Compositionality *
with an appendix by B. Partee

Theo M.V. Janssen
Dept. of Math. and Comp. Sc., Univerity of Amsterdam
Plantage Muidergracht 24, 1018TV Amsterdam
theo@fwi.uva.nl

Abstract

The first topic of the paper is to provide a formalization of the principle of compositionality of meaning. A mathematical model (based upon universal algebra) is presented, and its properties are investigated. The second topic is to discuss arguments from the literature against compositionality (of Hintikka, Higginbotham, Pelletier, Partee, Schiffer and others). Methods are presented that help to obtain compositionality. It is argued that the principle is should not be considered an empirical verifyable restriction, but a methodological principle that describes how a system for syntax and semantics should be designed. The paper has an appendix by B. Partee on the compositional treatment of genitives.

*To appear in J.F.A.K. van Benthem & A. ter Meulen (eds.), *Handbook of Logic and Linguistics, Elsevier Science Publishers, Amsterdam*

1 The principle of compositionality of meaning

1.1 The principle

The principle of compositionality reads, in its best known formulation:

> The meaning of a compound expression is a function of the meanings of its parts

The principle of compositionality of meaning has immediate appeal, but at the same time it arouses many emotions. Does the principle hold for natural languages? This question cannot be answered directly, because the formulation of the principle is sufficiently vague, that anyone can put his own interpretation on the principle. One topic of investigation in this chapter is providing a more precise interpretation of the principle, and developing a mathematical model for the principle. The second topic of investigation is to discuss challenges to the principle in the literature. It will be argued that the principle should not be considered an empirically verifiable restriction, but a methodological principle that describes how a system for syntax and semantics should be designed.

1.2 Occurrences of the principle

Compositionality of meaning is a standard principle in logic. It is hardly ever discussed there, and almost always adhered to. Propositional logic clearly satisfies the principle: the meaning of a formula is its truth value and the meaning of a compound formula is indeed a function of the truth values of its parts. The case of predicate logic will be discussed in more detail in section 2.

The principle of compositionality is a well-known issue in philosophy of language, in particular it is the fundamental principle of Montague Grammar. The discussions in philosophy of language will be reviewed in several sections of this chapter. In linguistics the principle was put forward by Katz and Fodor (Katz 1966, p.152), (Katz & Fodor 1963, p.503). They use it to design a finite system with infinite output: meanings for all sentences. There is also a psychological motivation in their argument, as, in their view, the principle can explain how a human being can understand sentences never heard before, an argument proposed by Frege much earlier (see section 1.3); see also the discussion in section 7.5.

The principle is also adhered to in computer science. Programming languages are not only used to instruct computers to perform certain tasks, but they are also used among scientists for the communication of algorithms. So they are languages with an (intended) meaning. To prove properties of programs, for example that the execution of the program terminates at some point, a formal semantics is required. A prominent school in this area , *Denotational Semantics* follows the methods of logic, and espouses therefore compositionality as a fundamental principle, see sections 4.2 and 10.1.

Another argument for working compositionally that is often put forward in computer science, is of a practical nature. A compositional approach enables the program designer to think of his system as a composite set of behaviors, which means that he can factorize his design problem into smaller problems which he can then handle one by one.

Above we have met occurrences of the principle of compositionality in rather different fields. They have a common characteristic. The problem to be dealt with is too difficult to tackle at once and in its entirety, therefore it is divided into parts and the solutions are combined. Thus compositionality forms a reformulation of old wisdom, attributed to Philippus of Macedonia: *divide et impera* (divide and conquer).

1.3 On the history of the principle

Many authors who mention compositionality call it *Frege's Principle*. Some assert that it originates with Frege (e.g. Dummett (1973, p.152)), others inform their readers that it cannot be found in explicit form in his writings (Popper 1976, p. 198). Below we will consider the situation in more detail.

In the introduction to *Grundlagen der Mathematik* (Frege 1884, p. xxii), Frege presents a few principles he promises to follow, one being:

> One should ask for the meaning of a word only in the context of a sentence, and not in isolation'

Later this principle acquired the name of 'principle of contextuality'. Contextuality is repeated several times in his writings and ignoring this principle is, according to Frege, a source of many philosophical errors. The same opinion on these matters is held by Wittgenstein in his Tractatus (Wittgenstein 1921).

Compositionality requires that words in isolation have a meaning and that from these meanings the meaning of a compound can be built. The formulation of contextuality given above disallows speaking about the meaning of words in isolation and is therefore incompatible with compositionality. This shows that Frege was (at the time he wrote these words) not an adherent of compositionality (for further arguments, see Janssen (1986*a*)). In Dummett (1973, p. 192-193) it is tried to reconcile contextuality with compositionality.

In Frege's later writings one finds fragments that come close to what we call compositionality of meaning. The most convincing passage, from 'Compound thoughts' (Frege 1923), is quoted here as it provides a clear illustration of Frege's attitude (in those days) with respect to compositionality. In the translation of Geach & Stoothoff:

> '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 thoughts corresponding to the parts of a sentence, so that the structure of the sentence serves as the image of the structure of the thoughts.'

In this passage one could read the idea compositionality of meaning. Yet it is not the principle itself, as it is not presented as a principle but as an argument in a wider discussion. Furthermore, one notices that Frege does not require that the ultimate parts of the thought have an independently given meaning (which is an aspect of compositionality).

