CHAPTER I

THE PRINCIPLE OF COMPOSITIONALITY OF MEANING

ABSTRACT

This chapter deals with various aspects of the principle of compositionality of meaning. The role of the principle in the literature is investigated, and the relation of the principle to Frege's works is discussed. A formalization of the principle is outlined, and several arguments are given in support of this formalization.
1. AN ATTRACTIVE PRINCIPLE

The starting point of the investigations in this book is the principle of compositionality of meaning. This principle says:

The meaning of a compound expression is built up from the meanings of its parts.

This is an attractive principle which pops up at a diversity of places in the literature. The principle can be applied to a variety of languages: natural, logical and programming languages. I would not know of a competing principle. In this section the attractiveness of the principle will be illustrated by means of many quotations.

In the philosophical literature the principle is well known, and generally attributed to the mathematician and philosopher Gottlob Frege. An example is the following quotation. It gives a formulation of the principle which is about the same as the formulation given above. THOMASON (1974, p.55) says:

Sentences [...] such as 'The square root of two is irrational', and 'Two is even', [...] ought to be substitutable salps veritate in all contexts obeying Frege's principle that the meaning of a phrase is a function of the meanings of its parts.

Another illustration of the fame of the principle is given by DAVIDSON (1967, p.306):

If we want a theory that gives the meaning (as distinct from reference) of each sentence, we must start with the meaning (as distinct from reference) of the parts.

Next he says:

Up to here we have been following Frege's footsteps; thanks to him the path is well known and even well worn.

Popper mentions a version of the principle which applies to whole theories (POPPER 1976, p.22):

[...] the meaning of a theory [...] is a function of the meanings of the words in which the theory is formulated.

Thereafter he says (ibid. p.22):

This view of the meaning of a theory seems almost obvious; it is widely held, and often unconsciously taken for granted.

(For completeness of information there is, according to Popper, hardly any truth in the principle). Concerning the origin of the principle, Popper remarks in a footnote (ibid. p.198):

Not even Gottlob Frege states it quite explicitly, though this doctrine is certainly implicit in his 'Sinn und Bedeutung', and he even produces there arguments in its support.
In the field of semantics of natural languages, the principle is found implicitly in the works of Katz and Fodor concerning the treatment of semantics in transformational grammars. An explicit statement of the principle is KATZ (1966, p.152):

The hypothesis on which we will base our model of the semantic component is that the process by which a speaker interprets each of the infinitely many sentences is a compositional process in which the meaning of any syntactically compound constituent of a sentence is obtained as a function of the meanings of the parts of the constituent.

Katz does not attribute the principle to Frege; his motivation is of a technical nature (ibid. p.152):

Accordingly, we again face the task of formulating an hypothesis about the nature of a finite mechanism with an infinite output.

The principle is mentioned explicitly in important work on the semantics of natural languages by logicians, and it is related there with Frege. Cresswell develops a mathematical framework for dealing with semantics, and having presented his framework he says (CRESSWELL 1973, p.19):

These rules reflect an important general principle which we shall discuss later under the name 'Frege's principle', that the meaning of the whole sentence is a function of the meanings of its parts.

For another logician, Montague, the principle seems to be a line of conduct (MONTAGUE 1970a, p.217):

Like Frege, we seek to do this [...] in such a way that [...] the assignment to a compound will be a function of the entities assigned to its components [...].

The principle is implicitly followed by all logic textbooks when they define, for instance, the truth value of \( p \land q \) as a function of the truth-values of \( p \) and of \( q \). In logic the enormous technical advantages of treating semantics in accordance with the principle are demonstrated frequently. For instance, one may use the power of induction: theorems with a semantic content can be proven by using induction on the construction of the expression under consideration. Logic textbooks usually do not say much about the motivation for their approach or about the principles of logic. In any case, I have not succeeded in finding a quotation in logic textbooks concerning the background of their compositional approach. Therefore, here is one from another source. In a remark concerning the work of Montague, Partee says the following about the role of compositionality in logic (PARTEE 1975, p.203):

A central working premise of Montague's theory [...] is that the syntactic rules that determine how a sentence is built up out of smaller syntactic parts should correspond one-to-one with the semantic rules that tell how the meaning of a sentence is a function of the meanings
of its parts. This idea is not new in either linguistics or philosophy; in philosophy it has its basis in the work of Frege, Tarski, and Carnap, and it is standard in the treatment of formalized languages [...].

Since almost all semantic work in mathematical logic is based upon Tarski, mathematical logic is indirectly based upon this principle of compositionality of meaning.

In the field of semantics of programming languages compositionality is implicit in most of the publications, but it is mentioned explicitly only by few authors. In a standard work for the approach called 'denotational semantics', the author says (STOY 1977, pp.11-13):

We give 'semantic valuation functions' which map syntactic constructs in the program to the abstract values (numbers, truth values, functions etc.) which they denote. These valuation functions are usually recursively defined; the value denoted by a construct is specified in terms of the values denoted by its syntactic subcomponents [...].

It becomes clear that this aspect is a basic principle of this approach from a remark of Tennent in a discussion of some proposals concerning the semantics of procedures. Tennent states about a certain proposal the following (NEUHOLD 1978,p.163):

Your first two semantics are not 'denotational' in the sense of Scott/Strachey/Milner because the meaning of the procedure call construct is not defined in terms of the meanings of its components; they are thus partly operational in nature.

Milner explicitly mentions compositionality as basic principle (MILNER 1975, p.167):

If we accept that any abstract semantics should give a way of composing the meanings of parts into the meaning of the whole [...].

As motivation for this approach, he gives a very practical argument (ibid. p.158):

The designer of a computing system should be able to think of his system as a composite of behaviours, in order that he may factor his design problem into smaller problems [...].

Mazurkiewics mentions naturalness as a motivation for following the principle (MAZURKIEWICS 1975, p.75).

One of the most natural methods of assigning meanings to programs is to define the meaning of the whole program by the meanings of its constituents [...].

We observe that the principle arises in connection with semantics in many fields. In the philosophical literature the principle is almost always attributed to Frege, whereas in the fields of programming and natural language semantics this is not the case. Authors in these fields give a practical motivation for obeying the principle: one wishes to deal with an
infinity of possibilities in some reasonable, practical, understandable, and (therefore) finite way.

2. FREGE AND THE PRINCIPLE

2.1. Introduction

In the previous section we observed that several philosophers attribute the principle of compositionality to Frege. But it is not made clear what the relationship is of the principle to Frege, and especially on what grounds the principle is attributed to him.

In his standard work on Frege, Dummett devotes a chapter to 'Some theses of Frege on sense and reference'. The first thesis he considers is (DUMMETT 1973, p.152):

The sense of a complex is compounded out of the senses of the constituents.

Since sense is about the same as meaning (this will be explained later), the thesis expresses the principle of compositionality of meaning. Unfortunately, Dummett does not relate this thesis to statements in the work of Frege, so it remains unclear on what writings the claim is based that it is a thesis of Frege. The authors quoted in the previous section who attribute the principle to Frege, do not refer to his writings either.

In the previous section we met a remark by Popper stating that the principle is not explicit in Frege's work, but that it is certainly implicit. The connection with Frege is, according to Cresswell, even looser. He says (CRESSWELL 1973, p.75):

For historical reasons we call this Frege's principle. This name must not be taken to imply that the principle is explicitly stated in Frege.

And in a footnote he adds to this:

The ascription to Frege is more a tribute to the general tenor of his views on the analysis of language.

However, Cresswell does not explain these remarks any further. So we have to conclude that the literature gives no decisive answer to the question what the relationship is of the principle to Frege. I will try to answer the question by considering Frege's publications and investigating what he explicitly says about this subject.
2.2. Grundlagen

The study of Frege's publications brings us to the point of terminology. Frege has introduced some notions associated with meaning, but his terminology is not the same in all his papers. Dummett says about 'Die Grundlagen der Arithmetik' (FREGE 1884) the following (DUMMETT 1973, p.193):

When Frege wrote 'Grundlagen', he had not yet formulated his distinction between sense and reference, and so it is quite possible that the words 'Bedeutung' and 'bedeuten', as they occur in the various statements [...] have the more general senses of 'meaning' and 'mean' [...].

This means that in 'Grundlagen' we have to look for Frege's remarks concerning the 'Bedeutung' of parts. He is quite decided on the role of their Bedeutung (FREGE 1884, p.XXII):

Als Grundsätze habe ich in dieser Untersuchung folgende festgehalten: [...] nach der Bedeutung der Wörter muss in Satzverhältnissen, nicht in ihrer Vereinselung gefragt werden [...] .

He also says (FREGE 1884, p.73):

Nur in Verhältnissen eines Satzes bedeuten die Wörter etwas.

Remarks like these ones are repeated, heavily underlined, several times in 'Grundlagen' (e.g. on p.71 and p.116). The idea expressed by them is sometimes called the principle of contextuality. Contextuality seems to be in conflict with compositionality for the following reason. The principle of compositionality requires that words in isolation have a meaning, since otherwise there is nothing from which the meaning of a compound expression can be built. A principle of contextuality would deny that words in isolation have a meaning.

Dummett discusses the remarks from 'Grundlagen', and he provides an interpretation in which they are not in conflict with compositionality (DUMMETT 1973, pp.192-196). A summary of his interpretation is as follows. The statements express that it has no significance to consider first the meaning of a word in isolation, and next some unrelated other question. Speaking about the meaning of a word makes only significance as preparation for considering the meaning of a sentence. The meaning of a word is determined by the role it plays in the meaning of the sentence.

