressions, from a given numeral or complex numerical designation, first append the full stop to the numeral '2', then append the given expression to this combination of signs, then append the numeral '3' as a superscript, then the plus sign, and finally append the given numeral to the combination of signs just formed.

Following Geach I call such a recipe or rule a linguistic function.<6> The expressions which are the ingredients of the recipe are called the arguments of the linguistic function and the results of applying the recipe to those ingredients are the values of the linguistic function. In the above example the arguments are numerals or complex numerical designations and the values are complex designations for numbers. The rule defining the function states that for any numerical designation X taken as argument, the value of this function is the linguistic expression $\ulcorner 2.X^{3} + X \urcorner$.<7>

Although in his later writings Frege does not refer to any linguistic entities as functions, I think that there can be no doubt that for him functional signs were indeed functions.

<6> Geach discusses linguistic functions in his papers "Frege" (pp.143-144), "Saying and Showing in Frege and Wittgenstein" (p.61) and "Names and Identity" (pp.139ff.). In the last mentioned paper he also says that a linguistic function is a 'rule of formation' (p.142).

I will not always qualify the noun 'function' with the adjective 'linguistic' when I am referring to a linguistic function and - later on - I will also call sense-functions simply 'functions' on occasion. The context will make clear what sort of function is being discussed.

<7> Here and throughout this thesis I use "corner" quotation marks to signify quasi-quotation. See Appendix 1 for an explanation.

This is because - as shown in the Section "A Hermeneutical Principle" in the Introduction - he uses exactly the same figurative and metaphorical language to characterise functions and functional signs. It would be very strange for Frege to do this if he did not think of those signs themselves as being functions. If someone interprets Frege's functional signs in some other way, then he is under an obligation to explain why Frege used the same terminology of functions and functional signs and yet with very different meanings in the two cases.

In the quotation from "Function and Concept" given on p.22 above, Frege introduces a particular numerical function using the functional sign '2.( )^3 + ( )'. Elsewhere he says that this 'is perhaps the most appropriate notation, and the one best calculated to avoid the confusion that arises from regarding the argument-sign as part of the functional sign' ("What is a Function?", p.664). However, he does sometimes make use of an alternative notation (though not in "Function and Concept") which employs the Greek letter xi. Using this notation the functional sign I have been using as an example can be written as '2.ξ^3 + ξ'. Under the usual convention of using quotation marks this should be understood as being a linguistic expression made up of an initial numeral '2', followed by a full stop, followed by the Greek letter xi to which is appended as a superscript the numeral '3', followed by the plus-sign and terminated by another occurrence of the Greek letter xi. But to so understand the expression '2.ξ^3 + ξ' is to treat it as a complete expression and

Frege is at pains to show that functional signs are incomplete and unsaturated. So, in this case we cannot understand the quotation marks in the usual way. I have shown that we have to understand the expression '$2.\xi^3 + \xi$' as meaning that recipe or rule of formation or linguistic function which, out of an arbitrary numerical designation X, makes the mathematical expression $\ulcorner 2.X^3 + X \urcorner$. From now on I shall always understand single quotation marks around expressions containing occurrences of the Greek letter xi in this way. If ever I want to refer to the complete linguistic expression rather than the rule I shall use double quotation marks, thus "$2.\xi^3 + \xi$".<8>

The functional sign '$2.\xi^3 + \xi$' refers to or stands for the numerical function $2.\xi^3 + \xi$. The functional sign or incomplete expression is a linguistic function which yields numerical designations when applied to numerical designations and the numerical function returns numbers when applied to numbers. It will be useful in what follows to have a succinct notation to express these facts. Mathematicians use the notation 'f: x --> y' to express the fact that the function f takes arguments of type x and returns values of type y.<9> Using 'J' to

---

<8> The notation that Frege uses for unsaturated expressions is related in an obvious way to arithmetical formulas. "$2.\xi^3 + \xi$" is obtained from '$2.x^3 + x$' simply by replacing the letter ex by the letter xi, but the way in which he uses it is very different from the uses to which the arithmetical notation is put. Frege employs it to express a rule.

<9> By using this notation I do not wish to suggest that the function f should be understood as being a particular subset of the Cartesian product of x and y such that if (d, r) and (d, s) are both elements of this subset, then r = s. Such a subset is a Fregean object and, thus, cannot be a Fregean function.

refer to the type which includes all the integers, and using the more accurate Fregean notation for functions, the fact that $2.\xi^3 + \xi$ is a function from numbers to numbers can more concisely be represented as:

(2) $2.\xi^3 + \xi$: J --> J.

Similarly, the fact that '$2.\xi^3 + \xi$' is a linguistic function from numerical designations to numerical designations is represented as:

(3) '$2.\xi^3 + \xi$': N --> N,

where 'N' refers to the category of singular terms, which includes numerals and complex numerical designations.

There are also linguistic functions whose values are not singular numerical terms, but are propositions. Just as in the case of the complex numerical designations displayed by Frege in the quotation on p.22 above, we can see that the propositions:

'7 = 2 + 5',
'9 = 2 + 5' and
'3 = 2 + 5',

are constructed in a uniform way from the numerals '7', '9' and '3', respectively. The rule of their formation goes as follows: Given any numerical singular term X, to obtain a proposition that

belongs with these three displayed propositions, append the expression '= 2 + 5' to X. Frege would write such an incomplete expression as 'ξ = 2 + 5'. As in the case of functional signs the quotation marks here must be understood in a special way. 'ξ = 2 + 5' does not refer to an expression whose first symbol is the Greek letter xi, followed by an equals-sign, which is followed by the numeral '2', to which is appended the plus-sign, followed by the numeral '5'. It is rather to be understood as that linguistic function which for any numerical singular term X taken as argument returns as value the expression X @ '= 2 + 5'.<10> If we let 'P' denote the category of propositions, then the category of this unsaturated expression is N --> P.

As is well known, Frege - in his later writings - assimilated propositions to singular terms and he called all such complete expressions proper names. This was a mistake - as Geach has conclusively shown in his paper "Czemu Zdanie nie jest Nazwą"<11> - but although mistaken there is a powerful consideration which makes the assimilation plausible. For Frege, numerals stand for numbers and functional signs stand for numerical functions. The existence of both functional signs and numerical functions is here quite unproblematical. As I said in the Introduction, Frege extended the notion of a functional sign by allowing functional signs to be constructed out of relational operators such as the

---

<10> The sign '@' is used to represent concatenation.

<11> I am grateful to Professor Geach for supplying me with an English translation of this paper.

equals-sign and the less-than-sign ("Function and Concept", pp.12ff.). Having done this it is natural to ask what a functional sign like 'ξ = 2 + 5' stands for. It seems reasonable to answer that it refers to some type of function and then the question concerning the type of this function arises. Because the model and prototype for all functional signs for Frege is the notion of a functional sign in arithmetic, it would be natural for him to assimilate the type of the referents of all functional signs to that of the referents of functional signs in arithmetic; that is to say, they are functions which make objects out of objects. (Because of his objections to piecemeal definitions, he would not think the type of the referent of a functional sign in arithmetic was a function from numbers to numbers. It had to be more general than that.) Thus, completing a functional sign like 'ξ = 2 + 5' results in an expression which refers to an object and then it was only a small step for Frege to take to identify the referent of a true proposition with the object which is the truth-value the True and also to identify the referent of a false proposition with the False. This makes <u>proposition</u> one species of the genus <u>singular term</u>.

The plausibility of regarding the truth-values as objects and propositions as singular terms referring to them comes from assimilating predicates to functional signs. I think Frege is wrong to assimilate propositions to singular terms and truth-values to objects, but he is right in thinking of predicates as

being functional signs. We should rather see the truth-values as forming a distinct type of complete entities, different from that of objects, and propositions should be conceived of as referring to these truth-values. The referents of predicates, such as 'ξ = 2 + 5', are then functions from numbers to truth values and - following Frege - we can call such functions concepts.

Functions of Two Arguments

So far I have only considered functions of a single argument, but functions of two arguments are common in mathematics. Such functions are 'doubly in need of completion' as Frege says (Grundgesetze, p.8; where this phrase occurs in italics). He uses the two Greek letters xi and zeta to mark these argument-places, as in 'ξ + ζ'. There are a number of ways of construing this. We can think of it as that linguistic function which for any two numerical singular terms X and Y taken as arguments has the value $\ulcorner$X + Y$\urcorner$. Thinking of it in this way we have a linguistic function which yields a complex numerical designation when simultaneously applied to two numerical designations in a particular order. This is represented in the following way:

(4) 'ξ + ζ': (N * N) --> N.

The use of the word 'simultaneously' here is not meant to suggest

that we are dealing with any sort of temporal phenomenon. The word is used metaphorically and the point of so doing is to deny that either of the arguments of this function is prior to the other and this is how the type-notation '(x * y) --> z' is to be understood. Neither the argument of type x nor that of type y can be present without the other. Any attempt to leave one of the argument-places of such a linguistic function unfilled results in a meaningless expression. From a certain perspective it looks as if we are dealing here with a function of <u>one</u> argument. The one argument, however, is a complex object, namely, an ordered pair of singular terms.

There is another way in which to think of two-place linguistic functions. Consider the incomplete expressions 'ζ + 2', 'ζ + 3' and 'ζ + 7'. These are built up in a uniform way. In order to see what this amounts to, let us spell out just what these three incomplete expressions are. They are, respectively:

(5) That linguistic function which for any numerical singular term X taken as argument has the value ⌜X + 2⌝.

(6) That linguistic function which for any numerical singular term X taken as argument has the value ⌜X + 3⌝.

(7) That linguistic function which for any numerical singular term X taken as argument has the value ⌜X + 7⌝.

The two-place function in question is such that when it is applied to the numerals '2', '3' and '7', respectively, it yields the above linguistic functions as values. It can be spelled out in this way:

(8) That function which for any numerical singular term Y taken as argument has for its value that linguistic function which for any numerical singular term X taken as argument has the value ⌜X + Y⌝.

This unsaturated expression can also be written as '$\xi + \zeta$' and we can represent its category as:

(9) '$\xi + \zeta$': N --> (N --> N).

It is also possible to construe addition in a third way to give another function of category N --> (N --> N). This time we take the arguments in a different order, beginning from the incomplete expressions '3 + ζ', '7 + ζ' and '9 + ζ', for example. This is also represented as '$\xi + \zeta$.

When introducing ontological functions of two arguments Frege explicitly says that they are to be construed as belonging to type J --> (J --> J). The corresponding linguistic functions therefore belong to category N --> (N --> N). He writes that functions of two arguments are doubly in need of completion

> in the sense that a function of one argument is obtained once completion by means of one argument has been effected. Only by means of yet another completion do we attain an object, and this is then called the <u>value</u> of the function for the two arguments. (<u>Grundgesetze</u>, p.8).

He goes on to consider the function of two arguments in '$(\xi + \zeta)^2 + \zeta$': 'By substituting (for example) "1" for "ζ", we saturate the function in such a way that in $(\xi + 1)^2 + 1$ we still have a function, but of one argument.' (<u>Ibid</u>., see also the

example on p.48.) And on the linguistic level - in discussing the first of the two ways in which names can be made out of names - he writes that there arises 'the name of a first-level function of one argument from a proper name and a name of a first-level function of two arguments.' (Ibid., p.47.)

Going against this textual evidence Dummett writes:

> It is important to observe that Frege does not allow for second-level functional expressions which yield, when their argument-places are filled, first-level functional expressions... Frege did not recognize the existence of functionals (second-level functions) whose values were themselves functions. (Frege, p.40.)

And in The Interpretation of Frege's Philosophy Dummett repeats this point while discussing the passage on p.47 of Grundgesetze that I quote above. He says that the 'admission of this first method of formation goes against Frege's own principles' (p.286). But is it credible that such a careful and rigorous thinker as Frege would contradict himself as Dummett suggests? I would not ascribe such an obvious contradiction to Frege. The conclusion to be drawn from the textual evidence is that Frege did not accept the principle that functions cannot be values of other functions. There is no statement of such a principle in his writings and there are several passages - like those quoted above - where he explicitly accepts as legitimate functions whose values are other functions.

The reason why Dummett says that Frege did not accept functions whose values are themselves functions is that he believes that 'functions can be understood only as the referents of functional signs' (Frege, p.248) and also that

> an incomplete expression may never be considered as derived from another incomplete expression by the removal of some constituent expression... (Frege, p.40).

Presumably, he thinks this about incomplete expressions because he regards them as features of propositions and not as their ingredients (ibid., pp.31 and 250). But as Frege clearly did accept functions of type J --> (J --> J) into his ontology, at least one of the relevant claims that Dummett makes must be false. Independently of this argument, I have already shown how an account of two-place unsaturated expressions can be given which falsifies Dummett's second claim and in the Section entitled "Functional Signs and their Referents" in Chapter 2 I will show that the first claim is false as well.

There are several terminological problems to sort out. (a) Frege refers to functions of type J --> (J --> J) as being of first-level, whereas Dummett calls these second-level functions. I do not think that Frege thought out very carefully the notion of level as applied to functions whose values are functions because his account of levels is entirely in terms of the type of argument that a function takes:

> We now call those functions whose arguments are objects _first-level functions_; on the other hand, those functions whose arguments are first-level functions may be called _second-level functions_. (_Grundgesetze_, p.37.)

Had he made more extensive use of functions whose values are functions he would, no doubt, have introduced a terminology that classified functions according to the types of both their arguments and their values.

Dummett's actual definition of level only applies to functions whose values cannot be functions (_Frege_, pp.44-45), so it is not entirely clear how it should be extended to functions whose values are themselves functions. I think that the reason why he talks about second-level functions in the passage I quote on p.33 above is that he there has operations like differentiation primarily in mind. That is to say, operations or functions both of whose arguments and values are functions which make real numbers out of real numbers. Such a function can be labelled second-level without knowing the type of its values, because its arguments are first-level.

(b) Furthermore, although it is common to say that a function of type J --> (J --> J) is a _two_-place numerical function or a function of _two_ arguments, this terminology is inaccurate. Such a function is really a _one_-place function. It has numbers as its arguments and it returns functions

from numbers to numbers as its values. I shall, however, continue to use this terminology because it is well established.

(c) A further terminological complication is caused by the fact that that branch of logic which deals with quantifiers of second-level is known as first-order logic and sometimes the quantifiers are referred to as first-order ones. Second-order logic, in this way of speaking, deals with quantifiers of third-level, which are then called second-order quantifiers.<12>

Unless I explicitly say otherwise I use the terminology of levels in Frege's way and the terminology of orders in the conventional way.

An Objection Met

Following Frege and Geach I have interpreted expressions for one- and two-place functions as particular sorts of linguistic function. But has all this trouble been worthwhile? Why interpret such functional signs as complicated rules for making expressions out of other expressions? Some people have indeed said that functional signs are (complete) expressions, such as 'sin' and 'log', which, however, belong to a different syntactic category from numerals and singular numerical terms.

<12> Church in Section 49 of his Introduction to Mathematical Logic (pp.288-294) discusses the origins of this terminology. This use of the word 'order' is different from Frege's use in Grundlagen.

They have seen the formation of complex numerical designations, like 'sin 3', as being rule-governed but they do not interpret the functional signs themselves as being rules.

This objection can seem quite forceful if we only consider a limited number of cases. For example, in the list of expressions 'sin 2', 'sin 3' and 'sin 4' there is indeed a common expression in each of them, namely 'sin'. But if we consider a wider selection of examples, we will see that this objection cannot be sustained. Let us consider the list of expressions '2 + 2', '3 + 3' and '4 + 4'. Frege would here say that the functional sign is '$\zeta + \zeta$' and - on my interpretation - that is the linguistic function which for any numerical designation X taken as argument has the value $\ulcorner X + X \urcorner$. And in the previous Section I introduced the functional sign '$\zeta + \zeta$'. But if we think of functional signs as being complete linguistic expressions, how are we to distinguish between these two different linguistic functions? In the first case what is important is not just that the plus sign makes a complex numerical designation when used as an infix operator between two numerical designations, but rather that the plus sign makes a complex numerical designation when used as an infix operator between two numerical designations which are the <u>same</u>; and <u>this</u> functional sign cannot be identified with the plus sign. If it were so identified, we would be unable to distinguish between the functional signs '$\zeta + \zeta$' and '$\zeta + \zeta$'. The same plus sign is used in formulating both of these functional signs and if we thought of the plus sign itself as the complete functional sign, we would have to identify these

distinct linguistic functions. Thus, functional signs cannot be interpreted as being complete expressions.

Another reason why this objection does not hold water is that in mathematics there are a wide variety of ways in which complex designations of numbers are constructed out of other numerical designations. For example, 'sin 3', '3!', and '5^5'. From these we can form a variety of incomplete expressions, such as, for example, 'sin $\}$', '$\}$!', '$5^{\}}$', '$\}^5$' and '$\}^{\}}$'. For each of these it is straightforward to formulate a rule which defines it as a linguistic function. Thus, corresponding to each of the five given incomplete expressions we have:

(10) That function which for any numerical designation X taken as argument has the value ⌜sin X⌝.

(11) That function which for any numerical designation X taken as argument has the value X concatenated with the exclamation sign.

(12) That function which for any numerical designation X taken as argument has the value '5' appended to which X is written as a superscript.

(13) That function which for any numerical designation X taken as argument has the value X appended to which is the numeral '5' written as a superscript.

(14) That function which for any numerical designation X taken as argument has the value X to which X is appended as a superscript.

In (10) we see the use of a prefix operator and in (11) we see a postfix operator being used, whereas in (12), (13) and (14) no operator appears at all. The functional sign in (12) can be

introduced in Frege's way by means of the list '5^{2}', '5^{5}' and '5^{7}', that in (13) by the list '2^{5}', '5^{5}' and '7^{5}', and that in (14) by the list '3^{3}', '5^{5}' and '9^{9}'.

In the functional sign 'ξ^{ξ}' there is no complete expression used which could be singled out as the (complete) functional sign; it is the rule for forming a particular spatial arrangement of numerical designations that refers to the numerical function involved.

A diehard opponent of the interpretation of functional signs as linguistic functions or rules could argue that we can introduce an operator to represent the functional signs (12), (13) and (14). In order to represent exponentiation he might suggest we write 'exp x y' instead of 'x^{y}'. Two objections can be brought against this. The first is exactly the same objection I brought to bear against thinking of the plus sign '+' as a functional sign. Even with the new exponentiation operator, how do we distinguish between 'exp $\xi\,\xi$' and 'exp $\xi\,\zeta$'? But secondly, the notation 'exp x y' is introduced as being <u>equivalent</u> to 'x^{y}'. We have to understand 'x^{y}' in the first place in order to establish the equivalence, by means of a definition, for example. And there is no candidate in the usual notational representation of exponentiation for the role of complete functional sign, because exponentiation is normally represented by a particular spatial arrangement of signs.<13>

<13> I use the forms 'exp x y', etc., here as if they came from an opponent of the position I am defending. It would be more confusing to do otherwise.

This is not an argument against the usefulness of introducing such notations as '$\exp \xi \zeta$' for 'ξ^{ζ}'. In Chapter 6 I shall show that for certain purposes the wholesale introduction of prefix operators is extremely useful.

An Example of a Second-Level Function

The usual examples given of second-level functions are differentiation and quantification, but - for variety - I shall discuss the arithmetical summation function. This is usually represented by the uncial Greek letter sigma. For example, in a mathematical text we might come across the following:

(15) $\sum_{i=1}^{5} i^2 = 55.$

This means that the sum of the squares of the first five positive numbers is 55.

Each of the following expressions are singular terms designating numbers:

'$\sum_{i=1}^{5} i^2$', '$\sum_{i=1}^{5} i + 3$' and '$\sum_{i=1}^{5} i!$'.

In fact, they designate the numbers 55, 30 and 153, respectively. These complex complete expressions are derived in a uniform way from the functional signs '$\}^2$', '$\} + 3$' and '$\}!$', respectively. Just as in the case of the first-level functional sign '$\} + 3$', it is possible to articulate a rule which expresses the way in which these designations of numbers have been formed. The rule states that in order to obtain a complex complete expression, which belongs with the above numerical designations, from a given incomplete expression or linguistic function of category N --> N decorate an uncial Greek letter sigma with the expression 'i = 1' below and the numeral '5' above, then append to this the value of the given linguistic function of category N --> N for the argument the letter 'i'.

Frege uses the Greek letter phi to mark the argument-place of such second-level functional signs. So, the linguistic function spelled out in the previous paragraph can be written as:

$$'\sum_{i=1}^{5} \varphi(i)'.$$

Just as in the case of the first-level functional sign '$\} + 3$' this is not to be understood as a linguistic expression built up out of an initial uncial Greek letter sigma decorated with the expression 'i = 1' below and the expression '5' above to which has been appended a Greek letter phi followed by the letter 'i'

enclosed in parentheses. What I have displayed does not refer to this expression; what it refers to is the function or rule I spelled out above. Thus, we can say that

$$'\sum_{i=1}^{5} i + 3' \text{ is the value of } '\sum_{i=1}^{5} \varphi(i)' \text{ for the argument } '\xi + 3'.$$

There is a slight difficulty in the above account. In spelling out the second-level linguistic function I included the phrase 'the value of the given linguistic function of category N --> N for the argument the letter "i"'; but when I was giving an account of the rules defining first-level functional signs, such as '$\xi + 3$', I did so in the following way: '$\xi + 3$' is that linguistic function which for any numeral or complex numerical designation X taken as argument has the value $\ulcorner X + 3 \urcorner$. The problem is that put in this way the rule is only defined for numerals and complex numerical designations. It is not defined for variables, such as the letter 'i', used in the second-level function above.

Frege is aware of this phenomenon but does not regard it as a problem. In discussing the existential quantifier (which in his formal system has to be built up out of the universal quantifier and negation signs) he says:

> Clearly, only names of functions of one argument - not proper names, nor names of functions of two arguments - may be substituted, for the combinations of signs being substituted must always have open argument-places to receive the letter '𝔄' ... (Grundgesetze, p.72).<14>

From this quotation we can see how to get round this difficulty. We just extend the account of first-level linguistic functions to allow variables to be their arguments. The operation of the functions in these cases should be obvious. Thus, for example, 'ξ + 3' becomes that function which for any numerical designation or numerical variable X taken as argument has the value ⌜X + 3⌝. In such cases we must be careful to think of the variable involved as a linguistic item.<15>

The linguistic function which refers to the summation function that I have just been describing yields a numerical designation when applied to a linguistic function which makes numerical singular terms out of numerical singular terms. Thus, the arguments of this linguistic function are of category N --> N. Hence, its category is (N --> N) --> N.

<14> Frege uses the Gothic letter ay as a bound individual variable.

<15> I realise that talking about "extending" the account that I gave of linguistic functions, if not interpreted charitably, opens me up to a criticism analogous to Frege's criticism of piecemeal definition. In order to simplify my discussion earlier in this Chapter I decided not to introduce linguistic functions as here given right from the outset. If I was being more rigorous, I would have done so.

This is a good place to warn the reader that the account given in this and the next Section of variable-binding operators will have to be further qualified after I introduce Frege's second way of making expressions. This will be necessary in order to avoid clashes between bound variables. See the Section "More about Variable-Binding Operators" below.

On the ontological level this version of the summation function applies the numerical function that is its argument to each of the numbers 1, 2, 3, 4 and 5; and it then adds together all five of these results. That is to say, the summation function I have described is equivalent to the function:

$$\varphi(1) + \varphi(2) + \varphi(3) + \varphi(4) + \varphi(5).$$

In mathematics it would be usual to define a three-place version of the summation function which took the lower and upper bounds of the summation as explicit parameters.

Third-Level Linguistic Functions

It is possible to introduce the second-level universal quantifier '(Ax)φ(x)' and the second-level existential quantifier '(Ex)φ(x)' in a way similar to that in which I introduced the summation function, and doing this helps us to see how third-level functions can be obtained. The two quantifiers are the following second-level linguistic functions:

(16) '(Ax)φ(x)' is that linguistic function which for any given first-level linguistic function of category N --> P taken as argument yields as value that expression which is formed by concatenating '(Ax)' to the result of applying that first-level linguistic function to the variable 'x'.

(17) '(Ex)φ(x)' is that linguistic function which for any given first-level linguistic function of category N --> P taken as argument yields as value that expression which is formed by concatenating '(Ex)' to the result of applying that first-level linguistic function to the variable 'x'.

An example of a third-level linguistic function would be one which mapped such second-level functions to propositions. Adapting Frege's notation (Grundgesetze, p.41) we can write the universal quantifier of category ((N --> P) --> P) --> P as

(18) '(Af) $(\mu_\beta)f(\beta)$'.

One's first thought on spelling out what this is runs as follows:

(19) The linguistic function (18) is to be understood as that linguistic function which for any linguistic function of category (N --> P) --> P taken as argument yields as value the expression consisting of an initial sign '(Af)' followed by the value of the second-level linguistic function for the argument 'f'.

The problem with this is right at the end, namely the phrase 'the argument "f"'. The arguments of second-level functions of category (N --> P) --> P are of category N --> P. Thus, we have to construct a "dummy" first-level linguistic function to fit the bill. A suitable one is 'fζ', namely, that linguistic function which for any singular term or appropriate variable X taken as argument yields $\ulcorner fX \urcorner$. Hence, putting all these things together, we have that '(Af) $(\mu_\beta)f(\beta)$' is:

(20) That linguistic function which for any given linguistic function of category (N --> P) --> P taken as argument yields as value the expression consisting of an initial sign '(Af)' followed by the value of the given function for the argument which is that first-level linguistic function which for any singular term or suitable variable X taken as argument yields ⌜fX⌝.

This is not very easy to take in. It may help to realise that the value of this linguistic function for the argument '(Ex)φ(x)' is '(Af) (Ex) f(x)'.

Having presented linguistic functions of first-, second- and third-level it should be clear how even higher-level functions could be introduced. As I have explained the method of construction, I will not give any more examples of its use.
I now turn to the Begriffsschrift of the Grundgesetze.
It is necessary to have some understanding of Frege's primitive signs in order to understand the formation rules that he gives for his formalised language.

The Begriffsschrift of the Grundgesetze

In Grundgesetze Frege uses the word 'name' very widely. In it names are not only numerals, singular terms and propositions, but also unsaturated expressions of various levels.<16> He also uses the phrase 'proper name' widely. It applies not only to all kinds of singular terms, but also to propositions. I will use

---

<16> This can be seen clearly on p.48 of Grundgesetze.

the letter 'E' as the name of the category of all those linguistic items that Frege calls proper names. In Grundgesetze (see Section 31) there are eight primitive or simple names (ursprünglichen oder einfachen Namen) and these are:

(a) Three names of first-level functions of one argument (that is to say, linguistic functions of category E --> E):

'$-\!\!-\!\!-\!\!-\, \xi$', '$-\!\!\top\!\!-\, \xi$' and '$\backslash \xi$'.

(b) Two names of first-level functions of two arguments (that is to say, linguistic functions of category E --> (E --> E)):

$$\text{'}\begin{array}{l} -\!\!\top\!\!-\, \zeta \\ \;\;\lfloor\!-\, \xi \end{array}\text{'}$$ and '$\xi = \zeta$'.

(c) Two names of second-level functions which take a single first-level function of one argument as their argument (that is to say, linguistic functions of category (E --> E) --> E):

'$-\!\!\overset{\mathfrak{a}}{\smile}\!\!-\, \varphi(\mathfrak{a})$' and '$\grave{\varepsilon}\varphi(\varepsilon)$'.

(d) The name of a third-level function (that is to say, a linguistic function of category ((E --> E) --> E) --> E):

'$-\!\!\overset{\mathfrak{f}}{\smile}\!\!-\, \mu_{\beta}(\mathfrak{f}(\beta))$'.

Explanation of Frege's Simple Names

I use the following notation in giving an explanation of Frege's simple names: 'T' is a name of the truth-value the True; 'F' is a name of the truth-value the False; and 'V $[\![\ldots]\!]$' is short for 'the referent (Bedeutung) of ...', where the ellipsis is to be replaced by an expression possibly containing metalinguistic variables, and the "fat" brackets behave exactly like quasi-quotation marks.

The referent of the first expression in class (a) is given by the equations (derived from Grundgesetze, pp.9-10):

$$V[\![\text{----}\Delta]\!] = T, \text{ if } V[\![\Delta]\!] = T,$$
$$= F, \text{ if } V[\![\Delta]\!] \neq T,$$

where 'Δ' is an expression of category E.

Frege calls the symbol '----' the horizontal. In Begriffsschrift he called it the content-stroke. It is sometimes said that the content-stroke played no role in that early work, but that is not true. Its effect there was to show that a proposition was not being asserted. It removed the assertoric force from any proposition that it preceded, as can be seen from the following passage:

[Let]

'⊢---- A'

mean the judgement: "Opposite magnetic poles attract each other." Then

'---- A'

will not express this judgement, but should simply evoke in the reader the idea of the reciprocal attraction of opposite magnetic poles, perhaps, say, in order to derive some conclusions from it and with these test the correctness of the thought. (Begriffsschrift, Section 2.)

The referents of the other two expressions in class (a) are given by the equations (derived from Grundgesetze, pp.10 and 19):

$$V\llbracket \text{--}\neg\text{--}\Delta \rrbracket = \text{F, if } V\llbracket \Delta \rrbracket = \text{T},$$
$$= \text{T, if } V\llbracket \Delta \rrbracket \neq \text{T},$$

$$V\llbracket \backslash \Gamma \rrbracket = \Delta \text{ , if } \Gamma = \overset{\smile}{\varepsilon}(\Delta = \varepsilon),$$
$$= \Gamma \text{ , if } \Gamma \neq \overset{\smile}{\varepsilon}(\Delta = \varepsilon).$$

The definition of the functional sign '$\backslash\}$' is given here because it is a first-level linguistic function, although it is defined in terms of higher-level signs. Frege gives the definitions in the correct order of dependence, but I give them in the order of levels. The sign '$\backslash\}$' is Frege's way of incorporating definite descriptions into the Begriffsschrift (as he says in Grundgesetze, p.19):

We have here a substitute for the definite article of ordinary language, which serves to form proper names out of concept-words. For example, we form from the words

'positive square root of 2',

which denote a concept, the proper name

'the positive square root of 2.'

The referents of the expressions in class (b) are given by the equations (derived from <u>Grundgesetze</u>, pp.20-21):

$$V[\![\;\text{---}\!\begin{array}{|l}\text{--}\Delta\\ \text{--}\Gamma\end{array}]\!] = F, \text{ if } V[\![\Gamma]\!] = T \text{ and } V[\![\Delta]\!] \neq T,$$

$$= T, \text{ otherwise.}$$

$$V[\![\Gamma = \Delta]\!] = T, \text{ if } \Gamma = \Delta,$$

$$= F, \text{ if } \Gamma \neq \Delta.$$

The referents of the expressions in class (c) are given by the equations (derived from <u>Grundgesetze</u>, pp.35 and 16-18):

$V\llbracket \text{---}\underset{\mathfrak{a}}{\smile}\text{---}\Phi(\mathfrak{a})\rrbracket$ = T, if for every fitting argument the value of the function $\Phi(\xi)$ is the True.

= F, otherwise.

$$V\llbracket \grave{\varepsilon}\Phi(\varepsilon) = \grave{\alpha}\Psi(\alpha)\rrbracket = V\llbracket \text{---}\underset{\mathfrak{a}}{\smile}\text{---}\Phi(\mathfrak{a}) = \Psi(\mathfrak{a})\rrbracket,$$
$$V\llbracket \grave{\varepsilon}(\text{----}\varepsilon)\rrbracket = T,$$
$$V\llbracket \grave{\varepsilon}(\varepsilon = (\text{--}\neg\text{-}\underset{\mathfrak{a}}{\smile}\text{--}\mathfrak{a} = \mathfrak{a}))\rrbracket = F.$$

The referent of the expression in class (d) is given by an equation analogous to that given for the universal quantifier of second-level. It is as follows (derived from Grundgesetze, p.42):

$V\llbracket \text{--}\underset{\mathfrak{f}}{\smile}\text{--}\Omega_{\beta}(\mathfrak{f}(\beta))\rrbracket$ = T, if for every fitting argument the value of the second-level function is the True,

F, otherwise.

In presenting Frege's explanation of his simple names I have deliberately employed the notation '$V\llbracket \ldots \rrbracket$ = ___' to bring out the similarity between what he is doing and the modern notion of a value-assignment.<17> Dummett makes a similar point (Frege, pp.89ff.) He, however, compares what Frege does to giving an interpretation to a language containing individual constants, unary function symbols, binary function symbols, one-place predicates, and (two-place) relational expressions. And he gives

---

<17> Defined, for example, in Hughes and Cresswell's An Introduction to Modal Logic, pp.10-11 (for the propositional calculus) and pp.135-137 (for the predicate calculus). See also Cresswell's Logics and Languages, pp.18ff.

the usual definitions of the notions of the completeness and the soundness of a logical system:

> A formalization of some part of logic is sound if, whenever a sentence A is a syntactic consequence of some set /⁻ of sentences, i.e. can be derived from them by means of a formal deduction, it is also a semantic consequence of them, i.e. is true under every interpretation of its non-logical constants under which all the sentences of /⁻ are also true. (Frege, p.82, emphasis added.)

He then goes on to say that these notions were within Frege's grasp. The main criticism I have of Dummett's interpretation of Frege in these passages is that he attributes to Frege a relative truth-theory rather than an absolute one.<18>

Dummett presents a modern account of the semantics of a logical system where truth is defined relative to an interpretation and the logical truth of a sentence is defined in terms of that sentence being true in every interpretation. Frege, however, presents an absolute truth-theory. He did not accept at least one of the central ideas in the standard modern account, namely, the idea of the multiple re-interpretability of the non-logical symbols of a formalised language. In fact, in the Begriffsschrift of the Grundgesetze there are no non-logical constants, but in the Preface to Begriffsschrift Frege does envisage the possibility - even the desirability - of adding non-logical constants; so that, for example, the subject-matter of chemistry, geometry and mechanics might be presented in his

---

<18> For further details on the distinction see Lewis's "Model Theory and Semantics", pp.278ff.

formal system. It is quite clear, however, from his disagreements with Hilbert that Frege would not have accepted that these non-logical constants are capable of being multiply re-interpreted, as is essential in a relative truth-theory.

Frege and Hilbert disagreed on a number of issues, but I just want to concentrate on the notion of multiple re-interpretability. In The Foundations of Geometry Hilbert gives an axiomatic presentation of Euclidean geometry in which words like 'between', 'point' and 'line' occur (Sections 1-8). It is most charitable to understand these words as being equivalent to the predicate-symbols in a first-order language, because Hilbert considers several different ways of interpreting them (Sections 9-12) in order to prove such things as the independence of one of the axioms from the others. He does this by constructing a model in which that axiom is false and the others true. In these various interpretations and models the three-place relation between, for example, is associated with different sets of ordered triples and the predicates point and line are associated with different sets of objects.

Hilbert did not express himself very well, especially when he wrote about definitions, and Frege criticises him for this.<19> But the most important aspect of Frege's criticism in this context is that he could not accept the way in which Hilbert re-interpreted words like 'between' and 'point' depending on his purposes. Frege kept insisting that Hilbert define these in such

---

<19> See, in particular, Frege's letter to Hilbert dated 6 January 1900 and the two parts of Frege's 1903 article "On the Foundations of Geometry".

a way that there be a determinate and unchanging answer to the question, for example, of whether Frege's pocket watch was a point.<20> Hilbert's reply, in effect, is that in some models it is and in others it is not, but Frege could not accept this. He keeps insisting that Hilbert "define" these notions properly. Hilbert's method, in this respect, is closer in spirit than Frege's to that branch of modern mathematical logic known as semantics, although that discipline only came to fruition with the work of Tarski. I therefore think that Dummett is wrong to say that the two notions of logical consequence, namely, the semantic and the syntactic 'lie ready to hand in his [Frege's] work' (Frege, p.82). The key semantic idea of the multiple re-interpretability of non-logical constants is at variance with Frege's way of thinking.<21>

<20> "On the Foundations of Geometry", pp.17-18.

<21> Resnik says much the same thing in his paper "Frege's Proof of Referentiality" (p.180): 'Frege's semantical conditions are like Tarski's in being recursive. Yet Frege's entire conception is unlike Tarski's in not relativizing interpretations to domains or to assignments to nonlogical symbols. The idea that a sentence could be true for one interpretation and false for an other, while familiar to Frege, was totally inimical to him. For him logic deals with one domain of each logical type - the universal domain of that type.'

Frege's First Way of Making Names

In Grundgesetze Frege tells us what the referents of his simple functional signs are and he gives an informal argument to show that an expression formed from referring expressions in certain ways also has a referent (Section 29).<22> He summarises the first method of formation as follows:<23>

> This formation is carried out in this way: a name fills the argument-places of another name that are fitting for it. Thus there arises
>
> [A] a proper name
>
> [1] from a proper name and a name of a first-level function of one argument,
>
> or [2] from a name of a first-level function and a name of a second-level function of one argument,
>
> or [3] from a name of a second-level function of one argument of type 2 and the name '$-\!\!\overset{\mathfrak{f}}{\smile}\!\!-\mu_{\beta}(\mathfrak{f}(\beta))$' of a third-level function;
>
> [B] the name of a first-level function of one argument
>
> [1] from a proper name and a name of a first-level function of two arguments.
>
> The names so formed may be used in the same way for the formation of further names, and all names arising in this way succeed in denoting if the primitive simple names do so.

<22> As is well known, it is the fact that Frege allows the principle of set-abstraction in its full generality that leads to the possibility of deriving Russell's paradox in the formal system of Grundgesetze. Some people have, therefore, supposed that Frege's referentiality proof must be faulty. But Resnik has established that referentiality does not imply consistency ("Frege's Proof of Referentiality", p.190).

<23> Grundgesetze, pp.46-47. The layout of this passage and the labelling of the various subdivisions are due to the translator of the opening Sections of the Grundgesetze.

(A name of a second-level function of one argument of type 2 is an expression of category (E --> E) --> E.)

Using the notation I introduced above for showing the category of an expression it is possible to present the different forms of Frege's first way of making names in the following way:

(A1') Applying an expression of category E --> E to one of category E results in an expression of category E.

(A2') Applying an expression of category (E --> E) --> E to one of category E --> E results in an expression of category E.

(A3') Applying an expression of category ((E --> E) --> E) --> E to one of category (E --> E) --> E results in an expression of category E.

(B1') Applying an expression of category E --> (E --> E) to one of category E results in an expression of category E --> E.

Dummett says of clause (B1) in Frege's account that it is 'quite redundant' (The Interpretation of Frege's Philosophy, p.286) and the reason he gives for this supposed redundancy is that

> in laying down the semantic principles governing the quantifiers, he concerns himself only with the truth-conditions of a quantified sentence, and not with the satisfaction-conditions of a predicate formed by attaching a quantifier to a relational expression. (Ibid., pp.286-287.)

Dummett is here quite wrong. The clause (B1), far from being redundant in giving an account of the formation rules of the Begriffsschrift, is essential, because without it propositions containing a functional sign for a function of two arguments

could not be formed. Assuming that the numerals '2' and '3' have been defined, then without clause (B1) it would be impossible to form, for example, the proposition '2 = 3', since none of the other clauses applies to the functional sign '$\zeta = \zeta$'.

The reason Dummett gives is irrelevant because clause (B1) applies to an expression of category E --> (E --> E) combining with one of category E to form one of category E --> E. It does not apply to a quantifier of category (E --> E) --> E combining with a two-place functional sign of category E --> (E --> E) to form an expression of category E --> E. (These forms of combination are very different.) In order to account for such a combination Frege introduces a second way of forming names, to which I now turn.

Frege's Second Way of Making Names

Frege explains the second way of forming names of first-level functions as follows:

> [We] begin by forming a name in the first way, and we then exclude from it at all or some places, a proper name that is part of it (or coincides with it entirely) - but in such a way that these places remain recognizable as argument-places of type 1. (Grundgesetze, p.47.)