The conclusion is that Frege rejected the principle of compositionality in the period in which he wrote Grundlagen der Mathematik, but may have accepted the principle later on in his life. It seems that nowhere in his published works he mentions compositionality as a principle. It is, therefore, inaccurate to speak of 'Frege's principle'. Compositionality is not Frege's, but it it might be called 'Fregean' because it is in the spirit of his later writings.

2 Illustrations of compositionality

2.1 Introduction

In this section the principle of compositionality is illustrated with four examples, in later sections more complex examples will be considered. The examples are taken from natural language, programming language and logic. All cases concern a phenomenon that at a first sight might be considered as non-compositional. But it turns out that there is a perspective under which they are compositional.

2.2 Time dependence in natural language

The phrase *the queen of Holland* can be used to denote some person. Who this is depends on the time one is speaking about. Usually the linguistic context (tense, time adverbials) give sufficient information about whom is meant, as in (1) or (2):

(1) The Queen of Holland is married to Prince Claus.

(2) In 1910 the Queen of Holland was married to Prince Hendrik.

In (1) the present tense indicates that the present queen is meant: Queen Beatrix. In (2) Queen Wilhelmina is meant, because she was queen in the year mentioned.

These examples might suggest that the meaning of *the queen of Holland* varies with the time about which one is speaking. This is, however, not in accordance with compositionality, which requires that the phrase, when considered in isolation, has a meaning from which the meaning of (1) and (2) can be build. The solution that leads to a single meaning for the phrase is to incorporate the source of variation into the notion of meaning. Accordingly, the meaning of *the queen of Holland* is a function from moments of time to persons. For other expressions there may other factors of influence (speaker, possible world, ...). Such factors are called *indices* and a function with indices as domain is called an *intension*. So compositionality leads us to consider intensions as meanings of natural language expressions. For a discussion, see Lewis (1970).

2.3 Identifiers in programming languages

Expressions like $x+1$ are used in almost every programming language. The expression denotes a number; which number this is, depends on the contents of a certain cell in the memory of the computer. For instance, if the value 7 is stored for x in the memory, then $x+1$ denotes the number 8. So one might say that the meaning of $x+1$ varies, which is not in accordance with compositionality. As in the previous example, the source of variation can be incorporated in the notion of meaning, so that the meaning of an expression like $x+1$ is a function from memory states of the computer to numbers. The same notion of meaning is given in the algebraic approach to semantics of programming languages, initiated by Adj (1977).

Interesting in the light of the present approach is a discussion in Pratt (1979). He distinguishes two notions of meaning: a static meaning (an expression gets a meaning once and for all) and a dynamic notion (the meaning of an expression varies). He argues that a static meaning has no practical purpose, because we frequently use expressions that are associated with different elements in the course of time. Therefore he developed a special language for the treatment of semantics of programming languages: dynamic logic. Compositionality requires that an expression has a meaning from which in all contexts the meaning of the compound can be built, hence a static notion of meaning. In this subsection we have seen that a dynamic aspect of meaning can be covered by a static logic by using a more abstract notion of meaning.

2.4 Tarski's interpretation of predicate logic

Compositionality requires that for each construction rule of predicate logic there is a semantic interpretation. It might not be obvious whether this is the case for predicate logic. Pratt (1979) even says that 'there is no function such that the meaning of $\forall x\phi$ can be specified with a constraint of the form $\mathcal{M}(\forall x\phi) = F(\mathcal{M}(\phi))$'. In a compositional approach such a meaning assignment $\mathcal{M}$ and an operator F on meanings has to be provided.

Let us consider Tarski's standard way of interpreting predicate logic in more detail. It roughly proceeds as follows. Let $\mathcal{A}$ be a model and g an $\mathcal{A}$-assignment. The interpretation in $\mathcal{A}$ of a formula ϕ with respect to g, denoted ϕ^g, is defined recursively. One of these clauses is:

$$[\phi \wedge \psi]^g \qquad \text{is } \textit{true} \text{ iff } \phi^g \text{ is } \textit{true} \text{ and } \psi^g \text{is } \textit{true}.$$

This suggests that the meaning of $\phi \wedge \psi$ is a truth value that is obtained from the truth values for ϕ and ψ. But another clause of the standard interpretation is not compatible with this idea:

$$[\exists x\phi]^g \qquad \text{is } \textit{true} \text{ iff if there is a } g' \sim_x g \text{ such that } [\phi(x)]^{g'} \text{is } \textit{true}.$$

(Here $g' \sim_x g$ means that g' is the same assignment as g except for the possible difference that $g'(x) \neq g(x)$). Since it obviously is not always possible to calculate the truth value of $\exists x\phi$ (for a given g) from the truth value of ϕ (for the same g), a compositional approach to predicate logic requires a more sophisticated notion of meaning.

Note that there is no single truth value which corresponds with $\phi(x)$. It depends on the interpretation of x, and in general on the interpretation of the free variables in ϕ, hence on g. In analogy with the previous example, we will incorporate the variable assignment into the notion of meaning. Then the meaning of a formula is a function from variable assignments to truth values, namely the function that yields true for an assignment in case the formula is true for that assignment. With this conception we can build the meaning of $\phi \wedge \psi$ from the meanings of ϕ and ψ: it is the function that yields true for an assignment if and only if both meanings of ϕ and ψ yield true for that assignment.