Following Dummett's interpretation, the remarks from Grundlagen have not to be considered as being in conflict with the principle of compositionality. It is quite well possible to build the meaning of a sentence from the meanings of its parts, and to base at the same time the judgments about the meanings of these parts on the role they play in the sentences in which they may occur. As a matter of fact, this approach is often
followed (for instance in the field of Montague grammar). In this way a bridge is laid between compositionality and contextuality. But even with his interpretation, the statements formulated in Grundlagen cannot be considered as propagating compositionality: nothing is said about building meanings of sentences from meanings of words.

Dummett's interpretation weakens the statements from 'Grundlagen' considerably. Unfortunately, Dummett hardly explains on which grounds he thinks that his interpretation coincides with Frege's intentions when writing 'Grundlagen'. He provides, for instance, no references to writings of Frege. I tried to do so, but did not find passages supporting Dummett's opinion. Dummett makes the remark that statements like the ones from 'Grundlagen' make no subsequent appearance in Frege's works. This is probably correct with respect to Frege's published works, but I found some statements which are close to those 'Grundlagen' in Frege's correspondence and in his posthumous writings. They do not express the whole context principle, but repeat the relevant aspect: that expressions outside the context of a sentence have no meaning. In a letter to E.V. Huntington, probably dating from 1902, Frege says the following (GABRIEL 1976, p.90).

Sollen Zeichenverbindungen wie "a+b", "f(a.b)" bedeuten also nichts, und haben für sich allein keinen Sinn [...] In 'Einleitung in die Logik', dating from 1906, he says (HERMES 1969, p.204):

Durch Zerlegung der singulären Gedanken erhält man Bestandteile der abgeschlossenen und der ungesättigten Art, die freilich abgesondert nicht vorkommen.

In an earlier paper (from 1880), called 'Booles rechnende Logik und die Begriffsschrift', he compares the situation with the behaviour of atoms (HERMES 1969, p.19).

Ich möchte diese mit dem Verhalten der Atome vergleichen, von denen man annimmt, dass sie eine allein vorkommt, sondern nur in einer Verbindung mit andern, die es nur verlässt, um sofort in eine andere eingezogen.

The formulation of the statements from 'Grundlagen' is evidently in conflict with the principle of compositionality. From our investigations it appears that related remarks occur in other writings of Frege. This shows that the formulation used in 'Grundlagen' is not just an accidental, and maybe unfelicitous expression of his thoughts. For this reason, and for the lack of evidence for Dummett's interpretation, I am not convinced that Frege's clear statements have to be understood in a weakened way. I think that they should be understood as they are formulated. Therefore I conclude
that in the days of 'Grundlagen' Frege probably would have rejected the principle of compositionality, and, in any case, the formulation we use.

2.3. Sinn und Bedeutung

In 'Ueber Sinn und Bedeutung' (FREGE 1892) the notions 'Sinn' and 'Bedeutung' are introduced. Frege uses these two already existing German words to name two notions he wished to discriminate. The subtle differences in the original meaning of these two words do not cover the different use Frege makes of them. For instance, it is very difficult to account for their differences in meaning in a translation. Frege himself has been confronted with these problems as appears from a letter to Peano (GABRIEL 1976, p.196):

[...]

Concerning the terminology DUMMETT (1973, p.84) gives the following information. The term 'Bedeutung' has come to be conventionally translated as 'reference'. Since 'Bedeutung' is simply the German word for 'meaning', one cannot render 'Bedeutung' as it occurs in Frege by 'meaning', without a special warning. The word 'reference' does not belie Frege's intention, though it gives it a much more explicit expression. Concerning 'Sinn', which is always translated 'sense', Dummett says that to the sense of a word or expression only those features of meaning belong which are relevant to the truth-value of some sentence in which it may occur. Differences in meaning which are not relevant in this way, are relegated by Frege to the 'tone' of the word or expression. In this way Dummett has given an indication what Frege intended with sense. It is not possible to be more precise about the meaning of 'Sinn'. As van Heyenoort says (Van HEYENOORT 1977, p.93):

As for the 'Sinn' Frege gives examples, but never presents a precise definition.

And Thiel states (THIEL 1965, p.165):

What Frege understood as the 'sense' of an expression is a problem that is so difficult that one generally weakens it to the question of when in Frege's semantics two expressions are identical in sense (synonymous).

I will not try to give a definition; it suffices for our purposes to conclude that the notion 'Sinn' is very close to the notion 'meaning'. Therefore we have to investigate Frege's publications after 1892 to see what he says about the compositionality of Sinn. What he says about compositionality of 'Bedeutung' is a different story (as illustration: he explicitly
rejected that in 1919 (HERMES 1969, p.275), but this is not the case for compositionality of 'Sinn', as will appear in the sequel).

In 'Über Sinn und Bedeutung', I found one remark concerning the relation between the senses of parts and the sense of the whole sentence. Fregé discusses the question whether a sentence has a reference, and, as an example, he considers the sentence Odyseus wurde tief schlafend in Ithaka an Land gesetzt. Fregé says that if someone considers this sentence as true or false, he assigns the name Odyseus a reference (Bedeutung). Next he says (FREGÉ 1892, p.33).

(Nach wären aber das Vordringen bis zur Bedeutung des Namens überflüssig; man könnte sich mit dem Sinn befassen, wenn man beim Gedanken stehenbleiben wollte. Kämme es nur auf den Sinn des Satzes, den Gedanken an, so wäre es unmöglich sich um die Bedeutung eines Satzteiles zu kümmern; für den Sinn des Satzes kann ja nur der Sinn, nicht die Bedeutung dieses Teiles in Betracht kommen.

So Fregé states that there is a connection between the sense of the whole sentence, and the senses of the parts. He does, however, not say anything about a compositional way of building the sense of the sentence. More in particular, the quotation is neither in conflict with the compositionality principle, nor with the statements from 'Grundlagen'. Therefore I agree with BARTSCH (1978), who says that, in 'Über Sinn und Bedeutung', Fregé does not speak, as is often supposed, about the contribution of the senses of parts to the senses of the compound expression.

2.4. The principle

Up till now we have not found any statement expressing the principle of compositionality. But there are such fragments. The most impressive one is from 'Logik in der Mathematik', an unpublished manuscript from 1914 (HERMES 1969, p.243).


This fragment expresses the compositionality principle. However, the fragment is not presented as a fragment expressing a basic principle. It is used as argument in a discussion, and does not get any special attention.
The quotation from 'Logik in der Mathematik', presented above, is considered very remarkable by the editors of Frege's posthumous works. They have added the following footnote in which they call attention to other statements of Frege which seem to conflict with the quotation in consideration (HERMES 1969, p.243).


They give two references to such statements in Frege's posthumous writings (i.e. the book they are editors of). One is from 'Booles rechnende Logik.' (1880), the other from 'Einleitung in die Logik' (1906). I have quoted these fragments in the discussion of 'Grundlagen'. In this way the editors suggest that the fragment from 'Logik in der Mathematik' is a slip of the pen, and a rather incomplete formulation of Frege's opinion concerning these matters.

The fragment under discussion does, however, not stand on its own. Almost the same fragment can be found in 'Gedankenfüge' (FREGE 1923). I present the fragment here in its English translation from 'Compound thoughts' by Geach and Stoothoff (p.55).

*It is astonishing what language can do. With a few syllables it can express an incalculable number of thoughts, so that even a thought grasped by a terrestrial being for the very first time can be put into a form of words which will be understood by someone to whom the thought is entirely new. This would be impossible, were we not able to distinguish parts in the thought corresponding to the parts of a sentence, so that the structure of the sentence serves as an image of the structure of the thought.*

Moreover, in a letter to Jourdain, written about 1914, Frege says (GABRIEL 1976, p.127):

*Die Möglichkeit für uns, Sätze zu verstehen, die wir noch nie gehört haben, beruht offenbar darauf, dass wir den Sinn eines Satzes aufbauen aus Teilen, die den Wörtern entsprechen.*

It is a remarkable fact that all quotations propagating compositionality are written after 1910: 'Gedankenfüge' (1923), 'Logik in der Mathematik' (1914), letter to Jourdain (1914). I have not succeeded in finding such quotations in earlier papers. But the statements which seem to conflict with compositionality are from an earlier period: 'Booles rechnende Logik ....' (1880), 'Grundlagen' (1884), letter to Huntington (1902), 'Einleitung in die Logik' (1906). This shows that, say after 1910, Frege has written about these matters in a completely different way than before. From this I conclude that his opinion concerning these matters changed. On the other hand, Frege never put forward the idea of compositionality as a principle.
It was rather an argument, although an important one, in his discussions. I would therefore not conclude to a break in his thoughts; rather it seems to me to be a shift in conception concerning a detail.

In the light of this change, the following information appears relevant. In 1902 Frege received a letter from Russell in which the discovery was mentioned of the famous contradiction in naive set theory, and, in particular, in the theory of classes in Frege's 'Grundgesetze'. About the influence of this discovery on Frege, Dummett says the following (DUMMETT 1973, p.657):

*It thus seems highly probable that Frege came quickly to regard his whole programme of deriving arithmetic from logic as having failed. Such a supposition is not only probable in itself; it is in complete harmony with what we know of his subsequent career. The fourth period of his life may be regarded as running from 1905 to 1913, and it was almost entirely unproductive.*

For this reason I consider it as very likely that in this period Frege was not concerned with issues related to compositionality. Then it is understandable that after this period he writes in a different way about the detail of compositionality (recall that it never was a principle, but just an argument).