(An argument-place of type 1 is one which is appropriate to admit a proper name; ibid., p.40.) Thus, this procedure makes an

expression of category E --> E from one of category E. He gives the following example, where 'Δ' is assumed to be of category E. First, from '$\zeta = \zeta$' and 'Δ', by clause (B1), we obtain '$\Delta = \zeta$'. Then, from '$\Delta = \zeta$' and 'Δ', by clause (A1), we obtain '$\Delta = \Delta$'. Then, by using the second procedure for making names of category E --> E from complete expressions, we obtain '$\zeta = \zeta$'. Finally, from the universal quantifier and '$\zeta = \zeta$', we obtain by clause (A2):

'$-\zeta - \zeta = \zeta$'.

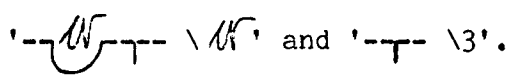
Using the notation for categories of expressions we can reformulate Frege's procedure as follows: Given an expression X of category E which contains an expression Y of category E, we obtain an expression of category E --> E by removing Y from X. It is allowable to remove several occurrences of Y from X.

Some Examples of the use of Frege's Formation Rules

In this Section I give a number of examples of the use of Frege's formation rules. In trying to transpose Frege's ideas to natural language these will be some of the difficult cases or combination problems. Each of the following four Subsections ends with an expression of category E --> E. Needless to say, in each case this could then be combined with an expression of category

(E --> E) --> E to form a complete expression.

Although I will talk extensively in Chapters 4, 5 and 6 about the problem, for example, of justifying the combination of negation with a predicate in natural language, it should be realised that what makes this problematical there is the absence of variables coupled with the fact that a negated predicate can occur both in a quantified proposition and also in a proposition that does not contain any quantifiers. In the Begriffsschrift there is no problem about constructing both of the propositions:

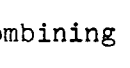


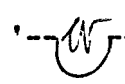
And if natural language contained variables, combination problems would not arise there either.<24>

I mention this here because I usually just talk of the problem of justifying the combination of negation with a predicate, say, without mentioning explicitly each time that it is the presence of quantifiers and the absence of variables in natural language that creates this combination problem there. This, however, should always be borne in mind.

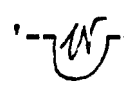
In what follows it should be assumed that the numerals have already been defined.

<24> In a modern account of a first-order language proper names and individual variables, say, must belong to the same substitution class and in Frege's scheme of things the domain of a second-level linguistic function must include singular terms and individual variables. And similarly for expressions of other categories in higher-order languages.

Quantifier and Relational Sign Having explained Frege's second way of making names I can now show how a quantifier and a relational sign combine in his Begriffsschrift. We begin by combining '  =  ' with '3', say, by clause (B1). This results in '3 =  '. This can then be combined with the second-level quantifier, by clause (A2), to yield:

'--  - 3 =  '.

From this we can form an expression of category E --> E, by removing '3' in accordance with the second way. This results in the required combination:

'--  -  = '.

Non-Propositional Negation In order to show how we can negate a functional sign of category E --> E, we have to begin by combining it with a complete expression. For expository purposes I shall use Frege's definite description operator and a numeral. This gives us '\3', by clause (A1). Combining this with negation, again in accordance with clause (A1), gives us '--¬-- \3'. From this '3' can be removed, as specified by the second way of making names, to yield '--¬-- \  '.

Note It might be useful in clarifying why combination problems do not occur in formalised languages to spell out what '--┬- \}' amounts to in terms of '-┬-}' and '\}'.

'-┬-}' is that linguistic function of category E --> E which given any expression or pseudo-expression X of category E as argument yields as value the expression '--┬--' @ X. The purpose of the clause 'or pseudo-expression' is to allow this negation operator to be the argument of a function of category (E --> E) --> E. A pseudo-expression here being something like '𝒩 + 3', that is to say, an open sentence in modern parlance.

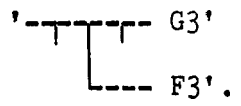
Similarly, '\}' is that linguistic function of category E --> E which given any expression or pseudo-expression X of category E as argument yields as value the expression '\' @ X.

Putting these accounts together we have that '-┬- \}' is that linguistic function which for any expression or pseudo-expression X of category E taken as argument returns as value the expression which is the result of applying the function '-┬-}' to the value of the function '\}' for the argument X. In other words, '--┬- \}' is that linguistic function of category E --> E which for any expression or pseudo-expression X taken as argument yields the value '-┬- \' @ X.

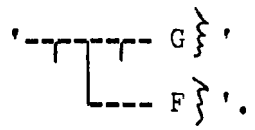
Hence, there is no problem in applying the universal quantifier '-(𝒩)-- 𝒫(𝒩)' to this, since '𝒩' is a pseudo-expression. The value of the linguistic function '-(𝒩)-- 𝒫(𝒩)' for the argument '-┬- \}' is simply '-(𝒩)-┬- \𝒩'.

Double Negation Combining negation with '3', in accordance with clause (A1), yields '--┬-- 3'. A second application of negation gives us '-┬┬-- 3'. From this '3' can be removed, in accordance with the second way of making names, to give us '-┬┬-- }'.

Internal Conjunction In order to show how two functional signs of category E --> E can be combined we begin by forming a complete expression from them. let 'F}' and 'G}' be the functional signs in question. Completing them both with '3' yields 'F3' and 'G3'. Negating the second gives us '--┬- G3'. Combining this with the conditional, by clause (B1), and then combining the result with 'F3', by clause (A1), and then negating that result gives us:



From this '3' can be removed, in accordance with the second way of making names, to yield:



More about Variable-Binding Operators

In the section "Third-Level Linguistic Functions" above I spelled out what the second-level universal quantifier '(Ax)φ(x)' is in these terms:

> It is that linguistic function which for any given first-level linguistic function of category N --> P taken as argument yields as value that expression which is formed by concatenating '(Ax)' to the result of applying that first-level linguistic function to the variable 'x'.

Having now explained Frege's second way of making expressions, it is possible to show that this account needs further qualification. In this discussion I shall not use Frege's notation, but the points I make apply to the Begriffsschrift as well.

In the account given so far the following is a legitimate series of constructions. (a) From the identity sign and the numeral '3' we form '3 = ξ'. (b) Applying '(Ax)φ(x)' to '3 = ξ'. we obtain '(Ax) 3 = x'. (c) Removing '3' yields '(Ax) ξ = x'. (d) Applying '(Ax)φ(x)' to '(Ax) ξ = x' results in '(Ax)(Ax) x = x'. The problem here is that the variable introduced by the outermost universal quantifier has become bound by the innermost quantifier. What we hoped to accomplish by this series of constructions is the proposition '(Ay)(Ax) y = x'. And reflecting on this shows us how to overcome this problem. Let us assume that our language has an infinite number of variables. The first few of which are 'x', 'y', 'z', ...

Corresponding to each of these is a universal quantifier:

'(Ax)φ(x)', '(Ay)φ(y)', '(Az)φ(z)', ...

The simplest way to resolve this difficulty is to stipulate that in constructing propositions involving universal quantifiers we use them in the order shown and, furthermore, we use each one only once. Such a qualification outlaws the construction step (d) given earlier.

I have illustrated this difficulty by talking about universal quantifiers, but similar qualifications would have to be made to all variable-binding operators of whatever level. Furthermore, if we have two or more variable-binding operators that use the same category of variables, then if we, say, have used '(Ax)φ(x)' in the construction of a proposition, then we must forbid '(Ex)φ(x)' being used subsequently, as well as outlawing the use of '(Ax)φ(x)'. It would be tedious to spell this out rigorously, but I have said enough to enable someone to do so if they wish.

The Hierarchy of Syntactic Categories

Frege uses an ad hoc terminology to distinguish the categories to which his simple functional signs belong and another related ad hoc terminology to differentiate the type to which the functions he considers belong. Thus, he talks, for example, of 'a second-level function of one argument of type 2' (Grundgesetze, p.42) or a third-level function which takes as its argument a second-level function of one argument of type 2 (ibid., p.41). He also only considers a small finite number of different types of function. I have introduced the familiar mathematical type-theoretic notation for both the categories of expressions that Frege considers and for the types of entities that he makes use of. In showing - in the Section "Third-Level Linguistic Functions" above - how to derive a second-level functional sign from considering the similarities between first-level functional signs and in showing how to derive a third-level functional sign by considering the similarities between second-level functional signs, it should have been made clear how a similar process could be used in deriving ever higher levels of functional signs. It will be useful in what follows to have a systematic and succinct notation for the categories to which such expressions can belong.

Let BAS be a set of basic category names. The set MAT of category names is defined as follows:

(i) Every member of BAS is a member of MAT.

(ii) If X and Y are members of MAT, then ⌜(X * Y)⌝ is a factor.

(iii) If X is either a member of MAT or a factor and Y is a member of MAT, then ⌜(X --> Y)⌝ is a member of MAT.

If BAS contains only the two basic category names 'P' and 'N', then the following are examples of category names: '(N --> P)', '(((N --> P) * (N --> P)) --> P)' and '(P --> ((P * P) --> P))'. In writing category names the outermost pair of parentheses will usually be left out.

Exactly mirroring this hierarchy of syntactic categories or linguistic functions there is a hierarchy of ontological entities. Indeed the two hierarchies are isomorphic to one another. Frege thought, in his later philosophy, that there was just a single type of complete entities and that that was the type of objects. He classified the two truth-values as being objects, but it is more appropriate to think of the truth-values as constituting a distinct type of complete entities.

The definition of the set ONT of all names of the different types of entity is very similar to that of MAT. Let FUN be the set of names of the basic types.

(1) Every member of FUN is a member of ONT.

(ii) If X and Y are members of ONT, then ⌜(X * Y)⌝ is a component.

(iii) If X is either a member of ONT or a component and Y is a member of MAT, then ⌜(X --> Y)⌝ is a member of ONT.

If 'P' and 'N' are the names of the basic categories, then the corresponding names of the basic types are 'H' and 'J'. 'J' is the name of the type of objects and 'H' is the name of the type of truth-values. If FUN contains only the two basic type names 'H' and 'J', then the following are examples of type names: '(J --> H)', '(((J --> H) * (J --> H)) --> H)' and '(H --> ((H * H) --> H))'.<25>

Prototypical Incomplete Expressions

The decision to introduce incomplete expressions by means of a discussion of mathematical functional signs was a deliberate one. I did this because such functional signs are for Frege the prototype of the general notion of an incomplete expression. This interpretation of Frege is at variance with Dummett's. He writes that:

> complex predicates form the prototype for Frege's general notion of an 'incomplete' expression. (Frege, p.31.)

There is much to be found in Frege's writings which contradicts this interpretation and nothing to support it. For example, when Frege gives examples of unsaturated expressions he invariably

---

<25> The names of the sets used in these definitions were chosen because they are the initial three letters of the following words: 'basic', 'mathematical', 'fundamental' and 'ontological'. Hopefully, they have mnenomic value.

starts with mathematical examples of functional signs.<26> And in "Function and Concept" (pp.12-13) he says that he has extended the notion of a function in such a way as to allow predicates and relational expressions to be used in the formation of functional signs. In an earlier part of the paper he gave an account of these as incomplete expressions. Such a procedure is incompatible with Dummett's claim.

In addition to the evidence to be found in Frege's writings contradicting Dummett, it is possible to give an ad hominem argument to show that this claim of Dummett's is at variance with other things he says. In Chapter 8 of The Interpretation of Frege's Philosophy Dummett mentions that Frege's understanding of concepts as functions embodies a genuine insight, because it provides a framework for giving different kinds of model to a first-order language. Thus, he admits that Frege's notion of a concept is modelled on his notion of a function. He admits that he seriously undervalued this idea in Frege, but adds that he does not think that anything he said there on this subject was actually wrong (ibid., p.166).

For Frege, concepts are the referents of predicates and functions are the referents of functional signs.<27> For Dummett, Frege's ontology is dependent upon his analysis of language and not the other way around. It is very strange - if not

---

<26> See, for example, the first few pages of "Function and Concept" and of Grundgesetze.

<27> For the sake of argument I am assuming that functional signs are limited to such expressions as 'sin ' and ' + 3' and that functions are limited to the referents of such expressions.

inconsistent - for Dummett to claim that for Frege predicates are prototypical incomplete expressions and yet to also say that Frege's account of concepts (which are the referents of predicates) is modelled on his account of functions (which are the referents of functional signs). In other words, on the linguistic level Dummett is assigning the priority to predicates, yet on the ontological level he is assigning the priority to functions. In the light of the close connection between Frege's analysis of language and his ontology, it cannot be correct to think that the direction of the relation of priority between predicates and functional signs should be reversed between their referents.

The reason why Dummett thinks that predicates are prototypical incomplete expressions is because of the central role that they play in his idea of the step-by-step construction of non-atomic sentences from atomic ones. I shall discuss that idea fully in Chapter 4.

Natural Language Unsaturated Expressions

Although Frege is mainly interested in mathematical and logical functional signs of various categories, he is aware that incomplete expressions occur in natural language:

> Statements in general, just like equations or inequalities or expressions in Analysis, can be imagined to be split up

> into two parts; one complete in itself, and the other in need of supplementation, or 'unsaturated.' Thus, e.g., we split up the sentence
>
> 'Caesar conquered Gaul'
>
> into 'Caesar' and 'conquered Gaul.' The second part is 'unsaturated' - it contains an empty place; only when this place is filled up with a proper name, or with an expression that replaces a proper name, does a complete sense appear. Here too I give the name 'function' to what this 'unsaturated' part stands for. In this case the argument is Caesar. ("Function and Concept", p.17.)

Just as in the mathematical examples considered above, the unsaturated expression mentioned by Frege can be represented as '$\hat{\xi}$ conquered Gaul', which is to be understood as that linguistic function which for any singular term X taken as argument yields as value the proposition X @ 'conquered Gaul'.

It is not as straightforward as this to give natural language examples of functional signs of categories higher than the first. If we extend natural language so that it contains variables (as found in a formalised first-order language), then it is possible to represent second-level expressions such as quantifiers, for example, by 'for every x,φ(x)' or 'for some x,φ(x)'. This is, in fact, what writers on Frege generally do. Similarly, if we add further suitable kinds of variables to natural language, then it becomes possible to express functional signs of other higher-level categories.

The necessity for extending natural language in these ways before it becomes possible to express higher-level functional signs should be apparent from a consideration of what

mathematical functional signs of the same category are. In spelling out what linguistic function the incomplete expression '(Ax)φ(x)' is I had to say that its value for a predicate 'F$\}$' is the sign '(Ax)' concatenated with the value of 'F$\}$' for the argument 'x'. And natural language does not contain any expression which could do the work of the variable 'x' here.<28> Similar points could be made about expressions belonging to other higher-level categories.

Logical Analysis

In the Preface to Begriffsschrift (in a passage that I have already quoted) Frege wrote:

> I believe that the replacement of the concepts of subject and predicate by argument and function will prove itself in the long run.

At first sight there does not appear to be much difference between analysing the proposition 'Jack loves Jill' into the subject 'Jack' and the predicate 'loves Jill' and analysing it into the argument 'Jack' and the linguistic function 'ξ loves Jill'. The usefulness of Frege's ideas comes from the fact that one and the same proposition can be the value of several

---

<28> I discuss the relationship between variables and pronouns in the Section "Pronouns and Variables" in Chapter 6.

different linguistic functions. The arguments may or may not be different. And the possibility of multiple analysis is very useful in logic, since the same propositions can occur in many different forms of derivation.<29> For example, we can see that the following derivations are all valid:

(21) Jack loves Jill, therefore Jack loves someone.

(22) Jack loves Jill, therefore someone loves Jill.

(23) Jack loves Jill, therefore Jack stands in some relation to Jill.

The unique grammatical analysis of 'Jack loves Jill' into the subject 'Jack' and the predicate 'loves Jill' might help us to explain the validity of (22), but it is useless in explaining the validity of either (21) or (23). By contrast, the Fregean apparatus of function and argument allows us to analyse the proposition 'Jack loves Jill' in the following ways:

(24) 'Jack loves ζ': 'Jill' |--> 'Jack loves Jill'.

(25) 'ζ loves Jill': 'Jack' |--> 'Jack loves Jill'.

(26) 'ψ(Jack, Jill)': 'ζ loves ζ' |--> 'Jack loves Jill'.

<29> I use the word 'derivation' here rather than the more natural 'argument' to avoid confusing the argument of a function with an argument understood as a group of propositions standing in an inferential relation to another proposition.

Whereas the notation

'Jack loves $\}$': N --> P

tells us that 'Jack loves $\}$' is a function from singular terms to propositions, the notation

'Jack loves $\}$': 'Jill' |--> 'Jack loves Jill'

tells us that the value of the function 'Jack loves Jill' for the particular argument 'Jill' is the specific proposition 'Jack loves Jill'. The arrow '|-->' understood like this is common in mathematics.

The different analyses of 'Jack loves Jill' shown in (24), (25) and (26) enable us to explain the validity of (21), (22) and (23), respectively. (21) is an instance of the schema

Fa, therefore (Ex) Fx,

where 'F$\}$' is interpreted as 'Jack loves $\}$' and 'a' as 'Jill'.<30> (22) is an instance of the same schema, but this time with 'F$\}$' interpreted as '$\}$ loves Jill' and 'a' as 'Jack'. (23) is an instance of

---

<30> I am assuming that the universe of discourse is restricted to people.

Rab, therefore (Ef) fab,

in which 'R$\xi\zeta$' is interpreted as 'ξ loves ζ' and 'a' and 'b' are 'Jack' and 'Jill', respectively.

Although the propositions of the Begriffsschrift can be analysed in different ways, it is possible to associate each proposition with a unique analysis in such a way that many of the other analyses can be seen as partial results in the process of constructing this unique analysis. I will refer to this as the total or canonical logical analysis. It is obtained by carrying on the process of analysis until the primitive names are reached. In doing this it may be necessary to expand definitions. Starting from an arbitrary proposition there will often be several definitions that can be expanded at any one stage, thus - typically - the partial results will form a tree. At the root of the tree the canonical analysis will be located and at each of the terminal nodes there will occur the proposition being analysed.

It is not clear whether natural language has this property. Let us assume that it has and that 'Jack' and 'Jill' are primitive names and that 'ξ loves ζ' is a primitive relational sign. Then the canonical logical analysis of 'Jack loves Jill' is that in which it is analysed into these three primitives. In this case 'Jack loves ξ' and 'ξ loves Jill' are partial analyses. Not every logical analysis of a proposition is a partial analysis. It is possible, for example, to see 'Jack loves Jill' as the value of the linguistic function 'φ(Jack)'

for the argument 'ξ loves Jill', where 'φ(Jack)' is that linguistic function of category (N --> P) --> P which for any linguistic function of category N --> P taken as argument yields as value the proposition which is the value of that first-level linguistic function for the argument 'Jack'. The analysis of 'Jack loves Jill' into 'φ(Jack)' and 'ξ loves Jill' is perfectly legitimate, but it is not a partial analysis.

Propositional Unity

Frege's account of functional signs as unsaturated expressions solves the problem of propositional unity.<31> A group of complete expressions cannot combine together to form a proposition. They will forever remain unconnected and isolated from each other. Only a group of expressions at least one of whose members is unsaturated can unite to form a proposition.

Although this is a necessary condition of propositional unity, it is not sufficient. The two unsaturated expressions 'ξ snores' and 'ξ hallucinates' cannot combine together. What is wrong here is that neither expression has an argument-place that is fitting for the other. This can be expressed by saying that their logical valencies are incompatible. An expression's valency is

---

<31> Bell discusses this and the related problem in the realm of sense extensively in Frege's Theory of Judgement. What I call propositional unity he refers to as sentential unity (for example, on p.14).

given by the syntactic category to which it belongs. So, a group of expressions can only combine if at least one of them is unsaturated and they have appropriately matching valencies. For the Begriffsschrift Frege lays down which valencies match in his discussion of the first way of making names. A complex expression which is made up out of simpler expressions with appropriately matching valencies is said to be syntactically coherent.

Discontinuous Expressions

One of the minor advantages of Frege's idea of an unsaturated expression is that it can easily account for discontinuous expressions. The functional sign 'ξ picked ζ up' of category (N * N) --> P, for example, is simply that linguistic function which, for any pair of singular terms X and Y taken as argument, yields the proposition ⌜X picked Y up⌝ as its value. Thus, for the argument ('Jack', 'Jill') (which is to be understood as an ordered pair of singular terms) it returns 'Jack picked Jill up'.

Conclusion

The purpose of this Chapter has largely been expository. Many of the ideas mentioned in it will be elaborated in future Chapters.

I began by giving an exposition of Frege's unsaturated expressions as linguistic functions. In doing this I was building on Geach's work and in Chapter 2 I will expose and rectify a lacuna in his account. And in Chapter 3 I will sort out the tangle that Geach and Dummett get themselves into when they discuss Frege's view that the concept horse is not a concept but an object.

Then I gave an account of the primitive expressions of Frege's Begriffsschrift and of his formation rules for that language. The way in which Dummett applies these ideas to natural language will be dealt with in Chapter 4 and how Geach tries to do this will be the subject of Chapter 5. In both of those Chapters I show why their respective attempts fail. My own attempt will occupy Chapter 6.

The Chapter ended with a few observations on how Frege's ideas can be transposed to deal with a natural language to which variables and variable-binding operators have been added. Many of the additions to those ideas that Dummett and Geach make concern the provision of operations which handle the phenomena that the quantifier-variable notation handles in a formalised language. My approach involves using combinatory logic to do those things.

Chapter 2: Pathological Linguistic Functions

Not All Linguistic Functions are Unsaturated Expressions

Frege extended the sense of the word 'function' in several directions. One of them was to allow certain symbols to be used in the construction of functional signs which previously no one had ever thought of allowing. For example, he let relational expressions such as the equals-sign and the less-than-sign be used in the construction of functional signs like '$\}$ = 2' and '5 < $\}$ + 2'. But there were limits beyond which Frege would not go. For example, he did not allow the assertion sign to be used in the construction of functional signs.

> The assertion sign [Urtheilsstrich] cannot be used to construct a functional sign; for it does not serve, in conjunction with other signs, to designate an object. '⊢ 2 + 3 = 5' does not designate anything; it asserts something. ("Function and Concept", p.22, fn.*.)

It is possible, however, to formulate linguistic functions which make use of the assertion sign. That linguistic function which

for any given numerical singular term X taken as argument yields as value the expression $\ulcorner \vdash\!\!\!- 2 + X = 5 \urcorner$ is a perfectly legitimate and respectable linguistic function. Although Frege would not call it a functional sign, we can express this linguistic function with the help of his xi-notation as '$\vdash\!\!\!- 2 + \xi = 5$'.

Therefore, although every functional sign is a linguistic function the converse is not true. There are perfectly legitimate linguistic functions which are not incomplete expressions. I call such linguistic functions pathological (by analogy with the use of the word 'pathological' in mathematical analysis), because their existence makes the characterisation of unsaturated expressions more difficult than it would otherwise have been.

In the light of Frege's remark it seems natural to formulate the following criterion of demarcation between unsaturated expressions and pathological linguistic functions: An unsaturated expression is a linguistic function that can be represented by means of Frege's xi-notation (or an extension of this notation in order to cope with functions of higher level or greater polyadicity) and which is capable of having as a referent an entity which is of a type that occurs somewhere in the Fregean hierarchy of types and which is not a basic type.

It should be noted that in the formulation of the criterion I have made one of the conditions for a linguistic function to be an unsaturated expression that it have the capability of having a referent. In the Begriffsschrift of the Grundgesetze it is

impossible to construct expressions which lack a referent, but this is not true of languages in general. By making the capability of having a referent one of the conditions in this criterion I allow certain linguistic functions to be unsaturated expressions even if they fail to have a reference. Allowing such unsaturated expressions is analogous to allowing a vacuous definite description, such as 'the least rapidly converging series', to be of the category of singular terms, although it lacks a referent.<1>

Some Examples of Pathological Functions

The reason why '⊢---- 2 + ζ = 5' is not a functional sign for Frege is that its value for any numerical singular term taken as argument is an assertion and assertions do not refer to anything. The only difference between '⊢---- 2 + 3 = 5' and '---- 2 + 3 = 5' is that the former has a symbolic indicator attached which indicates that we are dealing with an asserted proposition.<2> In ordinary language - and even in mathematical discourse - there are no mandatory assertoric force indicators.

<1> Dummett discusses how it is possible to construct predicates which lack a reference in Frege, pp.243-244. The method uses a higher level analogue of the definite description operator.

<2> It is a fairly common mistake to think that the assertion sign is the combination of symbols '⊢----'. The assertion sign is simply the vertical bar. In the Begriffsschrift Frege called the horizontal the content-stroke, but in Grundgesetze he simply referred to it as the horizontal.

The same string of words can be used as an asserted proposition or an unasserted one. It is only from the context that we can say whether '2 + 3 = 5', say, in a mathematical text is to be understood as having or lacking assertoric force.

There are indicators, however, in ordinary language which show whether sentences have forces other than the assertoric attaching to them. Interrogative sentences, such as 'Does Jill love Jack?', are used to ask questions and so can be said to be uttered with interrogative force.<3> This suggests the formation of linguistic functions which have interrogative sentences as their values. An example of such a linguistic function is the one which for any singular term X taken as argument has the value ⌜Does X love Jack?⌝ Quite clearly this has no reference and so could not be a functional sign for Frege, but it is a perfectly legitimate linguistic function.

In a similar way it is possible to construct linguistic functions whose values are derivations.<4> For example, there is that linguistic function whose value for any singular term X taken as argument is the derivation ⌜X loves Jill, therefore X is crazy⌝. This can be represented by means of Frege's xi-notation as 'ξ loves Jill, therefore ξ is crazy'.

The word 'therefore' can be used to construct pathological

<3> There are also other ways of indicating interrogative force in ordinary language which I shall ignore here. For example, accompanied by suitable prosodic and paralinguistic modulations the sentence 'Jill loves Jack' can be used to ask a question.

<4> A more natural word than 'derivation' here would be 'argument', but to use this here would be confusing because I also talk of the argument of a function.

functions. Often it looks as if 'therefore' makes a sentence out of two propositions.<5> For example, the sentence 'Jack snores, therefore someone snores' looks as if it has been made from the two propositions 'Jack snores' and 'someone snores' by means of the linguistic function '___ therefore ...', but clearly this is a very different type of operator from the binary truth-functional connectives. It does not have as its referent a function from pairs of truth-values to truth-values nor a function from truth-values to functions from truth-values to truth-values, and there is no other candidate for its referent in the Fregean hierarchy of types.

It might be suggested that we add a new class of complete entities to our ontology. These could be called validity-values and there would be two of them, namely, the Valid and the Invalid. Then, a derivation like 'Jack snores, therefore someone snores' would refer to the Valid and the derivation 'Jack snores, therefore Jill snores' would refer to the Invalid. And a function like '$\}$ snores, therefore someone snores' would refer to a function from objects to validity-values. This would, however, involve a wholesale revision of our notion of reference and I shall not pursue it further.

All of the examples of pathological linguistic functions mentioned so far fail to satisfy the natural criterion because they do not refer to anything. It is easy to construct lots more

---

<5> I am well aware that 'therefore' sometimes joins more than two propositions, but that does not affect my point. Similar considerations apply in those cases as well.

examples of such functions whose values are sentences which have indicators that relate to the force with which they are conventionally uttered, but there is little point in doing so.

There is another class of pathological linguistic functions which, although they cannot be expressed by means of Frege's auxiliary notation, do have a referent of the right type. One example of this class is that linguistic function which maps a person's name onto his father's name. This function takes, for example, the arguments 'Isaac', 'Jacob' and 'Judah' onto the values 'Abraham', 'Isaac' and 'Jacob', respectively. It is impossible to represent this function by means of Frege's xi-notation. In order to see this, consider the fact that the value of this function for the argument 'Isaac' is 'Abraham', but the word 'Isaac' does not occur in the value 'Abraham'; so, it is impossible to introduce this function in Frege's characteristic way, for this would have to include the sentence fragment 'people recognise the same function in "Abraham", "Isaac" and "Jacob" ...', but there is nothing to recognise. The referent of the linguistic function which maps a person's name onto his father's name is that ontological function which maps a person onto his father.

Another example of this class of pathological linguistic functions is that function which for any proper name X taken as argument has as value the name of the father of the bearer of X concatenated with the word 'loves' concatenated with the name of the mother of the bearer of X. For example, this function yields the value 'Abraham loves Sarah' for the argument 'Isaac'. It is

again impossible to represent this by means of Frege's xi-notation. The referent of this linguistic function is that ontological function which maps a person onto the truth-value the True if his father loves his mother and onto the False otherwise.

A different kind of example belonging to this class of pathological linguistic functions can be constructed by looking carefully at Frege's characteristic method of introducing unsaturated expressions. He says, for example, that people recognise the same numerical function in the expressions '2 + 3', '5 + 3' and '7 + 3' and that this numerical function is the referent of 'ξ + 3'. But what if someone did not recognise this numerical function, but rather recognised that function which is the referent of the linguistic function which for the argument X, where X is either '2', '5' or '7' returns X @ '+ 3' but for every other numerical designation returns the numeral '7'? The numerical function in question here is the one which maps the numbers 2, 5 and 7 to 5, 8 and 10, respectively, and maps every other number onto the number 7. The numeral '7' and the number 7 are here arbitrary. Any singular term and its referent could be substituted in the appropriate places with as much or as little justification. They are supplied by the person involved in the recognition, so the numerical function really involved here is the one that is undefined for every number other than 2, 5 and 7. And the linguistic function involved is undefined for every numeral other than '2', '5' and '7'.

This function cannot be expressed by means of Frege's auxiliary notation, because - as I have explained it - the expression '} + 3' means that linguistic function which given any numerical singular term X taken as argument yields as value X @ '+ 3'.

Concerning the issue of which function is recognised, I have always assumed that the most "general" or the most "informative" is always recognised as this seems the most reasonable assumption. The numerical function which maps 2, 5 and 7 to 5, 8 and 10, respectively, and every other number to 7 is less "general" or less "informative" than the one which maps every number to the sum of itself and three, because the latter function is properly defined for arguments other than 2, 5 and 7. Unlike the referent of the pathological linguistic function it does contain the information which would determinately tell us what its value is for all numerical arguments.

It is also possible to construct pathological linguistic functions which fail to satisfy both conditions of the natural criterion. They neither have a referent of the right type nor can they be expressed by means of Frege's auxiliary notation. An example in this class that I shall discuss is a linguistic function whose value is an infinitely long expression. It is that function which for any singular term X taken as argument yields as value the expression Y where Y is the same expression as X @ Y. Thus, for the argument 'Abraham' the value of this function is 'Abraham Abraham Abraham Abraham ...' That such a function actually exists should not be taken on trust. It is

something that can be proved to exist.<6>

So far all the pathological linguistic functions that I have considered, although they fail to satisfy the natural criterion of demarcation, are still fairly closely related to Frege's philosophy of language. There is no reason for this to be the case. It is possible to construct linguistic functions based upon the Aristotelian tradition of logic. I am thinking, for example, of that linguistic function which for any pair of terms T and U given as arguments yields as value the proposition ⌜every T is a U⌝.<7> This maps, for example, 'Greek' and 'philosopher' onto 'every Greek is a philosopher' and it maps 'man' and 'woman' onto 'every man is a woman'. It is even possible to extend Frege's auxiliary notation to allow it to express such functions. We could use the Greek letters tau and upsilon to have the same relation to terms as xi and zeta have to proper names. Then the linguistic function becomes 'every τ is a υ'. This is pathological because it does not have the correct type of referent.

<6> See, Stoy, Denotational Semantics, p.185. In fact, the entire Section entitled "Syntactic Lattices" is very instructive in this context: pp.182-190.

<7> Needless to say, the word 'term' in this paragraph is used differently from how I use it elsewhere. For example, it has no connection of meaning with the word as it occurs in the phrase 'singular term'.

Word-Forming Linguistic Functions

So far all the natural language linguistic functions that I have considered have kept words intact, but it is possible to devise linguistic functions which deal with word fragments. For example, the adjectives 'soldierly', 'priestly' and 'fatherly' are formed in the same way from the nouns 'soldier', 'priest' and 'father', respectively. The linguistic function involved here can be expressed as follows: Given a noun X as argument the value of this function is X & 'ly'.<8> As formulated here this linguistic function will apply to any noun in English. In some cases the value of this function is not an English adjective. For example, although, 'house' and 'bank' are nouns, the results of applying this function to them, namely, 'housely' and 'bankly', are not English adjectives.<9> In the cases when the value of this function is an adjective we can discern a connection of meaning between its argument and value: X means (roughly) ⌜of or pertaining to a X or Xs⌝.<10>

The prefix 'un' can also be understood as a linguistic function (which might be better symbolised as 'un-') which, for example, makes adjectives out of adjectives. So, for the

---

<8> Here the ampersand '&' is used to denote concatenation without the insertion of a space.

<9> They do not occur in either the OED or its supplements.

<10> This way of giving semantic rules for word-forming linguistic functions was suggested by the way in which dictionaries define words in conjunction with Radford's discussion of word-formation and lexical-redundancy rules in Chapter 4 of his book Transformational Syntax.

arguments 'annotated', 'catalogued' and 'inhabitable' it yields the values 'unannotated', 'uncatalogued' and 'uninhabitable'. Here the connection of sense between the arguments and values is that the values are negations of the arguments. The prefix 'un-' also makes verbs out of verbs. Thus, for the arguments 'block, 'cork' and 'zip' it returns the values 'unblock, 'uncork' and 'unzip', respectively. Here the semantic role of the prefix is to indicate the reversal of a process.

This example raises a number of questions. Are we dealing here with one linguistic function or two? In both cases the function involved adds the letters 'un' to the front of a word, but the semantic effect of this addition is different in the two cases. So, there are reasons for considering that two functions are involved.

Another question raised is how these functions are to be extrapolated. (Assuming that we are here dealing with two functions.) In the case of the function which makes adjectives out of adjectives are we interested in a linguistic operation which turns an adjective into its opposite - whether this is achieved by prefixing 'un' or 'in' or 'ir' or 'a' - or are we primarily concerned in the linguistic operation of prefixing the letters 'un' to an adjective irrespective of whether the result is or is not an English word? Given the adjective 'reflexive' the first linguistic operation mentioned would result in 'irreflexive', whereas the second would yield 'unreflexive', which - at present - is not a word in the English language.

Both sorts of operation mentioned in the previous paragraph

have their uses. This can be illustrated by one of Radford's examples. He considers what I would say was the linguistic function which for any verb X taken as argument yields as value the character-string X & 'ment'.<11> So, for the arguments 'abandon', 'curtail' and 'appease' it yields the values 'abandonment', 'curtailment' and 'appeasement', respectively. These are all English words, but this function for the argument 'repeal' yields 'repealment', which is not an English word. In my scheme of things - which is different from Radford's - in addition to the linguistic function mentioned, there is also the function which for any verb X taken as argument yields as value the noun which means ⌜act of Xing⌝.<12> This yields, for example, 'development', 'arrival' and 'criticism' for the arguments 'develop', 'arrive' and 'criticise', respectively. This rule is useful for describing the actual lexicon of English, whereas the other rule is useful if we are trying to coin new words. This is because the creation of new words tends to follow the dominant paradigm for word-formation in that area. If you need a noun that means ⌜act of Xing⌝ for a verb X and such a noun does not exist in English, you are more likely to construct it by adding the letters 'ment' to X than by forming it by analogy with the fact that 'destruction' and 'destroy' stand in this semantic relation.

<11> Transformational Syntax, pp.134-136.

<12> Such a linguistic function would not fit into Radford's scheme of things, because he assumes - as do other linguists influenced by Chomsky - that syntax is independent of and prior to semantics.

There are also many pathological linguistic functions in this area. For example, we can construct a linguistic function '-ife' which adds the letters 'ife' to whatever string is given it as its argument. Thus, it makes 'wife', 'strife' and 'life' out of 'w', 'str' and 'l', respectively. Another example is the function 'l-' which yields 'love', 'light' and 'life' when applied to 'ove', 'ight' and 'ife', respectively. These are perfectly respectable linguistic functions, but there is something pathological about them. What appears to be going wrong here is that the strings on which both the functions '-ife. and 'l-' operate are not morphemes, so it is impossible to formulate any sort of semantic rule to correspond to them.<13>

Unsaturated Expressions and Patterns

In this and the next Section I want to discuss two uses to which the nexus of ideas related to the notion of a pathological linguistic function can be put. (a) They can be used in the construction of an argument to show the incorrectness of an

---

<13> I do not wish to claim that either Wittgenstein or Husserl was concerned with precisely this point, but - in effect - both of them consider such pathological functions. In Philosophical Grammar Wittgenstein considers the function 'co-' which out of 'rn', 'al' and 'lt' makes 'corn', 'coal' and 'colt' (pp.316-317) and in Volume II of Logical Investigations Husserl considers the function 'fu-' which makes 'futile' out of 'tile', 'fugitive' out of 'gitive' and 'fuming' out of 'ming' (pp.502-503). It was by reflecting on these passages that I was led to formulate the notion of a pathological linguistic function.

exegesis of unsaturated expressions as patterns. (b) They can be used to show that in general it is false to say that we can only come to know an ontological function as the referent of some functional sign or other. In this Section I will discuss the first of these.

Dummett interprets Frege's incomplete expressions as patterns or features of sentences:

> There is no part in common to the sentences 'Brutus killed Brutus' and 'Cassius killed Cassius' which is not also a part of the sentence 'Brutus killed Caesar': yet the predicate '} killed {' is said to occur in the first two and not in the third. Such a complex predicate is, rather, to be regarded as a feature in common to the two sentences, the feature, namely, that in both the simple relational expression '... killed ...' occurs with the same names in both of its argument-places.<14>

By analogy with the notion of a pathological linguistic function it is possible to construct pathological patterns or to discern, say, a common pathological feature in two interrogative sentences. For example, there is no part in common to the interrogative sentences 'Did Brutus kill Brutus?' and 'Did Cassius kill Cassius?' which is not also a part of the interrogative sentence 'Did Brutus kill Caesar?', but the first two have a common feature, namely, that the expression 'Did ... kill ...?' occurs in both of them with the same proper name in both of its argument-places. Dummett does not raise - let alone try to answer - the question of how the pattern or feature, qua pattern or feature, '} killed {' differs from the pathological

---

<14> Frege, p.31. See also p.250, where the word 'pattern' appears.

pattern or feature 'Did ζ kill ζ?' He does not ask what property the first pattern possessess that the second lacks in virtue of which the first is and the second is not an incomplete expression. In fact, it is difficult to see how a criterion could be devised which distinguishes between legitimate patterns, such as 'ζ killed ζ', and pathological ones, such as 'Did ζ kill ζ?' Because of this difficulty, the interpretation of incomplete expressions as patterns or features cannot be correct.

Someone might try to devise a criterion by analogy with what I have called the natural criterion of demarcation. They might try saying that an incomplete expression is a pattern or feature which can be expressed by means of Frege's auxiliary notation and that is capable of having a functional referent. In this case a further argument can be brought to bear against the pattern-interpretation. By the principle of parallel interpretation if someone interprets unsaturated expressions as patterns or features, then he is committed to interpreting their referents as patterns or features. But although there is some plausibility in saying that 'ζ killed ζ' is a feature of the sentence 'Brutus killed Brutus', there is no plausibility in saying that the concept ζ killed ζ is a feature of both the True and the False. (A defender of this position would be committed to saying that the concept ζ killed ζ was a feature of the True when that concept was applied to a person who actually did kill himself, but that it was a feature of the False when applied to a person who did not kill himself.) Concepts are neither features nor patterns. They are functions from from objects to truth-values.

Therefore, this attempt to devise a criterion of demarcation between legitimate and pathological patterns fails.<15>

I have presented the above argument in terms of patterns and features, but - with suitable modifications - it can be turned into an argument against the interpretation of incomplete expressions as stencils or schemata.