The situation becomes more transparent if we use an another perspective: the meaning of a formula is the set of assignments for which the formula is true. Let M denote the function that assigns meanings to formulas. Then we have: $\mathcal{M}(\phi \wedge \psi) = \phi \cap \psi$. For the other connectives there are related operations on sets. For existential quantification the operation is: $\mathcal{M}(\exists x\phi) = \{h \mid h \sim_x g \text{ and } g \in \mathcal{M}(\phi)\}$. Let C_x denote the semantic 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'. Thus the requirement of compositionality is satisfied: the syntactic operation of writing $\exists x$ in front of a formula has a semantic interpretation: apply C_x to the meaning of ϕ. This view on the meaning of predicate logic (sets of assignments) is explicit in some textbooks on logic (Monk (1976, p.196), Kreisel & Krivine (1976, p.17)).

Note that same strategy can be followed for other logics. For instance, a compositional meaning assignment to propositional modal logic is obtained by defining the meaning of a proposition to be the set of possible worlds in which the proposition holds.

It is interesting to take another perspective on the conception of meaning besides as sets of variable assignments. An assignment can be seen as a infinite tuple of elements: the first element of the tuple being the value for the first variable, the second element for the second variable etc. So an assignment is a point in a infinite-dimensional space. If ϕ holds for a set of assignments, then the meaning of ϕ is a set of points in this space. The operator C_x applied to a point adds all points which differ from this point only in their x-coordinate. Geometrically speaking, a single point extends into an infinite line. When C_x is applied to a set consisting of a circle area, it is extended to a cylinder. Because of this effect, the operation C_x

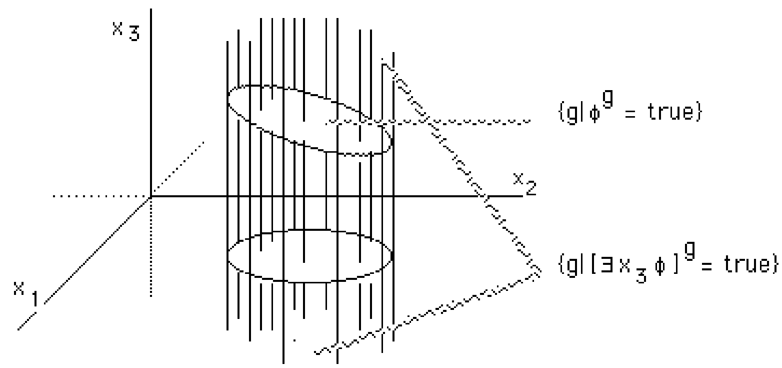

Figure 1: The interpretation of $\exists x$ as a cylindrification operation

is called the x-th cylindrification operation (see figure 1). The algebraic structure obtained for predicate logic with cylindrifications as operators, is called a cylindric algebra. The original motivation for studying cylindric algebras was a technical one: to make the powerful tools from algebra available for studying logic (Henkin, Monk & Tarski 1971).

The discussion can be summarized as follows. The standard (Tarskian) interpretation of predicate logic is not a meaning assignment but a recursive, parameterized definition of truth for predicate logic. It can easily be turned into a compositional meaning assignment by incorporating the parameter (viz. the assignment to variables) into the concept of meaning. Then meaning becomes a function with assignments as domain.

2.5 Situation semantics

Situation Semantics (Barwise & Perry 1983) presents an approach to meaning which differs from the traditional model-theoretic one. The basic new point is that a sentence conveys information (about the external world or about states of mind), formalized in their conception of meaning as a relation. The meaning of a declarative sentence is a relation between utterances of the sentence and the situation described by the utterance. More generally, the meaning of an expression is a relation between utterances and situations. The interpretation of an utterance at a specific occasion is the described situation.

To illustrate Situation Semantics, consider the following example (op. cit. p. 19):

(3) I am sitting.

The meaning of this sentence is a relation between utterance u and situation e which holds just in case there is a location l and an individual a such that a speaks at l, and in situation e this individual a is sitting at l. The parts of a sentence provide the following ingredients to build this meaning relation. The meaning of a referring noun phrase is a relation between an utterance and an individual; and the verb phrase is a relation between an utterance and a property. From the meanings of the subject and the verb phrase the meaning of the whole sentence is built in a systematic way. Thus, Situation Semantics satisfies the principle of compositionality of meaning.

This was a simple example because the domain of interpretation does not change. More challenging is sentence (4) with antecedent relations as indicated in (5) (Barwise & Perry 1983, p. 136-137):

(4) Joe admires Sarah and she admires him.

(5) Joe_1 admires Sarah_2 and she_2 admires him_1.

Sentence (4) has two parts (6) and (7):

(6) Joe admires Sarah

(7) She admires him.

Sentence (7), when considered in isolation, has two free pronouns for which suitable connections must be found. This is not the case for the whole sentence (4); so (7) has another domain for the interpretation of pronouns than (4). For this reason, the statement made with (4) cannot be considered as just a conjunction of two independent statements: somehow the meaning of the first part has to influence the meaning of the second part.