2.5. Conclusion

My conclusions are as follows. Before 1910, and in any case especially in the years when he wrote his most important and influential works, Frege would probably have rejected the compositionality principle, in any case the formulation we use nowadays. After 1910 his opinion appears to have changed, and he would probably have accepted the principle, in any case the basic idea expressed in it. However, Frege never put forward such an idea as a basic principle, it is rather an argument in his discussions. Therefore, calling the compositionality principle 'Frege's principle' is above all, honouring his contributions to the study of semantics. But it is also an expression of his final opinion on these matters.

3. TOWARDS A FORMALIZATION

In this section, I will give the motivation for a formalized version of the compositionality principle. It is not my purpose to formalize what Frege or other authors might have intended to say when uttering something like the principle. I rather take the principle in the given formulation
as a starting point and proceed along the following line; the (formalized version of) the principle should have as much content as possible. This means that the principle should make it possible to derive interesting consequences about those grammars which are in accordance with the principle, and at the same time it should be sufficiently abstract and universal to be applicable to a wide variety of languages. From the formalization it must be possible to obtain necessary and sufficient conditions for a grammar to be in agreement with the compositionality principle.

Consider a language $L$ which is to be interpreted in some domain $D$ of meanings. The kind of objects $D$ consists of depends on the language under consideration, and the use one wishes to make of the semantics. In this section such aspects are left unspecified. Defining the semantics of a language consists in defining a suitable relation between expressions in $L$ and semantic objects in $D$. Then the compositionality principle says something about the way in which this relation between $L$ and $D$ has to be defined properly.

In the formulation of the principle given in section 1, we encounter the phrase 'its parts'. Clearly we should not allow the expressions of $L$ to be split in some random way. In the light of the standard priority conventions, the expression $y + \varepsilon$ is not to be considered as a part of the expression $7.y + \varepsilon.x$; so the meaning of $7.y + \varepsilon.x$ does not have to be built up from the meaning of $y + \varepsilon$. It would also be pointless to try to build the meaning of some compound expression directly from the meanings of its atomic symbols (the terminal symbols of the alphabet used to represent the language). Since distinct expressions consist of distinct strings of symbols, there is always some dependence of the meanings of the basic symbols. Consequently such an interpretation would trivialize the principle. Another trivialization results by taking all expressions of the language to be 'basic', and interpreting them individually. The principle is interesting only in case the 'parts' are not trivial parts. Traditionally, the true decomposition of an expression into parts is described in the syntax for the language. Thus a language, the semantics of which is defined in accordance with the principle, should have a syntax which clearly expresses what the parts of the compound expressions are.

Let the language $L$, together with the set of expressions we wish to consider as parts, be denoted by $E$. In order to give the principle a non-trivial content, we assume that the syntax of the language consists of rules of the following form:
If one has expressions $E_1, \ldots, E_n$ then one can build the compound expression $S_j(E_1, \ldots, E_n)$.

Here $S_j$ is some operation on expressions, and $S_j(E_1, \ldots, E_n)$ denotes the result of application of $S_j$ to the arguments $E_1, \ldots, E_n$. If the rules have the above form, we define the notion 'parts of' as follows.

If expression $E$ is built by a rule $S_j$ from arguments $E_1, \ldots, E_n$, then the parts of $E$ are the expressions $E_1, \ldots, E_n$.

It often is the case that a rule does not apply to all expressions in $E$, and that certain groups of expressions behave the same in this respect. Therefore the set of expressions is divided into subsets. The names of these subsets are called *types* or *sorts* in logic, *categories* in linguistics, and *types* or *modes* in programming languages. Often the name of a set and the set itself are identified and I will follow this practice. Instead of speaking about elements of the subset of a certain type, I will speak about the elements of a certain type, etc. The use of names for subsets allows us to specify in each rule from which category its arguments have to be taken, and to which category the resulting expression belongs. Thus, a *syntactic rule* $S_j$ has the following form:

If one has expressions $E_1, \ldots, E_n$ of the categories $C_1, \ldots, C_n$ respectively, then one can form the expression $S_j(E_1, \ldots, E_n)$ of category $C_{n+1}$.

An equivalent formulation is:

Rule $S_j$ is a function from $C_1 \times \cdots \times C_n$ to $C_{n+1}$;

i.e. $S_j : C_1 \times \cdots \times C_n \to C_{n+1}$.

Suppose that a certain rule $S_j$ is defined as follows:

$S_j : C_1 \times C_2 \to C_3$, where $S_j(E_1, E_2) = E_1 \cdot E_2$.

This means that $S_j$ concatenates its arguments. Then our interpretation of the principle says that the meaning of $E_1 \cdot E_2$ has to be built up from the meanings of $E_1$ and $E_2$. A case like this constitutes the most elementary version of the principle. A compound expression is divided into real sub-expressions, and the meaning of the compound expression is built up from the meanings of these sub-expressions. In such a case the formalization coincides with the simplest, most intuitive conception of the principle: parts are visible as parts. But in some situations one might wish to consider as part an expression which is not visible as a part. We will meet several examples in later chapters. One example is the phenomenon of discontinuous constituents. The phrase *take away* is not a subphrase of *take the apple away*; it is not visible as a part. Nevertheless, one might here
wish to consider it as a unit which contributes to the meaning of take the apple away, i.e. as a part in the sense of the principle. The above definition of 'part' gives the possibility to do so. If the phrase take the apple away is produced by means of a rule which takes as arguments the phrases the apple and take away, then take away is indeed a part in the sense of the definition. The definition generalizes the principle for rules which are not just a concatenation operation, and consequently the parts need not be visible in the expression itself.

There are no restrictions on the possible effect of the rules \( S_j \). They may concatenate, insert, permute, delete, or alter (sub)expressions of their arguments in an arbitrary way. A rule may even introduce symbols which do not occur in its arguments. Such symbols are called synctegorematic symbols. These are not considered as parts of the resulting expression in the sense of the principle, and therefore they do not contribute a meaning from which the meaning of the compound can be formed. I will assume in general that the rules are total (i.e. they are defined for all expressions of the required categories). In chapter 6 partial rules will be discussed.

The abstraction just illustrated implies that we have lost the most intuitive conception of the principle. But it is not unlikely that several authors who mention Frege's principle only have the most intuitive version in mind. In order to avoid confusion, I will call the more abstract version not 'Frege's principle', but 'the compositionality principle'. As for the simple rules, where only concatenation is used, our interpretation of the principle coincides with the most intuitive interpretation. In more complex cases, where it might not be intuitively clear what the parts are, our interpretation can be applied as well. If one wishes to stick to the most intuitive interpretation of the principle, one must use only concatenation rules. In that way one would restrict considerably the applicability of grammars satisfying the framework (see chapter 2, section 5).

So far we have not considered the possibility of ambiguities. It is not excluded that some expression \( E \) can be obtained both as \( E = S_i(E_1', ..., E_n') \) and as \( E = S_j(E_1', ..., E_m') \). In practice such ambiguities frequently arise. In a programming language (e.g. in Algol 68), the procedure identifier \textit{random} can be used to denote the process of randomly selecting a number, as well as to denote the number thus obtained. The information needed to decide which interpretation is intended, is present in the production tree of the program, where the expression \textit{random} is either of type 'real' or not. In natural languages ambiguities arise even among expressions of the same category.
Consider for instance the sentence *John runs or walks and talks*. Its meaning depends on whether *talks* is combined with *walks*, or with *runs or walks*. Also here, the information needed to solve the ambiguity is hidden in the production tree. In the light of such ambiguities, we cannot speak in general of the meaning of some expression, but only of its meaning with respect to a certain derivational history. If we want to apply the compositionality principle to some language with ambiguities, we should not apply it to the language itself, but to the corresponding language of derivational histories.

In computer science it is generally accepted that the derivational histories form the real input for the semantical interpretation. SCHWARTZ (1972, p.2) states:

> We have sufficient confidence in our understanding of syntactic analysis to be willing to make the outcome of syntactic analysis, namely the syntax tree representation of the program, into a standard starting point for our thinking on program semantics. Therefore we may take the semantic problem to be that of associating a value [...] with each abstract program, i.e. parse tree.

In the field of semantics of natural languages, it is also common practice not to take the expressions of the language themselves as input to the semantical interpretation, but structured versions of them. KATZ & FODOR (1963, p.503) write:

> Fig. 6 shows the input to a semantic theory to be a sentence $S$ together with a structural description consisting of the $n$ derivations of $S$, $d_1, d_2, \ldots, d_n$, one for each of the $n$ ways that $S$ is grammatically ambiguous.

The book of Katz and Fodor is one of the early publications about the position of semantics in transformational grammars. There has been a lot of discussion in that field concerning the part of the derivational history which may actually be used for the semantic interpretation. In the so-called 'standard theory' only a small part of the information is used: the 'deep structure'. In the 'extended standard theory', one also uses the information which 'transformations' are applied, and what the final outcome, the 'surface structure', is. In the most recent proposals, the view on syntax and its relation with semantics is rather different.

As a matter of fact, neither Schwartz, nor Katz and Fodor use the same (semantic) framework we have. They are quoted here to illustrate that the idea of using information from the derivational history as input to the semantic component is not unusual.