I have just argued that Dummett's interpretation of unsaturated expressions as patterns or features is incorrect, but it will be profitable to explore how this exegesis fits in with some of his other interpretations of Frege's ideas. He writes that an unsaturated expression

> is not merely metaphorically but literally incomplete; it is something formed from a sentence by omission, rather than something that was assembled on its own in the course of constructing the sentence from which it can be so formed (Frege, p.63).

And he frequently talks about incomplete expressions being formed from complete expressions, such as complex numerical designations or propositions, by the omission or removal of another complete expression, such as a numeral or proper name.<16> For Dummett, whenever we are faced with a complex predicate or other unsaturated expression, there always has to be a proposition or other saturated expression from which it was formed. For him, it

---

<15> This objection cannot be sustained against my interpretation of incomplete expressions as linguistic functions, since I also regard their referents as being functions and I understand both sorts of functions as rules.

<16> For example, on pp.11, 16-17, 23, 30-31, 39 and 44-45.

is impossible to construct a predicative expression in any way other than by removing a complete expression from another complete expression. Such views do fit together well with his view that unsaturated expressions are patterns or features, since clearly a feature has to be a feature of something or other.

It is true that Frege on occasion uses similar language to Dummett's talk of omission and removal. For example, in explaining his second way of forming names he talks of excluding from an expression a proper name that is a part of it (Grundgesetze, p.47). It seems to me that such language has to be interpreted in context. Typically, it comes after passages in which Frege has explained what an unsaturated expression is by means of his characteristic method of introducing functions and the signs that refer to them. This consists in displaying a number of complex expressions and "isolating" or "abstracting" the functional sign in question.<17> The relevant functional sign - and the ontological function that it refers to - are determined by the context in which the "abstraction" takes place. For example, '2 + 2' is the result of saturating a number of unsaturated expressions, but in the context of '3 + 3' and '4 + 4' the most natural one to think of is '$\zeta + \zeta$' whereas in the context of '2 + 3' and '2 + 7' it is more natural to think of '2 + ζ'. Accepting this it is only a small step to take in order to construe functional signs as rules of formation. The language of omission and removal must be construed as being elliptical for

---

<17> Needless to say, abstraction here has nothing to do with psychological abstraction.

Frege's characteristic method of introducing functions and functional signs.

One of the most important differences between my interpretation of Frege on this point and Dummett's is that in my account it is possible to grasp an unsaturated expression without there being a complete expression from which it was formed. I have explained the unsaturated predicate '} snores' as being that linguistic function which out of any singular term X makes the proposition X @ 'snores'. In spelling out this rule no mention is made of any sentence containing the word 'snores'. In Dummett's account it is impossible to introduce a new unsaturated expression into our language, except by means of some complete expression in which it figures, but if you understand predicative expressions constructively - as I do - then it is possible to introduce new unsaturated expressions into our language without there being any complete expressions in which they figure. In fact, those unsaturated expressions will be used in the construction of the complete expressions in which they occur.

And this way of thinking of unsaturated expressions corresponds to Frege's description of the Begriffsschrift in Grundgesetze. He does not there introduce his primitive unsaturated expressions by first displaying complete propositions of the Begriffsschrift. What he does do is first explain his primitive unsaturated expressions and then use them in the construction of the Begriffsschrift's propositions.

I am not saying that it is possible to grasp the general notion of an unsaturated expression without ever having seen a

complete expression in which an unsaturated expression figures. Frege characteristically introduces the general notion by showing how a particular unsaturated expression, say, 'ξ sleeps', can be obtained from a number of sentences in which it occurs, say, 'Jack sleeps', 'Maxine sleeps' and 'Jill sleeps'. But once you have grasped the general notion in this way, there is no reason to go through an analogous process every time you want to introduce an unsaturated expression into the language. In Grundgesetze, after Frege has conveyed to the reader what an incomplete expression is, he does not then repeat the process every time he wants to introduce a new predicative expression. Once you have grasped that 'ξ sleeps' is an unsaturated expression by recognising it in 'Jack sleeps', 'Maxine sleeps' and 'Jill sleeps', you can then grasp the unsaturated expressions 'ξ snores', 'ξ smokes', 'ξ drinks', etc., by analogy with your understanding of 'ξ sleeps'. You do not have to go via the propositions 'Jack snores', 'Jack smokes', etc. (In Dummett's scheme of things you would have to do this.) In my account it is possible to grasp the general notion of an unsaturated expression by seeing just one unsaturated expression occurring in a complete expression, but you do not have to see every predicative expression occurring in some complete expression or other in order to recognise it as predicative. And once you have grasped the general notion of an incomplete expression you can construct predicative expressions without reference to any complete expressions in which they figure.

If you understand unsaturated expressions as being linguistic

functions, then it is possible to give a straightforward account of the phrases 'occurs in', 'omitted from' and 'fills the gap in', which Dummett uses a lot. An expression occurs in another if either it is a linguistic function which yields that other expression as value for some argument or it is the argument to some linguistic function which yields that other expression as value. An expression is considered to have been omitted from another if it is viewed as the argument of some linguistic function that has that other expression as its value. An expression fills the gap in another when that other expression is a linguistic function that has been applied to it.<18>

Dummett's view of unsaturated expressions as features of sentences leads him to say that incomplete expressions are literally incomplete. In order to further show that Dummett is mistaken, I will show that Frege understood the language of incompleteness metaphorically.

Frege uses only a small number of expressions to describe the essential peculiarity of functions and functional signs. He says that they are incomplete, unsaturated, in need of supplementation and predicative. He several times states that such language is to be understood metaphorically. For example, after saying that both functional signs and functions can be called 'unsaturated' he writes:

> Of course this is no definition; but likewise none is here

<18> These explanations should be compared with Dummett's on pp.45-48 of Frege. They are more straightforward and more general than his explanations.

> possible. I must confine myself to hinting at what I have in mind by means of a metaphorical expression, and here I rely on my reader's agreeing to meet me half-way. ("What is a Function?", p.665.)

Similarly, near the end of his paper "On Concept and Object" he writes:

> 'Complete' and 'unsaturated' are of course only figures of speech; but all that I wish or am able to do here is to give hints. (P.205.)

I think it is important to stress this point, because it is surprisingly easy to forget that such language - especially the language of incompleteness - is being used metaphorically and to think of functions and functional signs as literally incomplete. It is harder - if not impossible - to construe 'unsaturated' literally and this is Frege's commonest description of functions in his later writings, especially in Grundgesetze. This is because the notion of unsaturatedness is drawn from chemistry and it is unclear how something non-chemical, such as a numerical function, could be literally unsaturated.

Frege calls the language of incompleteness and unsaturatedness metaphorical and figurative. It is definitely not meant literally, but it does not seem to be truly and fully metaphorical. In the case of a proper metaphor, such as Homer's description of Achilles as a lion, we can learn about and grow in understanding of Achilles's character and behaviour by learning how Homer understood the sense of the word 'lion'; but in the case of Frege's metaphors this does not happen. Frege most

frequently refers to functional signs and functions as being unsaturated, yet when you find out what an unsaturated molecule is it does not help you to understand how Frege understood functions.<19> In fact - to the best of my knowledge - none of Frege's commentators go to the trouble of explaining what an unsaturated molecule is.<20> So, rather than dealing with true metaphors here, we are dealing with words whose meaning is to be derived from the contexts in which they occur rather than their conventional associations. And the characteristic way in which Frege introduces functional signs strongly suggests an interpretation in terms of rules of formation.

Functional Signs and their Referents

The view that we can only come to know an ontological function by knowing a functional sign that refers to it has some attraction,<21> but the existence of pathological linguistic

---

<19> An account of unsaturated molecules can be found in any standard textbook of organic chemistry which has not been published recently. (The term seems to have disappeared from modern treatments.) For example, an account is to be found on pp.118-119 of Durrant's Organic Chemistry.

<20> The fullest account in writings on Frege seems to be in a footnote to the glossary contained in the Geach-Black translations, where the editors write that Frege 'may well have had in mind unsaturated molecules, which, without dissolution of their existing structure, can take up more atoms' (p.x, footnote 2).

<21> It is a position that Dummett holds, see Frege, p.254.

functions shows that such a view must be - in general - false.

In order to establish this conclusion recall how the notion of a functional sign was explained. A functional sign is a linguistic function expressable by means of Frege's auxiliary notation which is capable of having a functional referent. We need to have some independent access to ontological functions - independent of knowing them as the referents of functional signs - in order to be able to use them to distinguish functional signs from pathological linguistic functions.

Considered as linguistic functions there is nothing to differentiate between 'ξ killed ξ' and 'Did ξ kill ξ?'. No examination of these expressions will reveal a property that one of them has that the other lacks, yet the first is and the second is not an incomplete expression. The difference between them is that one of them has an extra-linguistic correlate, but the other does not. We are thus frustrated in trying to characterise ontological functions as the referents of functional signs, since we need to know that a linguistic function has a referent in order to classify it as a functional sign.

It would be too much of a digression for me to explain how it is possible to come to know particular ontological functions other than as the referents of functional signs. I just want to say here that an attempt to do this would not involve me in a defence of the ludicrous position that we can come to know particular ontological functions unmediated by language.

Chapter 3: A Problem with Unsaturatedness

An Awkwardness of Language

In his paper "On Concept and Object" Frege replies to a number of criticisms of his views about concepts made by Kerry. The details of Kerry's criticisms of Frege's Grundlagen need not concern us here, but in the course of the discussion Frege comes across what he calls 'an awkwardness of language' (p.196) that arises when we try to talk about concepts. One of Kerry's examples was the proposition: 'The concept "horse" is a concept easily attained'. He said that the concept 'horse' was an object which fell under the concept 'concept easily attained'. This has the consequence that the concept 'horse' is not a concept, but an object. And this is the awkwardness that Frege refers to. But why does Frege accept this? The reason is to be found in his discussion of concepts and objects in Grundlagen.

In Grundlagen Frege laid down a criterion for distinguishing between expressions that stand for objects and those that stand for concepts. He, of course, expressed the criterion for German

phrases, but it can be reformulated for English phrases as follows: A singular substantival phrase governed by the definite article stands for an object, whereas a word 'used with the indefinite article or in the plural without any article' is a concept-word.<1> Dummett claims that this criterion is inexact both ways and he adds that

> Frege was perfectly well aware that there are expressions satisfying this criterion which he would not wish to admit as proper names, and others which fail the criterion which he would wish to admit: but he was content to allow the whole distinction between proper names and expressions of other kinds to depend upon intuitive recognition, guided only by the most rough and ready of tests. (Frege, p.54.)

That is a misrepresentation of Frege's attitude. He thought the criterion was almost watertight. The only exception he could think of as regards the indefinite article was an obsolete German formula for a councillor and as regards the definite article the only troublesome cases he mentions are propositions like 'the horse is a four-legged animal' ("On Concept and Object", pp.195-196). It is because Frege thought that the criterion had hardly any exceptions that he held the proposition that the concept 'horse' is an object to be true. If he really did have a low opinion of this criterion, there would have been little reason for him to categorise phrases like 'the concept "horse"' as singular terms. Although Frege accepted this consequence of his criterion it is clear that he was not entirely happy with it,

---

<1> Section 51. See also Section 57 and the footnotes to Sections 66 and 68.

for he wrote:

> I admit that there is a quite particular obstacle in the way of an understanding with my reader. By a kind of necessity of language, my expressions, taken literally, sometimes miss my thought; I mention an object, when what I intend is a concept. I fully realize that in such cases I was relying upon a reader who would be ready to meet me half-way - who does not begrudge a pinch of salt.<2>

This further shows the high regard he had for his criterion. He was willing to accept this consequence of it rather than give the criterion up.

This result has come to be known as Frege's paradox. When it is discussed it is usually presented as a problem to do with Frege's understanding of concepts, but it is important to realise that it is a problem which also affects his account of relations and, indeed, functions in general. In "On Concept and Object" Frege explicitly says that the problem arises for relations,<3> and also for functions.<4> In view of the fact that Frege's understanding of concepts and relations is modelled on his understanding of functions - a concept is a function from objects to truth-values and a relation is a function from objects to

---

<2> "On Concept and Object", p.204. Wright observes that between pages 64 and 67 of Grundlagen 'there are no less than eight occasions when salt is called for!' (Frege's Conception of Numbers as Objects, p.171, footnote 1 to Section iv of Chapter 1.)

<3> He says on p.205 that 'the words "the relation of an object to the concept it falls under" designate not a relation but an object'.

<4> He says that the expression 'the function f(x)' does not stand for a function (p.197, footnote †). Obviously, it stands for an object.

concepts - the primary version of the problem is that which occurs for what are usually called functions. The problem is normally, however, discussed mainly in terms of how it arises for concepts and I shall follow this tradition.<5>

The Geach-Dummett Solution

The proposition 'the concept horse is not a concept' is the negation of 'the concept horse is a concept' and this proposition, from its surface structure, appears to be analysable into the singular term 'the concept horse' and the predicate 'ξ is a concept'. Stated simply, the Geach-Dummett solution says that this analysis is incorrect, because the categorisation of the constituents is incorrect. I shall first look at the categorisation of the apparent predicate 'ξ is a concept' and then at that of the apparent singular term 'the concept horse'.<6>

<5> Up to now I have used expressions like 'the concept "horse"'. This was because I was following Kerry's usage in discussing his example, as indeed does Frege. From now on, however, I shall either use expressions of the form 'the concept horse' or of the form 'the concept ξ is a horse'. The second alternative is more precise but it is also more awkward. It is probably for this reason that Frege preferred the first alternative in cases where it was possible to isolate a concept-word.

<6> Dummett says in Frege, pp.211-217, that the solution was discovered by Frege himself after the publication of "On Concept and Object", but there is no textual support for this claim in Frege's writings. In the posthumously published paper "Comments on Sense and Meaning" Frege says a few things

'ξ is a concept' appears to be a predicate which has been constructed by analogy with 'ξ is an object'. This latter predicate has the property that when its argument-place is filled with a non-vacuous singular term it always yields a true proposition, irrespective of what object the singular term refers to. In constructing 'ξ is a concept' it was hoped to obtain a predicate which yields a true proposition whenever its argument-place is filled with an expression that stands for a concept, but expressions which stand for concepts are not fitting for its argument-place. In fact, it looks as if 'ξ is a concept' is a predicate which yields a <u>false</u> proposition whenever its argument-place is filled by a non-vacuous singular term.

In order to construct a linguistic function which yields a true proposition whenever its argument-place is filled by a non-vacuous predicate, we have to go up to the second-level and for this role Dummett suggests (<u>Frege</u>, pp.216-217):

$$(Ax)\ (\varphi(x)\ v\ -\varphi(x)),$$

which can be expressed in ordinary language as '... is something which everything either is or is not'. This is of category (N --> P) --> P and it has the required property. Dummett advocates the banishment of the phrase 'is a concept' from our language and its replacement with this second-level predicate.

---

which are superficially similar to parts of the Geach-Dummett solution. See also Anscombe's <u>An Introduction to Wittgenstein's Tractatus</u>, pp.111-112, and Geach's "On What There Is", pp.132-134, where the solution was first published.

The expression 'the concept horse' appears to be a singular term, but both Geach and Dummett argue against this categorisation. Surprisingly, rather than conducting their discussion in terms of the expression 'the concept horse', which Frege uses, they prefer the form of words 'what "} is a horse" stands for'. For their discussion to be a discussion of Frege's paradox we must assume that these two expressions are equivalent in every way.<7> Geach writes:

> The result of inserting an English expression in the blank between the quotes in the context:
>
> what ' ' stands for
>
> will stand for, bedeutet, whatever that very English expression stands for ... ("Saying and Showing in Frege and Wittgenstein", pp.56-57.)

As an example he considers the predicate '} killed Caesar'. He says that the expression 'what "} killed Caesar" stands for' is a predicate. It could be said against this position that the expression 'what "} killed Caesar" stands for' produces nonsense when substituted for '} killed Caesar' without alteration. For example, making this substitution in the sentence 'Brutus

---

<7> The form of words 'what "..." stands for' hardly ever occurs in Frege's writings. I know of only two occurrences of a similar form of words and they are both in the posthumously published article "Comments on Sense and Meaning", p.132. He there says that we should outlaw the expression 'the meaning of the concept-word A' and adds: 'It would be better to confine ourselves to saying 'what the concept-word A means', for this at any rate is to be used predicatively: "Jesus is, what the concept-word 'man' means" in the sense of "Jesus is a man".'

killed Caesar' results in the string of words 'Brutus what "ξ killed Caesar" stands for'. Geach says, however, that the string of words 'Brutus is what "ξ killed Caesar" stands for' is acceptable and nothing of logical significance hangs on the fact that grammar demands the copula 'is'.

Applying these insights to the expression 'the concept horse' we see that this expression, although looking on the surface like a singular term, is actually a predicate.

Putting together these accounts of the expressions 'the concept horse' and 'is a concept' we see that it is no longer possible to construct such pseudo-propositions as 'the concept horse is a concept' or 'what "ξ is a horse" stands for is a concept'. According to Dummett, what we hoped to convey by means of these pseudo-propositions is correctly expressed, respectively, by the propositions 'a horse is something which everything either is or is not' and 'what "ξ is a horse" stands for is something which everything either is or is not'.<8> It should be noted that, whereas for Frege - at least when he wrote "On Concept and Object" - it was true to say that 'the concept horse is not a concept', for Dummett the proposition 'a horse is not something which everything either is or is not' is false.

Dummett urges that the pseudo-predicate 'ξ is a concept' be banished from our language; then it will be impossible to construct such paradoxical sounding propositions like 'the

---

<8> Frege, pp.216-217. Dummett replaces the phrase 'the concept horse' here by 'a horse', no doubt for idiomatic reasons.

concept horse is not a concept' (Frege, p.217). But it is possible to rehabilitate this instead and turn it into a legitimate linguistic function. If we reject Frege's criterion for distinguishing between singular terms and concept-words, then just because the indefinite article occurs in the phrase 'is a concept' it does not follow that the word 'concept' is a concept-word. It is possible to construe the phrase 'is a concept' as a second-level predicate. In fact, we can regard the expression '... is a concept' as referring to the same second-level concept as the expression '... is something which everything either is or is not' refers to. Because this possibility has not been used by either Geach or Dummett I shall not mention it again.

Further Aspects of Dummett's Solution

Dummett says that the problem with talking about unsaturated entities begins 'with the words "concept", "relation" and "function"' (Frege, p.213), and that solutions similar to the one he gives for the version of the paradox involving the word 'concept' can be given for the versions of the paradox which make use of the words 'relation' and 'function' (ibid., pp.213 and 218). His solution does indeed work for the pseudo-predicate 'ζ is a relation', but not for 'ζ is a function'. It does, however, work for each element in the potentially infinite list of pseudo-predicates: 'ζ is a one-place first-level function',

'$\}$ is a two-place first-level function', '$\}$ is a three-place first-level function', and so on. It also works for all the specific pseudo-predicates which appear to be true of particular types of second-level functions and so on. Considering some of these examples shows us why the solution fails for the pseudo-predicate '$\}$ is a function'.

Dummett would have to replace the pseudo-predicate '$\}$ is a one-place first-level function' by a second-level predicate of category (N --> N) --> P, which is true of all one-place first-level functions. A suitable candidate for this second-level predicate is

$$(Ax)\varphi(x) = x \vee \varphi(x) \neq x.$$

And he would have to replace the pseudo-predicate '$\}$ is a two-place first-level function' by a second-level predicate of category (N --> (N --> N)) --> P, which is true of all two-place first-level functions.<9> A suitable candidate for this second-level predicate is

$$(Ax)\psi(x, x) = x \vee \psi(x, x) \neq x.$$

The generic term 'function' is used as if it were true of all one-place first-level functions and also true of all two-place first-level functions and also true of all three-place first-

---

<9> He would also have to find a suitable replacement for that pseudo-predicate which looks as if it is true of functions of category (J * J) --> J.

level functions and so on. But Dummett cannot account for this generic term being in our language. This is because he would require 'ξ is a function' to be replaced in (as well as many others) both the propositions: 'every one-place first-level function is a function' and 'every two-place first-level function is a function'. In the first of these 'ξ is a function' would have to be replaced by something of the same category as what 'ξ is a one-place first-level function' is replaced by and in the second by something of the same category as what 'ξ is a two-place first-level function' is replaced by. Quite clearly, these two replacement second-level predicates cannot be combined to form a single unified second-level predicate, because they have different sorts of argument-place. An expression which was fitting for one of them would not be fitting for the other. Thus, there is no consistent replacement for 'ξ is a function' which would work in both these cases. Hence, there can be no replacement for the pseudo-predicate 'ξ is a function', which appears to be true of all first-level functions and all second-level functions and so on. Unlike the pseudo-predicate 'ξ is a one-place first-level function', which he says should be banished from our language and replaced by a certain second-level predicate, in the case of the generic term 'function', he has to say just that it be banished from our language. His account cannot be modified in such a way as to find a suitable replacement for the generic pseudo-predicate 'ξ is a function'.<10>

<10> Consideration of this argument shows that there can be no

A similar argument shows that there can be no replacement for the pseudo-predicate 'ξ is an entity', which almost everyone who writes about Frege's ontology finds so useful. Currie, for example, says that according to Frege 'every entity is either a function or an object' (Frege, p.85). In an account such as Dummett's there cannot be a genus entity of which one-place first-level function and two-place first-level function are species, let alone function and object. To be fair to Dummett, he does recognise this, for he writes that 'the word "entity" is un-Fregean, in that it trespasses over the bounds dividing one level from another' (Frege, p.523), but rather than suggesting the banishment of the word 'entity' he suggests construing a proposition in which it occurs as being 'something like a typically ambiguous formula within the theory of types' (ibid.). A similar argument to that given above shows this to be impossible, since in the proposition 'every entity is either a one-place first-level function or an object' we would have to assign two distinct types to the term 'entity' simultaneously, which cannot be done.

But even if we banish the pseudo-predicate 'ξ is an entity' from our language, we will still be left with many troublesome pseudo-propositions. An example occurs in Frege's writings:

---

replacement in Dummett's scheme of things for the generic notion of a concept, namely, that notion which appears to be employed in the phrases 'is a first-level concept' and 'is a second-level concept'. Dummett's account does, however, work for the specific notion of a concept. That is to say, that notion which is true of all entities of type J --> H. My discussion of the notion of a concept in the body of the text is restricted to the specific notion and does not apply to the generic notion.

'I count as objects everything that is not a function ...' (Grundgesetze, pp.35-36). For this to make sense the domain of quantification must be understood as including both objects and functions and - on Fregean principles - this is clearly impossible. (This string of words would still be meaningless and not false if we replaced 'function' by 'one-place first-level function', because there is no logically acceptable translation along the lines suggested by Dummett.)

So, if we accept Dummett's solution to Frege's paradox, we would have to make radical changes to our ontological vocabulary. Generic terms like 'function' and 'entity' would simply have to go, whereas specific terms like 'one-place first-level function' and 'two-place first-level function' would have to be replaced by second-level predicates. Furthermore, we would have to outlaw many constructions like 'everything which is not an object is a function'.

After presenting his solution to Frege's paradox Dummett writes:

> The terminology that would be required for speaking, in a logically correct manner, about the referents of predicates and relational expressions is ... cumbrous and verbose; it is therefore best, when there is no danger of misunderstanding or of antinomies, to revert to the logically erroneous vocabulary of 'concept', 'relation' and 'function'. (Frege, p.217.)

And he carries on his exegesis of Frege in terms drawn from the logically erroneous vocabulary, but - no doubt - with the logically correct terminology in mind.

Further Aspects of Geach's Solution

Geach's position is slightly different. Unlike Dummett, he does not first urge us to banish the troublesome words 'concept', 'relation' and 'function' from our language and then lets us revert to the logically erroneous terminology, if we use it carefully. Right from the start Geach realises that many of the propositions that we would like to assert about the differences of type between entities in Frege's ontology violate the principles on which those very differences of type are based. As an example he considers the proposition 'there is a difference between what "Brutus" stands for and what the predicate "- killed Caesar" stands for' ("Showing and Saying in Frege and Wittgenstein", p.57). On his account of the expression 'what "..." stands for' this translates into the proposition 'there is a difference between Brutus and killed Caesar' and this is manifest nonsense. Therefore, the original proposition is also meaningless and it could not even be formulated in a language such as Frege's Begriffsschrift. Geach believes, however, that such nonsensical propositions can be 'didactically useful', as he puts it (ibid., p.58), for they can be helpful in teaching someone to understand the Begriffsschrift.

Semantic Ascent

Frege's paradox is usually formulated in ontological terms and so far I have discussed it in such terms. It might be thought that this paradox could easily be solved by means of the manoeuvre that Quine calls semantic ascent and which he describes as follows:

> It is the shift from talk of miles to talk of 'mile'. It is what leads from the material (inhaltich) mode into the formal mode, to invoke an old terminology of Carnap's. It is the shift from talking in certain terms to talking about them. (Word and Object, p.271.)

To use this manoeuvre in an attempt to solve Frege's paradox is the approach that a follower of Carnap, for example, would favour. Carnap distinguished between three kinds of sentence, namely, syntactical sentences, object sentences and pseudo-object sentences. (The Logical Syntax of Language, p.286.) An example of an object sentence is 'water is not an acid but an alkali'. One species of pseudo-object sentence occurs frequently in philosophy. An example is '5 is not a thing but a number'. Such sentences give the impression of being about objects, but - according to Carnap - they are really disguised syntactical sentences. He claims that it is more precise to say that 'the numeral "5" is not a thing-word but a number-word'. Carnap goes in for a wholesale translation of similar philosophical sentences. He translates Kronecker's statement that 'God created the natural numbers (integers); fractions and real numbers,

on the other hand, are the work of man' into the syntactical sentence that 'the natural number symbols are primitive symbols; but the fractional expressions and the real-number expressions are introduced by definition' (ibid., pp.304-305).

Someone standing in this philosophical tradition might think that if we stop talking about entities, and talk exclusively in terms of linguistic expressions and the categories to which they belong, then Frege's paradox will never arise. Unfortunately, this is not the case, similar difficulties also occur on the linguistic level.

This is not really very surprising. Geach well articulates in general terms what is wrong with such an approach:

> Language, after all, is not something set over against the whole world, like the Divine Mind; languages are part of the world, linguistic facts and structures are facts and structures in the world. This sets a limit to the usefulness of semantic ascent in solving philosophical problems. We cannot solve the problem of universals by talking about the word 'pig' instead of The Pig; for there is exactly the same problem about the relation of the word 'pig' to its tokens as there is about the relation of The Pig to Jones's pigs ... So also in our case; the business of function and argument and value cannot be shelved by talking about expressions of different category, because it reappears if we consider, as we must, the ways of forming expressions out of expressions. ("Names and Identity", pp.142-143.)

His own criticism of those people who believe that Frege's paradox can be solved by the manoeuvre of semantic ascent consists in pointing out that they think of predicates and functional signs as complete expressions and what he does is to

show that predicates and functional signs are unsaturated expressions. (For example, in "Saying and Showing in Frege and Wittgenstein", pp.58-61.)

It is more interesting, however, to construct a version of the paradox on the linguistic level analogous to the ontological version. Frege is aware that something like this is possible, but he expresses himself badly. He says:

> A similar thing happens when we say as regards the sentence 'this rose is red': The grammatical predicate 'is red' belongs to the subject 'this rose.' Here the words 'The grammatical predicate "is red"' are not a grammatical predicate but a subject. By the very act of explicitly calling it a predicate, we deprive it of this property. ("On Concept and Object", p.196, footnote †.)

He expresses himself badly because he talks of grammatical predicates. This is probably why Geach says that this comment is irrelevant (op.cit., p.60), but re-formulated in terms of Fregean predicates, that is to say, unsaturated expressions which make a proposition out of a singular term, it becomes as problematical as the ontological version. Doing this we have the paradoxical proposition: 'The predicate "ξ is a horse" is not a predicate but a singular term'. Dummett tacitly performs such a re-formulation yet he says:

> But this latter fact is no more paradoxical than the fact that the expression 'the city of Paris' is not a city: the case would be comparable with 'The concept horse is not a concept' only if we had reason for saying that the predicate 'ξ is a horse' is not a predicate. (Frege, p.212.)

However, we have exactly the same reason for saying that the expression 'the predicate "} is a horse"' appears to be a singular term as for saying that the expression 'the concept horse' appears to be one, namely, Frege's criterion for distinguishing between singular terms and concept-words. The definite article followed by a singular substantival phrase figures in both of them.

Not surprisingly, because of the parallelism between the linguistic and the ontological levels in Frege's thought, a solution similar to that offered by Geach and Dummett to the ontological version of the paradox can be constructed for the linguistic version. It would go something like this:

The problem begins with the predicate '} is a predicate'. The predicate '} is a singular term' is true of all singular terms and in constructing '} is a predicate' we were hoping to devise a predicate that was true of all predicates; but the argument-place of '} is a predicate' is not fitting for predicates, it is only fitting for saturated expressions. In order to construct a suitable predicate we have to go up to the second level and devise a second-level predicate which is true of all first-level predicates. This is no trickier than for the ontological predicate '} is a concept' and a suitable candidate is: '... is something which yields a proposition when applied to a singular term'.

And the apparent singular term 'the predicate "ζ is a horse"' is really an incomplete expression which refers to a predicate and is fully equivalent with '"ζ is a horse"'.

Thus, what we hoped to convey by means of the pseudo-proposition 'the predicate "ζ is a horse" is a predicate' is correctly conveyed by the logically more accurate 'the predicate "ζ is a horse" is something which yields a proposition when applied to a singular term' or '"ζ is a horse" is something which yields a proposition when applied to a singular term'.

To be made fully analogous to Dummett's solution to the ontological paradox this linguistic version would have to be combined with the view that predicates like 'ζ is a predicate', 'ζ is a one-place first-level functional sign', 'ζ is a two-place first-level functional sign', and so on, be banished from our language and be replaced by their logically correct translations. The generic pseudo-predicates 'ζ is a functional sign' and 'ζ is an incomplete expression' (and others) would simply have to be banished.

Nested Quotation Marks In constructing the above linguistic version of Frege's paradox and its solution I have assumed that an expression enclosed in two sets of quotation marks - such as '"ζ is a horse"' or 'the predicate "ζ is a horse"' - is unsaturated and that it refers to something unsaturated, namely, 'ζ is a horse'. Needless to say, the varieties of

unsaturatedness associated with such expressions are very different from those associated with expressions enclosed in only one set of quotation marks.

This seems to me to be the correct assumption to have made, but it differs from both Dummett's and Frege's views about expressions enclosed in two sets of quotation marks. Concerning the predicate 'ξ is what "ξ is a horse" stands for' Dummett writes:

> The first 'ξ' indicates the argument-place of this predicate; the second 'ξ', being, in this predicate, between quotation marks, does not indicate an argument-place, but is a constant part of the expression. (Frege, p.214.)

This cannot be correct, because it has the consequence that something complete, namely, '"ξ is a horse"', refers to something incomplete, namely 'ξ is a horse'.

Similarly, Frege writes concerning nested quotation marks:

> While '( ).3 + 4' is a function name, '"( ).3 + 4"' is a proper name, and its meaning is the function name '( ).3 + 4'.<11>

Again, this cannot be correct, because it has the consequence that something complete, namely, '"( ).3 + 4"', refers to something incomplete, namely, '( ).3 + 4'.

<11> This is from a letter to Russell dated 29 June 1902. In the English version there is a typographical mistake - the second occurrence of '4' is missing - which I have rectified.

Both Dummett's and Frege's accounts violate the principle that only saturated things can refer to saturated things and that only unsaturated things can refer to unsaturated things. And if we give up this principle when one of the things involved is enclosed in two sets of quotation marks and the other in one, then why should we retain it when one of the things involved is enclosed in one set of quotation marks and the other is not enclosed in any? There is nothing magical about nested quotation marks. Once the above principle is violated the isomorphism between expressions and their referents - which is characteristic of Frege's philosophy - breaks down as well. There is nothing special about nested quotation marks which creates an immunity from being involved in this isomorphism. Thus, if we want to retain this isomorphism, then we cannot accept either Dummett's or Frege's account of nested quotation marks.

Difficulties in the Geach-Dummett Solution

I want to consider the operator 'what "..." stands for' in greater detail. As explained by Geach and Dummett this is an operator that cannot be assigned to a single syntactic category. According to Geach's rule (quoted on p.106 above) the expression 'what "Socrates" stands for' is a singular term and is just a long-winded way of referring to Socrates. Similarly, the

expression 'that function of 2 which "the square of" stands for' is a long-winded way of referring to the square of 2 (ibid., p.57). Thus, we must regard 'what "the square of ζ " stands for' in this case as being an unsaturated functional sign, that is to say, an expression of category N --> N.<12> We also have that the expression 'what "ζ killed ζ " stands for' is a relational sign and 'what "the brother of ζ and ζ " stands for' is a two-place functional sign. Furthermore, 'what "everything" stands for' is a quantifier of category (N --> P) --> P. These categorisations can be presented as follows:

(1) 'what "Socrates" stands for': N,

(2) 'what "ζ is a horse" stands for': N --> P,

(3) 'what "ζ killed ζ " stands for': N --> (N --> P),

(4) 'what "the square of ζ " stands for': N --> N,

(5) 'what "the brother of ζ and ζ " stands for': N --> (N --> N),

(6) 'what "everything" stands for': (N --> P) --> P.

Generalising, we can say that given an expression of arbitrary category the operator 'what "..." stands for' yields an expression of the same category which is everywhere substitutable with the original expression salva congruitate and salva veritate.

<12> Geach changes 'what' to 'which', no doubt for idiomatic reasons, and does not use Frege's xi-notation in this example.

Although both Geach and Dummett make use of this operator neither of them stops to consider the implications of its introduction or whether its existence can be justified. Had they done so they would have realised that in a language constructed on Frege's principles, in which expressions are assigned to distinct non-overlapping syntactic categories, there can be no such operator. This is because if an unsaturated expression has an argument-place that is fitting for singular terms, then ipso facto it is not fitting for predicates and if an unsaturated expression has an argument-place that is fitting for one-place first-level functional signs, then it is not fitting for quantifiers and so on.

The upshot of this discussion is that the operator required by Geach and Dummett in their solution of Frege's paradox does not exist. They require a single operator, whereas the best that can be accomplished in a Fregean language is a large number of similar operators. In a Fregean language, each of the operators represented by the form of words 'what"..." stands for' in (1) to (6) has to be different. In order for the Geach-Dummett solution to be successful, they would require a trans-categorial operator of the form 'what "..." stands for, but such operators are illegitimate on Fregean principles.<13>

<13> The word 'trans-categorial' simply means that what it qualifies can belong to several categories. A proposition in which a trans-categorial operator occurs is, therefore, similar to a typically ambiguous formula in the theory of types. But as I partition the expressions of a language into syntactic categories (and their referents into types) I prefer the phrase 'trans-categorial' to 'typically ambiguous'.

Another example of a trans-categorial operator is '... refers to ___'. This is trans-categorial because we would - in our uncritical moments - accept the following propositions to be true:

(7) 'Socrates' refers to Socrates,

(8) 'The concept horse' refers to the concept horse,

(9) 'ξ loves ζ' refers to ξ loves ζ,

(10) The universal quantifier refers to a function from concepts to truth-values,

(11) Conjunction refers to a truth-function.

And we would also accept the following as being true:

(12) Proper names refer to objects,

(13) Predicates refer to concepts,

(14) Relational signs refer to relations,

(15) Second-level quantifiers refer to functions from concepts to truth-values,

(16) Binary truth-functional connectives refer to two-place truth-functions.

The best that Dummett can do faced with the propositions (7) to (11) and (12) to (16) is to banish the alleged trans-categorial operator '... refers to ___' from our language and replace it with lots of signs for all the different reference relations that are necessary. He would require at least as many signs as there are different categories of expression in our language. This is because the argument-place on the right of '... refers to ___'

in (7) is fitting for a proper name, whereas that on the right in (8) is fitting for a predicate. And that on the right in (9) is fitting for a relational sign and so on.<14> And the best that Geach can do is to say that whenever we use a proposition involving the words 'refers to' we are talking gibberish, which yet might be useful in teaching people about Frege's philosophy.

This investigation of the solutions offered by Geach and Dummett to Frege's paradox shows that the views they are forced to adopt are far worse philosophically than the paradox they use them to solve. Neither of them can account for the terms that are indispensable for talking about language and ontology. Dummett's solution of Frege's paradox involves banishing at least the following expressions from our language: 'refers to', 'entity', 'function', 'incomplete expression' and 'what "..." stands for'. Geach's solution has the consequence that whenever we use any of these terms we are talking gibberish. That would mean, for example, that almost every sentence in Dummett's book Frege is meaningless. It seems to me that in the light of these considerations we should investigate alternative ways of looking at Frege's paradox. And what I propose to do is to take seriously Frege's view that 'the concept horse' and similar expressions are genuine singular terms which refer to objects and explore its consequences. Dummett has claimed that Frege

---

<14> And even then he would have difficulty with "propositions" like '"Socrates" does not refer to a concept' and '"the concept horse" does not refer to a relation'.

repudiated this position, but it seems to me that there is some evidence to suggest that he continued to think in these terms. I will first present this evidence - which is only suggestive and not conclusive - and then I will look in greater detail at the consequences of treating expressions like 'the concept horse' as genuine singular terms.

Frege's Definition of Numbers

In Grundlagen Frege at one point gives the following definition:

> the Number which belongs to the concept F is the extension of the concept "equal to the concept F". (Section 68.)

If we accept that the expression 'the concept F' is a predicate, then the expression 'the Number which belongs to ...' is a linguistic function which makes singular terms out of predicates and the expression '... is equal to ___' is a linguistic function which makes a proposition out of two predicates. So, the concept which is the referent of the expression 'equal to the concept F' is a second-level concept and Frege is defining the expresssion 'the Number which belongs to the concept F' as a singular term which refers to a class of first-level concepts.

Or is he? In order to justify this definition he has been considering the definition of the direction of a line x, say, as the extension of the concept parallel to line x.

To understand the expression 'the Number which belongs to ...' as a linguistic function which makes singular terms out of predicates makes the correspondence with the definition of the direction of a line inexact. Since, in the numerical case the equivalence relation involved is one which holds between first-level concepts, whereas for directions the equivalence relation involved, namely, the relation parallel to, is one which holds between objects.

The interpretation of Frege's definition that I just gave is, in fact, anachronistic, for it is based on the Geach-Dummett account of the expression 'the concept horse' and similar expressions. As he makes explicit in "On Concept and Object", Frege considered such expressions to be singular terms, therefore they should be so understood in Grundlagen. If we so interpret them there, then the correspondence between the definition of numbers and directions becomes exact. The expression 'the Number which belongs to ...' is then a functional sign which makes singular terms out of singular terms and the equivalence relation equal to is then just a relation between objects.

In his discussion of the hermeneutical principles to be used in giving a correct exegesis of Frege, Dummett says that

> what Frege wrote in work he did not submit for publication bears less weight than what occurs in his published writings; and, above all, what is said in Grundgesetze carries more weight than anything else. (The Interpretation of Frege's Philosophy, p.8.)

But as a result of a careful structural analysis of Grundgesetze he adds that 'Grundlagen ... is in a somewhat different position' (ibid., p.9), that is to say, from Frege's published writings other than Grundgesetze. Simply stated, this is because Dummett regards Grundlagen, in effect, as a prose section of Grundgesetze. He says that the progression of ideas in the corresponding passages of the two books is very similar 'apart from the fact that in Grundgesetze numbers are defined as classes of classes, rather than as classes of concepts as in Grundlagen' (ibid., p.10). But as I have shown above, in Grundlagen numbers are defined as classes of objects of a particular kind, namely, those objects that can be the referents of expressions of the form 'the concept F'. That Frege did not go over this material again in prose in Grundgesetze leads me to presume that he still regarded expressions of the form 'the concept F' as singular terms.