The solution is based on the meaning of names. Initially (op. cit. p. 131), the meaning of a name β was defined as a relation that holds between an utterance u and an individual a_σ (in a discourse situation d) if and only if the speaker c of the utterance refers by β to that individual. For sentences like (4), the meaning of names is augmented to make them suitable antecedents for co-indexed pronouns (op. cit. p. 137), evoking a connection with the coindexed pronouns. In symbols:

$$d, c[[\beta_i]]a_\sigma, e \text{ iff } c(\beta_i) = a_\sigma,\ a_\sigma \text{ is named } \beta,\ \text{and if } c(he_i) = b \text{ then } b = a_\sigma$$

With this extension the meaning of a sentence of the form ϕ *and* ψ can be obtained from the meanings of ϕ and ψ in the following way:

$$d, c[[\phi \textit{ and } \psi]]e \text{ iff there is an extension } c' \text{ of } c \text{ such that } d, c'[[\phi]]e \text{ and } d, c'[[\psi]]e$$

Let us summarize the solution. The meaning of ϕ *and* ψ is a relation, and to find its value for the pair of coordinates d, c the value of the meanings of ϕ and ψ for these coordinates is not sufficient. Other coordinates c' have to be considered too, so the whole meaning relation has to be known. This illustrates that (op. cit. p. 32): 'a version of compositionality holds of meanings, but not of interpretations' . This is in analogy of the situation in Montague grammar, where there is compositionality of meaning, but not of extension.

This example illustrates that the relational approach to meaning is not an obstacle to compositional semantics. The problem was that the initial meaning of names was too poor to deal with coindexed pronouns, and the solution was to augment the concept of meaning. Again, the strategy was followed that if a given conception of meaning is not suitable for a compositional semantics, a richer conception of meaning is defined.

2.6 Conclusion

These examples illustrate that compositionality is not too narrow. Using a sufficiently abstract notion of meaning, it is flexible enough to cover many standard proposals in the field of semantics. The strategy was to incorporate a possible source of variation of meaning into a more abstract notion of meaning. In this way meanings not only capture the semantic intuitions, but do so in a compositional way. The classical advice of Lewis (1970, p.5) is followed: 'In order to say what a meaning is, first ask what a meaning does, and then find something that does that'.

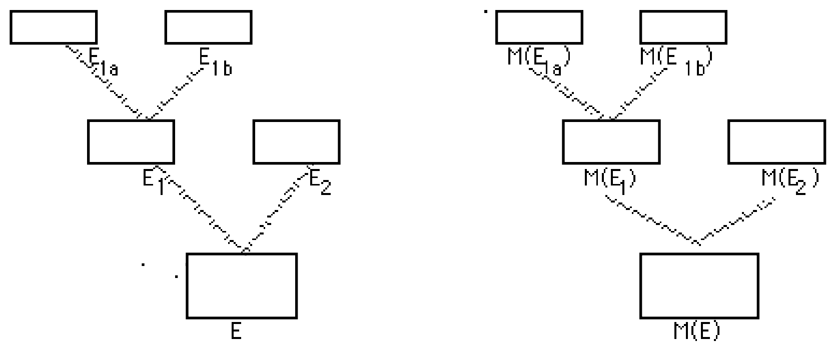

Figure 2: Compositionality: the compositional formation of expression E from its parts and the compositional formation of the meaning of E from the meanings its parts

.

3 Towards formalization

3.1 Introduction

The principle of compositionality of meaning is not a formal statement. It contains several vague words which have to be made precise in order to give formal content to the principle. In this section the first steps in this direction are made, giving us ways to distinguish compositional and non-compositional proposals (in section 4). In later sections (viz. 8, 9) mathematical formalizations are given, making it possible to prove certain consequences of the compositional approach.

Suppose that an expression E is constituted by the parts E_1 and E_2 (according to some syntactic rule). Then compositionality says that the meaning $M(E)$ of E can be found by finding the meanings $M(E_1)$ and $M(E_2)$ of respectively E_1 and E_2, and combining them (according to some semantic rule). Suppose moreover that E_1 is constituted by E_{1a} and E_{1b} (according to some syntactic rule, maybe another that the one used for E). Then the meaning $M(E_1)$ is in turn obtained from the meanings $M(E_{1a})$ and $M(E_{1b})$ (maybe according to another rule than the one combining $M(E_1)$ and $M(E_2)$). This situation is presented in figure 2.

3.2 Assumptions

The interpretation in 3.1 is a rather straightforward explication of the principle, but there are several assumptions implicit in it. Most assumptions on compositionality are widely accepted, some will return in later sections, when the principle is discussed further.

The assumptions are:

1. In a grammar the syntax and the semantics are distinguished components connected by the requirement of compositionality. This assumption excludes approaches, as in some variants of Transformational Grammar, with a series of intermediate levels between the syntax and the semantics.

2. It is assumed that the output of the syntax is the input for meaning assignment. This is for instance in contrast to the situation in Generative Semantics, where the syntactic form is projected from the meanings.

3. The rules specify how to combine the parts, i.e. they are instructions for combining expressions. So this gives a different perspective from the traditional view of a grammar as a rewriting system.
4. The grammar determines what the parts of an expression are. It depends on the rules whether *Mary does not cry* has two parts *Mary* and *does not cry*, or three *Mary, does not* and *cry*. This illustrates that **part** is a technical notion.
5. All expressions that arise as parts have meaning. This excludes systems in which only complete sentences can be assigned meaning (as in some variants of Transformational Grammar). Not only parts for which we have an intuitive meaning (as *loves* in *John loves Mary*), but also parts for which this is less intuitive (as *only* in *Only John loves Mary*). The choice what the meaning of a part is might depend on what we consider a suitable ingredient for building the meaning of the whole expression.
6. The meaning of an expression is not only determined by the parts, but also by the rule which combines those parts. From the same collection of parts several sentences can be made with different meanings (e.g. *John loves Mary* vs. *Mary loves John*). Several authors make this explicit in their formulation of the principle, e.g. Partee, ter Meulen & Wall (1990, p.318):

    > The meaning of a compound expression is a function of the meanings of its parts and of the syntactic rule by which they are combined.