Let us now turn to the phrase 'composed from the meanings of its parts'. Consider again a rule $S$, which allows us to form the compound
expression \( S_i(E_1, \ldots, E_n) \) from the expressions \( E_1, \ldots, E_n \). Assume moreover that the meanings of the \( E_k \) are the semantical objects \( D_k \). According to the principle, the information we are allowed to use for building the meaning of the compound expression consists in the meanings of the parts of the expression and the information which rule was applied. As usual, 'rule' is intended to take into account the order of its arguments. So the meaning of a compound expression is determined by an \( n \)-tuple of meanings (of its parts) and the information of the identity of the rule. This is in fact the only information which may be used. If one would be allowed to use other information (e.g. syntactic information concerning the parts), the principle would not express the whole truth, and not provide a sufficient condition. Thus the principle would tend to become a hollow phrase.

As argued for above, I interpret the compositionality principle as stating that the meaning of a compound expression is determined completely by the meanings of its parts and the information which syntactic rule is used. This means that for each syntactic rule there is a function on meanings which yields the meaning of a compound expression when it is applied to the meanings of the parts of that expression. So for each syntactic rule there is a corresponding semantic operation. Such a correspondence is not unusual; it can be found everywhere in mathematics and computer science. If one encounters for instance a definition of the style 'the function \( f \circ g \) is defined by performing the following calculations using \( f \) and \( g \)', then one sees in fact a syntactic and a semantic rule. The syntactic rule introduces the operator \( \circ \) between functions and states that the result is again a function, whereas the semantic rule tells us how the function should be evaluated.

If we use the freedom allowed by the \( p \)-principle at most, we may associate with each syntactic rule \( S_i \) a distinct semantic operation \( T_i \). So the most general description of the situation is as follows. The meaning of an expression formed by application of \( S_i \) to \( (E_1, \ldots, E_n) \) can be obtained by application of operator \( T_i \) to \( (D_1, \ldots, D_n) \), where \( D_j \) is the meaning of \( E_j \). These semantic operators \( T_i \) may be partially defined functions on the set \( D \) of meanings, since \( T_i \) has to be defined only for those tuples from \( D \) which may arise as argument of \( T_i \). These are those tuples which can arise as meanings of arguments of the syntactic rule \( S_i \) which corresponds with \( T_i \). In this way the set \( E \) leaves a trace in the set \( D \). The set of meanings of the expressions of some category forms a subset of \( D \) which becomes the set of possible arguments for some semantic rule. Thus the domain \( D \) obtains a structure which is closely related to the structure of the syntactic domain.
E. Our formalization of the compositionality principle may at this stage be summarized as follows.

Let \( S : C_1 \times \ldots \times C_n \rightarrow C_{n+1} \) be a syntactic rule, and \( M : E \rightarrow D \) be a function which assigns a meaning to an expression with given derivational history. Then there is a function \( T : M(C_1) \times \ldots \times M(C_n) \rightarrow M(C_{n+1}) \) such that \( M(S(E_1, \ldots, E_n)) = T(M(E_1), \ldots, M(E_n)) \).

In the process of formalizing the principle of compositionality we have now obtained a special framework. The form of the syntactic rules and the use of sorts give the syntax the structure of, what is called, a 'many-sorted algebra'. The correspondence between syntax and semantics implicates that the semantic domain is a many-sorted algebra of the same kind as the syntactic algebra. The meaning assignment is not based upon the syntactic algebra itself, but on the associated algebra of derivational histories. The principle of compositionality requires that meaning assignment is a homomorphism from that algebra to the semantic algebra. Note that the principle, which is formulated as a principle for semantics, has important consequences not only for the semantics, but also for the syntax.

The approach described here, is closely related to the framework developed by the logician Richard Montague for the treatment of syntax and semantics of natural languages (Montague 1970b). It is also closely related to the approach propagated by the group called 'Adj' for the treatment of syntax and semantics of programming languages (Adj 1977, 1979). Consequently frameworks related to the one described here can be found in the publications of authors following Adj (for references see Adj 1979), or following Montague (for references see Dowty, Wall & Peters 1981, or the present book). The conclusion that the principle of compositionality requires an algebraic approach is also given by Mazurkiewics (1975) and Milner (1975), without, however, developing some framework. The observation that there is a close relationship between the frameworks of Adj and Montague, was independently made by Markus & Szots (1981), Andreka & Sain (1981), and van Emde Boas & Janssen (1979).

4. AN ALGEBRAIC FRAMEWORK

In this section I will develop the framework sketched in section 3, and arguments concerning the practical use of the framework will influence this further development. The mathematical theory of the framework will be investigated in chapter 2.

The central notions in our formalization of the principle of
compositionality are 'many-sorted algebra' and 'homomorphism'. An algebra consists of some set (the elements of the algebra) and a set of operations defined on those elements. A many-sorted algebra is a generalization of this. It consists of a non-empty set $S$ of sorts (types, modes, or categories), for each sort $s \in S$ a set $A_s$ of elements of that sort ($A_s$ is called the carrier of sort $s$), and a collection $(F_{\gamma})_{\gamma \in \Gamma}$ of operations which are mappings from cartesian products of specified carriers to a specified carrier. So in order to determine a many-sorted algebra, one has to determine a 'sorted' family of sets and a collection of operators. This should explain the following definition.

4.1. **DEFINITION.** A many-sorted algebra $A$ is a pair $\langle (A_s)_{s \in S}, F \rangle$, where
1. $S$ is a non-empty set, its elements are called the sorts of $A$.
2. $A_s$ is a set (for each $s \in S$), the carrier of sort $s$.
3. $F$ is a collection of operators defined on certain $n$-tuples of sets $A_s$, where $n > 0$.

4.1. **END.**

Structures of this kind have been defined by several authors, using different names; the name 'many-sorted algebra' is borrowed from ADJ(1977). Notice that in the above definition there are hardly any restrictions on the sets and operators. The carriers may be non-disjunct, the operators may perform any action, and the sets involved (except for $S$) may be empty.

In order to illustrate the notion 'many-sorted algebra', I will present three examples in an informal way. I assume that these examples are familiar, and I will, therefore, not describe them in detail. The main interest of these examples is that they illustrate the notion of a many-sorted algebra. The examples are of a divergent nature, thus illustrating the generality of this notion.

4.2. **EXAMPLE:** Real numbers

Let us consider the set of real numbers as consisting of two sorts, $\text{Neg}$ and $\text{Pos}$. The carrier $R_{\text{Neg}}$ of sort $\text{Neg}$ consists of the negative real numbers, the carrier $R_{\text{Pos}}$ of sort $\text{Pos}$ of the positive real numbers, zero included. An example of an operation is $\text{sqrt}: R_{\text{Pos}} \rightarrow R_{\text{Pos}}$, where $\text{sqrt}$ yields the square root of a positive number. For $R_{\text{Neg}}$ there is no corresponding operation. Since we consider (in this example) the real numbers as a two-sorted algebra, there are two operations for squaring a number. One for squaring a positive number ($\text{sqpos}: R_{\text{Pos}} \rightarrow R_{\text{Pos}}$) and one for squaring a
negative number \((\text{sqneg} : \mathbb{R}_{\text{Neg}} \to \mathbb{R}_{\text{Pos}})\). Since these two operations are closely related, we may use the same symbol for both operations: 

\[ ( )^2 \]

4.3. **EXAMPLE: Monadic Predicate Logic**

Sorts are \(\text{Atom}, \text{Pred}\) and \(\text{Form}\). The carrier \(A_{\text{Atom}}\) of sort \(\text{Atom}\) consists of the symbols \(a_1, a_2, \ldots\) and the carrier \(A_{\text{Pred}}\) of the sort \(\text{Pred}\) consists of the predicate letters \(P_1, P_2, \ldots\). The carrier \(A_{\text{Form}}\) consists of formulas like \(P_1(a_1), \neg P_1(a_1),\) and \(P_1(a_2) \lor P_2(a_3)\). Two examples of operators are as follows.

1. The operation \(\text{Ap} : A_{\text{Pred}} \times A_{\text{Atom}} \to A_{\text{Form}}\). \(\text{Ap}\) assigns to predicate \(P\) and atom \(a\) the formula where \(P\) is applied to \(a\); viz. \(P(a)\).

2. The operation \(\text{Disj} : A_{\text{Form}} \times A_{\text{Form}} \to A_{\text{Form}}\). \(\text{Disj}\) assigns to two formulas \(\phi\) and \(\psi\) their disjunction \(\phi \lor \psi\).

Notice that (in the present algebraization) the brackets (,), and the disjunction symbol \(\lor\) are syncategorematic symbols.

4.4. **EXAMPLE: English**

Examples of sorts are \(\text{Sentence}, \text{Verb phrase},\) and \(\text{Noun phrase}\). The carrier of sort \(\text{Sentence}\) consists of the analysis trees of English sentences, and the carriers of other sorts of trees for expressions of other sorts.

An example of an operator is \(T_{\text{Neg}} : \text{Sentence} \to \text{Sentence}\). The operator \(T_{\text{Neg}}\) assigns to an analysis tree of an English sentence the analysis tree of the negated version of that sentence. An explicit and complete description of this algebra I cannot provide. This example is mentioned to illustrate that complex objects like trees can be elements of an algebra.