The correspondence between relevant sections of the two works suggests two possible candidates for the type of object that could be the referent of such an expression. It could either be the class of Fs or it could be the referent of the expression which is formed by applying the functional abstraction operator to the unsaturated expression '$\}$ is an F'. There is a real choice here, because the operator in the Begriffsschrift which is usually called the class abstraction operator can just as validly be thought of as a lambda abstraction operator. This is because in Grundgesetze Frege did not distinguish between singular terms and propositions. If the operator in question is applied to a

predicate, it makes sense to call it a class abstraction operator; but when it is applied to a functional sign, there is greater justification in calling it a functional abstraction operator.

As I said above, I do not regard this as conclusive proof that Frege still thought that expressions like 'the concept horse' were singular terms when he wrote Grundgesetze, but I do think that it is fairly strong. In any case, even if Frege did not think this, it is still possible that treating such expressions as singular terms has consequences that are preferable to those of the position that treats them as predicates. And I shall explore that possiblity in the next Section.

An Alternative Proposal

The solution that Dummett and Geach put forward in order to deal with Frege's paradox has - at first sight - a lot of plausibility, but upon investigation it leads to several unacceptable consequences. In the first place, the operator that they use, namely, 'what "..." stands for', cannot be accommodated in their accounts and yet it is essential to them. It cannot be accommodated because, in order to do the job required of it, it must belong to more than one syntactic category. It must, for example, make a proper name out of a proper name and also a predicate out of a predicate.

It thus violates Frege's requirement that no expression be assigned to more than one category. The alternative, in which there are as many operators of the form 'what "..." stands for' as there are syntactic categories in our language, does not have much to recommend it.

This is not the only unacceptable consequence of the Geach-Dummett solution. Dummett's account leads him to a position in which a large part of our vocabulary for talking about type and category differences is banished from the language. Alternatively, we are allowed to use this logically erroneous terminology, if we do so carefully. Neither of these positions are very attractive philosophically.

Geach's account, on the other hand, leads him to a position in which meaningless strings of words are given definite and precise uses in teaching people about Frege's philosophy. This - again - is not very attractive philosophically.

The inadequacies of both Geach's and Dummett's solution to Frege's paradox should encourage us to look for a better solution and in this Section I want to explore the possibility that a sensible theory can be formulated in which expressions like 'the concept horse' are genuine singular terms that really do refer to objects. The best place to start from in trying to construct such a theory is Frege's discussion of an attempted refutation of his doctrine that concepts are unsaturated.<15> He considers the

---

<15> "On Concept and Object", pp.204-205. The discussion is actually carried out by Frege in the realm of sense, but - by the principle of parallel interpretation - it quite clearly also applies in the realm of reference and on the linguistic level.

translation of the sentence 'the number 2 is a prime number' into 'the number 2 falls under the concept prime number'. The first sentence contains the predicate 'ξ is a prime number' which refers to an unsaturated concept, whereas the second contains the singular term 'the concept prime number' which refers to an object. So, it looks as if the troublesome notion of unsaturatedness has been removed. Frege points out, however, that it has not been removed. What has happened is that the second sentence contains the relational expression 'ξ falls under ζ' which is unsaturated and which refers to an unsaturated relation. He writes: 'It is thus easy for us to see that the difficulty arising from the "unsaturatedness" of one part of the thought can indeed be shifted but not avoided.' ("On Concept and Object", p.205.) This also applies to the linguistic and the ontological levels. The word 'shifted' is unfortunate, because the same expression cannot be both the unsaturated predicate 'ξ is a prime number' and the saturated singular term 'the concept prime number' (and similarly on the ontological level). What we are dealing with is a procedure which translates unsaturated predicates into their corresponding saturated singular terms.

The picture of language that Frege has here is that language can be divided into two parts. One part of it contains a rich multiplicity of unsaturated expressions. In this part of language there are predicates like 'ξ rides', relational signs like 'ξ loves ζ', functional signs of various levels, quantifiers of several levels, truth-functional connectives and so on.

Because of the variety of unsaturated expressions in this part of language, we can call it the rich part.

But there is another part of language. Here there are singular terms and little else. Corresponding to each variety of unsaturated expression in the rich part of language there is a singular term here. We might, therefore, call this the austere part of language. Clearly, the austere part cannot just consist of singular terms. There has to be at least one unsaturated expression. In fact, there is no need to have more than one unsaturated expression in the austere part of language and I shall assume that there is only one.<16> In order to avoid confusion, in this Chapter I will call the category of austere singular terms 'E'. In fact, it is best to consider E as being the category of all the complete expressions in the austere part of language. It, thus, contains propositions as well as singular terms like '2' and 'Jack' and also like 'the concept horse'. Singular terms like '2' and 'Jack' are, in fact, the only expressions common to the rich and austere parts of language. The single unsaturated expression in the austere part of language is of syntactic category E --> (E --> E) and it has a number of linguistic incarnations. In the example Frege considers, the rich proposition 'the number 2 is a prime number' is translated

---

<16> Although I shall often write this unsaturated expression as 'ξ falls under ζ', the expression I use is more general than Frege's operator. I have deliberately generalised it in the direction of the functional application operator of combinatory logic. (This will be explained more fully in Chapter 6.) This is justified because the two operators have the same meaning in all the cases that Frege discusses and he gives no indication of how his views should be generalised.

into the austere sentence 'the number 2 falls under the concept prime number' and the unsaturated expression from the austere part of language is represented as 'ξ falls under ζ'.<17> It can sometimes be more idiomatically expressed as 'ζ applies to ξ' and its most succinct representation is '$(\zeta\ \xi)$'.

One of the difficulties that results from allowing propositions like 'the number 2 is a prime number' to be translated into 'the number 2 falls under the concept prime number' and then letting the latter be analysed into the two singular terms 'the number 2' and 'the concept prime number' and the austere unsaturated expression is that it is possible to re-arrange the austere constituents in such a way that no proposition constructed from rich components corresponds to it. In order to illustrate this, it should be observed that as well as allowing the above translation, Frege would also have to allow the proposition 'Bucephalus is a horse' to be translated into 'Bucephalus falls under the concept horse'. And now it is possible to combine both the singular terms 'the concept prime number' and 'the concept horse' by means of the unsaturated

---

<17> Long argues that the decomposition of 'the number 2 falls under the concept prime number' into a relational sign 'ξ falls under ζ' and two singular terms is incorrect (see, for example, his articles "Universals" and "Formal Relations"). He says that it is much more plausible to decompose this sentence into the singular term 'the number 2' and the predicate 'ξ falls under the concept prime number' and to regard this predicate as just a long-winded version of 'ξ is a prime number'. Viewed as an account of what goes on in the rich part of language this position has much to recommend it, but it cannot be a correct account of the way in which the austere part of language functions. If there was no unsaturated expression there, then the austere singular terms could not combine with one another.

expression '$\}$ falls under ζ' to yield:

(17) The concept prime number falls under the concept horse.

But there is no way in which the corresponding unsaturated expressions '$\}$ is a horse' and '$\}$ is a prime number' can combine together because the argument-place of each of them is not fitting for the other.

Another example of this sort of austere proposition can be constructed from an example that Frege discusses. He considers the proposition 'the number 2 falls under the concept prime number' and says that a similar problem to that of 'the concept horse' can be constructed here, since the 'words "the relation of an object to the concept it falls under" designate not a relation but an object' ("On Concept and Object", p.205). We can, therefore, fill up both argument-places of the relational sign '$\}$ falls under ζ' by the singular term 'the relation of an object to the concept it falls under' to give us the proposition:

(18) The relation of an object to the concept it falls under falls under the relation of an object to the concept it falls under.

There is a way, however, of isolating those combinations of austere symbols that do have propositions constructed out of rich expressions corresponding to them. In order to do this I need to introduce the notion of an austere expression's functional

character and also that of stratification.<18>

Every linguistic function in the rich part of language has a category, for example, '$\{ > \}$' is of category N --> (N --> P). The austere saturated expression 'the relation greater than', which is its translation, is not of this category. It is in fact a singular term, but we can assign to it what I propose to call a functional character which tells us what its powers of significant combination are. In this case the name of the functional character is written 'N ==> (N ==> P)' and its extension to other cases should be obvious. To obtain the name of a functional character from a category name simply replace every single-shafted arrow '-->' by a double-shafted one '==>'.<19> The relation between category names and names of functional characters generated by this translation induces a relation between categories and functional characters.
A functional character x' corresponds to a syntactic category x iff the name of x' has been obtained from that of x by replacing every single-shafted arrow in the name of x by a double-shafted one.<20>

<18> The notion of stratification was introduced by Quine in "New Foundations for Mathematical Logic", but my account derives from that of Curry and Feys in Combinatory Logic, pp.289ff.

<19> In order to simplify the discussion I ignore the fact that in Chapter 1 I allowed the sign '*' to be used in the construction of category names. In any case, such category names only play a small role in this thesis.

<20> I use the expression 'iff' as shorthand for 'if and only if'.

The notion of stratification can now be defined as follows:

(i) Every saturated austere expression without any logically relevant internal structure which has a rich counterpart is stratified and its functional character corresponds to the syntactic category of its rich counterpart.

(ii) An expression ⌜X falls under Y⌝ or ⌜Y applies to X⌝ or simply ⌜(Y X)⌝ is stratified and has the functional character w iff X is stratified and has the functional character v and Y is stratified and has the functional character v ==> w.

An austere expression has a logically relevant internal structure if it is made up from austere expressions which have functional characters. Thus, 'the concept horse', under this account, has no logically relevant internal structure. If an expression made up from austere constituents is stratified, then the rich components that correspond to those constituents can combine together to form a syntactically coherent expression in the rich part of language.

Under this definition the strings (17) and (18) - although syntactically coherent expressions of the austere part of language - are unstratified. Therefore, the rich components that correspond to their austere constituents cannot combine together. In other words, there are no rich propositions corresponding to (17) and (18).

Similarly, it is possible to assign valencies to the referents of all expressions of category E in the austere part of language.<21> In order to avoid confusion, I shall call these

---

<21> This notion of valency is different from the one that I used in the Section "Propositional Unity" of Chapter 1.

entities obs (the word comes from combinatory logic). Every entity that is the referent of some expression from the rich part of language has a type. The referent of the linguistic function '{ > }', for example, is of type J --> (J --> H). The ob which is the referent of the austere counterpart of this unsaturated expression - namely, the relation greater than - is not of this type, but we can assign to it what I propose to call its valency. This tells us what its powers of meaningful combination are. In this case the name of the valency is written 'J ==> (J ==> H)' and its extension should be obvious. To obtain the name of a valency from that of a type simply replace every single-shafted arrow by a double-shafted one. The relation between names of types and valency names generated by this translation induces a relation between types and valencies. A valency x' corresponds to a type x iff the name of x' has been obtained from that of x by replacing every single-shafted arrow in the name of x by a double-shafted one.

Analogous to the notion of linguistic stratification defined earlier it is possible to define a notion of ontological stratification:

(i) Every referent of an expression of category E which has no logically relevant internal structure and which has a rich counterpart is stratified and its valency corresponds to the type of the referent of its rich counterpart.

(ii) The referent of an expression of the form ⌜X falls under Y⌝ or ⌜X applies to Y⌝ or simply ⌜(Y X)⌝ is stratified and has valency w iff the referent of X is stratified and has valency v and the referent of Y is stratified and has valency v ==> w.

Expressions like 'the concept prime number falls under the concept horse', which are not stratified do not have stratified referents.

It is now possible to give a coherent account of such terms as 'entity' and 'function'. To be precise, it is possible to give an account of the predicates '$\hat{\xi}$ falls under the concept entity' and '$\hat{\xi}$ falls under the concept function'.<22> It is possible to give a coherent account of these although we can neither assign a functional character to 'the concept entity' nor to 'the concept function'. These expressions do not have any logically relevant internal structure, but - as shown earlier in this Chapter - they do not have rich counterparts. Furthermore, their referents do not have valencies.

<22> The first of these is constructed by filling the second argument-place of '$\hat{\xi}$ falls under ζ' with the complete expression 'the concept entity' and the second results from filling the same argument-place with 'the concept function'.

The satisfaction-conditions of the predicates 'ζ falls under the concept entity' and 'ζ falls under the concept function' are given in this way. ⌜X falls under the concept entity⌝ or ⌜X is an entity⌝ is true iff X is stratified. ⌜X falls under the concept function⌝ or ⌜X is a function⌝ is true iff X is stratified and the name of its valency (a) contains at least one double-shafted arrow '==>' and (b) contains no occurrence of the valency name 'H'.

There is no problem about incorporating the operators '... refers to ___' and 'what "..." stands for' into the austere part of language. '... refers to ___' is just a relation between complete austere expressions with a functional character and obs that have a valency. And 'what "..." stands for' is an operator which makes complete austere expressions out of complete austere expressions. The satisfaction-conditions of these operators are only slightly more complicated than those of the predicates 'ζ is an entity' and 'ζ is a function'.<23> In order to make the following account concise I shall write '(ξ ζ)' instead of either 'ξ applies to ζ' or 'ζ falls under ξ'. I also make use of what Quine calls the disquotation operator. The disquotation of an expression X or Δ(X) is simply the thing named by the expression X.<24>

<23> Needless to say, the operator '... refers to ___' is built up out of two tokens of the unsaturated expression 'ζ applies to ξ' and a suitable singular term, which I propose to call the saturated reference operator.

<24> Quine, "Truth and Disquotation", p.5.

In order to give the satisfaction-conditions of '... refers to ___', we first require every proposition of the form

(19) X refers to $\Delta(X)$,

to be true, where X is an austere saturated expression without any logically relevant internal structure which has a rich counterpart. Then, we require the following to hold:

(20) $\ulcorner(Y\ X)\urcorner$ refers to $\Delta(\ulcorner(Y\ X)\urcorner)$,

for all stratified Y and X such that $\ulcorner(Y\ X)\urcorner$ is also stratified. The disquotation of $\ulcorner(Y\ X)\urcorner$ is simply the disquotation of Y applied to the disquotation of X.

The result of inserting a stratified expression in the gap of 'what "..." stands for' is an expression which is everywhere substitutable salva congruitate and salva veritate with the original expression.

Having explained how the predicates '$\hat{\}$ is an entity' and '$\hat{\}$ is a function', the relational sign '... refers to ___' and the operator 'what "..." stands for' can be accommodated in the austere part of language it should not be too difficult to see how other similar expressions can be handled there.

Before concluding I just want to make it explicit that the austere part of language has two distinct classes of complete expressions in it. The first is the class of all those austere expressions which have analogues in the rich part of language.

Members of this class are expressions like 'the concept horse' and 'the addition function' and 'the number 2 falls under the concept prime number'. Every expression in this class has a functional character and its referent has a valency. The other class of complete expressions in the austere part of language consists of such expressions as 'the concept function', 'the concept entity' and the saturated reference operator. These do not have rich counterparts, they do not have functional characters associated with them and their referents do not have valencies assigned to them.<25>

Conclusion

Frege's notion of unsaturatedness is very attractive. And so is the related idea of a language whose expressions are partitioned into a multiplicity of discrete non-overlapping syntactic categories. Associated with each category is a distinct kind of unsaturatedness which unambiguously determines the powers of combination of those expressions. Frege's Begriffsschrift is a language built on these ideas. Although these ideas are very attractive, we should be clear about the limitations of a language constructed in accordance with them. It is impossible

---

<25> I proved earlier in this Chapter in discussing Dummett's views that 'refers to' and 'entity' have to be banished from the rich part of language, although I had not then introduced the 'rich'/'austere' terminology.

in such a language to express the category differences on which it is built and it is impossible to accommodate trans-categorial operators in it. An important question to consider is: 'To what extent is natural language a Begriffsschrift?' Dummett assumes that it is and proposes a thoroughgoing revision of our ontological vocabulary. Geach assumes it is, but rather than outlawing all category differentiating "propositions" and all "propositions" involving trans-categorial operators, he retains them as meaningless strings of symbols. What I have shown is that by building a theory in which phrases like 'the concept horse' are genuine singular terms it is possible to accommodate these kinds of proposition as being meaningful. Such an advantage outweighs the initial unnaturalness of the theory. The resulting language still needs an unsaturated expression, but a single one will suffice. Because there is just one unsaturated expression, we can discuss it explicitly when necessary. Such a language does not violate any of Frege's principles. Its syntactic simplicity is achieved by not including the multiplicity of unsaturated expressions to be found in the Begriffsschrift. This language has much in common with the language of combinatory logic and some further advantages of that will be mentioned in Chapter 6. My own contention is that natural language is closer in syntax to such a combinatory language than to Frege's Begriffsschrift.

Chapter 4: Types of Analysis

Introduction

In Frege Dummett distinguishes between two types of non-grammatical analysis of propositions. I will call these the semantic and the logical. Semantic analysis is unique. In this form of analysis, a sentence is analysed into the simple linguistic expressions out of which it is constituted. These simple constituents are to be understood as saturated expressions. Logical analysis, by contrast, is not unique. In this form of analysis one and the same proposition can be seen as containing many different unsaturated expressions or complex components. Dummett uses the fact that there are two distinct types of analysis that can be applied to any sentence - one into simple constituents and the other into complex components - to resolve the tension between the obvious fact that we understand sentences we have never heard or seen before and Frege's insight that it is only in the context of a proposition that a word has

meaning.<1> Dummett uses the unique analysis of a sentence into its simple constituents in order to account for our ability to recognise the senses of new sentences. And he uses the many decompositions of a sentence into different collections of complex components in order to account for the validity of the inferences in which that sentence can figure and also in order to explain the contribution that the sense of an unsaturated expression makes to the senses of the sentences in which it can occur.

Corresponding to each of these two types of analysis there is an inverse process of synthesis. Dummett claims that his account of logical synthesis solves a number of combination problems. He says that it allows us to dispense with the idea that binary truth-functional connectives unite expressions smaller than sentences (Frege, p.15). And Dummett's distinction between two types of non-grammatical analysis solves some of the other combination problems, such as the problem of justifying the combination of a quantifier with a relational sign and that of justifying the combination of negation with a predicate.

The purpose of this Chapter is two-fold. In the first place, I show that what really justifies the combination of binary truth-functional connectives with predicates in Dummett's notion of logical synthesis is the presence of combinatory ideas there. I do not claim that Dummett consciously took over those ideas from combinatory logic. What I do claim is that they are ideas

---

<1> Made in the Introduction to Grundlagen and also in Sections 60, 62 and 106.

that are characteristic of combinatory logic. In the second place, I show that the arguments that Dummett uses in order to distinguish between logical and semantic analysis are bad. I also show that Frege's account of unsaturated expressions is different from both Dummett's account of simple constituents and also his account of complex components. On Dummett's account, simple constituents are complete and are used in the creation of sentences, whereas complex components are incomplete and have to be seen as derived from sentences.<2> But if you understand unsaturated expressions as constructive linguistic functions, then it is possible to see them both as being used in the creation of other sentences and also as being incomplete. His justifications of the combination of a quantifier with a relational sign and that of negation with a predicate, thus, fall apart.

The Orders of Recognition and Explanation

In the first chapter of Frege Dummett considers Quine's view that for Frege the unit of significance is not the word but the sentence. He says that this view is either truistic or nonsensical. It is truistic if interpreted to mean that we can

---

<2> I talk of the creation of sentences here, because on Dummett's account complex components are also used in the construction of sentences, but only from other sentences.

only perform linguistic acts by means of complete sentences.<3> It is nonsensical if interpreted to mean that the individual words of a sentence have no meaning. Dummett then adds that 'Frege's achievement was to give an account which acknowledged and explained' the unique role which sentences play in language (Frege, p.7).<4>

Supposedly to expound Frege's doctrine on the matter, Dummett distinguishes between an order of recognition of senses and an order of explanation of senses. The former is necessary in order to account for the obvious and essential fact that we can understand new sentences which we have never heard or seen before, so long as they are composed of constituents whose senses we know, put together in ways with which we are familiar (Frege, p.3). We recognise the sense of a sentence by recognising the senses of the constituents composing it and the way in which they are put together. This can be succinctly stated by saying that in the order of recognition, the senses of the constituents of sentences are primary and the senses of sentences secondary.

In the order of explanation the order is reversed. We can explain the sense of a sentence without reference to the senses of the components that make it up by means of the notion of the truth-conditions of that sentence: To grasp the sense of a

---

<3> This obviously has to be qualified to account for those cases where linguistic acts are apparently performed by the utterances of phrases that are smaller than sentences.

<4> Dummett continues his discussion in terms of the expression 'word', but in order to make the connection with his other views clear I make use of the expressions 'component' and 'constituent' in my exposition.

sentence is, in general, to know the conditions under which it is true and the conditions under which it is false (Frege, p.5).<5> We can then explain the senses of a sentence's components in terms of the contribution they make to the truth-conditions of that sentence. This can be succinctly stated by saying that in the order of explanation the senses of sentences are primary and the senses of their components secondary. In order to give a general account of what it is for a component of a sentence to have a sense we have to do so in terms of the senses of the sentences in which that component can occur (Frege, pp.4-5).

Atomic Sentences

According to Dummett one of the greatest steps forward that Frege made in the theory of meaning was his 'distinction between the two stages of sentence-formation - the formation of atomic sentences and their transformation into complex sentences' (Frege, p.195). In this Section I will look at the first stage of Dummett's scheme of sentence formation and in the next Section I will look at the second stage.

Atomic 'sentences are formed out of basic constituents none of which are, or have been formed from, sentences' (Frege, p.21) and

---

<5> Needless to say, in Dummett's scheme of things we do not associate a sentence with its truth-conditions without reference to that sentence's structure, but the only structure we need to see the sentence as having is that revealed by semantic analysis.

Dummett's initial list of basic constituents consists of logically simple proper names, functional signs, predicates and relational signs (Frege, p.23). He actually says that Frege considered atomic sentences as constructed out of expressions of these four kinds, but there are significant differences between Dummett's account and Frege's. To begin with there are no logically simple proper names in the Begriffsschrift of the Grundgesetze.<6> That is is a fairly minor point. More importantly, all the primitive signs of that Begriffsschrift are unsaturated expressions. Frege does refer to them as simple names, but by that he does not mean that they are saturated. He calls them simple because they are his primitive symbols, that is to say, everything else in the Begriffsschrift is defined in terms of them. There is nothing in Frege's writings to support Dummett's claim that he countenanced complete expressions other than singular terms (and propositions, which he also thought of as singular terms).

The construction of atomic sentences out of basic constituents is - for Dummett - a rule-governed construction (Frege, pp.32-33). For example, the atomic sentence 'Theaetetus flies' is formed out of the proper name 'Theaetetus' and the simple predicate 'flies'. The rule governing this construction is that an atomic sentence is formed when a proper name is prefixed to a simple one-place predicate. In other words, given a proper name X and a simple predicate Y, the result of applying this rule to X

<6> The context of Dummett's remarks in this part of Frege make it clear that he has that formalised language in mind in his discussion.

and Y is the atomic sentence $\ulcorner$X Y$\urcorner$.<7> As another example, we can consider the atomic sentence 'Peter envies John', which has been constructed from the logically simple proper names 'Peter' and 'John' and the simple relational sign 'envies'. The rule governing the construction here is that given any proper names X and Y and an infix relational sign R, then $\ulcorner$X R Y$\urcorner$ is an atomic sentence.<8>

The inverse of this construction process Dummett calls analysis in Chapter 15 of The Interpretation of Frege's Philosophy, but I prefer to call it semantic analysis, in order to bring out its connection with the order of recognition of senses. In Dummett's philosophy it is by seeing how an atomic sentence has been constructed from simple constituents that enables us to grasp its sense.

Dummett's account of the formation of atomic sentences is clearly derived from one clause of Frege's first method of making names out of names.<9> The main difference is that Frege's

---

<7> Dummett has difficulties in extending this rule to cover singular terms in general, because in order to define that general notion we have to make use of a number of unsaturated expressions. I shall not explore these difficulties here. See the Chapter "Alternative Analyses" in Dummett's The Interpretation of Frege's Philosophy.

<8> I have deliberately expressed these rules in a form which brings out their similarity to my account of linguistic functions of category N --> P and N --> (N --> P). Although there are similarities, the types of rule involved are different.

<9> Grundgesetze, Section 30, clause (A1), restricted to those cases in which the arguments are not propositions and the values are.

construction starts from incomplete expressions, whereas Dummett's starts from complete ones.

Step-By-Step Construction

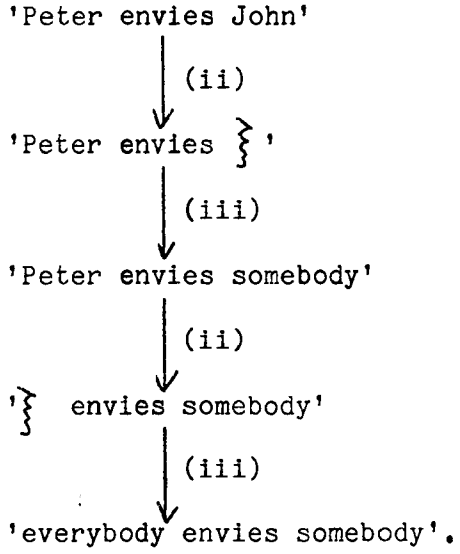
In the second chapter of Frege Dummett expounds the notion of the step-by-step construction of a sentence from a given stock of atomic sentences. He says that three operations are involved. These are: (i) The operation of constructing a sentence out of one or more sentences by means of the sentential connectives; (ii) the operation of constructing a complex predicate out of a sentence by means of replacing one or more occurrences of a single proper name in that sentence with the Greek letter xi; and (iii) the operation of constructing a sentence out of a complex one-place predicate by means of replacing the Greek letter xi with a sign of generality. (Ibid., pp.16 and 23.) Concerning step (ii) Dummett says that 'the general notion of a one-place predicate' cannot be thought of 'as synthesized from its components, but as formed by omission of a proper name from a sentence' (ibid., pp.22-23). Clearly, step (ii) is derived from Frege's second way of making names out of names and steps (i) and (iii) from his first way.<10> Thus, for example, from the

---

<10> See Grundgesetze, Section 30. Step (i) corresponds to clause (A1), restricted to those cases when the arguments and values are propositions, and step (iii) corresponds to clause (A2), restricted to the case when the second-level functional sign is a quantifier.

sentence 'Theaetetus flies' we can, by operation (ii), construct the complex predicate 'ξ flies', and from this, by operation (iii), we can construct the sentence 'everything flies'.

Although Dummett only explicitly mentions these three types of sentence-forming operations, when he comes to deal with the constructional history of more complicated sentences and sentences containing more than one sign of generality he makes use of a further two operations and he also substantially qualifies step (ii). These refinements can best be introduced by considering one of Dummett's examples, namely, the sentence 'everybody envies somebody'. This is constructed from the atomic sentence 'Peter envies John', which has been constructed from the simple components - 'Peter', 'envies' and 'John' - according to an appropriate rule. Its constructional history is:



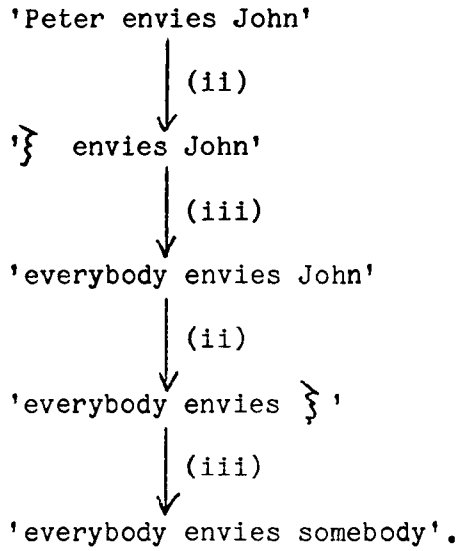
Here the labelled arrows indicate the application of one the steps in the account of the step-by-step construction of a sentence and the label indicates which particular step is being applied.

Quite clearly the choice of the proper names 'Peter' and 'John' is quite arbitrary. Any two names would serve just as well. In fact, it is even possible to start from the atomic sentence 'Peter envies Peter', since in the formulation of step (ii) it is not essential to remove all occurrences of the same proper name. Here, again, the choice of the name 'Peter' is arbitrary and any other proper name would do just as well.

In connection with sentences containing more than one sign of generality Dummett says that we employ the ad hoc convention that

> the order of construction corresponds to the inverse order of occurrence of the signs of generality in the sentence: when 'everybody' precedes 'somebody', it is taken as having been introduced later in the step-by-step construction, and conversely. (Frege, p.12.)

Therefore, the original version of the idea of a step-by-step construction must be modified to outlaw the following constructional history:



This is illegal because it violates the ad hoc convention that Dummett mentions: 'everybody' precedes 'somebody' in the final sentence, but it was introduced before it in the constructional history. In order to take account of this ad hoc convention it is necessary to modify the operation in step (ii) by adding the qualification that no occurrence of any proper name to the right of the rightmost occurrence of the proper name that is being replaced by the Greek letter xi in this application of step (ii) can be replaced by the Greek letter xi in further applications of step (ii). Only occurrences of proper names to the left of the rightmost occurrence replaced in an application of step (ii) can be replaced in further applications of step (ii).

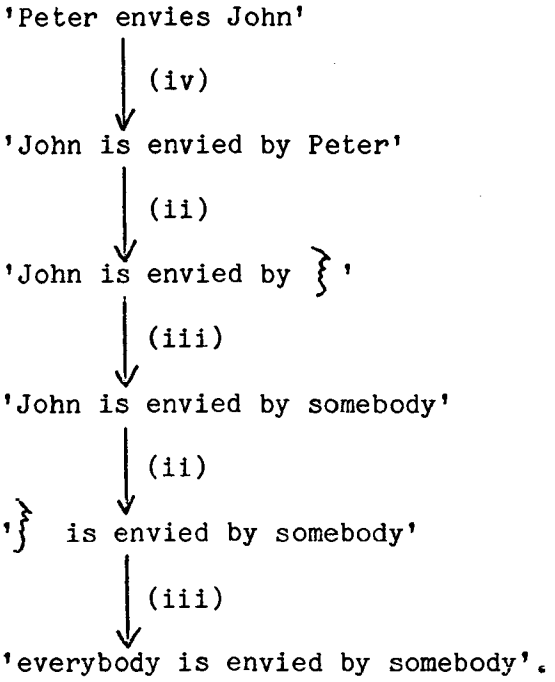
The sentence 'everybody envies somebody' corresponds to the pseudo-English sentence '(Ax) (Ey) x envies y', where the universe of discourse is taken to be the set of human beings. The illegal constructional history above can be seen as an attempt to obtain an English sentence which corresponds to the

hybrid '(Ey) (Ax) x envies y'. In order to obtain such an English sentence we need to observe that under the ad hoc convention mentioned by Dummett we have to start from a sentence logically equivalent to 'Peter envies John' but in which the names involved occur in reverse order. This is easy to achieve in this case by making use of the passive construction. The required sentence is 'John is envied by Peter'.

For sentences in English containing more than two occurrences of a proper name the situation is more complicated. Dummett says that in natural language there is a great deal of redundancy in the class of sentences that do not contain any sign of generality. By this he means that there are lots of equivalent sentences in this class. He says that it must be true that

> for any sentence of natural language containing any number of distinct proper names, there be an equivalent sentence containing the same names in any permutation of the original order of occurrence. (Frege, p.13.)

I will refer to the constructional step described here as step (iv). Thus, the constructional history of the English sentence corresponding to '(Ey) (Ax) x envies y' is as follows:



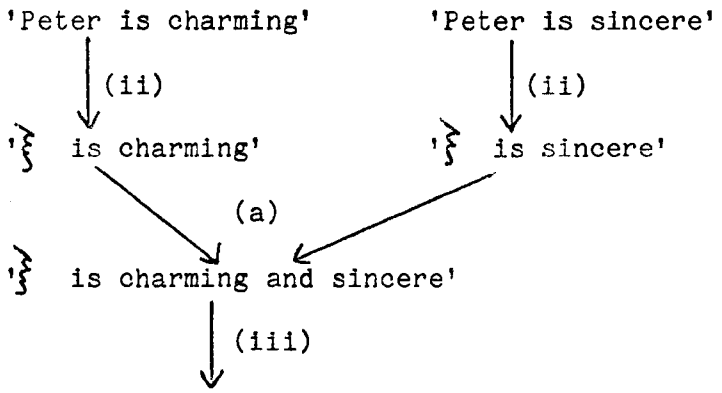
Dummett also says that it is true of natural language that:

> for any sentence containing two or more occurrences of some one proper name, there must be an equivalent sentence containing only one occurrence of that name (ibid.).

I will refer to this constructional step as (v). The simplest example of the use of this transformation is that which makes use of the reflexive pronoun. The sentence 'Peter envies Peter' can be transformed into 'Peter envies himself' by means of step (v). Dummett says that the reflexive pronoun is an operator which turns a relational sign into a predicate.

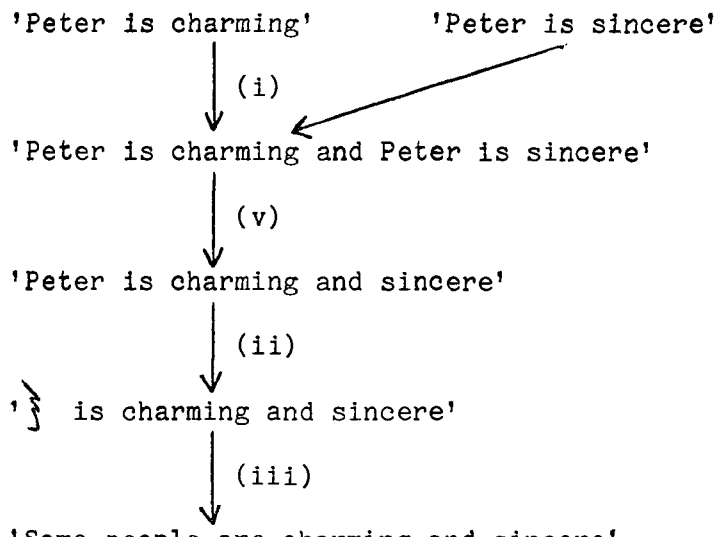
Dummett says of the notion of the step-by-step construction of propositions that it does away with the need of introducing operators analogous to the sentential operators, which combine logically unified expressions "smaller than" sentences (Frege,

p.15).<11> The example that he gives will make this clearer. Consider the sentence 'some people are charming and sincere'. Prima facie, this might be thought to have been constructed as follows:



Here,(a) cannot be thought of as step (i) because the 'and' involved makes a predicate out of two predicates. Dummett sees the sentence 'some people are charming and sincere' as having the following constructional history:

<11> In other words, this notion solves one group of combination problems.



It is the step that I have called (v) that does away with the need to account for the combination of the two predicates 'ξ is charming' and 'ξ is sincere' by means of conjunction to form the complex predicate 'ξ is charming and sincere'.

We thus see that Dummett's idea of the step-by-step construction of a sentence makes use not only of the three steps he explicitly states, but also of the steps (iv) and (v).<12> Concerning these Dummett says:

> Whether natural language actually has this power is not wholly clear: certainly it often cannot be accomplished without considerable clumsiness. (Frege, p.14.)

<12> It might be thought that step (iv) should not be seen as part of the step-by-step construction of non-atomic sentences, but rather as part of the construction procedure for making atomic sentences out of simple components. Just as 'Peter envies John' has been made out of 'Peter', 'envies' and 'John', it might be thought that 'John is envied by Peter' has been made out of 'John', 'is envied by' and 'Peter'. I decided against this interpretation because step (v) has to be part of the step-by-step construction of non-atomic sentences and Dummett firmly links (iv) and (v) as being the same sort of rule of formation (Frege, p.14).

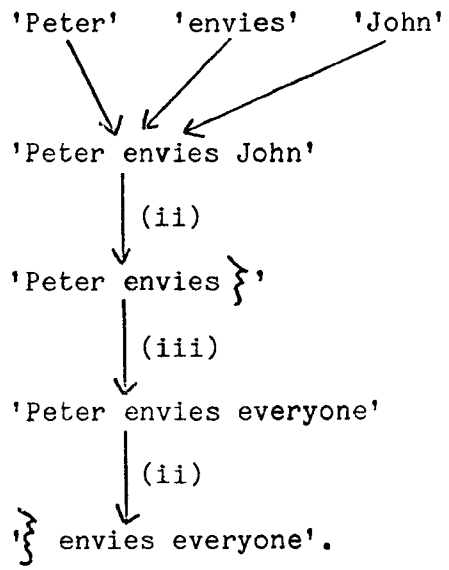
In Chapter 6 I will describe a formal system, namely, that of combinatory logic, which can be proved to contain operations analogous to those specified by Dummett for natural language. Steps (iv) and (v) introduce combinatory ideas into Dummett's notion of sentence-formation. The necessity of this is not really surprising, since Dummett is trying to account for phenomena that in a language with variable-binding operators would be handled by the apparatus of quantifiers and variables. Combinatory logic was devised to do these things without this apparatus.

Some Examples

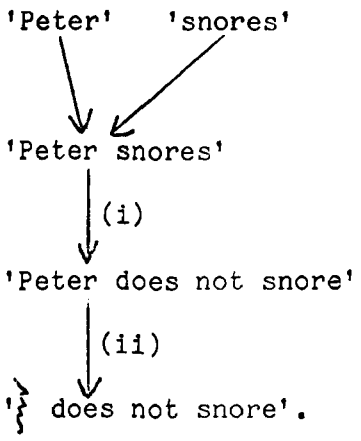
Dummett's account of sentence-formation presented in the previous two Sections is an attempt to apply Frege's views about the construction of the Begriffsschrift's propositions to natural language. Dummett tries to do this for a natural language which has <u>not</u> been augmented with the apparatus of quantifiers and variables. He thus comes across a number of combination problems. In order to solve some of these he is forced to make use of the combinatory transformations (iv) and (v), but others can be resolved without making use of these. In the previous Section I showed how step (v) was used to solve the problem of how the complex predicates '$\}$ is charming' and '$\}$ is sincere' combine with conjunction to form '$\}$ is charming and sincere'. In

this Section I will show how some other combination problems can be solved without using the combinatory transformations (iv) and (v). Their solutions just depend on the fact that Dummett's sentence-construction procedure has two stages. I will consider the problems of how to account for the combination of a quantifier with a simple relational sign and then the problem of how to account for the combination of a simple predicate with negation. Both these combinations of symbols can then be combined with a quantifier, say, to form a proposition.

Quantifier and Relational Sign



Predicate Negation



Are There Sense-Functions?

So far in this Chapter I have given an exposition of Dummett's two types of non-grammatical analysis and shown how he deals with a few combination problems. There is a single substantive reason why Dummett distinguishes between these two types of analysis and that is semantic. Because he thinks that the sense of an incomplete expression can only be explained in terms of the meaning of the propositions in which it figures, he needs the semantic synthesis of a proposition from its complete constituents in order to account for our ability to understand new sentences. In this Section I show that an account can be

given of the sense of an incomplete expression which does not make use of the specific senses of the propositions in which it can figure.

In Chapters 13 and 15 of The Interpretation of Frege's Philosophy Dummett criticises Geach's view that the senses of functional signs are themselves functions. In particular, he criticises the view that the sense of a (complex) predicate is a function whose arguments are the senses of singular terms and whose values are thoughts (in the Fregean sense, that is to say, the senses of propositions). Dummett explicitly says that the senses of functional signs are incomplete in a different way from how those functional signs themselves are incomplete (The Interpretation of Frege's Philosophy, p.270). This cannot be correct because it violates the principle of parallel interpretation, but it is also possible to show that Dummett's argument against Geach's position is faulty. Dummett's criticism is that

> before grasping the conception of functions whose values are thoughts, we must already know what thoughts are, and how to identify particular thoughts (ibid., p.267).