7. For each syntactic rule there is a semantic rule that describes its effect. In order to obtain this correspondence, the syntactic rules should be designed appropriately. For instance, semantic considerations may influence the design of syntactic rules. This correspondence leaves open the possibility that the semantic rule is a meaning-preserving rule (no change of meanings), or that different syntactic rules have the same meaning.
8. The meaning of an expression is determined by the way in which it is formed from its parts. The syntactic production process is, therefore, the only input to the process of determining its meaning. There is no other input, so no external factors can have an effect on the meaning of a sentence. If, for instance, discourse factors should contribute to meaning, the conception of meaning has to be enriched in order to capture this.
9. The production process is the input for the meaning assignment. Ambiguous expressions must have different derivations: i.e. a derivation with different rules, and/or with different basic expressions.

3.3 Options in syntax

In the above section it is not specified what the nature is of expressions and parts, i.e. what kind of objects are in the boxes in figure 2. Such a decision has to be based upon linguistic insights. Below some important options are mentioned.

Concatenation of words

Close to the most naive conception of compositionality is that the boxes contain strings of words (the terminal boxes single words), and that the syntactic rules concatenate their contents. However, all important theories of natural language have a more sophisticated view. Classical Categorical Grammar and classical Generalized Phrase Structure grammar (GPSG) do not to use real words, but more abstract word-forms with features. In all these cases the structure from figure 2 is isomorphic to the constituent structure of the involved expression.

Powerful operations on strings

In some theories the syntactic rules are more powerful than just concatenation. A small step is to allow a wrap-rule (a rule with two arguments, where the first argument is inserted in the second position of the second argument). In PTQ (Montague 1973) the syntactic rules are very powerful, for instance there is a rule that substitutes a string for a pronoun (e.g. the wide scope reading of *Every man loves a woman* is obtained by substituting *a woman* for *him* in *every man loves him*). In these cases the grammar generates strings, and the derivation does not assign a constituent structure to them (since the parts are not constituent parts).

Operations on structures

Most theories concern structures. Tree Adjoining Grammar, for instance, assumes as its basic elements (small) trees, and two kinds of rules: adjunction and substitution. Another example are the M-grammars, introduced by Partee (1973), and used in the translation system Rosetta (Rosetta 1994). The boxes contain phrase-structure trees as in Transformational Grammar, and the rules are powerful operations on such trees. In this situation the tree that describes the derivation might differ considerably from the tree describing the structure of the string, as illustrated below.

Consider the following sentence:

(1) *John seeks a unicorn.*

There are semantic arguments for distinguishing two readings: the *de re* reading which implicates the existence of unicorns, and the *de dicto* reading which does not. But there are no syntactic arguments for distinguishing two different constituent structures. In an M-grammar this unique constituent structure can be derived in two ways, one for each meaning. In figure 3 the derivation of the de re reading of (1) is given, using a tree-substitution rule.

3.4 Conclusion

Above it is argued that there are several options in syntax. In the previous section it has been shown that there are choices in defining what meanings are. The discussion whether natural language is compositional has to do with these options. If one has a definite opinion on what parts, meanings and rules should be like, then it may be doubted whether compositionality holds. But if one leaves one or more of these choices open, then the issue becomes: in which way can compositionality be obtained? These two positions will return in several discussions concerning the principle of compositionality.

4 Examples of non-compositional semantics

4.1 Introduction

In this section examples of essential non-compositional semantics are presented, where their non-compositional character is not caused by the nature of the phenomena, but by the fundamental aspects of the approach taken. It is not possible to turn these proposals into compositional ones without losing a fundamental aspect of the analysis. Thus the examples illustrate the demarcation line between compositional and non-compositional semantics. As in section (2), the examples deal with several types of languages: programming languages, natural languages and logic.

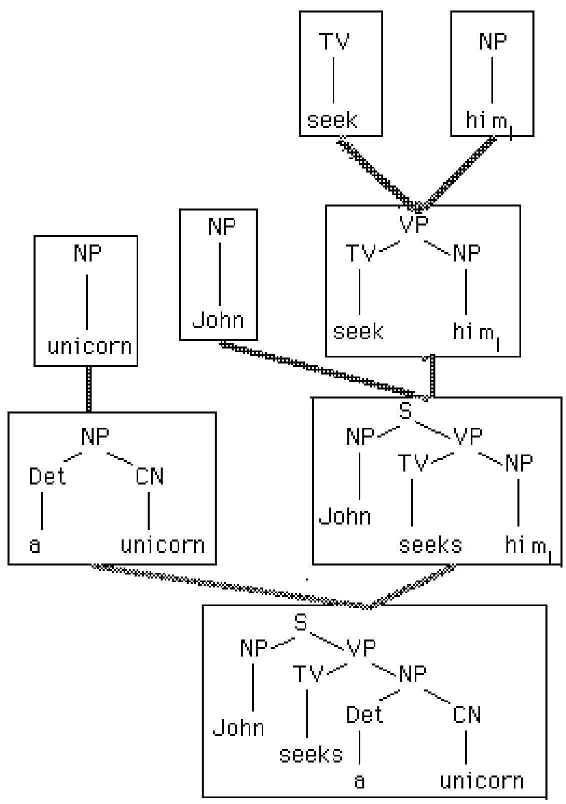

Figure 3: The production of the *de dicto* reading of *John seeks a unicorn* The resulting constituent structure is the same as the structure for the *de re* reading

4.2 Arrays in programming languages

In programming languages one finds expressions consisting of an array identifier with subscript, e.g. $a[7]$. Here a is an array identifier, it refers to a series of memory cells in the computer. Between the [-sign and] -sign the subscript is mentioned. That subscript tells which of the cells is to be considered, so the expression $a[7]$ refers to the contents of this cell (e.g. a number). The subscript can be a compound expression that denotes a number, e.g. $x + 1$, hence the syntax of this construction says that there are two parts: an array identifier, and an arithmetical expression.