4.4. END.

As explained in the previous section, we do not assign meanings to the elements of the syntactic algebra itself, but to the derivational histories associated with that algebra. These derivational histories form an algebra: if expressions \(E_1\) and \(E_2\) can be combined to expression \(E_3\), then the derivational histories of \(E_1\) and \(E_2\) can be combined to a derivational history of \(E_3\). So the derivational histories constitute a (many-sorted) set in which certain operations are defined. Hence it is an algebra. The nature of the operations of this algebra will become evident when we consider below representations of derivational histories.

Suppose that a certain derivational history consists of first an application of operator \(S_1\) to basic expressions \(E_1\) and \(E_2\), and of next an
application of $S_2$ to $E_3$ and the result of the first step. A description like this of a derivational history is not suited to be used in practice (e.g. because of its verbosity). Therefore formal representations for derivational histories will be used (certain trees or certain mathematical expressions).

In Montague grammar one usually finds trees as representation of a derivational history. The history described above is represented in figure 1. Variants of such trees are used as well. Often the names of the rules are not mentioned (e.g. $S_1, S_2$), but their indices (viz. 1, 2). Sometimes the rule, or its index is not mentioned, but the category of the resulting expression. Even the resulting expressions are sometimes left out, especially when the rules are concatenation rules (figure 2). The relation between the representations of derivational histories and the expressions of the languages is obvious. In figure 1 one has to take the expression labelling the root of the tree, and in figure 2 one has to perform the mentioned operations. (For this kind of trees, it usually amounts to a concatenation of the expressions mentioned at the leaves (i.e. end-nodes)).

![Figure 1. Representation of a derivational history](image1)

![Figure 2. Another representation of the same derivational history](image2)

An alternative representation originates from the field of algebra. Derivational histories are represented by a compound expressions, consisting of basic expressions, symbols for the operators, and brackets. The derivational history from figure 1 is represented by the expression:

$$S_2(S_1(E_1, E_2), E_3).$$

Such expressions are called terms. The algebra of terms corresponding with algebra $A$ is called the term algebra $T_A$. From a term one obtains an expression of the actual language by evaluating the term, i.e. by application of the operators (corresponding with the operator symbols) to the mentioned arguments. The sorts of the term algebra $T_A$ are identical to the sorts of $A$, the operators are concatenation operators on terms. Note that all these
different representations mathematically are equivalent.

MONTAGUE (1970b) introduced the name 'disambiguated language' for the algebra of derivational histories. The relation between the disambiguated language and the language under consideration (he calls it R) is completely arbitrary in his approach. The only information he provides is that it is a binary relation with domain included in the disambiguated language (MONTAGUE 1970b, p.226). From an algebraic viewpoint this arbitrariness is very unnatural, and therefore I restrict this relation in the way described above (evaluating the term, or taking the expression mentioned at the root). This is a restriction on the framework, but not on the class of languages that can be described by the framework (see chapter 2).

The tree representations are the most suggestive representations, and they are most suitable to show complex derivational histories. The term representations take less space and are suitable for simple histories and in theoretical discussions. In the first chapters I will mainly use terms, in later chapters trees. According to the framework we have to speak about the meaning of an expression relative to a derivational history. In practice one often is sloppy and speaks about the meaning of an expression (when the history is clear from the context, or when there is only one).

After this description of the notion of a many-sorted algebra, I will introduce the other central notion in our formalization of the principle of compositionality: the notion 'homomorphism'. It is a special kind of mapping between algebras, and therefore first mappings are introduced.

4.5. DEFINITION. By a mapping \( m \) from an algebra \( A \) to an algebra \( B \) is understood a mapping from the carriers of \( A \) to the carriers of \( B \). Thus:

\[
\begin{align*}
  m & : \bigcup A_s \rightarrow \bigcup B_s \\
  s & \in S_A \\
  & \text{and} \\
  s & \in S_B
\end{align*}
\]

4.5. END.

A mapping is called a homomorphism if it respects the structures of the algebras involved. This is only possible if the two algebras have a similar structure. By this is understood that there is a one-one correspondence between the sorts in the one algebra and in the other algebra, and between the operators in the one algebra and in the other algebra. The latter means that if an operator is defined for certain sorts in the one algebra, then the corresponding operator is defined for the corresponding sorts in the other algebra. This should describe the essential aspects of the technical
The notion of 'similarity' of two algebras; a formal definition will be given in chapter 2. Then the definition of a homomorphism given below, will be adapted accordingly, and the slight differences with the definitions in the literature (Montague, Adj) will be discussed.

4.6. DEFINITION. Let \( A = \langle A_s \mid s \in S \rangle \) and \( B = \langle B_t \mid t \in T \rangle \) be similar algebras. A mapping \( h \) from \( A \) to \( B \) is called a homomorphism if the following two conditions are satisfied:

1. \( h \) respects the sorts, i.e., \( h(A_s) = B_t \), where \( t \) is the sort of \( B \) which corresponds to sort \( s \) of \( A \).
2. \( h \) respects the operators, i.e., \( h(F(a_1, \ldots, a_n)) = G(h(a_1), \ldots, h(a_n)) \) where \( G \in \mathcal{G} \) is the operator of \( B \) which corresponds to \( F \in \mathcal{F} \).

4.6. END

Now that the notions of a many sorted algebra and of a homomorphism are introduced, I will present two detailed examples.

4.7. EXAMPLE: Fragment of English.

Syntactic Algebra

The syntactic algebra \( E \) consists of some English words and sentences.

I. Sorts
\[ S_E = \{\text{Sent, Subj, Verb}\} \]

II. Carriers
\[ E_{\text{Subj}} = \{\text{John, Mary, Bill}\} \]
\[ E_{\text{Verb}} = \{\text{runs, talks}\} \]
\[ E_{\text{Sent}} = \{\text{John runs, Mary runs, Bill runs, John talks, Mary talks, Bill talks}\} \]

III. Operations
\[ C: E_{\text{Subj}} \times E_{\text{Verb}} \rightarrow E_{\text{Sent}} \]
deﬁned by \( C(a, b) = ab \)

So \( C(\text{John, runs}) \) is obtained by concatenating \( \text{John} \) and \( \text{runs} \), thus yielding \( \text{John runs} \).

Semantic Algebra

The semantic algebra \( M \) consists of model-theoretic entities, such as truth values and functions.
I. Sorts

\[ S = \{ e, t, \langle e, t \rangle \} \]

So there are three sorts: two sorts being simple symbols (\( e \sim \) entity, 
\( t \sim \) truthvalue), and the compound symbol \( \langle e, t \rangle \) (function from \( e \) to \( t \)).

II. Carriers

\[ M = \{ \text{true, false} \} \]

The carrier \( M \) consists of two elements, the truth-values \text{true} and \text{false}.

\[ M = \{ e_1, e_2, e_3 \} \]

The set \( M \) consists of three elements: \( e_1, e_2, \) and \( e_3 \).

\[ M_{\langle e, t \rangle} = (M, M) \]

The carrier \( M_{\langle e, t \rangle} \) consists of all functions from \( M \) to \( M \). This set has 8 elements.

III. Operations

There is one operation in \( M \): the operation \( F \) of function application.

\[ F : M \times M_{\langle e, t \rangle} \rightarrow M, \]

where \( F(\alpha, \beta) \) is the result of application of \( \beta \) to argument \( \alpha \).

The algebras \( E \) and \( M \) are similar. The correspondence of sorts is

\( \text{Subj} \sim e, \text{Verb} \sim \langle e, t \rangle, \text{Sent} \sim t, \) and operation \( C \) corresponds to \( F \). Although the algebras \( E \) and \( M \) are similar, they are not the same. For instance, the number of elements in \( E_{\text{verb}} \) differs from the number of elements in \( M_{\langle e, t \rangle} \).

There are a lot of homomorphisms from \( T_E \) (the derivational histories in \( E \), to \( M \). An example is as follows.

Let \( h \) be defined by

\[ h(\text{John}) = e_1, \quad h(\text{Bill}) = e_2, \quad h(\text{Mary}) = e_3 \]

\( h(\text{runs}) \) is the function \( f_1 \) which has value \text{true} for all \( e \in M \).

\( h(\text{talks}) \) is the function \( f_2 \) which has value \text{false} for all \( e \in M \).

Furthermore, we define \( h \) for the compound terms.

\[ h(\text{C(John, runs)}) = h(\text{C(Mary, runs)}) = h(\text{C(Bill, runs)}) = \text{true} \]

\[ h(\text{C(John, talks)}) = h(\text{C(Mary, talks)}) = h(\text{C(Bill, talks)}) = \text{false} \]

The function \( h \), thus defined, is a homomorphism because

1. \( h(T_E, \text{Subj}) \in M \), \( h(T_E, \text{Verb}) \in M_{\langle e, t \rangle} \), \( h(T_E, \text{Sent}) \in M \)

2. \( h(\text{C}(\alpha, \beta)) = F(h(\alpha), h(\beta)) \) for all subjects \( \alpha \) and verbs \( \beta \).

It is easy to define other homomorphisms from \( T_E \) to \( M \). Notice that once

\( h \) is defined for \( T_E, \text{Subj} \) and for \( T_E, \text{Verb} \), then there is no choice left for

the definition of \( h \) for \( T_E, \text{Sent} \) (provided that we want \( h \) to be a homo-

morphism).
4.8. **EXAMPLE: Number denotations**

**Syntactic Algebra**

The algebra $\text{Den}$ of natural number denotations is defined as follows

I. **Sorts**

$S_{\text{Den}} = \{\text{digit}, \text{num}\}$

II. **Carriers**

$D_{\text{digit}} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$D_{\text{num}} = \{0, 1, 2, 3, \ldots, 10, 11, \ldots, 01, 02, \ldots, 001, \ldots, 007, \ldots\}$

So $D_{\text{num}}$ is the set of all number denotations, including denotations with leading zero's. Notice that $D_{\text{digit}} \subseteq D_{\text{num}}$.