And in context it is clear that Dummett has in mind a particular notion of what a function is. For Dummett's criticism to hold a function must be understood as being posterior to its domain and range.<13> The nature of this dependence can be illustrated by

---

<13> This is clearly stated on p.268, where Dummett says that Geach's position entails the proposition that 'we must already be able to pick out a given thought before we can

the familiar notion of a function in set theory. Here a function is defined as being a particular subset of the Cartesian product of the set which is its domain and the set which is its range. It must also satisfy the property that if $(d, r) \in f$ and $(d, s) \in f$, then $r = s$. I am not saying that Dummett is committed to understanding functions set-theoretically. What I am saying is that he is committed to accepting a notion of a function in which the relation of dependence between the function itself and its domain and range is the same as in the set-theoretic case.

There are, however, ways of construing functions in which a different dependence relation obtains, namely, a relation in which the range of the function depends upon the function and its domain. This is best illustrated on the linguistic level. A predicate is a function whose arguments are singular terms and whose values are propositions, but the predicate <u>constructs</u> its values out of its arguments. This is not to be understood as any sort of temporal process. The function is to be understood as a rule. In order to grasp the rule 'ξ snores', for example, we do not have to know that 'Socrates snores' is a proposition. In fact, we do not have to know <u>any</u> proposition of this form. All that is necessary in order to grasp this rule is a general notion of what it is to be a proposition and a general notion of what it is to be a singular term.

It is now possible to transfer all this to the realm of sense.

---

conceive of any function that has that thought as value for some argument'.

In order to grasp the rule which is the sense of the predicate 'ξ snores' it is not necessary to have grasped the sense of the proposition 'Socrates snores'. In fact, it is not necessary to have grasped <u>any</u> thought of this form. All that is needed is the possession of the general idea of what it is to be a thought and the general idea of what it is to be the sense of a singular term. This can also be expressed by saying that all that is necessary is to know the <u>form</u> that the thought takes and the <u>form</u> that the sense of a singular term takes.<14> Then, given the sense of a particular singular term as <u>content</u> the rule which is the sense of 'ξ snores' fills out the form of the thought with that sense together with a sense obtained from the rule. So, rather than the sense of a functional sign being derived from that of a proposition it is the sense of the proposition which is derived from that of the functional sign (together with the sense of a singular term).

Given the sense of a predicate understood in this way it is possible to determine its referent.<15> In order to make the account of how this is to be done easier to follow I shall

---

<14> No metaphysic of forms is here assumed. Knowing the form that the sense of a singular term takes is simply knowing that what is to count as a sense here must be something which is capable of determining an object as the referent of the singular term. And knowing the form that the sense of a proposition takes is simply knowing that what is to count as a sense here is something the knowledge of which in conjunction with how the world is must be capable of determining the truth-value of the proposition.

<15> It is not the sense of a predicate by itself that determines its referent, but also how the world is. This should be understood whenever I talk about sense determining reference (and not just for predicates). It would be tedious to always mention this explicitly.

discuss a particular example. I shall consider the predicate 'ξ snores' and the proper name 'Socrates'. The sense of 'ξ snores' yields the thought that Socrates snores for the argument the sense of 'Socrates'. And knowing the sense of 'Socrates' enables us to determine its referent and knowing the sense of 'Socrates snores' enables us to determine its truth-value. Thus, knowing the sense of 'Socrates' and the sense of 'ξ snores' enables us to determine both the referent of 'Socrates' and the truth-value of 'Socrates snores'. Thus, we can determine the value of the ontological function ξ snores in the case in which the argument is Socrates. But in this account the name 'Socrates' is arbitrary, so it also shows how the value of the referent of 'ξ snores' can be determined for any argument. Hence, it shows how the referent of 'ξ snores' can be determined from its sense. And this account could be applied to any predicate.

In this Section I have shown that Dummett's argument against the position that the sense of a functional sign is incomplete in the same way that the functional sign itself is incomplete is faulty. I have also shown how the referent of a functional sign can be determined from such a sense (and how the world is). Dummett's account of the sense of a functional sign is all to do with how knowing it enables us to determine the referent of that sign. The present account is, therefore, superior because it explains that as well as retaining Frege's insight that the sense of of unsaturated expression is itself unsaturated.

In Dummett's scheme of things both simple and complex

predicates have a sense. The role that the sense of a simple predicate plays in his philosophy is that it accounts for our ability to understand new sentences. When we recognise the sense of a proposition we do so by recognising the senses of its simple constituents. The sense of a proposition is built out of the senses of its constituents.<16> The sense of a complex predicate, on the other hand, is needed - amongst other things - to determine that predicate's referent. As Dummett says, 'to grasp the sense of a predicate is to grasp a means for determining a function from objects to truth-values' (ibid., p.270). In this Section I have shown that the sense of a predicate, thought of as a sense-function, can do both the jobs that the sense of a simple predicate and the sense of its corresponding complex predicate do in Dummett's scheme of things. Hence, I have undermined Dummett's reason for distinguishing between two types of non-grammatical analysis.

Understanding and Inferring

The topic I consider in this Section is Dummett's account of the relationship between our understanding of a proposition and our knowledge of its inferential powers. He thinks that these two types of knowledge are very different and, in fact, independent

---

<16> See Dummett's discussion of the B theses in The Interpretation of Frege's Philosophy, pp.261ff.

of one another. For example, he writes (concerning the sentence 'Brutus killed Caesar'):

> The representation of the sentence as consisting of 'Brutus' and 'ξ killed Caesar' is quite irrelevant to any explanation of the way in which the sense of the atomic sentence is determined from that of its constituents. (Frege, p.28.)

Dummett does think that unsaturated expressions do have a semantic role to play, but that relates to the way in which we explain an expression's meaning.

In arguing that these two types of knowledge should be kept separate Dummett considers the logical analysis of the proposition

(1) If Brutus killed Caesar, then Brutus's wife hated Brutus'

into the proper name 'Brutus' and the complex predicate

(2) If ξ killed Caesar, then Brutus's wife hated ξ.

This is necessary, for example, in order to explain the validity of the inference of this sentence from

(3) (Ax) if x killed Caesar, then Brutus's wife hated x.

(I express this in quasi-English in order to preserve the similarity of structure between it and (2).) Dummett comments:

> But the possibility of giving such an "analysis" of the sentence has no bearing on the process by which we form the sentence, or on that by which we come to grasp its sense. We understand the sentence by reference to the process by which it was formed, namely as being put together out of two atomic sentences joined by the connective 'if', those atomic sentences having in turn been constructed by linking singular terms and relational expressions; the thought that one might recognize the complex predicate cited above as occurring within that sentence could be utterly remote from the mind of someone who had the firmest grasp upon its meaning. (Op.cit., p.29.)

The reason why Dummett separates our understanding of a proposition from our knowledge of its inferential powers is that - in his scheme of things - complex predicates are derived from atomic propositions. In other words, complex predicates are posterior to those atomic propositions of which they are features. In order to account for even a straightforward inference - such as 'someone sings' from 'Toyah sings' - we have to see both the premise and the conclusion as sharing a common feature. And - by definition - this feature, namely, 'ξ sings', cannot occur in the construction of any atomic sentence.

The separation that Dummett makes follows almost immediately from his non-constructive account of complex predicates. When he says that the recognition of the complex predicate (2) in (1) might be utterly remote from someone's mind who had the firmest grip on the meaning of (1), he is just saying that (2) has to be obtained form (1) (or a trivial variant) and is, thus, posterior to it. In order to explain the inference of (1) from (3) we have to see both (1) and (3) containing (2) as a component. As (2) is dependent on (1) it is possible to understand (1) without seeing it as containing (2).

As the separation Dummett makes depends on his non-constructive understanding of complex predicates, all that it is necessary to do in order to show that that separation is untenable is to give a constructive account of complex predicates and their senses. And I gave a constructive account of unsaturated predicates in Chapter 1 and of their senses in the previous Section.

Grammatical Analysis

In Frege as well as distinguishing between two types of non-grammatical analysis Dummett also accepts the validity of grammatical analysis. This comes out in passages like the following:

> As far as the sentence-structure of natural language is concerned, signs of generality such as 'someone' and 'anyone' behave exactly like proper names - they occupy the same positions in sentences and are governed by the same grammatical rules; it is only when the truth-conditions are considered that the difference appears.<17>

<17> Frege, p.20, emphasis added. The actual claim made in this passage is false. There are grammatical differences between signs of generality and proper names, namely, to do with the postpositive use of adjectives. For example, 'anyone intelligent can do it' and 'I want to try on something larger' are both grammatical, but neither 'Jack intelligent can do it' nor 'I want to try on a coat larger' are. (See Quirk, et.al., A Grammar of Contemporary English, pp.248-250, where these examples are taken from.)

Elsewhere in the same book (p.34) he considers the two propositions:

(4) Was what Henry killed a man?

(5) The man wearing my coat tore up the letter.

And he makes use of some grammatical rules which can be expressed as rewrite rules as follows:

(6) Sentence --> Subject + copula + complement,

(7) Sentence --> Subject + transitive verb + object.

He adds that the grammar would also contain rewrite rules beginning: Subject --> ... . He uses this notion of grammatical analysis to explain why 'killed a man' in (4) and 'my coat tore' in (5) are not wholes having unified sense. These expressions are not constituent phrases of those sentences (though they might be constituent phrases of some other sentences). He goes on to point out some of the inadequacies of the grammatical analysis, but nowhere does he suggest that it is radically misconceived.

Whereas I think that Dummett is wrong to distinguish between two types of non-grammatical analysis, I agree with him that we should draw a distinction between grammatical and non-grammatical analysis. I say more about how this distinction should be drawn in Section "The Foundations of Categorial Grammar" in Chaper 5 and in the Section "Outline of a Combinatory Logic" in Chapter 6.

Conclusion

It will be profitable at this stage to spell out what has been achieved by this discussion of Dummett's attempt to apply Frege's ideas on sentence-formation to natural language. As I have shown - apart from minor differences - Dummett makes two significant additions to Frege's formation rules. The first is that he splits the sentence-forming process more rigidly into two stages than did Frege. Dummett's first stage involves the formation of atomic sentences from complete constituents and the second stage involves the formation of non-atomic sentences and incomplete components from atomic sentences, by means of the procedure of step-by-step construction. The second is that in the step-by-step construction process he introduces some combinatory operations, which I have called steps (iv) and (v). These are clearly a different sort of transformation from that involved in steps (i) to (iii) and they have no analogues in Frege's formation rules.

My criticism of Dummett has focussed on the first of these additions. In particular, I have attacked his notion of a complex predicate as something essentially non-constructive. He thinks of it in these terms because for him it is derived from a proposition rather than used in the construction of propositions. In the Section "Unsaturated Expressions and Patterns" in Chapter 2 I criticised this notion from the linguistic point of view, whereas in this Chapter I criticised Dummett's views about the senses of such predicates. By showing that it is possible to

understand unsaturated predicates and their senses in a constructive way, I have shown that there is no need to distinguish - as Dummett does - between two types of non-grammatical analysis. The types of synthesis involved in the two stages of sentence formation are essentially the same.

The additions that Dummett makes to Frege's ideas solve several combination problems. His rigid separation of logical analysis from semantic analysis solves the problems of how to account for the combination of a quantifier with a simple relational sign and also that of how to account for the combination of negation with a simple predicate. But as I have shown that the way in which he makes that separation is untenable and un-Fregean, it is still necessary to explore other solutions to those problems. I will discuss the solutions offered by Geach and Potts in the next Chapter.<18>

I have not criticised Dummett's use of combinatory ideas in solving the problem of how to justify the combination of two predicates with a binary truth-functional connective. In fact, I think that the best way in which to solve all combination problems is by using combinatory logic wholeheartedly and this I will do in Chapter 6.

<18> The problems that Geach and Potts actually discuss are slightly different, since they do not accept the existence of simple relational signs and simple predicates.

Chapter 5: Combination Problems in Categorial Grammar

Introduction

In this Chapter I look at the attempt of Geach and Potts to extend Frege's ideas about sentence-formation to natural language. In doing this they have constructed a version of categorial grammar. A substantial amount of their work revolves around a number of combination problems. Such problems occur when a collocation of natural language expressions is intuitively thought to be syntactically coherent, but it cannot be shown to be so by the rules of that grammar. The purpose of this Chapter is to show why the way in which they choose to solve these problems - by adding extra grammatical rules - fails.

Although in this Chapter I talk about expressions belonging to lots of different syntactic categories, I do not make much use of Frege's auxiliary notation. In this respect I follow the example of Geach. Although he advocates Frege's views on incompleteness (in such articles as "Frege"), when he is dealing with issues relating to categorial grammar (as in "A Program for Syntax" and

"Should Traditional Grammar be Ended or Mended?") he leaves out the auxiliary notation most of the time. The justification of this is pragmatic. It is very cumbersome to always include all of the auxiliary notation explicitly, especially when dealing with expressions of levels higher than the second with several arguments (as I do in this Chapter). Such an attitude may appear objectionable from a theoretical point of view, but it works well in practice.

A Succinct Notation for Category Names

In Chapter 1 I introduced the standard mathematical notation for the type of a function in order to show which syntactic category a given linguistic function belongs to. Thus,

'ζ + 3': N --> P,

means that the linguistic function 'ζ + 3' takes arguments of category N and returns values of category P. In other words, it makes propositions out of singular terms. I used this notation there in order to stress the importance of the mathematical notion of a function on Frege's philosophy of language. In theory I could continue to use this notation for category names, but in practice it becomes very cumbersome for the names of higher-level categories of incomplete expressions which take

several arguments. For example, in this Chapter I have occasion to discuss what Potts calls a pro-verb. This is of the syntactic category:

((N --> P) --> ((N --> P) --> P)) --> ((N --> P) --> P).

It would take up a great deal of space to use this notation, so I have decided to use Potts's more succinct notation. In this the category of the pro-verb is represented as:

11P1PN11P1PN1PN.

The only change that I have made to Potts's notation is that I do not use the subscripts and superscripts that he uses.<1>

The set CAT of category names is the smallest set satisfying the following conditions:

(i) An element of BAS is a basic category name.

(ii) If I is a numeral which is the name of the number i such that $i > 0$ and $X_1, \ldots, X_{i+1}$ are category names, then so is $\ulcorner IX_1 \ldots X_{i+1} \urcorner$.

Here the set BAS is a non-empty set of category names. I will use capital letters for the names of basic categories. For example, if BAS = {'E'}, then the following are examples of category names: 'E', '1EE', '2EEE', '11EE1EE', '2EE1EE' and '31EE1EE1EE1EE'.

<1> "Fregean Grammar", pp.10-11.

It is straightforward to translate the notation used previously into this notation and vice versa. If the set of basic category names used in both cases is the same, then the two notations are, in fact, isomorphic and this can be shown by displaying a structure preserving bijection from CAT to MAT and another from MAT to CAT. I will write the translation function from CAT to MAT as a postfix operator '#' and it is defined as follows:

(i) A basic name is translated to itself. That is to say, if X is a member of BAS, then X# = X.

(ii) The translation of $\ulcorner IX_1 \ldots X_{i+1} \urcorner$ is $\ulcorner X_2\# * \ldots * X_{i+1}\# \dashrightarrow X_1\# \urcorner$.

Here I is a numeral which names the number i.

The reverse mapping from MAT to CAT is written as a postfix operator '%' and is defined as follows:

(i) A basic category name is translated to itself. That is to say, if X is a member of BAS, then X% = X.

(ii) The translation of $\ulcorner X_2 * \ldots * X_{i+1} \dashrightarrow X_1 \urcorner$ is $\ulcorner IX_1\% \ldots X_{i+1}\% \urcorner$.

Here, again, I is a numeral which names the number i.<2>

On the whole I shall restrict myself to talking only about category names constructed from the basic category names 'P' and 'N' and the numeral '1'. Unless I explicitly say otherwise, it

---

<2> It is easy to prove that these translations are indeed bijections by mathematical induction on the number i.

should be assumed that I am using this restricted class of category names in what follows. I shall also refer to it as the standard class of category names.

It should be noted that 'N --> P' becomes '1PN' in Potts's notation and in general the order of occurrence of the names of the categories of the argument and value of the linguistic function in question are - in Potts's notation - the reverse of their order in the standard mathematical notation. This reversal may seem merely perverse at first, but it makes the formulation of the recursive rules used by Geach and Potts straightforward.

The Foundations of Categorial Grammar

The fundamental rule of combination of any system of categorial grammar is a generalisation of Frege's first way of making names out of names (discussed in Section 30 of Grundgesetze). Using the letter 'E' as the name of the category of all the complete expressions of the Begriffsschrift we can paraphrase what he there says in the following terms. An expression of category E is formed either by combining an expression of category 1EE with one of category E, or by combining one of category 1E1EE with one of category 1EE, or by combining one of category 1E1E1EE with one of category 1E1EE. He also says that an expression of category 1EE is formed by combining one of category 11EEE with one of category E. Bringing these four methods of combining expressions

together and generalising to include expressions of any syntactic category in the infinite hierarchy of categories we obtain the general rule that an expression of category x is obtained by combining an expression of category 1xy with one of category y. It is easy to see that Frege's four methods are instances of this general rule. For example, to obtain the second method we substitute '1EE' for 'y' and 'E' for 'x' in the general rule.

Each version of categorial grammar contains this general rule in one form or another and it is known by a variety of names. Ajdukiewicz, for example, talks of cancelling in this connection ("On Syntactical Coherence", p.643), because in his notation an application of this rule looks like the procedure of cancelling-out when we multiply fractions by whole numbers. Geach refers to it as the multiplying-out rule ("A Program for Syntax", p.484). Although I and others call it a rule, it is important to realise that it is a rule <u>schema</u> which has an infinite number of rules as its instances. One for each pair of substitutions of category names for 'x' and 'y'.

I now introduce the notation and terminology that will be used throughout this Chapter in discussing linguistic structures.<3>

A <u>list</u> of categories is either the empty list or it is formed by appending a category to a list of categories. The empty list is represented by '()'. A list containing a single category is represented by writing the name of that category enclosed in

---

<3> There is no standard terminology for the concepts used in categorial grammar. I have selected terms from various authors and made up some of my own. My treatment is most closely related to Lambek's in "On the Calculus of Syntactic Types".

parentheses and a list with more than one component is represented by writing the names of those categories between parentheses and separated by commas. Using the standard class of category names, the following are examples of lists of categories: (P), (N, 1PN), (P, P, P, P, 1NN), and (11PN1PN, 1PN, N). Lists of other sorts of entity are to be understood analogously. It should be noted that lists are not sets. Lists are essentially ordered collections, so we can talk, for example, of the first and second component of a list with more than two members.

A pair of things is a list consisting of just two things and a structure of categories is either a category or a pair of categories. I shall usually just talk of a structure rather than a structure of categories. Examples of structures are: (P, (P, 1NN)), ((N, N), (P, P)) and (1NN, (1PN, N)).

Contraction is a relation between structures and it is represented by the arrow '-->'. The letters 'S', 'T', 'U' and 'V' stand for arbitrary structures. 'S --> T' can be read as 'S contracts to T'.

I shall actually consider a number of contraction relations in this Chapter, but they all have the following structural properties:

$$(S1)\ \frac{S \dashrightarrow T \qquad T \dashrightarrow U}{S \dashrightarrow U}$$

$$(S2)\ \frac{S \dashrightarrow T}{(S, U) \dashrightarrow (T, U)}$$

$$(S3)\ \frac{S \dashrightarrow T}{(U, S) \dashrightarrow (U, T)}$$

(S1) states that contraction is transitive. (S2) and (S3) say that it is right and left monotonic, respectively. It should be noted that contraction - as defined here - is not reflexive. The horizontal line in these structural rules is to be read as 'therefore', since they are rules of inference. I shall also write:

$$\frac{S \dashrightarrow T}{U \dashrightarrow V}$$

as

$$S \dashrightarrow T \ \vdash\ U \dashrightarrow V.$$

As well as having these structural properties, all the contractions I consider obey the multiplying-out rule:<4>

(R1) (1xy, y) --> x.

Because we have the rules (S2) and (S3) it is possible to apply (R1) in the context of a larger structure, for example, as follows:

(S, (1xy, y)) --> (S, x).

In such cases, I shall simply talk of applying (R1), rather than always explicitly mentioning that it is being done in the context of a larger structure.

A structure which cannot be contracted to another structure is said to be in normal form. P, (1PN, 1PN) and (N, (N, P)) are all examples of structures in normal form; whereas the following are not in normal form: (1PN, N), ((11PPP, P), P) and (P, (1NN, N)).

This terminology and notation should become clearer by considering an extended example. I shall look at one of Ajdukiewicz's examples, which (in Geach's translation) is:<5>

(1) Lilac smells very powerfully and roses bloom.

<4> It should be noted that this use of the word 'rule' is different from its use in 'rule of inference'.

<5> "On Syntactical Coherence", p.640.

The general problem of deciding or finding out which category an expression belongs to is not relevant to my concerns here. For my purposes it is sufficient to leave the categorisation of most expressions at an intuitive level. The problems I raise later are independent of such issues. So, I shall assume the following categorisations:

'lilac, 'roses': N,
'smells', 'bloom': 1PN,
'very': 111PN1PN11PN1PN,
'powerfully': 11PN1PN,
'and': 11PPP.

(The notation '"lilac", "roses": N' means that both 'lilac' and 'roses' are of category N. And it is to be understood similarly if more than two expressions occur on the left separated by commas.)

Given this categorisation and Ajdukiewicz's example it is possible to form the following list of categories:

(2) (N, 1PN, 111PN1PN11PN1PN, 11PN1PN, 11PPP, N, 1PN).

This list is obtained by looking - to begin with - at the last or rightmost word in (1) and then moving through the sentence expression by expression. We begin with the empty list and as each expression of known category is encountered in the process just mentioned we append its category to the front of our growing

list of categories.

Before we can apply the multiplying-out rule (R1) to this we have to transform it into a structure. Clearly, there are many ways in which this could be done and the multiplying-out rule will not apply to all the possible structures corresponding to the list (2). In this Chapter, when I say that a list of categories cannot be re-arranged to a structure which contracts to a single category, this can always be proved by exhaustively enumerating all the possible structures and showing that none of them contracts to a single category. In order to prove that a list does contract to a single category we try to prove that it does not. If it really does contract, then we will not be able to prove otherwise. In the case of Ajdukiewicz's example, it is possible to re-arrange (2) into a structure that contracts to a single category and this structure is:

((11PPP, (((111PN1PN11PN1PN, 11PN1PN), 1PN), N)), (1PN, N)).

We can now apply the multiplying-out rule (R1) to this structure. This results in the new structure:

(3) ((11PPP, ((11PN1PN, 1PN), N)), (1PN, N)),

Repeating this process we obtain the following sequence of structures:

(4) ((11PPP, (1PN, N)), (1PN, N)),

(5) ((11PPP, P), (1PN, N)),

(6) ((11PPP, P), P),

(7) P.

I need to introduce two further technical terms before I can begin discussing combination problems. In his version of categorial grammar Potts's distinguishes between the semantic structure of a proposition and its phonetic structure.<6> He justifies the distinction by observing that different sentences can have the same meaning and also that a single sentence can have several meanings (because it is ambiguous). An example of the first possibility is given by the pair of sentences:

(8) Rommel was defeated by Montgomery at El Alamein.

(9) At El Alamein, Montgomery defeated Rommel.

The second possibility can be exemplified by Chomsky's sentence:

(10) Flying planes can be dangerous.

Hence, according to Potts every meaningful sentence has at least two structures, which - as I have already mentioned - he calls the phonetic and the semantic. A proposition's phonetic

---

<6> In order to avoid confusion with what I have called a structure of categories, after this Section I will never use the word 'structure' by itself when I mean either 'phonetic structure' or 'semantic structure'.

structure is that in virtue of which it qualifies as a sentence of a particular language; and its semantic structure is that in virtue of which it has a meaning.<7> The sentences (8) and (9) have different phonetic structures, but the same semantic structure, whereas (10) has two different semantic structures associated with it. As further support for the distinction, he observes that:

> An expression may, indeed, have a semantic structure but no phonetic structure (a foreigner says something which is ungrammatical but we nevertheless understand what he means); equally, it may have a phonetic but no semantic structure (mastery of the grammar of a language is no guarantee that every sentence one formulates has a meaning). ("Case-Grammar as Componential Analysis", pp.400-401.)

The term 'phonetic' is used by Potts because languages are, in the first instance, spoken; but it must here be understood to include written equivalents and also other means of encoding sentences. The phonetic structure of sentences is what has, on the whole, been studied by traditional grammar and syntax. Potts stresses that the differences between phonetic and semantic structure should not be confused with the difference between surface and deep structure in transformational grammar:

> in particular, there is nothing "deep" about semantic structures and semantic structures are not related to phonetic ones by transformation rules or by any kind of computation ("Case-Grammar as Componential Analysis", p.400).

<7> "The Place of Structure in Communication", pp.99ff. "A General Theory of the Meaning of Anaphoric Pronouns", p.141.

He sees the relationship as one of projection.<8>

Having introduced the basic ideas of a categorial grammar, I now turn to a discussion of a number of combination problems and to the solutions offered to these by Geach and Potts. The combination problems I discuss only occur when we try to describe natural language by means of a categorial grammar. They do not occur, for example, in Frege's formalised language.

Predicate Negation

Negation is an operator which makes propositions out of propositions. In natural language it takes a variety of linguistic forms, but I shall simply express it here as 'not Π'. The Greek letter pi here stands in almost the same relation to a proposition as the letter xi does to a singular term. Under the conventions explained in Chapter 1 this should be understood as a linguistic function which prefixes the word 'not' to a proposition in order to form a proposition, but I shall use it to denote that linguistic operation which constructs the negation of a proposition whatever form that negation might take. Thus, the value of 'not Π' for the argument 'Socrates flies' is 'Socrates does not fly' rather than 'not Socrates flies'. Clearly, the exact formulation of such a linguistic function would be tricky for English, but such convolutions are logically irrelevant.

<8> "The Place of Structure in Communication", p.113.

The structure of categories for the proposition 'Henry does not snore' is:

(11) (1PP, (1PN, N)).

This contracts to P in two steps. There are times, however, when we have to regard negation as an operator on predicates. In order to justify the inference of 'Henry does not snore' from 'everyone does not snore' we have to see negation as forming a syntactically coherent expression with the predicate '$\}$ snores'. The structure of categories for such a construct is:

(12) ((1PP, 1PN), N).

These differences can also be shown by means of bracketing. The proposition corresponding to (12) can be written 'Henry (does not snore)', whereas for (11) we have to resort to the quasi-English 'not (Henry snores)'. The problem with structure (12) is that it does not contract to anything. It is in normal form. It is impossible to apply rule (R1) to it. One way round this problem is to introduce a negation operator on predicates. That is to say, an operator of category 11PN1PN. Then, we would have the structure for 'Henry (does not snore)':

(13) ((11PN1PN, 1PN), N),

which does contract to P. This however, does not help us in

explaining the validity of the inference of 'not (Henry snores)' (that is to say, the proposition 'Henry does not snore', where the negation involved is propositional negation) from 'everyone does not snore' (where the negation has to be understood as predicate negation), because we have given no account of how predicate negation relates to our ordinary propositional negation. It looks as if we have to introduce two new rules of inference to relate the two. One of these rules would justify the inference of 'not (Henry snores)' from 'Henry (does not snore)' and the other would justify the inference of 'Henry (does not snore)' from 'not (Henry snores)'. Using '-' for propositional negation and 'N' for predicate negation, these two rules could be expressed, respectively, as:

(14) (NF)a ⊢ -(Fa),

(15) -(Fa) ⊢ (NF)a.

Furthermore, we have to assume that these rules are tacitly applied in many cases. Consider, for example, the following propositions:

(16) Everyone does not snore,

(17) Henry does not snore,

(18) Either Henry snores or Maxine hallucinates,

(19) Maxine hallucinates.

The inference of (17) from (16) and the inference of (19) from

(17) and (18) are both straightforward; but in order to explain the validity of the inference of (17) from (16) we have to interpret the negation involved as being predicate negation, whereas in order to explain the validity of (19) from (17) and (18) the negation involved has to be understood as propositional negation. Thus - if we differentiate between propositional and predicate negation - we cannot unite these two arguments by means of the transitivity thema.<9> We have to first add the inference of 'Henry does not snore' (where the negation involved is propositional negation) from the proposition 'Henry does not snore' (where the negation involved is predicate negation).

As well as adding these two new rules of inference to our logic we would also have to add some sort of semantic rule to give the meaning of predicate negation. Something along the lines of: The predicate 'NF$\}$' is true of an arbitrary object iff 'F$\}$' is not true of it.

If we decide to distinguish between predicate and propositional negation in this way, we will soon find that we have to add a large number of different negations to our language and, at least, a corresponding number of new inference rules and new semantic rules. This is because we would have to introduce a version of negation that has arguments and values that are two-place predicates and another version whose arguments and values are three-place predicates and so on. In fact, we would have to

---

<9> The word 'thema' is taken over from the Stoic logicians, who used it to describe the process of turning one or more arguments into another in such a way that if all the original arguments were valid, then so was the conclusion. Geach discusses the method in Chapter 14 of Reason and Argument.

introduce at least as many versions of negation as there are different polyadicities of predicate in our language. This would lead to a very complicated logic.

Rather than distinguishing between propositional and predicate negation (and other sorts of negation) Geach considers the negation involved in a proposition like 'everyone does not snore' to be propositional negation and in order to justify the syntactic coherence of this negation and a predicate he introduces a recursive multiplying-out rule,<10> which can be formulated in the notation I am using as:

(R2) (w, x) --> y ⊢ (w, 1xz) --> 1yz.

Substituting '1PP' for 'w', 'P' for 'x' and 'y' ,and 'N' for 'z' gives us:

(20) (1PP, P) --> P ⊢ (1PP, 1PN) --> 1PN.

The premise in (20) holds by substituting into the original multiplying-out rule (R1), giving us:

(21) (1PP, P) --> P.

From (20) and (21) it follows that:

<10> "A Program for Syntax", p.485 and "Should Traditional Grammar be Ended or Mended?", p.21. See also Potts's "Fregean Grammar", p.11.

(22) (1PP, 1PN) --> 1PN.

This justifies the syntactic coherence of '$\}$ does not snore'. It also shows us that 'Henry (does not snore)' is syntactically coherent. Its structure is given by

(23) ((1PP, 1PN), N).

And (22) allows us to contract (23) to

(24) (1PN, N),

and (24) contracts to P by means of the original multiplying-out rule.

Geach's recursive rule also justifies the syntactic coherence of propositional negation and a two-place predicate. To see this, we need to substitute '1PP' for 'w', '1PN' for 'x' and 'y' and 'N' for 'z' in (R2). This gives us:

(25) (1PP, 1PN) --> 1PN ⊢ (1PP, 11PNN) --> 11PNN.

The premise has already been shown to hold as (22), so it follows that:

(26) (1PP, 11PNN) --> 11PNN.

The recursive rule (R2) also justifies the syntactic coherence

of propositional negation and a three-place predicate. Substituting in (R2) gives us:

(27) $\dfrac{\text{(1PP, 11PNN) --> 11PNN}}{\text{(1PP, 111PNNN) --> 111PNNN.}}$

And the premise has been shown to hold as (26).

The way in which (R2) can be used to justify the syntactic coherence of negation and a predicate of any polyadicity should be obvious. Assuming that we have justified the syntactic coherence of negation and a predicate of polyadicity i, we make appropriate substitutions in (R2) to ensure that the premise of the resulting inference is the same as the contraction which justified negation combining with the predicate of polyadicity i. The conclusion is then the required justification for the syntactic coherence of negation and a predicate of polyadicity i + 1.

Double Negation

A similar argument to that used to justify the syntactic coherence of propositional negation and a predicate shows how two negations can combine together.<11> The following is a substitution instance of (R2):

(28) (1PP, P) --> P |- (1PP, 1PP) --> 1PP.

And the premise of this holds by (R1).

A Derived Contraction Schema

The similarity mentioned in the previous Section can be shown by the following derived contraction schema:

(D1) (1xy, 1yz) --> 1xz.

This follows from (R2) by substituting '1xy' for 'w' and observing that the premise always holds by (R1). In order to use (D1) to justify the syntactic coherence of both propositional

---

<11> Frege thought that two negations could combine. See "Negation", p.156, where he makes this point about the senses of two negations. By the principle of parallel interpretation, however, it also holds on the linguistic level.

negation with a predicate and also of two negations, we substitute 'P' for 'x' and 'y'. In order to justify the combination of propositional negation and a predicate we also substitute 'N' for 'z' and to justify two negations combining we substitute 'P' for 'z' instead.

(D1) can also be used to show the syntactic coherence of a quantifier of category 1P1PN with a relational sign of category 11PNN. Substituting 'P' for 'x', '1PN' for 'y' and 'N' for 'z' in (D1) shows that a quantifier and a relational sign combine to give a predicate:

(1P1PN, 11PNN) --> 1PN.

Internal Conjunction

Conjunction is a binary truth-functional connective, so it is a linguistic function which makes a proposition out of two propositions. Thus, from 'Henry snores' and 'Maxine swears' it makes 'Henry snores and Maxine swears'. But on occasion it looks as if it yields a predicate when applied to two predicates. For example, in the proposition 'Joseph drinks and smokes'. Corresponding to this is the list of categories:

(11PPP, 1PN, 1PN, N).

We would like to transform this to the structure

(((11PPP, 1PN), 1PN), N),

which we would like to contract to P, but if rule (R1) is our only rule, then this structure cannot be contracted to anything. Thus, with just the basic multiplying-out rule the proposition 'Joseph drinks and smokes' cannot be shown to be syntactically coherent, yet intuitively it is perfectly acceptable.

It is possible to "solve" this problem by introducing a conjunction operator that makes a predicate out of two predicates, that is to say, a function of category 111PN1PN1PN, but this leads to problems similar to those discussed in connection with predicate negation.

Within the framework of a categorial grammar there are two different ways of handling this problem depending on whether you distinguish between the syntactic categories 11PPP and 2PPP and on whether you categorise conjunction as belonging to 11PPP or 2PPP. I shall first present the solution offered by Potts (which treats conjunction as belonging to the category 11PPP) and then I shall present Geach's solution (which treats conjunction as belonging to category 2PPP).<12>

<12> Potts's solution is presented in "A General Theory of the Meaning of Anaphoric Pronouns", pp.155-157, 166-171 and 194-195. Geach's solution is in "A Program for Syntax", pp.485-486.

Potts's Solution Potts first shows how conjunction can combine with two predicates to form a relational sign by means of the basic multiplying-out rule. The use of this rule should be clear now, so I will just present the skeleton of the argument.

(29) (11PPP, P) --> 1PP	R1
(30) (11PPP, 1PN) --> 11PPN	R2 29
(31) (1PP, 1PN) --> 1PN	D1
(32) (1PN, 1PP) --> 1PN	switch 31
(33) (1PN, 11PPN) --> 11PNN	R2 32
(34) (11PPN, 1PN) --> 11PNN	switch 33.

This justifies the following contraction:

(35) ((11PPP, 1PN), 1PN) --> 11PNN,

since (35) follows from (30) and (34) by the fact that contractions are allowed in context, and (35) justifies the syntactic coherence of the expression '$\}$ drinks and $\{$ smokes'.

It should be observed that in the argument just given I have been forced to make use of a new structural inference rule, namely, switch. This is given by the following schema:

(x, y) --> z |- (y, x) --> z.

Although the use of this structural rule is essential in justifying the syntactic coherence of many combinations of

expressions, Potts never makes this fact explicit.

Having shown how conjunction combines two predicates into a relational sign Potts then introduces an operator which turns a relational sign into a predicate. It is, thus, of category 11PN11PNN. According to Potts this operator corresponds to the anaphoric use of various kinds of pronoun in English and in this example it is represented by 'he'. Thus, we have that '$\{$ drinks and he smokes' is of category 1PN. Potts has thus shown the syntactic coherence of 'Joseph drinks and he smokes' and he then uses his distinction between the phonetic structure of a proposition and its semantic structure to show how 'Joseph drinks and smokes' is an acceptable projection of 'Joseph drinks and he smokes'.

Geach's Solution In this Subsection I allow the numeral '2' to be used in the formation of category names. Geach uses the difference between expressions of category 11PPP and those of category 2PPP to distinguish between subordinating and coordinating two-place connectives. Subordinating connectives, like 'if', belong to category 11PPP, whereas coordinating connectives, such as 'and' and 'or', belong to category 2PPP.<13> Thus, Geach would accept Potts's solution in order to show the syntactic coherence of propositions like 'if Jack is asleep, then he is at home', but he has to provide a different solution in order to deal with internal conjunction. He introduces an extra multiplying-out rule and an extra recursive rule to help us deal

---

<13> "A Program for Syntax", p.485.

with expressions belonging to categories of the form 2xyz. These are:

(G1) (2xyy, y, y) --> x,

(G2) (u, v, v) --> x |- (u, 1vy, 1vy) --> 1xy.

The multiplying-out rule (G1) justifies the syntactic coherence of a proposition like 'Henry snores and Maxine swears', since this has the following structure:<14>

(2PPP, (1PN, N), (1PN, N)).

This contracts to:

(36) (2PPP, P, P).

And this is just a substitution instance of (G1). The structure of the proposition 'Joseph drinks and smokes' is:

(37) ((2PPP, 1PN, 1PN), N).

Substituting '2PPP' for 'u', 'P' for 'v' and 'x' and 'N' for 'y' in (G2) gives us:

<14> Because I allow category names of the form '2xyz' in this Subsection, the notion of a structure is slightly different. Here, a structure is either a category or a pair of structures or a triple of structures. Clearly, the contraction relation is also slightly altered.

(38) (2PPP, P, P) --> P |- (2PPP, 1PN, 1PN) --> 1PN.

The premise of (38) has been shown to hold as (36), therefore we can infer:

(39) (2PPP, 1PN, 1PN) --> 1PN.

And this justifies the syntactic coherence of 'Joseph drinks and smokes'.

Other Combination Problems Involving Conjunction

Conjunction - and two-place connectives in general - give rise to a whole group of combination problems. In this Section I shall discuss the ways in which the propositions 'Jack loves and hates Jill' and 'Jill suspects and fears that Jack loves Maxine' can be shown to be syntactically coherent.

'Jack loves and hates Jill' The strategy behind this derivation is as follows. (A) We show that conjunction combines with two relational expressions to form a four-place relational sign. (B) This four-place relational sign is combined with a pronoun operator to obtain a three-place relational sign. (C) This three-place relational sign is combined with another pronoun to form a two-place relational sign. (D) This two-place relational sign is combined with two names to yield a

proposition.

In order to carry out stage (A) we first need to show that conjunction and a relational sign contract to an expression of category 111PNNP.

(40) (1PP, P) --> P	R1
(41) (1PP, 1PN) --> 1PN	R2 40
(42) (1PN, 1PP) --> 1PN	switch 41
(43) (1PN, 11PPN) --> 11PNN	R2 42
(44) (11PNN, 1PP) --> 11PNN	switch 43
(45) (11PNN, 11PPP) --> 111PNNP	R2 44
(46) (11PPP, 11PNN) --> 111PNNP	switch 45.

To complete stage (A) we have to show that an expression of category 111PNNP combines with a relational sign to form a four-place relational sign. This is achieved in this way:

(47) (111PNNP, P) --> 11PNN	R1
(48) (111PNNP, 1PN) --> 111PNNN	R2 47
(49) (111PNNP, 11PNN) --> 1111PNNNN	R2 48.

Putting (46) and (49) together justifies the contraction:

(50) ((11PPP, 11PNN), 11PNN) --> 1111PNNNN,

Stage (B) is easier to carry out. It is obtained as follows:

(51) (11PN11PNN, 11PNN) --> 1PN R1

(52) (11PN11PNN, 111PNNN) --> 11PNN R2 51

(53) (11PN11PNN, 1111PNNNN) --> 111PNNN R2 52.

Stage (C) was achieved in the course of stage (B), namely, (52). Combining (50), (53) and (52) justifies this contraction:

(11PN11PNN, (11PN11PNN, ((11PPP, 1PNN), 1PNN))) --> 11PNN.