In the semantics of programming languages one often interprets programs in an abstract computer with abstract memory cells. Then expressions like $a[7]$ and $a[x + 1]$ have as interpretation the value stored in such a memory cell (or alternatively a function to such a value). The array identifier itself cannot be given an interpretation, since in the abstract computer model there is nothing but cells and their contents, and a does not correspond to anyone of them. As a consequence every time the array identifier arises, it has to be accompanied by a subscript. This leads to complicated proof rules (e.g. in de Bakker (1980))

This interpretation is not in accordance with compositionality which requires that all parts have a meaning; in particular the array identifier should have a meaning. Although in the given computer model an appropriate meaning is not available, it is easy to define one: a function from numbers to cells. Changing the model in this way, allows a simpler reformulation of the proof rules, because array identifiers without subscripts can be used (see e.g. Janssen & van Emde Boas (1977)).

4.3 Syntactic rules as conditions

In several theories syntactic rules are formulated as conditions, or they are accompanied by conditions. First we will consider a simple example. A context sensitive rule allows us to rewrite a symbol in a certain context. A context sensitive grammar is a grammar with such rules. An example is one with the rules $S \to AA$, $Ab \to bb$, $bA \to bb$. This grammar does not produce any strings, because after application of the first rule, no further rules are applicable. MacCawley (1986) proposed to consider context sensitive rules as 'node-admissability conditions'. These specify which configurations in trees are allowed. For instance, the last rule says that an b immediately dominated by an A is allowed, if there is an b immediately to the left of this b. With this interpretation, the tree in figure 4 is allowed by the given grammar. So the string bb belongs to the language of the grammar, although it cannot be generated in the classical way. In this conception of grammar there are no rules, only conditions. Hence there is no syntactic algebra with operations, and an admissible structure has no derivation. Consequently, a compositional meaning assignment (in the sense of the principle) is not possible.

A similar situation arises in the variant of Transformational Grammar known as 'Principles and Parameters'. Conditions form also the central part of the theory, but formally the situation is slightly different. One single transformation, called $move - \alpha$, can in principle move any constituent to any position controlled by various conditions on movement. So the interesting aspect of the theory does not lie in this transformation, but in the conditions. An algebraic formulation of this theory is possible, with one partial rule which takes one argument as input. Since this single rule has to account for all phenomena, there is no semantic counterpart for this rule. So 'Principles and Parameters' is a theory where compositionality of meaning is impossible.

In Generalized Phrase Structure Grammar (GPSG) syntactic rules are considered as expressing a tree admissibility condition, i.e. they say which trees are allowed given an ID-rule or an LP-rule. This form of admissibility conditions does

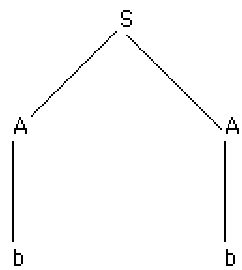

Figure 4: The context sensitive rules $S \to AA$, $Ab \to bb$, $bA \to bb$ used as node admissibility conditions

not disturb compositionality: a rule can be considered as an abbreviation for a collection of rules, each generating one of the admissible structures, and all rules from the collection have the same semantic interpretation (the one associated with the original rule).

4.4 Discourse representation theory

Pronominal references in discourses may depend on previous sentences, as illustrated by the following two discourses which have identical second sentences.

(1) A man walks in the park. He whistles.

(2) Not all men do not walk in the park. He whistles.

In (1), the pronoun *he* in the second sentence is interpreted as anaphorically linked to the term *a man* in the first sentence. This is not possible in (2), where *he* has to refer to a third party. The meanings of discourses (1) and (2) are, therefore, different.

Since their second sentences are identical, their first sentence (3) and (4) must contain the source of the meaning difference.

(3) A man walks in the park.

(4) Not all men do not walk in the park.

However, (3) and (4) have identical truth-conditions, hence the discourses (1) and (2) seem to provide an argument against compositionality.

Discourse representation theory (henceforth 'DRT') is a theory about semantic representations of texts, especially concerning pronominal references in texts (Kamp 1981, Kamp & Reyle 1993). There are explicit rules how these representations are formed, and these rules follow the syntactic rules step by step. Parts of sentences provide building blocks for discourse representations. However, no semantic interpretation is provided for these parts of discourse representations. Furthermore, the instructions may require specific information concerning already built parts of representations, and may change them. So the representation plays an essential role in the system and cannot be eliminated. DRT is a system for compositionally constructing representations, but not for compositional semantics (then the representations should not be essential, see also the discussion in section (5)). This is intended: the name of the theory states explicitly that it is about representations, and claims psychological relevance of the representations. The solution DRT provides for the discourses we started with roughly is as follows. Different representations are assigned to (3) and (4), and the two negations in (4) cause a difference that triggers a difference in interpretation strategy, hence a difference in the pronominal reference.