III. **Operators**

There is one operation.

$C: D_{\text{num}} \times D_{\text{digit}} \rightarrow D_{\text{num}}$

where $C$ is defined by $C(a, b) = ab$.

So $C$ concatenates a number with a digit.

**Semantic Algebra**

The algebra $\text{Nat}$ of natural numbers is defined as follows

I. **Sorts**

$S_{\text{Nat}} = \{d, n\}$.

II. **Carriers**

$N_d$ consists of the natural numbers up to nine (zero and nine included)

$N_n$ consists of all natural numbers.

III. **Operations**

There is one operation:

$F: N_n \times N_d \rightarrow N_n$

where $F$ is defined as multiplication of the element from $N_n$ by ten, followed by addition of the element from $N_d$.

A natural homomorphism $h$ from $\text{Den}_{\text{Den}}$ to $\text{Nat}$ is the mapping which associates with the derivational history of a digit or number denotation the corresponding number. Then $h(C(0, 7))$ and $h(7)$ are both mapped onto the number seven. That this $h$ is a homomorphism follows from the fact that $F$ describes the semantic effect of $C$, e.g. $h(C(2,7)) = F(h(2), h(7))$.

4.8. END
Syntax is an algebra, semantics is an algebra, and meaning assignment is a homomorphism; that is the aim of our enterprise. But much work has to be done in order to proceed in this way. Consider the two examples given above. The carriers were defined by specifying all their elements, the homomorphisms were defined by specifying the image of each element, and the operations in the semantic algebra were described by means of full English sentences. For larger, more complicated algebras this approach will be very impractical. Therefore a lot of technical tools will have to be introduced before we can deal with an interesting fragment of natural language or programming language. Consider again the first example (i.e. 4.7). The semantic operation corresponding to the concatenation of a Subj and a Verb was described as the application of the function corresponding to the verb to the element corresponding to the subject. One would like to use standard notation from logic and write something like Verb(Subj). Thus one is tempted to use some already known language in order to describe a semantic operation. This is precisely the method we will employ. If we wish to define the meaning of some fragment of a natural language, or of a programming language, we will not describe the semantic operations in the meta-language (for instance a mathematical dialect of English), but use some formal language, the meaning of which has already been defined somehow: we will use some formal or logical language. Thus the meaning of an expression is defined in two steps: by translating first, and next interpreting, see figure 3.

![Diagram](image)

**Figure 3** Meaning assignment in two steps.

Figure 3 illustrates that the semantics of the fragment of English is defined in a process with two stages. But is this approach in accordance with our algebraic aim? Is the mapping from the term algebra corresponding
with the syntax of English to the algebra of meanings indeed a homomorphism? The answer is that we have to obey certain restrictions, in order to be sure that the two-stage process indeed determines a homomorphism. The translation should be a homomorphism from the term algebra for English to the logical language and the interpretation of the logical language should be a homomorphism as well. Then, as is expressed in the theorem below, the composition of these two mappings is a homomorphism.

4.9. **Theorem.** Let \( A, B, \) and \( C \) be similar algebras, and \( h: A \rightarrow B \) and \( g: B \rightarrow C \) homomorphisms. Define the mapping \( h \circ g: A \rightarrow C \) by \( (h \circ g)(a) = g(h(a)) \). Then \( h \circ g \) is a homomorphism.

**Proof.**

1. \( (h \circ g)(A_s') \subseteq g(h(A_s')) \subseteq C_{s''} \), where \( s' \) and \( s'' \) are the sorts in \( B \) and \( C \) corresponding with \( s \).

2. Let \( C, H \), be the operators in \( B \) and \( C \) corresponding with \( F, \gamma \). Then
   \[(h \circ g)(F, \gamma(a_1, \ldots, a_n)) = g(h(F, \gamma(a_1, \ldots, a_n))) = g(C(h(a_1), \ldots, h(a_n))) = H(g(h(a_1), \ldots, g(h(a_n)))) = H((h \circ g)(a_1), \ldots, (h \circ g)(a_n))\]

4.9. **End.**

The semantical language does not always contain basic operators which correspond to the operators in the syntax. In the example concerning natural number denotations there is no basic arithmetical operator which corresponds to the syntactic operation of concatenation with a digit. I described the semantic operator by means of the phrase 'multiplication of the element from \( N \) with ten; followed by an addition with the element of \( N' \). One is tempted to indicate this operation not with this compound phrase, but with something like '10 \times \text{number} + \text{digit}'. One wishes to use a compound expression from the language of arithmetic for the semantic operation which corresponds to the concatenation operation, i.e. to build new operations from old ones.

The situation I have just described, is the one which almost always arises in practice. One wishes to define the semantics of some language. The set of semantic objects has some 'natural' structure of its own, and a 'natural' semantical language which reflects this structure. So this 'natural' semantical language has not the same algebraic structure as the language for which we wish to describe the semantics. Therefore we use the semantical language (usually some kind of formal or logical language) to build a new algebra, called a derived algebra. We make new operations by
forming compound expressions which correspond with the syntactic operations of the language for which we wish to describe the semantics. This situation is presented in figure 4; the closed arrows denote mappings, the dotted arrows indicate the construction of a new algebra by means of the introduction of new operations (built from old ones).

![Diagram](image)

**Figure 4.** Meaning assignment using derived algebras

In this way, we have derived a new syntactic algebra from the syntactic algebra of the logical language. The syntactic algebra of which we wish to define the semantics is translated into this derived algebra. Now the question arises whether this approach is in accordance with our aim of defining some homomorphism from the syntactic algebra to the collection of meanings. The theorem that will be mentioned below guarantees that under certain conditions this is the case. The interpretation of the logical language has to be a homomorphism, and the method by which we obtain the derived algebra is restricted to the introduction of new operators by composition of old operators. Such operators are called polynomials; for a formal description see chapter 2. If these conditions are satisfied, then the interpretation homomorphism for the logical language is also an interpretation homomorphism of the derived algebra (when restricted to this algebra). Composition of this interpretation homomorphism with the translation homomorphism gives the desired homomorphism from the language under consideration to its meanings. The theorem is based upon MONTAGUE (1970b), for its proof see chapter 2.
4.10. **THEOREM.** Let $A$ and $B$ be similar algebras and $h: A \rightarrow B$ a homomorphism onto $B$. Let $A'$ be an algebra obtained from $A$ by means of introduction of polynomially defined operators over $A$.

Then there is a unique algebra $B'$ such that $h$ is a homomorphism from $A'$ onto $B'$.

4.10. END

Finally I wish to make some remarks about the translation into some logical language. As I explained when introducing this intermediate step, it is used as a tool for defining the homomorphism from the syntactic algebra to the semantic one. If we would appreciate complicated definitions in the meta language, we might omit the level of a translation. It plays no essential role in the system, it is there for convenience only. If convenient, we may replace a translation by another translation which gets the same interpretation. We might even use another logical language. So in a Montague grammar there is nothing which deserves the name of the logical form of an expression. The obtained translation is just one representation of a semantic object, and might freely be interchanged with some other representation. **KIFNAN & FALTZ (1978),** in criticizing the logical form obtained in a Montague grammar, criticize a notion which does not exist in Montague grammar.

5. MEANINGS

5.1. **Introduction**

In this section some consequences are considered of the requirement of a homomorphic mapping from the syntactic term algebra to the semantic algebra. These consequences are considered for three kinds of language: natural languages, programming languages and logical languages. It will appear that the requirement of associating a single meaning with each expression of the language helps us, in all three cases, to find a suitable formalization of the notion of meaning. Furthermore, an example will be considered of an approach where the requirement of a homomorphic relation between syntax and semantics is not obeyed.
5.2. Natural Language

Consider the phrase \textit{the queen of Holland}, and assume that it is used
to denote some person (and not an institution). Which person is denoted, de-
pends on the moment of time one is speaking about. This information can
usually be derived from the linguistic context in which the expression oc-
curs. In

(1) \textit{The queen of Holland is married to Prince Claus}.
Queen Beatrix is meant, since she is the present queen. But in
(2) \textit{In 1910 the queen of Holland was married to Prince Hendrik}.
Queen Wilhelmina is meant, since she was the queen in the year mentioned.

So one is tempted to say that the meaning of the phrase \textit{the queen of Holland}
varies with the time one is speaking about. Such an opinion is, however,
not in accordance with our algebraic (compositional) framework. The approach
which leads to a single meaning for the phrase under discussion is to in-
corporate the source of variation into the notion of meaning. In this way
we arrive at the conception that the meaning of the phrase \textit{the queen of Holland}
is a function from moments of time to persons. For other expressions
(and probably also for this one) there are more factors of influence (place
of utterance, speaker,...). Such factors are called indices; a function with
the indices as domain is called an intension. So the meaning of an expres-
sion is formalized by an intension: our framework leads to an intensional
conception of meaning for natural language. For a more detailed discussion
concerning this conception, see LEWIS 1970. A logical language for dealing
with intensions is the language of 'intensional logic'. This language will
be considered in detail in chapter 3.