And an expression of category 11PNN combines with two singular terms to form a proposition. Thus, we have just shown that the structure:

(((11PN11PNN, (11PN11PNN, ((11PPP, 11PNN), 11PNN))), N), N)

contracts to P. This is the structure of the proposition:

(54) Jack loves Jill and he hates her,

which is a semantic structure of the proposition with phonetic structure:

(55) Jack loves and hates Jill.

'Jill suspects and fears that Jack loves Maxine' In the proposition

(56) Jill suspects and fears that Jack loves Maxine

the operators 'suspects' and 'fears' are both of category 11PNP, that is to say, they both make a proposition out of a singular term and a proposition.<15> The strategy involved in justifying the syntactic coherence of (56) is the following. (A) We first show how conjunction combines with two expressions of category 11PNP to form an expression of category 1111PNNPP. (B) We then show how such an expression combines with a pronoun to form an expression of category 111PNPP. This is then shown to combine with a singular term to give an expression of category 11PPP. (C) The expression of category 11PPP is then shown to combine with a "pro-proposition" operator to yield an expression of category 1PP, which combines with a proposition to give the result. The "pro-proposition" is of category 11PP11PPP.

Stage (A) is carried out in two parts. First, we show how conjunction and an expression of category 11PNP combine to yield an operator of category 111PNPP:

<15> Prior argues for such a categorisation in Objects of Thought, pp.16-21.

(57) (1PP, 1PN) --> 1PN D1

(58) (1PN, 1PP) --> 1PN switch 57

(59) (1PN, 11PPP) --> 11PNP R2 58

(60) (11PPP, 1PN) --> 11PNP switch 59

(61) (11PPP, 11PNP) --> 111PNPP R2 60.

In the second part of stage (A) we show how an operator of category 111PNPP combines with an expression of category 11PNP to yield an expression of category 1111PNNPP:

(62) (11PNP, P) --> 1PN R1

(63) (11PNP, 1PN) --> 11PNN R2 62

(64) (11PNP, 11PNP) --> 11PNNP R2 63

(65) (11PNP, 111PNPP) --> 111PNNPP R2 64

(66) (111PNPP, 11PNP) --> 111PNNPP switch 65.

Combining these two results, that is to say, (61) and (66) gives us:

(67) ((11PPP, 11PNP), 11PNP) --> 111PNNP.

Stage (B) is easier to carry out. It is done in this way:

(68) (11PN11PNN, 11PNN) --> 1PN R1

(69) (11PN11PNN, 111PNNP) --> 11PNP R2 68

(70) (11PN11PNN, 1111PNNPP) --> 111PNPP R2 69.

It is slightly more tricky to show how an expression of category 111PNPP combines with a singular term:

(71) (1PN, N) --> P	R1
(72) (N, 1PN) --> P	switch 71
(73) (N, 11PNP) --> 1PP	R2 72
(74) (N, 111PNPP) --> 11PPP	R2 73
(75) (111PNPP, N) --> 11PPP	switch 74.

Stage (C) is entirely straightforward:

(76) (11PP11PPP, 11PPP) --> 1PP,

and 1PP combines with a proposition to yield a proposition. Putting (67), (70), (75) and (76) together justifies the contraction of the structure

(11PP11PPP, ((11PN11PNN, ((11PPP, 11PNP), 11PNP)), N), P)

to a proposition. And this is the structure of:

(77) Jill suspects that Jack loves Maxine and she fears it,

which is a semantic structure of the proposition with phonetic structure:

(78) Jill suspects and fears that Jack loves Maxine.

The pro-proposition introduced in the course of justifying that this example is syntactically coherent is similar to Potts's pronoun. The category of the pronoun is 11PN11PNN, that is to say, it makes a predicate out of a relational sign by identifying the relational sign's two argument-places. And the category of the pro-proposition is 11PP11PPP. It makes an expression of category 1PP out of one of category 11PPP by identifying the latter expression's two argument-places.

Quantifier and Negation

It is possible to construct examples in which we have to see a quantifier and negation forming a syntactic unit. For example, this has to be the case in the inference of 'there is something not everyone is' from 'not everyone is a hospital porter'.<16> In the context of this inference the latter proposition has the structure:

(79) ((1PP, 1P1PN), 1PN).

This can be shown to be syntactically coherent by rule (R2). Substituting in this we get:

<16> This example is derived from Dummett's Frege, p.69.

(80) (1PP, P) --> P |- (1PP, 1P1PN) --> 1P1PN.

The premise of this holds by (R1), therefore (79) contracts to:

(81) (1P1PN, 1PN),

and this contracts to P by (R1).

Other modes of combination of a quantifier and negation are more difficult to deal with. Consider the inference of 'there is something someone is not' from 'someone is not a hospital porter'. In this context the latter proposition has the structure:

(82) ((1P1PN, 1PP), 1PN).

If we only have rules (R1) and (R2) at our disposal, then this does not contract to anything. It is in normal form. Intuitively, the problem is that we cannot show the "internal" 'P' of '1P1PN' combining with the first 'P' in '1PP' to contract to 1P1PN. In order to solve this problem Potts introduces a further recursive rule, which I shall refer to as his recursive rule (R3).<17> In the notation I am using it is formulated as follows:

<17> "Fregean Grammar", pp.16-17.

(R3) (1uv, w) --> 1xy |- (1u1vz, w) --> 1x1yz.

Given this rule we can show how the structure (82) contracts to P. Clearly, it is only necessary to show how (1P1PN, 1PP) contracts to 1P1PN. Substituting 'P' for 'u', 'v', 'x' and 'y', '1PP' for 'w' and 'N' for 'z' in rule (R3) gives us:

(1PP, 1PP) --> 1PP |- (1P1PN, 1PP) --> 1P1PN.

And the premise of this can be shown to hold by making appropriate substitutions in (D1).

Further Structures Underiveable Without Potts's Rule

There are at least two other combinations of expressions that cannot be shown to be syntactically coherent without Potts's recursive rule (R3). These are the combination of two quantifiers of category 1P1PN to form an expression of category 1P11PNN and the combination of conjunction and a personal pronoun to form a relative pronoun.<18> I shall look at each of these in turn.

<18> These are discussed in Potts's "Fregean Grammar", p.17, and his "A General Theory of the Meaning of Anaphoric Pronouns", pp.194-195.

Two Quantifiers In order to justify, for example, the inference of 'there is some relation that everyone stands in to someone' from 'everyone loves someone' the two quantifiers of category 1P1PN involved must form a syntactically coherent subexpression. Thus, we have to show that (1P1PN, 1P1PN) contracts to 1P11PNN. Substituting 'P' for 'u', 'v' and 'x', '1P1PN' for 'w' and '1PN' for 'y' in (R3) gives us:

(1PP, 1P1PN) --> 1P1PN |- (1P1PN, 1P1PN) --> 1P11PNN,

and the premise of this holds by (D1).

The Relative Pronoun Potts argues that relative pronouns such as 'who' are of category 111PN1PN1PN, that is to say, they make a predicate out of two other predicates. Furthermore, he believes that the relative pronoun 'who' is composed of conjunction and a personal pronoun of category 11PN11PNN.<19> Intuitively, it does seem as if the following two propositions have the same semantic structure:

(83) Socrates, who was a philosopher, lived in Athens,

(84) Socrates lived in Athens and he was a philosopher.

But in order to show that (11PN11PNN, 11PPP) reduces to 111PN1PN1PN we have to make use of the recursive rule (R3), since

---

<19> This position is also argued for by Geach, in "Quine's Syntactical Insights", pp.152ff., who calls it the Latin prose theory of relative clauses.

it cannot be shown otherwise. The derivation is as follows:

(85) (11PNP, P) --> 1PN	R1
(86) (11PNP, 1PP) --> 11PNP	R2 85
(87) (1PP, 11PNP) --> 11PNP	switch 86
(88) (1PP, 11PN1PN) --> 11PN1PN	R3 87
(89) (11PN1PN, 1PP) --> 11PN1PN	switch 88
(90) (11PN1PN, 11PPP) --> 111PN1PNP	R2 89
(91) (11PPP, 11PN1PN) --> 111PN1PNP	switch 90
(92) (11PPP, 11PN11PNN) --> 111PN1PN1PN	R3 91.

The list of categories of proposition (84) is

(N, 1PN, 11PPP, 11PN11PNN, 1PN).

This can be transformed to the structure

((((11PN11PNN, 11PPP), 1PN), 1PN), N),

which contracts - as a result of (92) - to

(((111PN1PN1PN, 1PN), 1PN), N),

and this is the structure of proposition (83).

This concludes my discussion - in this Chapter - of combination problems and the ways in which Geach and Potts solve them. Before criticising their solutions, I want to explore some

of the ideas involved in constructing a categorial grammar for natural language a bit further.

Syntactic Completion Analysis

'Syntactic completion analysis' is the name that Hiż gives to the principle that if we remove an expression of category x from an expression of category y, then the resulting expression is of category 1yx.<20> Potts calls this Frege's analytical procedure.<21> And Dummett calls it the extraction of functions, adding that here he is 'using "function" in its Begriffsschrift sense'.<22> A particular instance of it is also one of the component steps allowed in the step-by-step construction of a sentence, which Dummett discusses thoroughly in Frege. This principle is clearly derived from Frege's second way of making names out of names.<23> Frege restricts this second procedure to making names of first-level functions with argument-places of

---

<20> In his paper "Syntactic Completion Analysis". I have been unable to see a copy of this paper, so my knowledge of its content is derived from Lehrberger's discussion of it in his book Functor Analysis of Natural Language, pp.48-49. Hiż's principle is actually more complicated because his notation for categories makes allowance for word order. Thus, he assigns 'Caesar killed' and 'killed Caesar' to different categories.

<21> "Fregean Grammar", p.7.

<22> The Interpretation of Frege's Philosophy, p.281.

<23> Grundgesetze, Section 30.

type 1, but later writers have generalised it.

A simple consequence of the principle of syntactic completion analysis is that every proposition has a countably infinite number of distinct decompositions. Consider, for example, the proposition 'Socrates flies'. We can assume that 'Socrates' is a singular term. Therefore, by this principle, we can say that 'ξ flies' is of category 1PN. But removing 'ξ flies' from 'Socrates flies' results in the expression 'φ(Socrates)' of category 1P1PN. (Put '1PN' for 'x' and 'P' for 'y' in the above formulation of this principle.) Frege is aware of this possibility, since in Grundgesetze (Section 22) he writes about the referents of such expressions:

> We also have a second-level function in $\varphi(2)$... This second-level function is distinct from the number 2 itself, since, like all functions, it is unsaturated.

This possibility is misinterpreted by Baker and Hacker, who write:

> Even if the judgeable-content $\vdash\Phi(A)$ is first described as ascribing a property to an object named by 'A', it can with equal propriety be characterised as stating that the concept Φ falls under a second-level concept; the content $\vdash\Phi(A)$, as it were, contains a second-level concept... On this second interpretation, Frege apparently concluded through an argument by elimination, the symbol 'A' must itself be viewed as the name of a second-level concept.<24>

Just because 'Socrates flies', for example, can be decomposed

---

<24> Frege, p.166.

into the predicate 'ξ flies' and the singular term 'Socrates' as well as into the same predicate and the second-level linguistic function 'φ(Socrates)', it does not follow that 'Socrates' refers to the same entity that this second-level linguistic function refers to.

But these are not the only ways in which 'Socrates flies' can be decomposed. As 'φ(Socrates)' occurs in 'Socrates flies' it is possible to remove it to obtain an expression of category 1P1P1PN, which we might symbolise temporarily as '... flies'. (Put '1P1PN' for 'x' and 'P' for 'y' in the above principle.) But this is still not the end of the matter. As '... flies' occurs in 'Socrates flies' we can remove it to obtain an expression of category 1P1P1P1PN and, quite clearly, this process can be carried on indefinitely.

Thus, we can see that singular terms have analogues on every even level in the infinite hierarchy of expressions and predicates have analogues on every odd level. A similar argument would show that quantifiers of category 1P1PN have analogues on every higher even category. In fact, in general, an expression of category x of level i has analogues on every level j if j is greater than i and the difference between j and i is a multiple of 2.

Reflecting on the possibility of decomposing one and the same proposition in a number of distinct ways as discussed above shows us that the original formulation of the principle of syntactic completion analysis is imprecise, since removing a predicate, say 'ξ flies', from a proposition, say 'Socrates flies', can result

either in a proper name 'Socrates' or in a second-level linguistic function 'φ(Socrates)'. What is missing from the formulation is that we should only regard the principle as being correct if the expression being removed is always thought of as being the argument of the resulting linguistic function and this is how I shall think of it from now on.

Potts's Pro-Verb

It is theoretically possible to decompose the proposition 'Socrates flies' into a second-level linguistic function 'φ(Socrates)' and a predicate 'ξ flies', but is there any reason to think that this is ever practically necessary? Consideration of examples involving the operator that Potts calls the pro-verb suggests that there are occasions when it is helpful. The pro-verb is a third-level analogue of the second-level pronoun and its category is 11P1PN11P1PN1PN. Whereas Potts's pronoun operator makes a predicate out of a relational sign by identifying the relational sign's two argument-places, the pro-verb makes an expression of category 1P1PN out of one of category 11P1PN1PN by identifying the latter expression's two argument-places. It is needed in the following sort of proposition:<25>

<25> I owe this example to Dr Potts.

(93) If all of the Asian ambassadors refuse to come,
then so will some of the African ambassadors.

It is not necessary to categorise all the individual words of this example. It is sufficient to see that both the phrases 'all of the Asian ambassadors' and 'some of the African ambassadors' are quantifiers of category 1P1PN and that 'refuse to come' is a predicate. The pro-verb is represented in the phonetic structure (93) by 'so'. The list of categories of this proposition is:

(94) (11PPP, 1P1PN, 1PN, 11P1PN11P1PN1PN, 1P1PN)

and its structure is:

(95) ((11P1PN11P1PN1PN, ((11PPP, 1P1PN), 1P1PN)), 1PN).

To show that this cancels to P we need to use Geach's first recursive rule (R2) to show that:

(96) ((11PPP, 1P1PN), 1P1PN) --> 11P1PN1PN.

This is done as follows:

(97) (1PP, 1P1PN) --> 1P1PN D1

(98) (1P1PN, 1PP) --> 1P1PN switch 97

(99) (1P1PN, 11PPP) --> 11P1PNP R2 98

(100) (11PPP, 1P1PN) --> 11P1PNP switch 99

(101) (11P1PNP, 1P1PN) --> 11P1PN1PN D1.

The result that (96) holds follows from (100) and (101). It is then straightforward to show that (95) contracts to P.

But with such a pro-verb the following sentence cannot be shown to be syntactically coherent:

(102) If Maxine refuses to come, then so will Jack.

It has the list of categories

(103) (11PPP, N, 1PN, 11P1PN11P1PN1PN, N)

and there is no way to rearrange this list so that the resulting structure contracts to P. Yet, intuitively the proposition (102) is perfectly acceptable. Potts's solution to this problem is to categorise the proper names that occur in (102) as being of category 1P1PN. Then, it is possible to show that (102) is

syntactically coherent, since it has the same structure as (93), namely (95).<26>

I shall not consider the ramifications of systematically treating all occurrences of proper names as being of category 1P1PN. The purpose of this Section was to present some evidence to show that doing this does have some advantages.

The Use of Syntactic Completion Analysis

The principle of syntactic completion analysis is usually used to determine the unknown category of an expression or phrase on the assumption that we know the categories of other expressions or phrases. Some examples will make this procedure clear. Say, we want to find out the category of the reflexive pronoun 'himself'. In the proposition 'Canning killed himself' we know that 'Canning' is of category N and 'ζ killed ζ' is of category 11PNN. It is also the case that 'ζ killed himself' is of category 1PN. Therefore, letting 'x' stand for the category of 'himself', we have either that (11PNN, x) --> 1PN or that (x, 11PNN) --> 1PN. The first alternative is the case in which we think of 'himself' as being another argument to 'ζ killed ζ'. In this case x must

---

<26> Potts discusses the pro-verb in his paper "A General Theory of the Meaning of Anaphoric Pronouns", pp.190ff. Montague also treats proper names as belonging to the same category as quantifiers, for example, in his paper "The Proper Treatment of Quantification in Ordinary English". But his reasons for doing so are different from Potts's.

be N, but we know that 'himself' is not a singular term. So, we have to pick the second alternative. In this case 'himself' is a linguistic function which turns a relational sign into a predicate, therefore x is 11PN11PNN and this is the correct categorisation of 'himself'.

The method of syntactic completion analysis is a powerful way of extending our knowledge of how various expressions should be categorised, given that we know how to categorise a number of basic expressions. Without it the programme of devising a categorial grammar for natural language would be crippled. It would be too much of a digression for me to give lots of examples of its usefulness, so I will give just one more example.<27>

Say we are interested in finding out how to treat prepositions in a categorial grammar. Let us consider the proposition:

(104) Raleigh smoked in London.

Our intuition suggests that the syntactically coherent subexpressions of this are shown by the bracketing:

(105) Raleigh (smoked (in London)).

Let us use 'y' to represent the category of 'in London'. Thus, the structure of (104) is given by

---

<27> This is adapted from Geach's "A Program for Syntax", pp.490-492.

(106) ((y, 1PN), N)

and so we know that

(107) (y, 1PN) --> 1PN

must hold. Therefore, y is 11PN1PN. Now let us use 'x' to represent the category of 'in'. We have established that 'in London' is of category 11PN1PN. We know that 'London' is a proper name. Assuming that 'in' is the function and 'London' the argument, by the principle of syntactic completion analysis we can infer that 'in' is of category 111PN1PNN. That is to say, it takes a proper name into an operator that makes predicates out of predicates.

Difficulties with the Geach-Potts Solutions

In this Section I turn my attention to criticising the method adopted by Geach and Potts to solve the various combination problems. The recursive rules that they introduce for this purpose are unacceptable for at least four reasons. (a) The introduction of their rules has the consequence that there are a number of propositions and other expressions in the justification of whose syntactic coherence occur intermediate structures which cannot be interpreted. (b) The introduction of their rules

destroys Frege's partitioning of the expressions of a given language into discrete non-overlapping syntactic categories. (c) The introduction of their rules destroys the universal validity of the principle of syntactic completion analysis. (d) The introduction of their rules has the consequence that one and the same structure may contract to two distinct normal forms. I will now explain each of these criticisms in greater detail.

Uninterpretable Intermediate Results In justifying the syntactic coherence of various combinations of expressions by means of the rules (R2) and (R3) in this Chapter I have been forced to make use of the structural rule of inference switch, which has the following property:

$$(x, y) \rightarrow z \vdash (y, x) \rightarrow z,$$

but such a rule violates one of the fundamental assumptions of a categorial grammar. The notation

(108) $(x, y) \rightarrow z$

means that an expression of category x, considered as a linguistic function, yields an expression of category z when applied to an argument of category y. Similarly, the notation

(109) $(y, x) \rightarrow z$

means that an expression of category y, considered as a

linguistic function, yields an expression of category z when applied to an argument of category x. Understood in this way, a rule converting (108) into (109) cannot be allowed. This is because it destroys the asymmetry between function and argument.

Overlapping Syntactic Categories Potts adopts Geach's solution to the problem of how propositional negation and a predicate can combine together and he draws out one of its consequences as follows:

> The effect of (R2) is to increase the possibilities of combination of functorial category symbols. Given a series of category symbols of the form
>
> $1\alpha\beta$, $2\alpha\gamma 1\beta\gamma$, $3\alpha\gamma\delta 2\beta\gamma\delta$, ...
>
> if (R1) is our only rule, no symbol in the series will do the work of any other; once (R2) is added, any symbol will do the work of any subsequent symbol in the series, but not conversely. ("Fregean Grammar", p.5.)

But how are we to understand the notion of an expression belonging to one category doing the work of an expression belonging to another category? It seems that this has to be understood as meaning that one and the same linguistic function - negation, say - takes as arguments expressions belonging to categories P, 1PN, 11PNN and also expressions belonging to lots of other categories. But this clearly violates Frege's type theory, for a fundamental principle of that theory is that if expressions X and Y belong to different categories, then there is no incomplete expression which has an argument-place that is

fitting for both X and Y. Given Frege's principles a totality consisting of, say, propositions, predicates and relational signs cannot be coherently formulated. Thus, the introduction of rule (R2) destroys one of the assumptions on which Frege's hierarchy is built, namely, that one and the same expression cannot belong to more than one syntactic category.

Another way to establish this conclusion is to recall that rule (R2) justifies, for example, the following contraction:

(1PP, 1PN) --> 1PN.

I made use of this in discussing the syntactic coherence of negation and a predicate. As mentioned in the previous Subsection, the notation (108) means that a linguistic function of category x, when applied to an expression of category y, yields an expression of category z. Hence, the above contraction should mean that a linguistic function of category 1PP yields values of category 1PN when applied to arguments of category 1PN, but this is precisely what expressions of category 11PN1PN do. So, rule (R2) has the consequence that expressions of category 1PP also belong to category 11PN1PN and this violates Frege's type theory.<28>

Syntactic Completion Analysis Fails The principle of syntactic completion analysis is a powerful tool in determining

---

<28> Another way of looking at what rule (R2) does is to see it as conflating functional application with functional composition.

the categories of various expressions and phrases in a language, but it does not hold in a categorial grammar that contains Geach's recursive rule (R2).

The structure of the proposition 'Socrates (does not snore)' is ((1PP, 1PN), N), but let us - for the sake of argument - assume that we do not know the category of negation. We thus have to solve the following "equation":

(110) (x, 1PN) --> 1PN.

In a grammar with only rule (R1) x has to be 11PN1PN, but in a grammar with both (R1) and (R2) x might be either 1PP or 11PN1PN and this is inconsistent with the principle of syntactic completion analysis.

This criticism can be generalised. I showed above how the contraction (D1) can be derived:

(D1) (1xy, 1yz) --> 1xz.

Substituting '1xz' for 'x' and '1yz' for 'y' in (R1) gives us another derived contraction schema:

(D2) (11xz1yz, 1yz) --> 1xz.

Let us assume that we have ascertained the following three facts: (a) 'alpha beta' belongs to the category 1xz; (b) 'beta' belongs to the category 1yz; and (c) 'beta' is the argument of 'alpha

beta'. Then, if our categorial grammar contains (R2) - and hence (D1) - it is impossible to conclusively ascertain the category of 'alpha'. It could be either 1xy or 11xz11yz.

(R3) also destroys the universal validity of syntactic completion analysis. Let us assume that we have established the following three facts: (a) 'gamma delta' belongs to category 1P1PN; (b) 'delta' belongs to category 1PP; and (c) 'delta' is the argument of 'gamma delta'. Then, 'gamma delta' can have either of the following structures:

(111) (1P1PN, 1PP),

(112) (11P1PN1PP, 1PP).

Both of these contract to 1P1PN. (111) requires (R3), whereas (112) just requires (R1). And this criticism can obviously be generalised.

The loss of syntactic completion analysis would greatly impair the value of a categorial grammar. This is because of the methodological principles underlying the way in which categorial grammarians extend the categorisation of the expressions of a language. Typically, they begin from proper names, propositions and predicates. Then, they try to include other expressions in the lexicon they are building up by looking at how they combine with expressions of known category. So, if they are interested in adverbs, such as 'slowly', they would look at sentences like 'Jack drives slowly'. After ascertaining its organisation, they would use syntactic completion analysis to tell them the category

of 'slowly'. And this method would be extended to more types of expression and increasingly complex sentences. Without syntactic completion analysis, this would not produce determinate categorisations.

Multiple Normal Forms Consider the contraction:

(113) (1PP, P) --> P.

This is a simple substitution instance of (R1). It can be used in the following way:

(114) (1PP, 1PN) --> 1PN	R2 113
(115) (1PN, 1PP) --> 1PN	switch 114
(116) (1PN, 11PPP) --> 11PNP	R2 115
(117) (11PPP, 1PN) --> 11PNP	switch 116
(118) (11PPP, 11PNN) --> 111PNPN	R2 117.

But it can also be used in a different way:

(119) (P, 1PP) --> P	switch 113
(120) (P, 11PPP) --> 1PP	R2 119
(121) (11PPP, P) --> 1PP	switch 120
(122) (11PPP, 1PN) --> 11PPN	R2 121
(123) (11PPP, 11PNN) --> 111PPNN	R2 122.

The right hand sides of both (118) and (123) are in normal form, but they are distinct. This, therefore, shows that one and the

same structure, namely, the right hand side of (113), has at least two distinct normal forms.

Conclusion

In this Chapter I looked at the attempt made by Geach and Potts to apply Frege's views about the way in which the propositions of the Begriffsschrift are formed to natural language. In doing this they come across combination problems. Such problems do not occur in a formalised language. Geach and Potts solve these problems by adding extra syntactic rules. The most serious criticism of the rules they add is that they destroy the partition of the expressions of natural language into discrete non-overlapping syntactic categories. In effect, these rules create trans-categorial operators and such operators cannot be accounted for in a Fregean language.

In Chapter 3 I discussed other kinds of trans-categorial operators and I showed how they could be accommodated in a language which treated expressions like 'the concept horse' as genuine complete expressions. This suggests that it might be possible to solve combination problems in a similar sort of language. The way in which this can be done is the topic of the next Chapter.

Chapter 6: Combinatory Grammar

Introduction

In this Chapter I give a brief introduction to the main ideas of combinatory grammar. This is based on that branch of illative combinatory logic known as the theory of functionality.<1> I then solve all the combination problems discussed in the previous Chapter in the framework of such a grammar.

Because of the unfamiliarity of combinatory logic I have decided to provide an impressionistic account of some of its fundamental concepts. I have done this in the idioms of informal mathematics and I am well aware that many Fregean criticisms could be made of it. It is not essential to my argument and readers who want a Fregean account of combinatory logic are strongly advised to skip this informal account, which is contained in the next Section.

<1> Whereas pure combinatory logic just investigates the properties of combinators, the various systems of illative combinatory logic contain extra primitives which correspond to such logical notions as implication, generality and functionality.

A Plea for Combinatory Logic

Pure combinatory logic is that branch of mathematical logic concerned with investigating the properties of combinators. These are very general functions which are implicitly made use of in many branches of mathematics and even in some elementary mathematics. A very simple example is functional composition. This is often represented in mathematical books by means of a small circle written as an infix operator between the expressions for its two functional operands. It is defined in the following way:

$$(f \circ g)(x) \hat{=} f(g(x)).$$

In combinatory logic functional composition is represented by the capital letter '$\underset{\sim}{B}$', which is written as a prefix operator before the expressions for its functional arguments. Its fundamental property is:

$$\underset{\sim}{B}\ f\ g\ x = f\ (g\ x).$$

It is necessary at this stage to mention some of the assumptions and notational conventions of combinatory logic.

It only deals with one-place functions, but as functions are allowed to be both the arguments and values of other functions it is possible to "simulate" a function of type (J * J) --> J, say, by one of type J --> (J --> J). Concerning notation, in combinatory logic all functional signs are prefix operators and juxtaposition represents functional application. Brackets are used to distinguish between ((a b) c) and (a (b c)), for example. In order to avoid having to write lots of brackets they can be left out on the assumption that they associate to the left. That is to say, (a b c d) is the same as (((a b) c) d).

I will now introduce some more combinators. This can be done conveniently by analysing a suitable piece of mathematical discourse. I shall consider the proposition:

$$(1)\quad \frac{d}{dx}(16x) = 16.$$

Differentiation is here understood as being a higher-level function whose arguments and values are both functions from real numbers to real numbers. The first thing to note about (1) is that it looks as if the right hand side (RHS) is just a number, but mathematicians would construe the RHS of (1) as that function which for any real number taken as argument has the value 16. In informal mathematics there is no notational difference between the representation of this function and that of the number 16. Menger, in his interesting article "On Variables in Mathematics and in Natural Science", uses the combination of symbols '$\underline{\underline{16}}$' to stand for that function which for any real number taken as

argument has the value 16. In combinatory logic there is a combinator $\underset{\sim}{K}$ which makes the function $\underline{\underline{16}}$ out of the number 16. That is to say, the combinator $\underset{\sim}{K}$ takes a real number, say 16, as its argument and returns as its value that function which for any real number taken as argument has the value 16. Thus, ($\underset{\sim}{K}$ 16 7) is 16 and ($\underset{\sim}{K}$ 16 11) is 16 and ($\underset{\sim}{K}$ 16 17) is 16. So, it is easy to see that the combinator $\underset{\sim}{K}$ has the following fundamental property:

(2) $\underset{\sim}{K}$ a b = a.

In informal mathematics the argument of the differential operator in (1) would be said to be 16x. This is because it is conventional in informal mathematics to name a function after its value for the indeterminate x.<2> 16x is understood as being that function which for any given number taken as argument returns 16 times that number. Juxtaposing symbols has many meanings in mathematics and here it represents multiplication. Multiplication is a two-place function which returns the product of its arguments, but here it looks as if its value is a function from reals to reals. Moreover, the arguments to the multiplication here are both functions and not numbers. Let us use the symbol 'M1' to represent this function. Now we have to investigate what its two arguments are.

<2> There are logical scruples against saying this without qualification and in Chapter 1 I pointed out Frege's objections to such ordinary mathematical discourse. For the time being I shall couch my discussion in the terms of informal mathematics rather than translating everything into Fregean terminology.

'16' represents the function ($\underset{\sim}{K}$ 16 x) and 'x' represents the identity function, which returns its argument unchanged. In combinatory logic this is called the combinator $\underset{\sim}{I}$ and its fundamental property is:

(3) $\underset{\sim}{I}$ a = a.

Summarising, the notation '16x' represents the combination by means of M1 of the two functions ($\underset{\sim}{K}$ 16 x) and ($\underset{\sim}{I}$ x). M1 is closely related to multiplication. Let us write 'M' for multiplication. If we think about what the function M1 does, we see that the following equivalence holds:

(4) M ($\underset{\sim}{K}$ 16 x) ($\underset{\sim}{I}$ x) = M1 ($\underset{\sim}{K}$ 16) $\underset{\sim}{I}$ x.

So, 16x is more appropriately expressed as (M1 ($\underset{\sim}{K}$ 16) $\underset{\sim}{I}$ x). There is a combinator $\underset{\sim}{\Phi}$ which makes M1 out of M. Its fundamental property is:

(5) $\underset{\sim}{\Phi}$ a b c d = a (b d) (c d).

Putting all this together we have instead of (1) the logically preferable (from the point of view of combinatory logic):

(6) $\frac{d}{dx}$ ($\underset{\sim}{\Phi}$ M ($\underset{\sim}{K}$ 16) $\underset{\sim}{I}$ x) = $\underset{\sim}{K}$ 16.

Variables-binding operators can be eliminated in combinatory

logic. In (6) we see that the variable 'x' bound by the differential operator is at the extreme right of the expression for the argument of the differential operator on the LHS of the equation. I will later show that in combinatory logic it is always possible to move variables to this position and to coalesce multiple occurrences of the same variable. So, we can replace the usual differential operator by Arbogast's symbol for differentiation 'D' and we can leave off the rightmost variable 'x' on the LHS of the equation, giving:

(7) D ($\underset{\sim}{\Phi}$ M ($\underset{\sim}{K}$ 16) $\underset{\sim}{I}$) = $\underset{\sim}{K}$ 16.

There are a number of other useful combinators, but here I will only introduce one more. To justify the combinator $\underset{\sim}{W}$ let us consider the connection between the multiplication function M and the square function, which I shall denote by 'Q' for the time being. These are related as follows:

(8) Q a = M a a.

$\underset{\sim}{W}$ is the combinator that makes Q out of M and its fundamental property is

(9) $\underset{\sim}{W}$ a b = a b b.

Thus, we have that Q is ($\underset{\sim}{W}$ M).

The Foundations of Combinatory Logic

The language of combinatory logic is particularly easy to describe, because it contains no variables whatsoever. Given a set ATM of constants, the set CMB of all the terms of the language is the smallest set satisfying the following two conditions:

(i) Every member of ATM is a member of CMB.

(ii) If X and Y are members of CMB, then so is $\ulcorner(X\ Y)\urcorner$.

The set ATM is assumed to contain symbols for all the basic combinators, so all of the following belong to it: '$\underset{\sim}{S}$', '$\underset{\sim}{K}$', '$\underset{\sim}{B}$', '$\underset{\sim}{I}$', '$\underset{\sim}{W}$', '$\underset{\sim}{C}$', '$\underset{\sim}{\Phi}$' and '$\underset{\sim}{\Psi}$'.

The entities that combinatory logic deals with are known as obs. There is a single class of these and all the combinators are in it. In Fregean terms, the obs are complete entities. There is only a single way of making obs out of obs and that is application. If X and Y are obs, then so is (X Y). In Fregean terms, application is an incomplete entity. It is that unsaturated function which makes an ob (X Y) out of the two obs X and Y. So, it could be represented as $(\xi\ \zeta)$. Quite clearly application is not itself an ob. Brackets can be left out on the assumption that they associate to the left, so (a b c d) is the same as (((a b) c) d). The outermost pair of brackets can also be omitted.

Curry coined the word 'ob' in order to be completely

noncommittal about the nature of the objects that combinatory logic deals with.<3> It is possible to interpret obs in a number of different ways, but as my interests are to do with language, most of the obs I shall consider will be linguistic expressions.

The intuitive understanding of application is as functional application, but every ob can appear both in the function-place and the argument-place of an application. Indeed, it is possible for the same ob X, say, to appear both as function and as argument in a single application: (X X). The fact that every ob can be the argument of every other ob and that every ob can occur both in function-place and argument-place of an application is the reason why models for combinatory logic are such complicated mathematical structures.<4>

At this stage I need to say something about my use of the word 'function'. In combinatory logic the only unsaturated entity used in the construction of obs is application. Hence, in Fregean terms, this is the only function involved, but I have been referring to certain obs as functions as well. Clearly, this is a different use of the word from that which I have used in talking about Frege's views, since - for him - functions are unsaturated entities. No confusion should arise, however, since the context will make it clear which meaning the word should have. In Fregean terms, these saturated functions are the

---

<3> "The Elimination of Variables by Regular Combinators", p.128.

<4> Barendregt's The Lambda Calculus and Stoy's Denotational Semantics contain discussions of the nature of such models. Stoy's method - making use of lattices rather than complete partial orders - is considered old-fashioned nowadays.

objects which 'go proxy for' Frege's unsaturated functions and concepts (and which I discussed in Chapter 3).

Two of the most important ideas in combinatory logic are those of reduction and conversion. Both of these are two-place relations between obs. Before defining them precisely I need to say a few things about relations over obs in general. I make use of the following properties of relations:

(R) a R a,

(S) If a R b, then b R a,

(T) If a R b and b R c, then a R c,

(M) If a R b, then (c a) R (c b),

(N) If a R b, then (a c) R (b c).

If a relation R has the property (R), it is said to be reflexive and if it has the property (S) it is symmetric. If it has (T) it is transitive, if (M) it is right monotonic and if (N) it is left monotonic.

If Q is a given relation, then the monotone quasi-ordering generated by Q is the relation R such that if b Q c, then b R c and also R has the properties (R), (T), (M) and (N). Furthermore, the monotone equivalence generated by Q is the relation R such that if b Q c, then b R c and also R has the properties (R), (S), (T), (M) and (N).<5>

<5> For further details see Curry and Feys, Combinatory Logic, pp.59ff. and Barendregt, The Lambda Calculus, pp.50ff. and 150-151. Their presentations are more formal and rigorous than mine.

Let the relation ==> be given by all the substitution instances of the following schemata:

(K) K a b ==> a,

(S) S a b c ==> a b (a c).

Then the monotone quasi-ordering generated by ==>, which is denoted by '-->', is the two-place reduction relation mentioned earlier and the monotone equivalence, denoted by '=', is the two-place conversion relation mentioned earlier.

The most fundamental theorem in combinatory logic is the (first) Church-Rosser Theorem, which states that if b = c, then there exists a d such that b --> d and c --> d. This theorem is important because it establishes the consistency of combinatory logic. It would be too much of a digression to explain what 'consistency' means here and how it follows from the Church-Rosser Theorem. The interested reader is referred to Chapter 4 of Curry and Feys's book, where there is also a proof of the theorem.

All the combinators can be defined in terms of $\underset{\sim}{S}$ and $\underset{\sim}{K}$ and here I give the definitions of those used in this Chapter:

$\underset{\sim}{B} \hat{=} \underset{\sim}{S}\ (\underset{\sim}{K}\ \underset{\sim}{S})\ \underset{\sim}{K}$,

$\underset{\sim}{I} \hat{=} \underset{\sim}{S}\ \underset{\sim}{K}\ \underset{\sim}{K}$,

$\underset{\sim}{W} \hat{=} \underset{\sim}{S}\ \underset{\sim}{S}\ (\underset{\sim}{K}\ \underset{\sim}{I})$,

$\underset{\sim}{C} \hat{=} \underset{\sim}{S}\ (\underset{\sim}{B}\ \underset{\sim}{B}\ \underset{\sim}{S})\ (\underset{\sim}{K}\ \underset{\sim}{K})$,

$\underset{\sim}{\Phi} \hat{=} \underset{\sim}{B}\ (\underset{\sim}{B}\ \underset{\sim}{S})\ \underset{\sim}{B}$,

$\underset{\sim}{\Psi} \hat{=} \underset{\sim}{\Phi}\ (\underset{\sim}{\Phi}\ (\underset{\sim}{\Phi}\ \underset{\sim}{B}))\ \underset{\sim}{B}\ (\underset{\sim}{K}\ \underset{\sim}{K})$.

These have the following reduction properties:

$\underset{\sim}{B}$ a b c --> a (b c),

$\underset{\sim}{I}$ a --> a,

$\underset{\sim}{W}$ a b --> a b b,

$\underset{\sim}{C}$ a b c --> a c b,

$\underset{\sim}{\Phi}$ a b c d --> a (b d) (c d),

$\underset{\sim}{\Psi}$ a b c d --> a (b c) (b d).

Outline of a Combinatory Grammar

It would be a very large task to construct a combinatory grammar for a non-trivial subset of natural language, but my goal in the next Section is just to show how combinatory ideas can be used to solve combination problems. So, in this Section I will just give enough information about such a grammar to make those solutions comprehensible. In what follows the set ATM will consist of all

the names of the combinators that I have introduced together with names for all the expressions I mention. Thus, for example, '"snores"' and '"Jack"' are members of ATM. The set of simple obs is the set of all those things that these names name. So, the following are examples of obs: B, K, 'snores' and 'Jack'.

The Base Component A combinatory grammar consists of two parts. These are the base component and the transformational component. The base component consists of a lexicon and a single rule schema. The lexicon consists of a set of propositions, each of which is of the form X: x, where X is either an expression or a combinator and x is its functional character. The term 'functional character' has roughly the same meaning here as it did in Chapter 3. These functional characters "correspond" to the syntactic categories of unsaturated expressions and when there is no danger of confusion I will also call the functional character of a complete expression its syntactic category. Furthermore, rather than introducing yet another notation, I shall say that 'snores', for example, has the functional character 1PN and write this as:

'snores': 1PN.

Used in this way the notation '1PN' corresponds to the 'N ==> P' of Chapter 3, rather than to the 'N --> P' of Chapter 1 (and similarly for other category names). I think that there is little danger of confusion, since I have stated several times that there is only a single unsaturated expression in combinatory

logic which makes the name of an ob out of two ob names. The set of category names that I use in this Chapter is the standard class of category names that I defined in Chapter 5.

The following is part of the lexicon used in this Chapter:

'smokes', 'drinks', 'drives': 1PN,

'slowly', 'carefully': 11PN1PN,

'and': 11PPP,

'Jack', 'Joseph': N.

The first line of this means, for example, that the expressions 'smokes', 'drinks' and 'drives' all have the functional character 1PN. It should be noted that all the expressions which occur in propositions belonging to the lexicon are complete expressions.