However, a compositional treatment for this kind of discourse phenomena is quite feasible. In fact, the principle of compositionality itself points to a solution. Since (3) and (4) have identical truth-conditions, a richer notion of meaning is required if the principle of compositionality is to be saved for discourses. Truth-conditions of sentences (which involve possible worlds and assignments to free variables) are just one aspect of meaning. Another aspect is that the preceding discourse has a bearing on the interpretation of a sentence (and especially of the so called discourse pronouns). Moreover, the sentence itself extends this discourse and thus has a bearing on sentences that follow it. Hence a notion of meaning is required which takes the semantic contribution into account that a sentence makes to a discourse. Sentences (3) and (4) make different contributions to the meaning of the discourse, especially concerning the interpretation of later discourse pronouns. These ideas have led to Dynamic Predicate Logic (henceforth 'DPL'). It is a compositional theory that accounts not only for the phenomena that are treated in DRT, but for other phenomena as well, see Groenendijk & Stokhof (1991). Thus we see that the program to require compositionality has suggested a particular solution.

The difference in compositionality between in DRT and DPL was initially a central point in the discussion, see Groenendijk & Stokhof (1991). Later developments made the difference less crucial, because several reformulations of DRT were given that adhered to compositionality. Examples are Zeevat (1989) and Muskens (1993), and the chapter in this handbook on DRT. The concepts of meaning used in these proposals are illuminating. For instance, in Zeevat's proposal the meanings are pairs consisting of sets of assignments (as in predicate logic), and a set of variables (discourse markers). So syntactic symbols act as component in the semantics, which reflects the special role of representations in DRT.

4.5 Substitutional interpretation of quantifiers

For the interpretation of $\exists x \phi$ an alternative to the Tarskian interpretation has been proposed that is not compositional. It is called the *substitutional interpretation*, and says: $\exists x \phi(x)$ is true if and only if there is some substitution a for x such that $\phi(a)$ is true. Of course, the substitutional interpretation is only equivalent to the standard interpretation if there is a name for every element in the domain. The substitutional interpretation can be found in two rather divergent branches of logic: philosophical logic and in proof theory, both considered below.

In philosophical logic the substitutional interpretation is advocated by Marcus (1962) with an ontological motivation. Consider

(5) Pegasus is a winged horse.

Marcus argues that one might believe (5) without believing

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

At the same time she accepts that (5) entails (7):

(7) $\exists x(x$ is a winged horse$)$

This view implies that the quantification in (7) cannot be considered quantification in the ontological sense. The substitutional interpretation of quantifiers allows us to accept (7) as a consequence of (5), without accepting (6) as a consequence.

The substitutional interpretation is discussed more formally by Kripke (1976). As a syntax for the logic he presents the traditional syntax: $\exists x\phi(x)$ is produced from $\phi(x)$ by placing the quantifier in front of it. According to that grammar $\phi(x)$ is a part of $\exists x\phi(x)$, and $\phi(a)$ is not a part of $\exists x\phi(x)$. Hence in this case the substitutional interpretation is not compositional: the meaning of $\exists x\phi(x)$ is not obtained from the meaning of its part $\phi(x)$.

In proof theory the substitutional interpretation is given by e.g. Schütte (1977). According to his syntax $\forall x\phi(x)$ is formed from $\phi(a)$, where a is arbitrary. So the formula $\forall x\phi(x)$ is syntactically ambiguous: there are as many derivations as there are expression of the form $\phi(a)$. It is in general not possible, given one such a, to find the interpretation of $\forall x\phi(x)$ from the interpretation of that $\phi(a)$, because $\forall x\phi(x)$ can be false, whereas $\phi(a)$ is true for some a's. Hence also in this case the substitutional interpretation does not satisfy the compositionality principle.

If one wishes to have the substitutional interpretation, and at the same time meet the principle of compositionality, then the syntax has to contain an infinitistic rule which says that all expressions of the form $\phi(a)$ are a part of $\forall x\phi(x)$. But such an infinitistic rule has not been proposed.

4.6 Conclusion

The examples illustrate that compositionality is a real restriction in the sense that there are theories that are essentially non-compositional. Moreover, it illustrates that compositionality is crucial in evaluating theories: not in the sense that it discriminates good from bad (such arguments are not given above, but will be given in later sections), but in the sense that it exhibits a special aspect of those theories. The fact that there was no compositional treatment of arrays exhibits that the semantic model used was ontologically sparse (or too poor, if you prefer). It exhibits that the substitutional interpretation of quantifiers avoids assignments to variables with the price of introducing an infinitistic aspect in the syntax. In DRT the rules refer in several ways to the particular form of the partial discourse representations that occur as their inputs. The compositional reformulations exhibit in which respect this is essential. This brings us to the following advice: if you encounter a new proposal, and wish to find the innovative or deviant aspect, then look for the point where it departs from compositionality.

5 Logic as auxiliary language

5.1 Introduction

The principle of compositionality of meaning expresses that meanings of parts are combined into the meaning of a compound expression. Since meanings are generally formalized as model-theoretic entities, such as truth values, sets of sets etc., functions have to be specified which operate on such meanings. An example of such an operation is (Montague (1970*a*); p.194 in Thomason (1974)):

(1) G_3 is that function $f \in ((2^I)^{A\times A})^{A^\omega}$ such that, for all $x \in A^\omega$, all $u, t \in A$ and all $i \in I : f(x)(t,u)(i) = 1$ if and only if $t = u$

Such descriptions are not easy to understand, nor convenient to work with. Therefore almost always a logical language is used to represent meanings and operations on meanings. The main exception is Montague (1970*a*). So in practice associating meanings with natural language amounts to translating sentences into logical formulas. The operation described above is represented in intensional logic with the formula ${}^\wedge\lambda t\lambda u[t = u]$. This is much easier to grasp than the formulation in (1).