5.3. Programming Language

Consider the expression \texttt{x+1}. This kind of expressions occurs in almost
every programming language. It is used to denote some number. Which number
is denoted depends on the internal situation in the computer at the moment
of consideration. For instance, in case the internal situation of the com-
puter associates with \texttt{x} the value seven, then \texttt{x+1} denotes the number eight.
So one is tempted to say that the meaning of \texttt{x+1} varies. But this is not
in accordance with the framework. As in example 1, the conflict is resolved
by incorporating the source of variation into the notion of meaning. As
the meaning of an expression like \texttt{x+1} we take a function from computer
states to numbers. On the basis of this conception a compositional treatment
can be given of meanings of computer languages (see chapter 10). States of
the computer can be considered as an example of an index, so also in this
case we use an intensional approach to meaning. In the publications of Adj
a related conception of the meaning of such expressions is given, although
without calling it an intension (see e.g. AJJ 1977, 1979).

Interesting in the light of the present approach is a discussion in
PRATT 1979. Pratt discusses two notions of meaning: a static notion (an ex-
pression obtains once and for all a meaning), and a dynamic notion (the
meaning of an expression varies). He argues that (what he takes as) a static
notion of meaning has no practical purpose because we frequently use expres-
sion obtains once and for all a meaning), and a dynamic notion (the
of time. Therefore he develops a special logic for the treatment of seman-
tics of programming languages, called 'dynamic logic'. But on the basis of
our framework, we have to take a 'static' notion of meaning. By means of
intensions we can incorporate all dynamics into such a framework. Pratt's
dynamic meanings might be considered as a non-static version of intensional
logic.

5.4. Predicate Logic

It is probably not immediately clear how predicate logic fits into the
algebraic framework. PRATT (1979,p.55) even says that 'there is no function
F such that the meaning of \( \forall x p \) can be specified with a constraint of the
form \( \mu(\forall x p) = F(\mu(\rho))' \). In our algebraic approach we have to provide for
such a meaning function \( \nu \) and operator \( F \).

Let us consider the standard (Tarskian) way of interpreting logic. It
roughly proceeds as follows. Let \( \Theta \) be a model and \( g \) be an \( \Theta \)-assignment. The
interpretation in \( \Theta \) of a formula \( \phi \) with respect to \( g \), denoted \( \phi^g \), is then
recursively defined. One clause of this definition is as follows (here \( I \)
denotes the truth value for truth).

\[ [\phi \land \psi]^g = 1, \text{if } \phi^g \text{ is } 1 \text{ and } \psi^g \text{ is } 1. \]

This suggest that the meaning of \( \phi \land \psi \) is a truth value, which is obtained
out of the truth values for \( \phi \) and for \( \psi \). Another clause of the standard way
of interpretation is not compatible with this idea.

\[ [\exists x \phi(x)]^g = 1, \text{if there is a } g' \leadsto g \text{ such that } [\phi(x)]^{g'} = 1. \]

(Here \( g' \leadsto g \) means that \( g' \) is the same assignment as \( g \) except for the
possible difference that \( g'(x) \neq g(x) \)).

This clause shows that the concept of meaning being a truth value is too
simple for our algebraic framework. One cannot obtain the truth value of
$\exists x \phi(x)$ (for a certain value of $g$) out of the truth value of $\phi(x)$ (for the same $g$). If we wish to treat predicate logic in our framework, we have to find a more sophisticated notion of meaning for it.

Note that there is not a single truth value in the semantic domain which corresponds with $\phi(x)$. Its interpretation depends on the interpretation of $x$, and in general on the interpretation of the free variables in $\phi$, and therefore on $g$. In analogy with the previous examples, we incorporate the variable assignment into the conception of meaning. The meaning of a formula is a function from variable assignments to truthvalues, namely the function which yields 1 for an assignment in case the expression is true for that assignment. With this conception, we can easily build the meaning of $\phi \land \psi$ out of the meaning of $\phi$ and of $\psi$: a function which yields 1 for an assignment iff both the meanings of $\phi$ and of $\psi$ yield 1 for that assignment. The formulation becomes simpler by adopting a different view of the same situation. A function from assignments to truthvalues can be considered as the characteristic function of a set of assignments. Using this, we may formulate an alternative definition: the meaning of a formula is a set of variable assignments (namely those for which the formula gets the truth value 1). Let $M$ denote the meaning assignment function. Then we have:

$$M(\phi \land \psi) = M(\phi) \cap M(\psi).$$

For the other connectives there are related set theoretical operations. Thus this part of the semantic domain gets the structure of a Boolean algebra.

For quantified formulas we have the following formulation.

$$M(\exists x \phi) = \{ h \mid h \models x \Rightarrow g \text{ and } g \in M(\phi) \}.$$

Let $C_x$ denote the semantical operation described at the right hand side of the = sign, i.e. $C_x$ is the operation 'extend the set of assignments with all $x$ variants'. The syntactic operation of writing $\exists x$ in front of a formula now has a semantic interpretation: namely apply $C_x$ to the meaning of $\phi$.

$$M(\exists x \phi) = M(\exists x)(M(\phi)) = C_x M(\phi).$$

In this algebraization there are infinitely many operations which introduce the existential quantifier. One might wish to go one step further and produce $\exists x$ from $\exists$ and $x$. This would require that given the meaning of a variable (being a function from assignments to values) we are able to determine of which variable it is a meaning. This is not an attractive algebraic operation, and therefore this last step is not made. I conclude that we have obtained a compositional interpretation of predicate logic: a homomorphism to some semantic algebra. One might say that it shows how we have to look
at the Tarakian interpretation of logic in order to give it a compositional perspective.

The view on the semantics of predicate logic presented here is not new. Some logic books are based on this approach in which the meaning of a quantified formula is a set of assignments (MOOK 1976, p.196, KREISEL & KRIVINE 1976, p.17). The investigations on the algebraic structure of predicate logic constitute a special branch of logic: the theory of cylindric algebras. It requires a shift of terminology to see that the kinds of structures studied there is the same as those introduced here. An assignment can be considered as an infinite tuple of elements in the model: the first element of the tuple is the value for the first variable, etcetera. Thus an assignment can be considered as a point in an infinite dimensional space. So if \( \phi \) holds for a set of assignments, then \( \phi \) is interpreted as the set of corresponding points in this universe. The operator \( C_x \) applied to a point \( p \) causes that all points are added which differ from \( p \) only in their \( x \)-coordinate. Geometrically speaking, a single point extends to an infinite stick. If \( C_x \) is applied to a set consisting of a circle area, then this is extended to a cylinder. Because of this effect, \( C_x \) is called a cylindrification operator, and in particular, the \( x \)-th cylindrification. (see fig.5) The algebraic structure obtained in connection with predicate logic is called a cylindric set-algebra. These algebras and their connection with logic are studied in the theory of cylindric algebras (see HENKIN, MOOK & TARSKI 1971).

The original motivation for studying cylindric algebras was a technical one. Cylindric algebras were introduced to make the application of the powerful tools of algebra possible in studying logics, as can be read in HENKIN, MOOK & TARSKI (1971, p.1):

This theory [...] was originally designed to provide an apparatus for an algebraic study of first order, predicate logic.

New in the above discussion was the motivation which led us towards cylindric algebras. In my opinion, the compositional approach gives rise to a more direct introduction to this field than the existing one. Moreover, on the basis of the approach given above, it is not too difficult to find algebras for other order logics, such as intensional logic.

It cannot be said that the theory of cylindric algebras itself is a flourishing branch of logic nowadays. But the use of algebra is widespread in model theory (i.e. the branch of logic which deals with interpretations). Often one uses the terminology and techniques from universal algebra, as is evidenced by the amount of universal algebra in 'Model theory' by CHANG &
KEISLER (1973), and by the amount of model theory in 'Universal algebra' by
GRAETZER (1968). Results from one field are proven using methods from the
other field in VAN BENTHEM (1979b). Algebraic interpretations of several non-
classical logics are given by RASTIWA (1974). Important results concerning
modal logics are obtained, using algebraic techniques, by Blok (e.g. BLOK
1980).

![Diagram]

Figure 5. A cylindrification

The present discussion should not be understood as claiming that the
only legitimate way of studying (predicate) logic is by means of (cylindric)
algebras. There are a lot of topics concerning logic that can be studied,
and each has a natural viewpoint. For instance, if one is studying deduc-
tion systems, a syntactic point of view is the natural approach. One should
take that view which is the best for one's current aims. What I claim is
that, if one is studying semantics, then there has to be an algebraic
interpretation existing in the background, and one should take care that this interpretation is not violated by what one is doing.

5.5. **Strategy**

In all three examples discussed above, we followed the strategy of first investigating what a meaning should do, and then defining such an entity as the formalized notion of meaning which does that and which satisfies the compositionality principle. In all examples such an entity was obtained by giving the notion of meaning a sufficient degree of abstraction. By proceeding in this way (first investigating, then defining) we follow the advice of Lewis (1970, p. 5)

In order to say what a meaning is, we may first ask what a meaning does and then find something that does that.

5.6. **Substitutional Interpretation**

Next I will discuss an approach to the semantics of predicate logic which is not compositional with respect to the interpretation of quantifiers. For the interpretation of $\exists x \phi(x)$ an alternative has been proposed which is called the 'substitutional interpretation'. It says:

$\exists x \phi(x)$ is true iff there is some substitution $a$ for $x$ such that $\phi(a)$ is true.

Whether this definition is semantically equivalent to the Tarskian definition depends, of course, on whether the logical language contains a name for every element of the semantic domain or not. A definition like the above one can be found in two rather divergent branches of logic: in philosophical logic, and in proof theory.