As well as containing propositions which assign expressions to particular syntactic categories, the lexicon also contains propositions which assign the combinators to generic categories:

$\underset{\sim}{K}$: 11xyx,

$\underset{\sim}{S}$: 111zx1yx11zyx,

$\underset{\sim}{B}$: 111zx1yx1zy,

$\underset{\sim}{I}$: 1xx,

$\underset{\sim}{W}$: 11yx11yxx,

$\underset{\sim}{C}$: 111zyx11zxy,

$\underset{\sim}{\Phi}$: 1111zu1yu1xu11zyx,

$\underset{\sim}{\Psi}$: 1111zxx1yx11zyy.

The proposition '$\underset{\sim}{K}$: 11xyx', for example, means that $\underset{\sim}{K}$ belongs to every syntactic category whose name can be obtained from '11xyx' by substituting category names for 'x' and 'y'.<6>

The rule in a combinatory grammar corresponding to the basic multiplying-out rule in a system of categorial grammar is the rule of $\underset{\sim}{F}$-elimination,<7> which is:

(Fe) If X: 1xy and Y: y, then (X Y): x.

An application of rule (Fe) is known as an $\underset{\sim}{F}$-inference and it can be written as:

$$\frac{\text{X: 1xy} \qquad \text{Y: y}}{\text{(X Y): x.}}$$

The premise on the left (that is, X: 1xy) is the major premise and the one on the right (that is, Y: y) is the minor premise. A deduction all of whose inferences are $\underset{\sim}{F}$-inferences is called an $\underset{\sim}{F}$-deduction. When such a deduction is arranged as a genealogical tree, this is done in such a way that the major premise of each constituent $\underset{\sim}{F}$-inference is always on the left (as in the

---

<6> Clearly, if I was trying to devise a proper combinatory grammar for English, then I would have to include more information in the lexicon about each expression in it.

<7> The reason why it is called $\underset{\sim}{F}$-elimination is that what I write as '1xy' Curry writes as 'Fyx'. Note that the order of 'x' and 'y' is different in the two notations.

diagram).<8>

If L is a lexicon, that is to say, a set of propositions of the form X: x, then a proposition Y: y is said to be an F-consequence of L if it is the conclusion of an F-deduction all of whose premises are in L. We also say that Y: y is F-deducible from L and this can be written as:<9>

L ⊢ Y: y.

An ob Z is syntactically coherent if a proposition of the form Z: z can be derived from the lexicon, for some z. And those obs which belong to the category P are the combinatory deep structures of language.

As an example of these ideas I shall show how the deep structure of the sentence

(10) Joseph drinks and Joseph smokes,

can be derived. Rather than writing F-deductions as genealogical trees, however, I will write them in a linear fashion.

<8> This account of F-deductions is based on the discussion in Curry and Feys, Combinatory Logic, pp.280-281.

<9> The use of the turnstile symbol '⊢' here is deliberate, because the similarity between F-deductions and natural deduction tree-proofs involving only implication can be shown to be a formal analogy. See Appendix 3 for details.

(11) 'Joseph': N lexicon

(12) 'drinks': 1PN lexicon

(13) 'drinks' 'Joseph': P Fe 12 11.

Lines (11) and (12) are obtained from the lexicon and (13) follows from them by rule (Fe). Similarly, it can be shown that:

(14) 'smokes' 'Joseph': P.

From (13) and (14) together with the fact that 'and' is of category 11PPP we can conclude (by using rule (Fe) twice) that:

(15) 'and' ('drinks' 'Joseph') ('smokes' 'Joseph'): P.

And (15) is the deep structure of (10).

As propositions involving combinators occur in the lexicon, combinators can figure in the deep structures. In these cases the generic category of the combinator involved has to be instantiated to a particular category. An example of a derivation involving a combinator is as follows:

(16) $\underset{\sim}{\Phi}$: 1111PN1PN1PN11PPP lexicon

(17) 'and': 11PPP lexicon

(18) $\underset{\sim}{\Phi}$ 'and': 111PN1PN1PN Fe 16 17

(19) 'drinks': 1PN lexicon

(20) $\underset{\sim}{\Phi}$ 'and' 'drinks': 11PN1PN Fe 18 19

(21) 'smokes': 1PN lexicon

(22) $\underset{\sim}{\Phi}$ 'and' 'drinks' 'smokes': 1PN Fe 20 21

(23) 'Joseph': N lexicon

(24) $\underset{\sim}{\Phi}$ 'and' 'drinks' 'smokes' 'Joseph': P Fe 22 23.

The ob in (24) is the deep structure of

(25) Joseph drinks and smokes.

Intuitively, one thinks that this must be related to the ob in (15) and there is, in fact, a connection. They are convertible:

(26) 'and' ('drinks' 'Joseph') ('smokes' 'Joseph')
= $\underset{\sim}{\Phi}$ 'and' 'drinks' 'smokes' 'Joseph'.

In fact, it is generally true that if two obs X and Y are convertible and X has the functional character x, then Y also has the functional character x.<10>

The Transformational Component What I have described so far is the base component of a combinatory grammar. From the lexicon

---

<10> See Curry and Feys, Combinatory Logic, pp.279ff.

by means of rule (Fe) we can derive propositions that tell us that a particular ob belongs to a definite syntactic category. Those obs which belong to category P are the deep structures of language, but in order to arrive at acceptable surface structures the combinatory grammar has to be supplemented with a set of transformations. These would transform ('snores' 'Jack') into 'Jack snores' and they would transform the ob involved in (15) to 'Joseph drinks and Joseph smokes'. If my goal here was to devise a comprehensive grammar for English, then obviously I would have to say a great deal about the nature of such transformations; but as my goal is just to show how combination problems can be solved in a combinatory grammar, I can get away without saying anything more about them. This is because - in a combinatory grammar - combination problems are solved in the base component and not by means of transformations. In fact, many linguistic phenomena that require transformations to account for them in a transformational-generative grammar can be taken care of in the base component of a combinatory grammar. An obvious example is the passive transformation. This is not required in a combinatory grammar, because of the presence of the combinator C. For example, the surface structure of ('loves' 'Jack' 'Maxine') is 'Jack loves Maxine', but that of (C 'loves' 'Jack' 'Maxine') is 'Jack is loved by Maxine'.

Semantics For the same reason as given in the previous Subsection I will say very little about the semantics of a combinatory grammar. A knowledge of such a semantics is not necessary in order to understand the solutions of combination

problems. I just want to say here that the sort of semantics involved would be a denotational semantics, in which every ob is assigned an entity of some sort as its denotation.<11> Thus, the ob 'Socrates' would be assigned the person Socrates as its denotation and the ob 'snores' would be assigned that (complete) function which maps individuals to truth-values as its denotation. Furthermore, the denotation of every syntactically coherent (complete) expression ⌜(X Y)⌝ would be the value of the (complete) function which is the denotation of X for the argument which is the denotation of Y. The kinds of model that this would give rise to are discussed in Part V of Barendregt's The Lambda Calculus, (pp.463-553). They are quite complicated mathematical structures because it is possible to define fixed-point operators in a combinatory grammar. For example, let us define $\underset{\sim}{Y}$ as follows:

$$\underset{\sim}{Y} \hat{=} \underset{\sim}{W}\ \underset{\sim}{S}\ (\underset{\sim}{B}\ \underset{\sim}{W}\ \underset{\sim}{B}).$$

This has the reduction property:

$$\underset{\sim}{Y}\ f \dashrightarrow f\ (\underset{\sim}{Y}\ f).$$

$\underset{\sim}{Y}$ is called a fixed-point operator because the value of the function f for the argument $(\underset{\sim}{Y}\ f)$ is the same as that very

---

<11> This applies to the base component of the grammar.

argument.<12> $\underset{\sim}{Y}$ is needed in combinatory logic in order to define recursive functions. The only remaining point that I want to make is that if two obs are convertible, then they have the same value in every model.

Syntactic Coherence of Specific Examples Justified

In Chapter 5 I gave a number of examples of combinations of expressions which are intuitively syntactically coherent, but which cannot be shown to be so by means of the basic multiplying-out rule alone. As I have criticised the recursive rules of Geach and Potts, I am under an obligation to show how those structures can be justified with the help of some of the combinators and this I propose to do in this Section.

Non-Propositional Negations One of the combinatory deep structures of the proposition 'everyone does not snore' is:

(27) 'everyone' ($\underset{\sim}{B}$ 'not' 'snores').

The generic category of $\underset{\sim}{B}$ is 111xy1zy1xz. In this context it has to be instantiated to 111PN1PN1PP and then it is easy to show that the subcomponent ($\underset{\sim}{B}$ 'not' 'snores') of (27) is syntactically

---

<12> It is possible to define fixed-point operators in the formal system of Frege's Grundgesetze. See the Section "The Deriveability of Fixed-Points" in Appendix 2 for more details.

coherent. We reason as follows:

(28) $\underset{\sim}{B}$: 111PN1PN1PP	lexicon
(29) 'not': 1PP	lexicon
(30) $\underset{\sim}{B}$ 'not': 11PN1PN	Fe 28 29
(31) 'snores': 1PN	lexicon
(32) $\underset{\sim}{B}$ 'not' 'snores': 1PN	Fe 30 31.

The expression 'everyone' is of category 1P1PN and so the category of the combination (27) is P.

It is also easy to show how negation combines with a two-place predicate, say, 'killed', to form a syntactically coherent unit. The combination in this case is:

$\underset{\sim}{B}$ ($\underset{\sim}{B}$ 'not') 'killed'.

And this can be shown to be syntactically coherent as follows:

(33) $\underset{\sim}{B}$: 1111PNN11PNN11PN1PN	lexicon
(34) $\underset{\sim}{B}$ ($\underset{\sim}{B}$ 'not'): 111PNN11PNN	Fe 33 30
(35) 'killed': 11PNN	lexicon
(36) $\underset{\sim}{B}$ (B 'not') 'killed': 11PNN	Fe 34 35.

In order to show how to construct other kinds of non-propositional negation it will be necessary to introduce some

more terminology.<13> The composite product of two obs X and Y is written X.Y and is defined in this way:

$$X.Y \hat{=} \underset{\sim}{B}\ X\ Y.$$

Powers of obs are defined recursively as follows:

$$X^{0} = I,$$
$$X^{1} = X,$$
$$X^{i} = X.X^{i-1}, \qquad \text{for } i > 1.$$

If we think of a proposition as a zero-place predicate, then ($\underset{\sim}{B}^{i}$ 'not') is the operator which negates an i-place predicate. Here i can be any non-negative whole number.

Double Negation Two negations form a syntactic unit of category 1PP when combined with the combinator $\underset{\sim}{B}$ as in ($\underset{\sim}{B}$ 'not' 'not'). Here the category of B has to be taken to be 111PP1PP1PP.<14>

Quantifier and Relational Sign In the inference of 'someone loves everyone' from 'Maxine loves everyone' the expression 'loves everyone' has to be syntactically coherent. One of the deep structures underlying 'Maxine loves everyone' is:

<13> See Curry and Feys, Combinatory Logic, pp.163-165 for both these definitions.

<14> See also Curry and Feys, Combinatory Logic, p.164.

(37) B 'everyone' 'loves' 'Maxine'.

This can be shown to be a proposition in this way:

(38) B: 111PN11PNN1P1PN	lexicon
(39) 'everyone': 1P1PN	lexicon
(40) B 'everyone': 11PN11PNN	Fe 38 39
(41) 'loves': 11PNN	lexicon
(42) B 'everyone' 'loves': 1PN	Fe 40 41
(43) 'Maxine': N	lexicon
(44) B 'everyone' 'loves' 'Maxine': P	Fe 42 43.

Line (42) shows that (B 'everyone' 'loves') is a syntactically coherent subcomponent of (37).

One of the deep structures of 'someone loves everyone' is

(45) 'someone' (B 'everyone' 'loves').

And (B 'everyone' 'loves') is also a syntactically coherent part of this.

In these examples it should be realised that the surface structure of ('loves' 'Maxine' 'Jack') would be 'Maxine loves Jack'. The surface structure of ('loves' 'Jack' 'Maxine') would be, however, 'Jack loves Maxine'. As already mentioned, the combinator C corresponds to the passive transformation in a transformational-generative grammar, so a deep structure of 'someone is loved by everyone' would be

(46) 'someone' (B 'everyone' (C 'loves')).

Internal Conjunction Both Geach's and Potts's solution to the problem of conjunction (and other binary truth-functional connectives) joining expressions smaller than propositions can be handled within a combinatory grammar. The example I considered in Chapter 5 was 'Joseph drinks and smokes'. Geach's solution is represented by the deep structure:

(47) S (B 'and' 'drinks') 'smokes' 'Joseph'.

To show that this is syntactically coherent we reason as follows:

(48) B: 1111PPN1PN11PPP	lexicon
(49) 'and': 11PPP,	lexicon
(50) B 'and': 111PPN1PN	Fe 48 49
(51) 'drinks': 1PN	lexicon
(52) B 'and' 'drinks': 11PPN	Fe 50 51
(53) S: 111PN1PN11PPN	lexicon
(54) S (B 'and' 'drinks'): 11PN1PN	Fe 53 52
(55) 'smokes': 1PN	lexicon
(56) S (B 'and' 'drinks') 'smokes': 1PN	Fe 54 55
(57) 'Joseph': N	lexicon
(58) S (B 'and' 'drinks') 'smokes' 'Joseph': P	Fe 56 57.

This is not exactly the same as Geach's solution because there is

no category in a combinatory grammar corresponding to the category that Geach uses, namely, 2PPP. The reason I say that this is the combinatory analogue of Geach's solution is that it shows how conjunction combines with two predicates (and some combinators) to yield a predicate. This is apparent in line (56).

Potts's solution - in which conjunction combines with two predicates to yield a two-place relational sign - is taken care of by means of the combination:

(59) 'he' (C (B B (B 'and' 'drinks')) 'smokes') 'Joseph'.

And this can be shown to be syntactically coherent in a similar way to the previous examples. I say that this corresponds to Potts's solution because in the course of justifying that (59) is a proposition, we would show that:

(60) C (B B (B 'and' 'drinks')) 'smokes'

is of category 11PNN. Thus, in this case conjunction combines with two predicates (and a few combinators) to form a relational expression. The pronoun 'he', of category 11PN11PNN, then turns this into a predicate. Understood in this way the pronoun 'he' is just the combinator W instantiated in a particular way. This has the generic category 11xy11xyy and substituting 'P' for 'x' and 'N' for 'y' yields the category of Potts's pronoun.

I derived the deep structures (47) and (59) of the proposition 'Joseph drinks and smokes' in such a way as to mimic the solutions given by Geach and Potts, respectively, to the combination problem involved here. It is possible, however, to give a much neater solution if we use the combinator $\underset{\sim}{\Phi}$. In this case another deep structure of this proposition is:

(61) $\underset{\sim}{\Phi}$ 'and' 'smokes' 'drinks' 'Joseph'.

This was shown to be a proposition earlier by the derivation (16) to (24).

It should be noted that each of the obs (47), (59) and (61) can be converted to both of the others. This is proved by observing that each of them reduces to the the same thing, namely:

(62) 'and' ('drinks' 'Joseph') ('smokes' 'Joseph').

Thus, in a combinatory grammar each of the propositions:

Joseph drinks and Joseph smokes,
Joseph drinks and he smokes,
Joseph drinks and smokes,

has a different deep structure, but the various deep structures involved are all interconvertible. Thus, they are all assigned the same meaning in the denotational semantics of the language.

'Jack loves and hates Jill' A combinatory deep structure of this proposition is:

(63) $\underset{\sim}{\Phi}^2$ 'and' 'loves' 'hates' 'Jack' 'Jill'.

It is easy to show that (63) is a proposition if we realise that here $\underset{\sim}{\Phi}^2$ is of category 11111P11PPP11PNN11PNNNN. It is possible to give a deep structure of this proposition which mimics the solution I gave in Chapter 5 - that is to say, a solution which makes use of the pronouns 'he' and 'her', both of category 11PN11PNN - but it is horrendously complicated, so I will not give it here.

'Jill suspects and fears that Jack loves Maxine' One of the combinatory deep structure of this proposition is:

(64) $\underset{\sim}{\Phi}^2$ 'and' 'suspects' 'fears' 'Jack loves Maxine' 'Jill'.

Here the category of $\underset{\sim}{\Phi}^2$ is 11111P11PPP11PNP11PNPNP. As in the previous example, it is possible to give a deep structure that mimics the solution I gave in the previous Chapter. In this case it makes use of a pronoun 'she' and a pro-proposition of category 11PP11PPP, but it is also very complicated.

Quantifier and Negation There is no problem in showing how a quantifier and negation combine together to form a syntactically coherent expression. In order to show that 'there is something not everyone is' is a proposition, we argue as follows:

(65) $\underset{\sim}{B}$: 111P1PN1P1PN1PP lexicon

(66) 'not': 1PP lexicon

(67) $\underset{\sim}{B}$ 'not': 11P1PN1P1PN Fe 65 66

(68) 'everyone' 1P1PN lexicon

(69) $\underset{\sim}{B}$ 'not' 'everyone': 1P1PN Fe 67 68

(70) 'there is something': 1P1P1PN lexicon

(71) 'there is something' ($\underset{\sim}{B}$ 'not' everyone'): P Fe 70 69.

And in order to show that 'there is something someone is not' is a proposition we argue in this way:

(72) $\underset{\sim}{B}^2$: 1111P1PN1PP111PN1PN1PP1P1PN lexicon

(73) 'someone': 1P1PN lexicon

(74) $\underset{\sim}{B}^2$ 'someone': 111P1PN1PP111PN1PN1PP Fe 72 73

(75) $\underset{\sim}{B}$: 111PN1PN1PP lexicon

(76) $\underset{\sim}{B}^2$ 'someone' $\underset{\sim}{B}$: 11P1PN1PP Fe 74 75

(77) 'not': 1PP lexicon

(78) $\underset{\sim}{B}^2$ 'someone' $\underset{\sim}{B}$ 'not': 1P1PN Fe 76 77

(79) 'there is something': 1P1P1PN lexicon

(80) 'there is something' ($\underset{\sim}{B}^2$ 'someone' $\underset{\sim}{B}$ 'not'): P Fe 79 78.

Two Quantifiers Combining A deep structure of 'there is some relation that everyone stands in to someone' is:<15>

(81) 'there is some relation' ($\underset{\sim}{B}^3$ 'everyone' $\underset{\sim}{B}^2$ 'someone' $\underset{\sim}{C}$).

Relative Pronouns Potts's relative pronoun of category 111PN1PN1PN is slightly more difficult to define in combinatory logic, but the following suffices:

'who' = $\underset{\sim}{B}$ ($\underset{\sim}{B}$ 'he') (($\underset{\sim}{B}$ $\underset{\sim}{C}$)($\underset{\sim}{B}$ ($\underset{\sim}{B}$ $\underset{\sim}{B}$)($\underset{\sim}{B}$ 'and'))).

A deep structure of the proposition 'Socrates, who was a philosopher, lived in Athens' is, therefore:

'who' 'was a philosopher' 'lived in Athens' 'Socrates'.

The Pro-Verb In Chapter 5 I considered the following proposition, which is there labelled (39):

(82) If all of the Asian ambassadors refuse to come, then so will some of the African ambassadors.

A deep structure of this is the following:

<15> See also Curry, Hindley and Seldin, Combinatory Logic, p.429.

(83) W (C (B B (B 'if' $\prod$)) $\sum$) 'refuse to come'.

Here '$\prod$' is a symbol for 'all of the Asian ambassadors' and '$\sum$' for 'some of the African ambassadors'. Here the combinator $\underset{\sim}{W}$ has to be assigned the category 11P1PN11P1PN1PN and this is the same category as Potts's pro-verb.

The deep structure (83) was derived in order to mimic Potts's account of this proposition, but it is possible to give a much neater solution in this way:

(84) $\underset{\sim}{\Phi}$ 'if' $\prod$ $\sum$ 'refuse to come'.

One reason for discussing this example in the previous Chapter was to provide a justification for sometimes treating proper names as being of category 1P1PN. There is no need to do this in a combinatory grammar as a deep structure of the proposition 'if Maxine refuses to come, then so will Jack' is given by:

(85) $\underset{\sim}{\Psi}$ 'if' 'refuses to come' 'Maxine' 'Jack'.

But it is also possible in a combinatory grammar to define an ob which turns a proper name into an operator of category 1P1PN. I will write this as 'up'. For example, although 'Socrates' is of category N, (up 'Socrates') is of category 1P1PN. The following definition suffices:

$$up \hat{=} \underset{\sim}{C}\ \underset{\sim}{I}.$$

Using this new operator, we can derive another deep structure of 'if Maxine refuses to come, then so will Jack', but this time one that mimics Potts's solution. The deep structure in question is (83), where this time $\overline{\Pi}$ is (up 'Maxine') and $\overline{\Sigma}$ is (up 'Jack'). In both of these cases the new operator is of category 11P1PNN.

In Chapter 5 I showed that a proper name like 'Socrates' has analogues on every even level in the hierarchy of syntactic categories. Each of these has a counterpart in a combinatory grammar. If we think of a proper name as being of level 0, then the analogue of 'Socrates' on level 2.i is:

$$up^{\dot{\imath}}\ \text{'Socrates'}.$$

This account is neater than that which I gave for a categorial grammar. In particular, the relation of all the analogues is more neatly and precisely spelled out.

I have now shown how all the examples of combination problems that I discussed in Chapter 5 can be solved in the framework of a combinatory grammar. I will, therefore, now go on to discuss syntactic completion analysis in this type of grammar.

Combinatory Analogue of Syntactic Completion Analysis

In Chapter 5 I devoted a lot of space to discussing the principle of syntactic completion analysis. One of the most serious shortcomings of the Geach-Potts style of solution to the combination problems is that the rules that they introduce result in a system in which it is impossible to use the method of syntactic completion analysis in order to determine the categories of expressions. If there was no analogue of this principle in the theory of functionality, then that would severely limit its usefulness, but fortunately there is. It is the rule of F-introduction. In order to explain how it works I have to introduce some more ideas from combinatory logic.

Certain set of combinators possess the property of combinatorial completeness. There is only one mode of combination used in combinatory logic and that is functional application which is symbolised by juxtaposition. Thus, if f is a function and b is its argument, then the value of f for the argument b is (f b). This is also known as a combination of f and b, and in general if X' has been obtained by repeated uses of functional application out of $b_1,\ldots,b_n$, then X' is said to be a combination of $b_1,\ldots,b_n$.

Let X' be a combination of the constants (none of which are combinators) $b_1,\ldots,b_n$, and combinators from the set G*. The set G* is said to be combinatorially complete iff there exists a combination of combinators from the set G*, say X, such that:

(86) X b_1 $b_2 \ldots b_n$ = X'.

Many sets possess this property, I shall here consider the set G* consisting of the combinators $\underset{\sim}{K}$, $\underset{\sim}{I}$, $\underset{\sim}{B}$, $\underset{\sim}{C}$ and $\underset{\sim}{S}$.<16> To prove that the set G* is combinatorially complete it is obviously sufficient to prove that if X' is a combination of b and the elements of G*, then there exists a combination X of the elements of G* (hence X does not contain b) such that:

(87) X b = X'.

This is proved by giving an algorithm ALG such that given any combination X', applying ALG to X' yields X. The result of applying the algorithm ALG to X' is [b] X' and the process is known as bracket abstraction. One such algorithm is the following:

<16> I shall not attempt to fill in all the details of the proof that G* is combinatorially complete. The reader is referred to Chapter 6 of Curry and Feys, Combinatory Logic and Section C of Chapter 11 of Curry, Hindley and Seldin, Combinatory Logic for a more detailed proof.

(a) [b] X = $\underset{\sim}{K}$ X,

(b) [b] b = $\underset{\sim}{I}$,

(c) [b] X b = X,

(d) [b] Y Z' = $\underset{\sim}{B}$ Y Z,

(e) [b] Y' Z = $\underset{\sim}{C}$ Y Z,

(f) [b] Y' Z' = $\underset{\sim}{S}$ Y Z.

Here it is to be understood that Y = [b] Y' and Z = [b] Z' and that b occurs neither in Y nor Z. Each of the six clauses (a) to (f) specifies a replacement for a component of the form [b] X. Given an arbitrary combination involving b we try to apply one of the clauses to it, always trying the clauses in alphabetical order. An example should make this algorithm's operation clear. Consider the combination (b b ($\underset{\sim}{B}$ b)). The only clause we can apply to this is (f), giving us $\underset{\sim}{S}$ ([b] b b) ([b] $\underset{\sim}{B}$ b). We now proceed from left to right. The component [b] b b of the partial result is $\underset{\sim}{S}$ ([b] b) ([b] b), by clause (f) again, and as [b] b is $\underset{\sim}{I}$, by clause (b), this is equivalent to $\underset{\sim}{S}$ $\underset{\sim}{I}$ $\underset{\sim}{I}$. The component [b] $\underset{\sim}{B}$ b of the partial result is simply $\underset{\sim}{B}$, by clause (c). Putting all this together we have that:

[b] b b ($\underset{\sim}{B}$ b) = $\underset{\sim}{S}$ ($\underset{\sim}{S}$ $\underset{\sim}{I}$ $\underset{\sim}{I}$) $\underset{\sim}{B}$.

This can be presented in the following way:

$$[b]\ b\ b\ (\underset{\sim}{B}\ b) = \underset{\sim}{S}\ ([b]\ b\ b)\ ([b]\ \underset{\sim}{B}\ b) \qquad (f),$$

$$= \underset{\sim}{S}\ (\underset{\sim}{S}\ ([b]\ b)\ ([b]\ b)\ ([b]\ \underset{\sim}{B}\ b) \qquad (f),$$

$$= \underset{\sim}{S}\ (\underset{\sim}{S}\ \underset{\sim}{I}\ ([b]\ b)\ ([b]\ \underset{\sim}{B}\ b) \qquad (b),$$

$$= \underset{\sim}{S}\ (\underset{\sim}{S}\ \underset{\sim}{I}\ \underset{\sim}{I})\ ([b]\ \underset{\sim}{B}\ b) \qquad (b),$$

$$= \underset{\sim}{S}\ (\underset{\sim}{S}\ \underset{\sim}{I}\ \underset{\sim}{I})\ \underset{\sim}{B} \qquad (c).$$

The analogue of bracket abstraction in the lambda calculus is lambda abstraction. The main difference, however, is that bracket abstraction is not a variable-binding operation.<17>

So far the only rule from the theory of functionality that I have used is that of $\underset{\sim}{F}$-elimination:

(Fe) If X: 1xy and Y: y, then (X Y): x.

Corresponding to the procedure of syntactic completion analysis in a system of categorial grammar is the rule of F-introduction in the theory of functionality:

(Fi) If L, X: x $\vdash$ Y :y, and if X: x does not occur in L, then L $\vdash$ [X] Y :1yx.

In other words, if Y: y is an $\underset{\sim}{F}$-consequence of the lexicon L together with the proposition X: x and X: x does not occur in L, then there exists an $\underset{\sim}{F}$-deduction of [X] Y: 1yx from L. It is further assumed that X is simple, that is to say, it is not

---

<17> See Chapter 7 of Barendregt's The Lambda Calculus for a precise statement of the nature of the analogy for various bracket abstraction algorithms.

constructed from any other expressions.

To illustrate the use of this rule consider the F-deduction:

$$\dfrac{\dfrac{\text{'snores': 1PN} \qquad \text{'Tom': N}}{\text{'not': 1PP} \qquad \text{'snores' 'Tom': P}}}{\text{'not' ('snores' 'Tom'): P.}}$$

Making use of bracket abstraction we have that

['Tom'] 'not' ('snores' 'Tom') = $\underset{\sim}{B}$ 'not 'snores',

and the rule of $\underset{\sim}{F}$-introduction tells us that ($\underset{\sim}{B}$ 'not' 'snores') is of category 1PN. This can also be shown by constructing an explicit $\underset{\sim}{F}$-deduction:

$$\dfrac{\dfrac{\underset{\sim}{B}\text{: 111PN1PN1PP} \qquad \text{'not': 1PP}}{\underset{\sim}{B}\text{ 'not': 11PN1PN} \qquad \text{'snores': 1PN}}}{(\underset{\sim}{B}\text{ 'not') 'snores': 1PN.}}$$

In Chapter 5 I explained how the principle of syntactic completion analysis is used by categorial grammarians in extending the lexicon of a categorial grammar. Much the same can be said about its usefulness in a combinatory grammar; since, such a grammar is basically a categorial grammar with only the basic multiplying-out rule, but with combinators added. Rather

than repeating what I said there, I will conclude this Section by discussing two other issues related to bracket abstraction. The first of these is to do with Dummett's claims (a) that in natural language corresponding to every sentence containing more than one occurrence of some proper name there exists an equipollent sentence containing just one occurrence of that proper name, and (b) that in natural language corresponding to every sentence containing several different proper names there exists an equipollent sentence in which those proper names occur in any permutation of their original order. The second issue I want to discuss concerns constant functions, that is to say, functions which return the same value no matter what their argument is.

Dummett's Claims about Natural Language Dummett says that in order to be capable of unambiguously expressing what can be formulated in a first-order language by means of quantifiers and variables natural language has to have the property that there exists,

> for any sentence containing any number of occurrences of each of any number of proper names, an equivalent sentence containing only one occurrence of each of those names, in any arbitrary specified order. (Frege, p.14.)

(The two claims that I said Dummett makes about natural language are just particular instances of this more general claim.) Dummett goes on to say that he is not certain if natural language possesses this property he claims for it, but it is a property that combinatory logic has. Consideration of the bracket

abstraction algorithm establishes this. In proposition (86) the constant obs b_1 , ..., b_n can occur in any order in X' and each of them can occur any number of times, but in the ob on the left hand side of the equation each of them occurs only once and they occur in the order $b_1, b_2, \ldots, b_n$.

As an example, I shall consider the ob:

(87) 'and' ('loves' 'Jill' 'Jack') ('loves' 'Jack' 'Maxine').

This is a deep structure of:

(88) Jill loves Jack and Jack loves Maxine.

Applying the bracket abstraction algorithm ALG to (87), and abstracting on the constant 'Jack', yields the ob:

(89) $\underset{\sim}{S}$ ($\underset{\sim}{B}$ 'and' ('loves' 'Jill')) ($\underset{\sim}{C}$ 'loves' 'Maxine').

We thus have that:

(90) ['Jack'] (87) = (89) 'Jack'.

The usefulness of doing this, as Dummett points out, is to do with adding quantifiers. We can prefix the existential quantifier to (89) to give us:<18>

<18> I am assuming here that the universe of discourse is restricted to people.

(91) 'someone' (S̃ (B̃ 'and' ('loves' 'Jill'))
(C̃ 'loves' 'Maxine')).

And this could be expressed in natural language as:

(92) Jill loves someone who loves Maxine.

It should be noted that (91) does not reduce to anything except itself. Furthermore, it is not convertible with anything that might serve as a deep structure of:

(93) Jill loves someone and someone loves Maxine.

This has very different deep structures from (91). One of them is:

(94) 'and' ('someone' ('loves' 'Jill'))
('someone' (C̃ 'loves 'Maxine')).

Although it is possible to abstract 'someone' from (94), the resulting ob will not be convertible with (89). And, hence, prefixing 'someone' to it will not give us an ob convertible with (91). Thus, (94) and (91) are not convertible.

Constant Functions There is nothing in the account of bracket abstraction that I have given that the constant abstracted from a combination actually occurs in that combination. It is, therefore, possible to abstract 'Jack' from ('sings' 'Maxine'):

(95) ['Jack'] ('sings' 'Maxine') = $\underset{\sim}{K}$ ('sings' 'Maxine').

The function ($\underset{\sim}{K}$ ('sings' 'Maxine')) returns the same value, namely, ('sings' 'Maxine'), for whatever is taken as its argument.

Frege has great difficulty in accommodating such constant functions in his scheme of things. He suggests, for example, representing the numerical function that returns the number 2 no matter what is given it as argument as:<19>

(96) 2 + ξ - ζ.

The difficulty with this is that it is not easy to see how it can be extended to other cases. Not all mathematical systems have inverse operators. It would, for example, be impossible to construct constant functions in Frege's way in any algebraic system which only had the structure of a monoid. In particular, it is impossible to construct an unsaturated expression or to derive a linguistic function (as I explained in Chapters 1 and 2) which yields a constant expression when applied to all the expressions in some syntactic category or other. '2 + ξ - ζ' obviously will not work, because its value for the argument '3' is '2 + 3 - 3' and for the argument '7' is '2 + 7 - 7' and these are different expressions.

<19> "Function and Concept", p.8.

Pronouns and Variables

In this Chapter I have presented the outline of a grammar for English which generates deep structures in which variables do not occur and which treats pronouns in surface structures as being derived from occurrences of the combinator $\underset{\sim}{W}$ in the corresponding deep structures. I, therefore, feel under an obligation to discuss Geach's view that indefinite and anaphoric pronouns

> are closely connected with the modern quantifier-notation. The indefinite pronouns would be a natural means of rendering quantifiers into the vernacular - "anything" or "everything" being used for the universal, and "something" for the existential, quantifier; and pronouns with antecedents strictly correspond to the letters used as bound variables. (Reference and Generality, p.136.)

As an example of the correspondence he considers the following two propositions:<20>

(97) (Ax)((Ay)(y hurts x => x hurts y) => x hurts x),

(98) If there is anybody who, if there is anybody who hurts him, hurts him in turn: then he hurts himself.

(In this and the other examples in this Section I assume that the universe of discourse is restricted to people.) He adds that the pieces of (97) and (98) 'stand in strict mutual correspondence' (ibid., p.137) and he spells this out in considerable detail.

<20> I have altered the numbering of these propositions to agree with mine and I have changed the symbols for some of the logical constants for the same reason.

But just because this example illustrates Geach's general claim so well, it does not follow that that claim is correct. An examination of a wider class of examples suggests that it breaks down in many cases.

(a) As already mentioned Geach believes that 'pronouns whose antecedents are applicatival phrases correspond strictly in their syntax to variables bound by quantifiers'.<21> Given an open sentence like 'Fx & Gx' it is possible to prefix it with any quantifier and the result is well-formed, but given a phrase like 'smokes and he drinks' the result of prefixing this with an applicatival phrase often results in strings of words that are not sentences. For example, none of the following are well-formed sentences of English:

(99) No one smokes and he drinks,

(100) Every man smokes and he drinks,

(101) Almost every man smokes and he drinks.

If pronouns with antecedents really corresponded strictly to bound variables, then all of these three sentences would be well-formed. But as they are not, the correspondence cannot be as strict as Geach thinks.

(b) Here I am going to consider the case of ambiguity. The proposition:

<21> "Quine's Syntactical Insights", p.148. An applicative is an expression like 'some', 'any', 'each', 'just one' and 'almost every'. And an applicatival phrase is an expression formed by appending a substantival general term to an applicative.

(102) Some man loves a man and he is very happy,

could be translated into the quantifier-variable notation in either of the following two ways:<22>

(103) (E!x ¦ man)(E!y ¦ man)(x loves y & x is very happy),

(104) (E!x ¦ man)(E!y ¦ man)(x loves y & y is very happy).

Thus, whereas it is impossible not to know what the antecedent of a bound variable is, it is possible for a pronoun to have one or other of two preceding indefinite pronouns as its antecedent. It is likely that the context in which (102) occurs would disambiguate it, but you do not need to know anything about the context in which either (103) or (104) occurs to know which man is happy. Geach cannot reply to this by saying that (103) and (104) are two of the possible deep structures of (102), because he does not distinguish between deep and surface structure. This example again shows that there are significant differences between pronouns with antecedents and bound variables.

(c) As well as saying that pronouns with antecedents that are applicatival phrases correspond to bound variables, Geach also claims that pronouns whose antecedents are proper names so correspond as well.<23> He would, thus, translate:

<22> I use unique restricted existential quantifiers here rather than normal ones. Nothing in my argument would be altered by using the other type.

<23> My source for this is Evans in "Pronouns, Quantifiers, and Relative Clauses (I)", p.468.

(105) Jack owns a DX7 but he plays badly,

into

(106) (Ax)(x = Jack => x owns a DX7 & x plays badly).

But here the correspondence between pronouns and variables is far from being strict as Geach claims. There are four bound variables in (106) and a proper name, whereas there is only a single pronoun in (105). Furthermore, the antecedent of 'he' in (105) is 'Jack', but the antecedent of all but the first occurrence of the variable 'x' in (106) is the quantifier '(Ax)' and not 'Jack'.

(d) The fourth difference between pronouns with antecedents and bound variables that I want to mention here concerns what might be called gapping. By this I mean that quite often in English it is possible to delete a pronoun from a sentence and still be left with an equipollent sentence, as in these examples:

(107) Jack schemes and he deceives,

(108) Jack schemes and deceives.

As explained in the Subsection "Internal Conjunction" earlier in this Chapter, propositions like these have different, though convertible, deep structures. But what can Geach say about them? Presumably he would see (107) as corresponding to:

(109) (Ax)(x = Jack => x schemes and x deceives).

But if he sees (108) as corresponding to:

(110) (Ax)(x = Jack => x (schemes and deceives)),

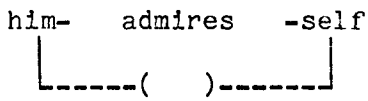
then he owes us an account of the internal conjunction that is being used here. In the last Chapter I showed that his solution in the framework of a categorial grammar fails. So, here is a common type of sentence which does not correspond straightforwardly to anything in the predicate calculus.

As well as there being the differences between pronouns with antecedents and bound variables that I have just mentioned, it is possible to use Geach's account of reflexive pronouns against him here. Reflexive pronouns are one species of pronouns with antecedents, as the following sentences show:

(111) Every man admires himself,

(112) Jack admires himself.

Concerning the way in which reflexive pronouns function, Geach writes that:<24>

> I maintain ... that it is wrong to regard "himself" as turning a two-place into a one-place predicable by filling up one place; rather, a reflexive pronoun fills up both places of the two-place predicable into which it is inserted, but itself has an incompleteness tantamount to there being one empty place - an incompleteness that appears in grammar as the need of the pronoun for an antecedent. In passing from "___ admires ..." to "___ admires himself"; the real logical structure is better brought out by this sort of diagram:
>
> 
>
>
> where the place between the parentheses is to be filled with the antecedent of "himself". (Reference and Generality, pp.162-163.)

But this account describes what the combinator W does more accurately than what bound variables do. A deep structure of the proposition 'Jack admires himself' is:

(113) W 'admires' 'Jack'.

And a deep structure of 'every man admires himself' is:

<24> For Geach a one-place predicable is what I have been calling a predicate. He reserves the term 'predicate' for a predicable that is actually used in a sentence by being attached to a subject. He illustrates the difference by means of these two sentences: 'Jim broke the bank at Monte Carlo' and 'the man who broke the bank at Monte Carlo died in misery'. He says that here 'we have two occurrences of the same predicable, but only in the first sentence is it actually a predicate attached to the subject "Jim".' (Reference and Generality, p.50.)

(114) 'every man' ($\underset{\sim}{W}$ 'admires').

In both of these deep structures the subcomponent ($\underset{\sim}{W}$ 'admires') is of category 1PN and 'admires' is of category 11PNN, therefore, $\underset{\sim}{W}$ is of category 11PN11PNN.<25> The role of $\underset{\sim}{W}$ here is precisely as Geach describes the role of 'himself'. It fills up both argument-places of 'admires' and yet the resulting combination is still a function from singular terms to propositions.

Dummett shares Geach's views about reflexive pronouns (as I said in the Section "Step-By-Step Construction" in Chapter 4). Potts, however, takes them even further in the direction of combinatory logic, since he argues that the function of all anaphoric pronouns 'is to reduce by one the polyadicity of the expressions upon whose meaning they work' ("A General Theory of the Meaning of Anaphoric Pronoun", p.143). As the combinator $\underset{\sim}{W}$ is of generic category 11xy11xyy it can fulfill this function perfectly.