In philosophical logic the substitutional interpretation has been put forward by R. Marcus (e.g. Marcus 1962). Her motivation was of an ontological nature. Consider sentence (3).

(3) *Pegasus is a winged horse.*

According to standard logic, (4) is a consequence of (3), and Marcus accepts this consequence.

(4) $\exists x (x$ is a winged horse).

She argues, however, that one might believe (3), without believing (5).

(5) There exists at least one thing which is a winged horse.

This opinion has as a consequence that the quantification used in (4) cannot be considered as an existential quantification in the ontological sense. The substitutional interpretation of quantifiers allows her to have (4) as
a consequence of (3), without being forced to accept (5) as a consequence.

Kripke (1976) discusses this approach in a more formal way. As syntax for the logic he gives the traditional syntax: $\exists x \phi(x)$ is produced from $\phi(x)$ by placing $\exists$ in front of it. According to such a grammar $\phi(a)$ certainly is not a part of $\exists x \phi(x)$. This means that the substitution interpretation is not a compositional interpretation (this was noticed by Tarski, as appears from a footnote in Partee (1973, p. 74)).

In proof theory the substitutional interpretation is given e.g. in Schütte 1977. In his syntax he constructs $\forall x \phi(x)$ from $\phi(a)$, where $a$ is arbitrary. So the formula $\forall x \phi(x)$ is syntactically rather ambiguous: It has as many derivations as there are expressions of the form $\phi(a)$. Given one such production, it is impossible to define the interpretation of $\forall x \phi(x)$ on the basis of the interpretation of the formula $\phi(a)$ from which $\forall x \phi(x)$ was built in the parse under consideration. It may be the case that $\forall x \phi(x)$ is false, and $\phi(a)$ is true for some $a$, but false for another one. So we see that the truth value of $\forall x \phi(x)$ cannot depend on the truth value of $\phi(a)$ for any single $a$. Hence in this case the substitutional interpretation does not satisfy the compositionality principle.

If one wishes to define the semantics in a compositional way, and to follow at the same time the substitutional interpretation of quantifiers, then the syntax has to contain an infinitistic rule which says that all expressions of the form $\phi(a)$ are part of $\forall x \phi(x)$. Such an infinitistic rule has not been proposed by authors which follow the substitutional interpretation.

6. MOTIVATION

In this section I will give several arguments for accepting the compositionality principle and the formalization given for it. I will give three kinds of arguments. The first kind is very general and argues for working within some mathematically defined framework. The second kind of arguments lists benefits of working with the present framework, and is based upon the properties of the framework. The third kind concerns the principle itself. As a matter of fact, this entire book is intended as a support for the algebraic formalization of the compositionality principle, and many of the arguments will be worked out in the remainder of this book.

Regarding the first kind of arguments: it is very useful to work with-
In some mathematically well defined framework. Such a standard framework gives rise to a language in which one can formulate observations, relations and generalizations. It is a point of departure for formulating extensions, restrictions and deviations. If one has no standard framework, then whenever one considers a new proposal, one has to start anew in obtaining intuitions concerning properties of the system, and to check whether old knowledge still holds. It is then difficult to see whether the proposals within some framework are in accordance with those in other frameworks, and whether they can be combined into a coherent treatment. If one wishes to design a computer program for Montague grammars, then one has to design for each proposed extension or variant a completely new program, unless all proposals fit into a single framework. This experience was my original motivation for the whole research presented in this book. But the final result is independent of this motivation: only at a few places programming considerations are mentioned (viz. here and in chapters 7 and 8).

The second kind of arguments is based upon the quality of the framework.

a) Elegance

The framework presented here is mathematically rather elegant. This is apparent especially from the fact that it is based upon two simple mathematical notions: many-sorted algebra and homomorphism. The important tool of a logical language is combined in an elegant way with these algebraic notions. One should, however, not confuse the notion of 'elegant' with 'elementary' or 'easy to understand'. That the system is elegant, is due to its abstractness, and this abstractness might be a source of difficulties in understanding the system. The insight obtained from the abstract view on the framework led to an answer to a question of PARTEE 1973 concerning restrictions on relative clause formation, see chapter 9 or JANSSEN 1981a. It also led to an application in a rather different direction by providing a semantics for Dik's functional grammar, see JANSSEN 1981b.

b) Generality

The framework can be applied to a wide variety of languages: natural, programming and logical languages. See chapter 3 for an application to logic, chapter 10 for an application to programming languages, and the other chapters of this book for applications to natural languages.

c) Restrictiveness

The framework gives rise to rather strong restrictions concerning the organization of syntax and semantics, and their mutual relation. The use
of polynomial operators especially constitutes a concrete, practical restriction. For a discussion of several deviations from the present framework, see chapters 5 and 6.

d) Comprehensibility

The argument given by Milner (see section 1) for designers of computing systems can be generalized to: 'If someone describes the semantics of some language, he should be able to think of the description as a composite of descriptions, in order that he may factor a semantic problem into smaller problems'. And what is said here for the designer of a system, holds at least as much for someone trying to understand the system. This property of the system is employed in the presentation of the fragment in chapter 4.

e) Power

The recursive definitions used in the framework allow us to apply the technique of induction. Statements concerning structures and expressions can be proved by using induction to the complexity of the elements involved. Especially in chapters 2 and 3 this power is employed.

f) Heuristic tool

A most valuable argument in favor of the principle and its formalization is its benefit for the practice of describing semantics of languages. Examples of this benefit, however, would require a detailed knowledge of certain proposals. Therefore some quotations have to suffice.

ADJ 1979 (p. 85) say about the algebraic approach:

The belief that the ideas presented here are key, comes from our experience over the last eight years in developing and applying these concepts.

Furthermore they say (op. cit. p. 88):

When one becomes familiar with such concepts (and the results concerning them) they provide a guide as to what one should look for, and as to how to formulate one's definitions and results.

Van Emde Boas & Janssen 1979 (p. 112) claim:

It will turn out that quite often some complicated description in a semantic treatment actually hides a deviation from the principle. Confronted with such a violation the principle sometimes suggests an alternative approach to the problematic situation which does obey the principle and solves the problem easier than thought to be possible. Such cases establish the value of the principle as a heuristic tool.

Both papers contain a lot of evidence for their claims. I will present several examples supporting them: concerning programming languages in chapter 10, and concerning natural languages in the other chapters.

The last kind of arguments concerns the principle itself.
g) No alternative

An important argument in favor of the principle is that there is no competing principle. Authors not working in accordance with the principle do not, as far as I know, put forward an alternative general principle with a mathematical formalization. The principles one finds in the literature are language-specific, or specific for a certain theory of languages, but never principles concerning a framework.

h) Widespread

As demonstrated in section 1, the principle of compositionality is widespread in sciences dealing with semantics; it arises in philosophy, linguistics, logic and computer science.

i) Psychology

An argument sometimes put forward is that the principle reflects something of the way in which human beings understand natural language. The principle explains how it is possible that a human being, with his finite brain, can understand a potentially infinite set of sentences. Or to say it in Frege's words (as translated by Geach & Stoothof (FREGE 1923, p.35)):

[... even a thought grasped by a terrestrial being for the first time can be put into a form of words which will be understood by someone to whom the thought is entirely new. This would be impossible, were we not able to distinguish parts in the thought corresponding to the parts of a sentence [...].

The last two arguments I do not consider as very strong. As for argument h), I think that the principle is so popular because it is so vague. There are many undefined words in the formulation of the principle, so that everybody can find his own interpretation in it. As for argument i), we know so little about the process in the human brain associated with learning or understanding natural language, that arguments concerning psychological relevance are no more than speculations. I would not like to have the mathematical attractiveness of the framework disturbed by further speculations of this nature. The most valuable arguments are, in my opinion, those concerning the elegance and power of the framework, its heuristic value, and the lack of a mathematically well defined alternative. So I adhere to the principle for the technical qualities of its formalization.

An argument not found above is the truth of the principle: a statement like 'The semantics of English is compositional'. Such an argument would not be convincing since it is circular. In section 5, I gave examples which illustrated that the principle, and especially the requirement of similarity, may lead us to a certain conception of meaning. And in section 3 I gave a
definition of the notion 'parts' which made it possible to have 'abstract parts'. So there is a large freedom: we may choose what the parts are of an expression, and what the meanings are of those parts. In such a situation it is not surprising that there is some choice which gives rise to a compositional treatment of the semantics. In the next chapter I will prove that it is possible within this framework to generate every recursively enumerable language, and to relate with every sentence any meaning we would like. If someone wishes to doubt the principle, this only seems possible if he has some judgements at forehand about what the parts of an expression are, and what their meanings are. In the light of the power and flexibility of the framework, it cannot be refuted by pointing out in some language a phenomenon which requires a non-compositional treatment. I expect that problematic cases can always be dealt with by means of another organization of the syntax, resulting in more abstract parts, or by means of a more abstract conception of meaning. The principle only has to be abandoned if it leads too often to unnecessarily complicated treatments.

As appears from this discussion, the principle of compositionality is not a principle about languages. It is a principle concerning the organization of grammars dealing both with syntax and semantics. The arguments given above for adhering to the principle, are not based on phenomena in languages, but on properties of grammars satisfying the framework. If one is not pleased with the power of the grammars, one might formulate severe restrictions within the framework. In the light of the examples to be given in chapter 5, it seems that the framework as it is, gives, from a practical viewpoint, already more than enough restrictions.