To end I just want to mention Evans's view that all that Geach's claim about the correspondence of pronouns with variables amounts to is that just as variables cannot be said to refer to objects neither can pronouns.<26> If that is all that there is to Geach's claim, then I agree with him entirely.

<25> In other contexts $\underset{\sim}{W}$ would be assigned a different particular category.

<26> "Pronouns, Quantifiers, and Relative Clauses (I)", p.467.

Conclusion

In this, the final Section of this thesis, I want to review a few of the things that I have achieved in the course of it.

(a) In Chapter 1 I gave an exposition of the primitive unsaturated expressions of Frege's Begriffsschrift as linguistic functions and I also showed how such an account could be extended to natural language if that was augmented with various categories of variables. In viewing the universal quantifier, for example, as a linguistic function we have to consider the category of singular terms together with the set of individual variables as forming a unified class. This is because the predicates in the class that forms the domain for the universal quantifier have to be defined in such a way that their domain contains not only singular terms, but also individual variables. Many of the problems we encounter in trying to apply Frege's views to natural language arise because natural language does not contain variables or substitutes for variables.

(b) If my exposition of unsaturated expressions as linguistic functions in Chapter 1 seemed obvious, then I am glad. This is because the acceptance of my interpretation has far-reaching implications for an understanding of Frege's philosophy. In particular, my interpretation is radically different from Dummett's. As explained in Chapter 4, in order to account for the workings of language Dummett has to distinguish between constructive saturated expressions and non-constructive

unsaturated expressions.<27> The notion of a linguistic function, however, is that of a constructive unsaturated expression and it can do all the jobs that are done by Dummett's two types of expression.

(c) In his writings Frege advocated, in effect, a simple theory of types. Its distinctive feature was that combinations of symbols whose referents did not have matching types could not be unified into a single expression. So, for example, two proper names could not be combined to form any sort of coherent expression, nor could two predicates. Frege saw this distinctive feature as a strength of his formalised language. And - as I showed in Chapter 1 - it is possible to give a coherent account of all the different sorts of unsaturatedness that occur in the Begriffsschrift. In trying to extend Frege's ideas to natural language, however, we come across a whole battery of combination problems. Natural language makes use of operators, like conjunction, which can unite expressions belonging to lots of different categories. For example, all of the following are legitimate propositions of natural language:<28>

<27> Non-constructive because they are obtained from larger complete expressions, rather than being used in their formation.

<28> For simplicity I ignore certain forms of combination involving conjunction, such as 'Maxine has to choose between Jack and smoking'.

(115) Jack snores and Jill swears.

(116) Jack smokes and drinks.

(117) Jack dislikes fat and ugly women.

(118) Jack drives slowly and carefully.

(119) Jack loves and hates Jill.

(120) Jill suspects and fears that Jack loves Maxine.

(121) Jack and Jill went up the hill.

(122) Some men and all women snore.

(123) Jack arranged the golf balls on and around the coffin.

Such trans-categorial operators cannot be accommodated in a Fregean language, that is to say, one in which every expression belongs to only one category and the unsaturatedness associated with each category is unique to that category. In such a Fregean language each of the conjunctions in the sentences (115) to (123) would have to be a distinct operator. Rather than multiplying conjunctions in this way, in this Chapter I outlined an approach involving combinatory logic which can account for the syntactic coherence of (115) to (123) and in which conjunction is assigned a single functional character. I use the phrase 'functional character' rather than 'syntactic category' because in combinatory logic there is only one unsaturated expression which makes terms out of terms.<29> But the category of terms is partitioned into various classes. Each class has an associated functional character which reveals the powers of meaningful

---

<29> In this respect it is very similar to the language I outlined at the end of Chapter 3.

combination of the expressions belonging to it. How this is to be understood was explained above in the Section "Outline of a Combinatory Grammar".

Although every expression is assigned a functional character, some of these are generic. This means that the expressions in question have greater powers of significant combination than ordinary expressions. Expressions or entities with generic functional characters are also called polymorphic operators and they accomplish what the alleged trans-categorial operators in the Geach-Potts version of categorial grammar were introduced to accomplish.

In a combinatory grammar rather than letting conjunction and similar expressions be polymorphic operators, it is more convenient only to let the combinators have generic functional characters. So, although each of the conjunctions in (115) to (123) is the same expression, the deep structures underlying those sentences involve combinators which, for example, in (116) turn conjunction from being an operator which makes a proposition out of two propositions into an operator which makes a "predicate" out of two "predicates".<30>

<30> I put 'predicate' in scare quotes because in the combinatory grammar that I have introduced predicates are considered to be complete expressions.

(d) The main conclusion of the investigations reported in this thesis is that if you begin with the idea that the way in which the expressions of a language combine should be understood by analogy with the way in which a function combines with its argument to yield its value, then the way in which this idea works out in the case of a formalised language is very different from how it works out in the case of a natural language. In the case of a formalised language - such as Frege's Begriffsschrift - it results in a system which makes use of a multiplicity of unsaturated expressions, whereas - in the natural language case - it results in a system which uses only a single unsaturated expression which makes terms out of terms. The main reason for this difference is that a large number of combination problems occur in natural language, whereas no such problems occur in first- and second-order languages. And the only satisfactory way in which to solve such problems is to make use of a combinatory grammar. Such a grammar has been sketched in this Chapter, but - obviously - it has to be much refined before it can seriously compete with rival accounts. The resulting grammar would be similar to the grammars proposed by Montague and Cresswell.<31> This is not very surprising because of the known connection between combinatory logic and the lambda calculus. A combinatory grammar, however, has one significant advantage over those grammars and that is that its base component is a context-free

---

<31> In saying this I do not mean to commit myself to the details of their approach. In particular, I disagree with many of their categorisations of natural language expressions and I have little sympathy for their employment of possible worlds semantics.

grammar, whereas the language in which the lambda calculus is expressed cannot be generated by means of a context-free grammar.<32>

<32> The phenomenon of variable-binding in general cannot be captured by means of a context-free grammar. The proof is by the result known as the Pumping Lemma or the uvwxy Theorem. This was first proved by Bar-Hillel, Perles and Shamir, in "On Formal Properties of Simple Phrase Structure Grammars".

Appendix 1: Quasi-Quotation

Throughout this thesis I use "corner" quotation marks to signify quasi-quotation. Because there exist divergent accounts of quasi-quotation in the literature I will make precise what I mean by it.<1> Let 'X' be a metalinguistic variable and 'A' a metalinguistic constant. For example, A might be 'Socrates'. When placed around an expression which is not metalinguistic, corners function in exactly the same way as ordinary quotation marks. Thus, there is no difference between ⌜Plato⌝ and 'Plato'. It is only when enclosing the complex designation of an expression containing metalinguistic components that they differ from ordinary quotation marks. In the case of metalinguistic variables and constants ⌜X⌝ and X are the same and so are ⌜A⌝ and A. In the latter case both are the same as 'Socrates'. When corners enclose an expression made up out of metalinguistic components and ordinary expressions, it is equivalent to the concatenation of the constituents individually enclosed in corners. Thus, ⌜A loves A⌝ is identical with ⌜A⌝ @ ⌜loves⌝ @ ⌜A⌝, which - in turn - is the same as A @ 'loves' @ A, where the at-sign signifies concatenation. So,

---

<1> See Geach's "Quotation and Quantification", p.206, and Quine's "Logic Based on Inclusion and Abstraction", pp.101-102, for differing accounts of quasi-quotation.

if A is 'Socrates', then ⌜A loves A⌝ is 'Socrates loves Socrates'. Metalinguistic variables can only be understood in context, but the same expansion holds. Thus, to say that the value of the linguistic function '{ loves }' for the argument X is ⌜X loves X⌝ is identical with saying that its value is ⌜X⌝ @ ⌜loves⌝ @ ⌜X⌝. And this is equivalent to saying that its value for the argument X is X @ 'loves' @ X. Thus for the argument 'Aristotle' the value of '{ loves }' is 'Aristotle loves Aristotle'.

Appendix 2: Objections to Combinatory Logic

In this Appendix I want to examine some of the philosophical objections that have been made against combinatory logic and I show that none of them can be sustained. I shall consider the following two objections:<1> (1) It is possible to derive fixed points in combinatory logic. (2) Curry's paradox is deriveable in it.

The Deriveability of Fixed Points Geach actually makes this criticism of the formal calculus that Frege developed in Grundgesetze, but it is equally applicable to combinatory logic and the lambda calculus. ("Frege", pp.149ff.)

In the Grundgesetze there is only one basic category of expressions, namely, singular terms and Frege calls these proper names. In reading this book most people make the necessary changes to translate what Frege says into a language based on two basic categories, namely, those of propositions and singular terms. In doing this is it natural to interprets Frege's notation '$\zeta^{\wedge}\zeta$' as meaning '$\zeta \in \zeta$' and the notation '$\grave{\varepsilon}\varphi(\varepsilon)$' as '$\{\varepsilon \mid \varphi(\varepsilon)\}$'. But, it is just as legitimate to interpret '$\zeta^{\wedge}\zeta$' as functional application '$(\zeta\ \zeta)$' and '$\grave{\varepsilon}\varphi(\varepsilon)$' as functional

---

<1> I am grateful to Professor Geach for drawing my attention to these.

abstraction or lambda abstraction '$\lambda\xi.\varphi(\xi)$'. In doing this we have to understand functions in a non-Fregean way as complete entities. Geach says that 'the doctrine of these functions, as stated by Frege, involves a contradiction' (ibid., p.150) and he derives the "contradiction" as follows. Recall that Frege's Basic Law V is:

$$\vdash F(x) = x^{\wedge}\grave{\alpha}\, F(\alpha).$$

Geach considers an arbitrary function $G(\xi)$ and substitutes '$G(\xi^{\wedge}\xi)$' for '$F(\xi)$' in Basic Law V giving:

$$\vdash G(x^{\wedge}x) = x^{\wedge}\grave{\alpha}\, G(\alpha^{\wedge}\alpha).$$

Then substituting '$\grave{\alpha}\, G(\alpha^{\wedge}\alpha)$' for 'x' in this gives:

$$\vdash G(\grave{\alpha}\, G(\alpha^{\wedge}\alpha)^{\wedge}\grave{\alpha}G(\alpha^{\wedge}\alpha)) = \grave{\alpha}\, G(\alpha^{\wedge}\alpha)^{\wedge}\grave{\alpha}G(\alpha^{\wedge}\alpha).$$

He concludes:

> The purport of this assertion is that, starting from any arbitrary first-level function $G(\xi)$, we can always specify in terms of it an argument ... for which the value of $G(\xi)$ is equal to that argument. But this conclusion is absurd; for we can specify any number of first-level functions whose value is never the same as the corresponding argument. Thus no second-level function will in fact exactly fulfil the role Frege assigned to his value-range function. How to mend Frege's system at this point is an important technical problem of logic; but it is of no philosophical importance how this is done. (Ibid., p.150.)

Let us consider a particular example for the function $G(\xi)$, say, $\xi + 3$. It is easy to show that there is no number which is the value of the function $\xi + 3$ when it itself is its argument. For $\xi + 3$ to have a numerical fixed-point we should have to have that for some x, $x + 3 = x$. By simple algebraic transformations, we should have for some number x, $3 = 0$. This is clearly impossible.

But in Frege's logical system every function must be defined for every object taken as argument. Informally, he argues that propositions such as

Julius Caesar + 3 = Brutus,

must make sense and be either true or false. Geach derives a contradiction from the supposition that there exist numerical functions, such as $\xi + 3$, which do not have numerical fixed-points. But that is not a proof of the impossibility of an object which is not a number being a fixed-point of this function.

Similar considerations apply in combinatory logic and the lambda calculus. In these systems it makes sense to apply any ob to any other ob. So, it is not a theorem of combinatory logic that there is a number which is the fixed-point of the function which adds 3 to its argument. But - as in the Grundgesetze - there is some strange non-numerical entity which has this property. We can capture our informal intuition about functions such as $\xi + 3$ not having numerical fixed-points by putting

conditions on the propositions we intuitively think correct or by restricting the quantifiers involved. For example, it is true in Frege's system - and in combinatory logic - that there does not exist a number x such that x + 3 = x.

Curry's Paradox Let us add the ob (which is an illative atom) $\underset{\sim}{P}$ to our system of combinatory logic. Intuitively $\underset{\sim}{P}$ is meant to represent material implication. (To make the following argument easier to follow I shall sometimes write '$\underset{\sim}{P}$ x y' as 'x => y'.) Along with the ob $\underset{\sim}{P}$ we add the following axioms:

(1) x => x,

(2) (x => (x => y)) => (x => y).

and modus ponens understood as a rule of proof. This is also called rule (P):

If |- x => y and |- x, then |- y.

It is now possible to show that the resulting system of illative combinatory logic is inconsistent, in the sense that every ob can be proved to be a theorem. The argument makes use of the so-called paradoxical combinator $\underset{\sim}{Y}$, which is defined as follows:

$$\underset{\sim}{Y} \hat{=} \underset{\sim}{W}\ \underset{\sim}{S}\ (\underset{\sim}{B}\ \underset{\sim}{W}\ \underset{\sim}{B}).$$

It has the following reduction property:

$\underset{\sim}{Y}$ f --> f ($\underset{\sim}{Y}$ f).

To show that this system is inconsistent, we first define N to be ($\underset{\sim}{C}$ $\underset{\sim}{P}$ A) and Q to be ($\underset{\sim}{Y}$ N). Q is then the same ob as (N Q). Using the arrow notation this becomes:<2>

(3) Q = Q => A.

Now we argue as follows:

(4) |- Q => Q	1,
(5) |- Q => (Q => A)	3 4,
(6) |- (Q => (Q => A)) => (Q => A)	2,
(7) |- Q => A	mp 6 5,
(8) |- Q	3 7,
(9) |- A	mp 7 8.

The formulas on lines (4) and (6) are just substitution instances of (1) and (2), respectively. In this proof A was an arbitrary ob. Thus, it proves the inconsistency of any system which contains P, the axioms (1) and (2), modus ponens and substitution.<3>

This paradox is very easy to resolve. It is only deriveable on the assumption that Rule (P) and the axioms (1) and (2) hold universally for all obs, but there is no reason to think that

---

<2> Recall that the equals-sign represents conversion.

<3> Curry and Feys, Combinatory Logic, pp.258-259.

modus ponens holds if the obs it is applied to are not truth-values. Similarly, we cannot give a meaning to an ob (P x y) if x and y are not truth-values. Thus, the way out of the paradox is to restrict rule (P) and the axioms (1) and (2) to just those obs that are truth-values.

This solution of the paradox is different from Curry's. He restricts rule (P) and the axioms to obs which are propositions, which he understands as metaphysical entities. But it is more in line with the use to which I am putting combinatory logic not to follow Curry in his solution.

Appendix 3: The Positive Implicational Calculus

If the lexicon M only contains propositions of the form A: x, where A is a combinator (but it must include a combinatorially complete set of combinators), then corresponding to every $\underset{\sim}{F}$-deduction constructable from such a lexicon there corresponds a derivation in the positive implicational propositional calculus of Hilbert (Curry and Feys, Combinatory Logic, pp.312ff.). The derivation is obtained by first deleting the "subject" part of every category assigning proposition (that is to say, X: x becomes x) and then everywhere translating '1xy' into 'y => x'. Thus, corresponding to the $\underset{\sim}{F}$-deduction:

$$\frac{\underset{\sim}{S}\text{: } 111xx1yx11xyx \qquad\qquad \underset{\sim}{K}\text{: } 11xyx}{(\underset{\sim}{S}\ \underset{\sim}{K})\text{: } 11xx1yx,}$$

there is the derivation:

$$\frac{(x \Rightarrow (y \Rightarrow x)) \Rightarrow ((x \Rightarrow y) \Rightarrow (x \Rightarrow x)) \qquad x \Rightarrow (y \Rightarrow x)}{(x \Rightarrow y) \Rightarrow (x \Rightarrow x).}$$

The positive implicational calculus has two axiom-schemas:

(1) x => (y => x),

(2) (x => (y => z)) => ((x => y) => (x => z)),

and modus ponens is its only rule of inference. Not every theorem of the propositional calculus in the construction of which the only logical connective used is the material conditional can be proved in the positive implicational calculus. For example, Peirce's Law - that is to say, the formula ((x => y) => x) => x - cannot be derived.<1>

In this analogy with the positive implicational calculus, the rule of $\underset{\sim}{F}$-introduction corresponds to the rule of implication introduction.

<1> For further details see Church, An Introduction to Mathematical Logic, pp.140 and 115. For the positive implicational calculus the analogy with combinatory logic provides not only a decision procedure, but also an elegant proof procedure as well, see Meredith, "A Positive Logic Proof Procedure".

Appendix 4: Work related to Combinatory Grammar

Although the combinatory grammar I develop in Chapter 6 is still very rudimentary, no one has developed such a grammar in greater detail. But others have realised that combinatory logic can be used to analyse grammatical structures and in this Appendix I want to mention their work. I restrict myself to attempts to use the distinctive ideas of combinatory logic in this way. So, for example, I do not say anything here about the literature on categorial grammar, although there are similarities between categorial grammar and the theory of functionality.

In Curry and Feys's Combinatory Logic (1958) the authors mention that the illative primitive $\underset{\sim}{F}$ can be given a grammatical interpretation.<1> And they identify $\underset{\sim}{F}xy$ with Ajdukiewicz's y/x. They give some examples of how different expressions should be categorised (pp.274-275), but do not develop any sort of grammar on this basis.

In his article "Some Logical Aspects of Grammatical Structure" (1961) Curry repeats and slightly elaborates the material contained in Curry and Feys's Combinatory Logic, but nothing substantial is added.

<1> P.274. Recall that '$\underset{\sim}{F}xy$' is what I call '1yx' in Chapter 6.

Geach in his article "A Program for Syntax" (1972) gives an example of how an English sentence can be translated into a version of Quine's predicate-functor logic.<2> Quine claims that predicate-functor logic is preferable to standard combinatory logic because that presupposes an abstract universe equivalent to that of higher set theory, whereas his logic makes no ontological demands ("Variables Explained Away", p.233, footnote 1). Curry, however, has shown how Quine's system can be represented within combinatory logic and he argues that that does not have the ontological commitments that Quine claims ("The Elimination of Variables by Regular Combinators").

Lewis's "Combinators and Deep Structure" (1973) is a survey article which mentions some of the above references. It concentrates, however, on Quine's predicate-functor logic and how that can be used in linguistics.

Self in his Leeds PhD thesis A Combinatory Base for a Natural Language System (1973) constructs a grammar for a restricted and normalised subset of English which makes use of Quine's predicate-functor logic, the theory of functionality and Geach's categorial recursive rules. Such a system is subject to the criticisms that I made of those recursive rules in Chapter 5.

Šaumjan's article "The Genotype Language and Formal Semantics" (1973) takes Curry's work - mentioned above - as its starting-point, but as it makes no use of the distinctive ideas of

---

<2> Introduced in Quine's paper "Variables Explained Away" and elaborated in the subsequent papers "Algebraic Logic and Predicate Functors", "Truth and Disquotation" and "Predicate Functors Revisited".

combinatory logic it is really indistinguishable from an approach based on categorial grammar.

In his book Elements of Combinatory Logic (1974) Fitch writes:

> It should also be mentioned that the system Q, and the extension of it introduced in Chapter 6, appear to be well suited for representing salient features of the deep structure of English. The author has research interests in this direction at present. (P.viii.)

But the only published account of the fruits of that research is in his paper "The Relation between Natural Languages and Formalized Language" (1976). The approach taken in that paper is the closest to my own of all the references mentioned in this Appendix. His distinction between deep structure and surface structure is similar to mine and he allows combinators to occur in combinatory deep structures. He does not, however, make any use of the theory of functionality and his treatment of pronouns is very different from mine.

The above list of references is not very big, but - to the best of my knowledge - it is complete. I am grateful to Professor Curry for drawing my attention to Šaumjan's article and to Professor Fitch for informing me that he has not published anything else relevant to combinatory grammar.

Bibliography

Frege's Writings (in German)

"Über Begriff und Gegenstand", Vierteljahrsschrift fur wissenschaftliche Philosophie, vol.16 (1892), pp.192-205.

Begriffsschrift: Eine der arithmetischen nachgebildete Formelsprache des reinen Denkens, [Halle, Nebert, 1879].

Funktion und Begriff, [Jena, Pohle, 1891].

Grundgesetze der Arithmetik: Begriffsschriftlich abgeleitet, Band 1, [Jena, Pohle, 1893].

Die Grundlagen der Arithmetik: Eine logisch-mathematische Untersuchung uber den Begriff der Zahl, [Breslau, Koebner, 1884].

Kleine Schriften, (ed.), I.Angelelli, [Darmstadt, Wissenschaftliche Buchgesellschaft, 1967].

Nachgelassene Schriften, (eds. H.Hermes, F.Kambartel and F.Kaulbach), [Hamburg, Felix Meiner,1969].

"Über die Grundlagen der Geometrie", Jahresbericht der Deutschen Mathematiker-Vereinigung, vol.XII (1903), pp.319-324 and 368-375.

"Die Verneinung: Eine logische Untersuchung", Beitrage zur Philosophie des deutschen Idealismus, vol.1(1918), pp.143-157.

"Was ist eine Funktion?", in Festschrift Ludwig Boltzmann gewidmet zum sechzigsten Geburtstage, 20. Februar 1904, [Leipzig, Barth, 1904].

Wissenschaftlicher Briefwechsel, (eds. G.Gabriel, H.Hermes, F.Kambartel, C.Thiel and A.Veraart), [Hamburg, Felix Meiner,1976].

Frege's Writings (Translations)

The Basic Laws of Arithmetic: Exposition of the System, (tr. and ed. M.Furth), [Berkeley, University of California Press, 1964].

"A Brief Survey of My Logical Doctrines", in Posthumous Writings, pp.197-202.

"Comments on Sense and Meaning", in Posthumous Writings, pp.118-125.

Conceptual Notation and Related Articles, (tr. and ed. T.W.Bynum), [Oxford, Oxford University Press, 1972].

The Foundations of Arithmetic, A Logico-Mathematical Enquiry into the Concept of Number, (tr. J.L.Austin), [Oxford, Blackwell, 1950, 1953].

"Function and Concept", in Translations from the Philosophical Writings, pp.21-41.

"Negation", in Translations from the Philosophical Writings, pp.117-135.

"On Concept and Object", in Translations from the Philosophical Writings, pp.42-55.

"On Schoenflies: Die logischen Paradoxien der Mengenlehre", in Posthumous Writings, pp.176-183.

"On the Foundations of Geometry", in On the Foundations of Geometry and Formal Theories of Arithmetic, pp.22-37.

On the Foundations of Geometry and Formal Theories of Arithmetic, (tr. E-H.W.Kluge), [London, Yale University Press, 1971].

Translations from the Philosophical Writings of Gottlob Frege, (tr. and ed. P.T.Geach and M.Black), [Oxford, Blackwell, 1952, 1960].

Philosophical and Mathematical Correspondence, (ed. B.McGuinness, tr. H.Kaal), [Oxford, Blackwell, 1980].

Posthumous Writings, (tr. P.Long and R.White), [Oxford, Blackwell, 1979].

"What is a Function?", in Translations from the Philosophical Writings, pp.107-116.

"What may I regard as the Result of my work?", in Posthumous Writings, p.184.

Other Authors

Abraham, W. (ed.),

Valence, Semantic Case and Grammatical Relations: Studies in Language Companion Series, vol.1, [Amsterdam, Benjamins, 1978].

Ajdukiewicz, K.,

"On Syntactical Coherence", (tr. P.T.Geach), The Review of Metaphysics, vol.XX (1966-1967), pp.635-647.

"Syntactic Connexion", (tr. H.Weber), in McCall (ed.), Polish Logic, pp.207-231.

"Die Syntaktische Konnexität", Studia Philosophica, vol.1 (1935), pp.1-27.

Anscombe, G.E.M.,

An Introduction to Wittgenstein's Tractatus, [London, Hutchinson University Library, 1959, 1963, 1967, 1971].

Anscombe, G.E.M., and P.T.Geach,

Three Philosophers, [Oxford, Blackwell, 1963].

Baker, G.P., and Hacker, P.M.S.,

Frege: Logical Excavations, [Oxford, Blackwell, 1984].

Bar-Hillel, Y.,

Language and Information: Selected Essays on their Theory and Application, [Reading (Massachusetts), Addison-Wesley, 1964].

Bar-Hillel, Y., C.Gaifman and E.Shamir,

"On Categorial and Phrase Structure Grammars", The Bulletin of the Research Council of Israel, vol 9F (1960), pp.1-16. Also in Bar-Hillel, Language and Information, pp.99-115.

Bar-Hillel, Y., M.Perles and E.Shamir,

"On Formal Properties of Simple Phrase Structure Grammars", Zeitschrift fur Phonetik, Sprachwissenschaft und Kommunikationsforschung, vol.14 (1961), pp.143-172. Also in Bar-Hillel, Language and Information, pp.116-150.

Barendregt, H.P.,

The Lambda Calculus: Its Syntax and Semantics, [Amsterdam, North-Holland, 1981, 1984].

Bartsch, R.,

"Categorial Syntax in Grammatical Theory", in Kasher (ed.), Language in Focus, pp.503-539.

Bell, D.,

Frege's Theory of Judgement, [Oxford, Oxford University Press, 1979].

Bell, D.R.,

"Critical Notice" of Dummett's The Interpretation of Frege's Philosophy, Mind, vol.XCIII (1984), pp.276-293.

Binnick, R.I., A.Davison, G.M.Green and J.L.Morgan (eds.),

Papers from the Fifth Regional Meeting of the Chicago Linguistic Society, [Chicago, Department of Linguistics, 1969].

Bogdan, R.J., and I.Niiniluoto (eds.),

Logic, Language, and Probability, [Dordrecht (Holland), Reidel, 1973].

Brandom, R.B.,

"Frege's Technical Concepts: Some Recent Developments", in Haaparanta and Hintikka (eds.), Frege Synthesized, pp.253-295.

Bunge, M. (ed.),

The Critical Approach to Science and Philosophy, [London, Collier-Macmillan, 1964].

Carnap, R.,

The Logical Syntax of Language, (tr. A.Smeaton), [London, Kegan Paul, Trench, Trubner & Co.Ltd., 1937].

Logische Syntax der Sprache, [Wien, Springer, 1934].

Church, A.,

Introduction to Mathematical Logic, vol.1, [Princeton, Princeton University Press, 1956].

Cresswell, M.J.,

Logics and Languages, [London, Methuen, 1973].

Structured Meanings: The Semantics of Propositional Attitudes, [Cambridge (Massachusetts). The MIT Press, 1985].

Currie, G.,

Frege: An Introduction to his Philosophy, [Brighton (Sussex), Harvester, 1982].

Curry, H.B.,

"The Elimination of Variables by Regular Combinators", in Bunge (ed.), The Critical Approach to Science and Philosophy, pp.127-143.

"Some Logical Aspects of Grammatical Structure", in Jakobson (ed.), Structure of Language and its Mathematical Aspects, pp.56-68.

Curry, H.B., J.R.Hindley and J.P.Seldin,

Combinatory Logic, vol.II, [Amsterdam, North-Holland, 1972].

Curry, H.B., and R.Feys,

Combinatory Logic, vol.I, [Amsterdam, North-Holland, 1958].

Davidson, D., and G.Harman (eds.),

Semantics of Natural Language, [Dordrecht (Holland), Reidel, 1972].

Davidson, D., and J.Hintikka (eds.),

Words and Objections: Essays on the Work of W.V.Quine, [Dordrecht (Holland), Reidel, 1969].

Dilman, I. (ed.),

Philosophy and Life: Essays on John Wisdom, [The Hague, Martinus Nijhof, 1984].

Dummett, M.,

Frege: Philosophy of Language, [London, Duckworth, 1973, 1981].

"Frege on Functions: A Reply", The Philosophical Review, vol.LXIV (1955), pp.96-107. Also in Klemke (ed.), Essays on Frege, pp.268-283.

The Interpretation of Frege's Philosophy, [London, Duckworth, 1981].

"Note: Frege on Functions", The Philosophical Review, vol.LXV (1956), pp.229-230. Also in Klemke (ed.), Essays on Frege, pp.295-297.

"An Unsuccessful Dig", The Philosophical Quarterly, vol.34 (1984), pp.377-401.

Durrant, P.J.,

Organic Chemistry, [London, Longmans, Green and Co., 1950].

Evans, G.,

"Pronouns, Quantifiers, and Relative Clauses (I)", Canadian Journal of Philosophy, vol. VII (1977), pp.467-536.

"Pronouns, Quantifiers, and Relative Clauses (II): Appendix", Canadian Journal of Philosophy, vol. VII (1977), pp.777-797.

Fitch, F.B.,

Elements of Combinatory Logic, [London, Yale University Press, 1974.].

"The Relation between Natural Languages and Formalized Languages", in Korner (ed.), Philosophy of Logic, pp.183-190.

Geach, P.T.

"Back Reference", in Kasher (ed.), Language in Focus, pp.25-39.

"Critical Notice" of Dummett's Frege, Mind, vol.LXXXV (1976), pp.436-449.

"Czemu Zdanie nie jest Nazwą", Studia Semiotyczne, vol.III (1972), pp.13-21.

"Frege", in Anscombe and Geach, Three Philosophers, pp.127-162.

Logic Matters, [Oxford, Blackwell, 1972].

"Names and Identity", in Guttenplan (ed.), Mind and Language, pp.139-158.

"On What There Is", Aristotelian Society, Supplementary Volume, XXV (1951), pp.125-134.

"A Program for Syntax", in Davidson and Harman (eds.), Semantics of Natural Language, pp.483-497.

"Quine's Syntactical Insights", in Davidson and Hintikka (eds.), Words and Objections, pp.146-157.

"Quotation and Quantification", in Geach, Logic Matters, pp.205-209.

Reason and Argument, [Oxford, Blackwell, 1976].

Reference and Generality: An Examination of Some Medieval and Modern Theories, [London, Cornell University Press, 1962, 1968, 1980].

"Saying and Showing in Frege and Wittgenstein", Act Philosophica Fennica, vol.28 (1976), pp.54-70.

"Should Traditional Grammar be Ended or Mended?", Educational Review, vol.22 (1969-1970), pp.18-25.

Grossmann, R.,

"Frege's Ontology", The Philosophical Review, vol.LXX (1961), pp.23-40. Also in Klemke (ed.), Essays on Frege, pp.79-98.

Guttenplan, S. (ed.),

Mind and Language: Wolfson College Lectures 1974, [Oxford, Oxford University Press, 1975].

Haaparanta, L., and J.Hintikka (eds.),

Frege Synthesized, [Dordrecht (Holland), Reidel, 1986].

Hatcher, W.S.,

The Logical Foundations of Mathematics, [Oxford, Pergamon, 1982].

Heny, F., and H.S.Schelle (eds.),

Syntax and Semantics, vol.10, Selections from the Third Groningen Round Table Conference, [New York, Academic Press, 1979].

Hilbert, D.,

The Foundations of Geometry, (tr. E.J.Townsend), [La Salle (Illinois), Open Court, 1947].

Grundlagen der Geometrie, [Leipzig, Teubner, 1903].

Hintikka, J., J.Moravcsik and P.Suppes (eds.),

Approaches to Natural Language, [Dordrecht (Holland), Reidel, 1973].

Hiż, H.,

"Grammar Logicism", The Monist, vol.51 (1967), pp.110-127.

"Syntactic Completion Analysis", Transformations and Discourse Analysis Papers, 21, (University of Pennsylvania, 1961).

Hughes, G.E., and M.J.Cresswell,

An Introduction to Modal Logic, [London, Methuen, 1968].

Husserl, E.G.,

Logical Investigations, vol.2, (tr. J.N.Findlay), [London, Routledge & Kegan Paul, 1970].

Logische Untersuchungen, Band II, Teil 1, [Halle, Niemeyer, 1928].

Logische Untersuchungen, Band II, Teil 2, [Halle, Niemeyer, 1922].

Jakobson, R. (ed.),

Structure of Language and its Mathematical Aspects, Proceedings of the Twelfth Symposium in Applied Mathematics, [Providence (Rhode Island), American Mathematical Society, 1961],

Karttunen, L.,

"Definite Descriptions with Crossing Coreference", Foundations of Language, vol.7 (1971), pp.183-198.

"Pronouns and Variables", in Binnick, et.al., Papers from the Fifth Regional Meeting of the Chicago Linguistic Society, pp.108-116.

Kasher, A. (ed.),

Language in Focus: Foundation, Methods and Systems: Essays in Memory of Yehoshua Bar-Hillel, [Dordrecht (Holland), Reidel, 1976].

Kiefer, F. (ed.),

Trends in Soviet Theoretical Linguistics, [Dordrecht (Holland), Reidel, 1973].

Klemke, E.D., (ed.),

Essays on Frege, [Urbana, University of Illinois Press, 1968].

Kline, M.,

Mathematical Thought from Ancient to Modern Times [New York, Oxford University Press, 1972].

Körner, S. (ed.),

Philosophy of Logic, [Oxford, Blackwell, 1976].

Lambek, J.,

"On the Calculus of Syntactic Types", in Jakobson (ed.), Structure of Language and its Mathematical Aspects, pp.166-178.

Lehrberger, J.,

Functor Analysis of Natural Language, Janua Linguarum, Series Minor, vol.197, [The Hague, Mouton, 1974].

Lewis, D.,

"General Semantics", in Davidson and Harman (eds.), Semantics of Natural Language, pp.169-218.

Lewis, H.A.,

"Combinators and Deep Structure: Syntactic and Semantic Functions", in Bogdan and Niiniluoto (eds.), Logic, Language, and Probability, pp.213-222.

"Model Theory and Semantics", Theoretical Linguistics, vol.1 (1974), pp.271-285.

Long, P.,

"Universals: Logic and Metaphor", in Dilman, (ed.), Philosophy and Life, pp271-289.

"Formal Relations", The Philosophical Quarterly, vol.32 (1982), pp.151-161.

Lyons, J.,

Introduction to Theoretical Linguistics, [Cambridge, Cambridge University Press, 1968].

"Towards a 'Notional' Theory of the "Parts of Speech'", Journal of Linguistics, vol.2 (1966), pp.209-236.

McCall,S., (ed.),

Polish Logic: 1920-1939, [Oxford, Oxford University Press, 1967].

Marshall, W.,

"Frege's Theory of Functions and Objects", The Philosophical Review, vol.LXII (1953), pp.374-390. Also in Klemke (ed.), Essays on Frege, pp.249-267.

Menger, K.,

"On Variables in Mathematics and in Natural Science", The British Journal for the Philosophy of Science, vol.V (1954-1955), pp.134-142.

Meredith, D.,

"A Positive Logic Proof Procedure", in Seldin and Hindley (eds.), To H.B.Curry, pp.503-510.

Monna, A.F.,

"The Concept of Function in the 19th and 20th Centuries: in Particular with Regard to the Discussions between Baire, Borel and Lebesgue", Archive for History of Exact Sciences, vol.9 (1972-1973), pp.57-84.

Montague, R.,

Formal Philosophy, (ed. R.H.Thomason), [London, Yale University Press, 1974].

"The Proper Treatment of Quantification in Ordinary English", in Hintikka, Moravcsik and Suppes (eds.), Approaches to Natural Language, pp.221-242. Also in Montague, Formal Philosophy, pp.247-270.

Potts, T.C.,

"Case-Grammar as Componential Analysis", in Abraham (ed.), Valence, Semantic Case and Grammatical Relations, pp.399-457.

"Fregean Categorial Grammar", in Bogdan and Niiniluoto (eds.), Logic, Language, and Probability, pp.245-284.

"Fregean Grammar: A Formal Outline", Studia Logica, vol.XXXVII (1978), pp.7-26.

"The Place of Structure in Communication", in Vesey (ed.), Communication and Understanding, pp.91-115.

"A General Theory of the Meaning of Anaphoric Pronouns", in Heny and Schelle (eds.), Syntax and Semantics, vol.10, pp.150-198.

Prior, A.,

Objects of Thought, (eds. P.T.Geach and A.J.P.Kenny), [Oxford, Oxford University Press, 1971].

Quine, W.V.O.,

"Algebraic Logic and Predicate Functors", in Rudner and Scheffler (eds.), Logic and Art, pp.214-238.

"Logic Based on Inclusion and Abstraction", Journal of Symbolic Logic, vol.2 (1937). Also in W.V.O.Quine, Selected Logic Papers, pp.100-109.

Mathematical Logic, [Cambridge, Harvard University Press, 1940, 1951].

"New Foundations for Mathematical Logic", American Mathematical Monthly, vol.44 (1937), pp.70-80.

"Predicate Functors Revisited", The Journal of Symbolic Logic, vol.46 (1981), pp.649-652.

Selected Logic Papers, [New York, Random House, 1966].

"Truth and Disquotation", unpublished typescript, 1971.

"Variables Explained Away", Proceedings of the American Philosophical Society, vol.104 (1960), pp.343-347. Also in W.V.O.Quine, Selected Logic Papers, pp.227-235.

The Ways of Paradox and Other Essays, [Cambridge (Massachusetts), Harvard University Press, 1976].

Word and Object, [Cambridge (Massachusetts), The MIT Press, 1960].

Quirk,R., S.Greenbaum, G.Leech and J.Svartik,

A Grammar of Contemporary English, [London, Longman, 1972].

Radford, A.,

Transformational Syntax, [Cambridge, Cambridge University Press, 1981].

Richards, B.,

"On Interpreting Pronouns", Linguistics and Philosophy, vol.7 (1984), pp.287-324.

Resnik, M.D.,

"Frege's Proof of Referentiality", in Haaparanta and Hintikka (eds.), Frege Synthesized, pp.177-195.

Rudner, R., and I.Scheffler (eds.),

Logic and Art: Essays in Honor of Nelson Goodman, [Indianapolis, Bobbs-Merrill, 1972].

Russell, B.,

An Introduction to Mathematical Philosophy, [London, Allen and Unwin, 1919].

Saumjan, S.K.,

"The Genotype Language and Formal Semantics", in Kiefer (ed.), Trends in Soviet Theoretical Linguistics, pp.251-333.

Seldin, J.P., and J.R.Hindley (eds.),

To H.B.Curry: Essays on Combinatory Logic, Lambda Calculus and Formalism, [London, Academic Press, 1980].

Self, J.A.,

A Combinatory Base for a Natural Language System, PhD thesis, The University of Leeds, 1973.

Stoy, J.E.,

Denotational Semantics: The Scott-Strachey Approach to Programming Language Theory, [Cambridge (Massachusetts), The MIT Press, 1977].

Tarski, A.,

"The Concept of Truth in Formalized Languages", in Tarski, Logic, Semantics, Metamathematics, pp.152-278.

Logic, Semantics, Metamathematics: Papers from 1923 to 1938 (tr. J.H.Woodger), [Oxford, Oxford University Press, 1956].

Turner, R.,

"Nominalization and Scott's Domains", Linguistics and Philosophy, vol.6 (1983), pp.259-288.

Vanderveken, D.R.,

"The Lesniewski-Curry Theory of Syntactical Categories and the Categorially Open Functors", Studia Logica, vol.XXXV (1976), pp.191-201.

Vesey, G., (ed.),

Communication and Understanding: Royal Institute of Philosophy Lectures, vol.10 (1975-1976), [Hassocks (Sussex), Harvester, 1977].

Wittgenstein, L.,

Philosophical Grammar, (ed. R.Rhees, tr. A.Kenny), [Oxford, Blackwell, 1974].

Philosophische Grammatik, (ed. R.Rhees), [Oxford, Blackwell, 1969].

Tractatus Logico-Philosophicus, (tr. C.K.Ogden), [London, Routledge and Kegan Paul, 1922].

Wright, C.,

Frege's Conception of Numbers as Objects, [Aberdeen, Aberdeen University Press, 1983].