This example illustrates that such translations into logic are used for good reasons. In the present section the role of translations into a logical language is investigated.

5.2 Restriction on the use of logic

Working in accordance with compositionality of meaning puts a heavy restriction on the translations into logic, because the goal of the translations is to assign meanings. The logical representations are just a tool to reach this goal. The representations are not meanings themselves, and should not be confused with them. This means for instance, that two logically equivalent representations are equally good as representation of the associated meaning. A semantic theory cannot be based upon accidental properties of meaning representations, since it would then be a theory about representations, and not about the meanings themselves. Therefore the logical language should only be auxiliary tool and, in principle, be dispensable.

If one has a logic for representing meanings, this logic will probably not have all the operations on meanings one needs. For instance, logic usually has only one conjunction operator (between formulas of type t), whereas natural language requires several (not only between sentences, but also between verbs, nouns, etc.). So the logic has be extended with new operations. We will consider two methods.

A new semantic operation can be introduced by introducing a new basic operator symbol, together with a model theoretic interpretation for it. Such an interpretation can be given directly, speaking e.g. about functions from functions to functions. Another method is to denote the intended interpretation with a logical expression. Then one should not forget that this expression stands for its interpretation, see the example below (in section (5.3)).

Another method is to describe the effects of the new operation using already available ones. An example we have met above (in section 5.1) is $^\wedge\lambda t \lambda u[t = u]$. This is an example of the standard method (introduced by Montague (1970*b*)): using polynomials. Probably anyone has encountered polynomials in studying elementary mathematics; an example (with two variables) is $x_1^2 + x_1 + 3 \times x_2$. This polynomial defines a function on two arguments; the resulting value is obtained by substituting the arguments for the variables and evaluating the result. For the arguments 2 and 1 it yields $2^2 + 2 + 3 \times 1$, being 9. The method of polynomials can be used in logic as well. For instance, a polynomial over intensional logic with variables X_1 and X_2 is:

(2) $\lambda y[X_1(y) \wedge X_2(y)]$

Note that y is not a variable in the sense of the polynomial. Polynomial (2) is an operation which takes two predicates as inputs and yields a predicate as result. It can be used to describe, for instance, the semantic effect of verb phrase conjunction. Usually greek letters are used to indicate variables, in PTQ (Montague 1973) one finds for the above polynomial:

(3) $\lambda y[\gamma'(y) \wedge \delta'(y)]$

In section (5.4) more examples of polynomials and non-polynomials will be given.

5.3 A new operator: CAUSE

Dowty (1976) presents a treatment of the semantics of factive constructions like *shake John awake*. For this purpose intensional logic is extended with an operator CAUSE. In order to define its interpretation the semantic apparatus is extended with a function that assigns to each well-formed formula ϕ and each possible world i a possible world $f(\phi, i)$. Intuitively speaking, $f(\phi, i)$ is the possible world that is most like i with the possible exception that ϕ is the case. Then the interpretation of CAUSE reads:

(4) If $\phi, \psi \in ME_t$ then $(\phi\, CAUSE\, \psi)^{\mathcal{A},i,j,g}$ is 1 if and only if $[\phi \wedge \psi]^{\mathcal{A},i,j,g}$ is 1 and $[\neg\phi]^{\mathcal{A},f(\neg\psi,i),j,g}$ is 1.

The first argument of f is a formula, and not the interpretation of this formula. Hence CAUSE, which is based upon this function, is an operator on formulas, and not on the meanings they represent. This suggests that the logic is not dispensable, that it is an essential stage and that the proposed solution is not compositional. This is shown as follows. Let f be such that $f(\neg[\phi \wedge \eta], i) \neq f(\neg[\eta \wedge \phi], i)$. Then it may be the case that $[(\phi \wedge \eta)\, CAUSE\, \psi]^{\mathcal{A},i,j,g}$ holds whereas this does not hold for $\eta \wedge \phi$. So the two equivalent formulas $\phi \wedge \eta$ and $\eta \wedge \phi$ cannot be substituted for each other without changing the resulting truth value; a consequence that was not intended. This illustrates that the introduction of a new operator in a way that violates compositionality bears the risk of being incorrect in the sense that the intended semantic operation is not defined.

The proposal can be corrected by defining f for the *meaning* of its first argument (i.e. its intension). Then the last clause of the definition becomes $[\neg\phi]^{\mathcal{A},k,j,g}$ is 1, where $k = f([{}^\wedge[\neg\eta \wedge \phi]]^{\mathcal{A},i,j,g}, i)$.

5.4 An operation on logic: relative clause formation

The syntactic rule for restrictive relative clause formation in PTQ (Montague 1973) roughly is as follows:

(5) $R_{3,n}$: If α is a CN and β a sentence, then α *such that* $\beta^\star$ is a CN, where $\beta^\star$ comes from β by replacing each occurrence of he_n by the appropriate pronoun.

The corresponding semantic rule reads (neglecting intensions and extensions):

(6) If α' is the translation of the common noun α, and β' of the sentence β, then the translation of the CN with relative clause is $\lambda x_n[\alpha'(x_n) \wedge \beta']$.

The rule above forms from *man* and he_2 *loves Mary* the common noun phrase *man such that he loves Mary*. Suppose that the meanings of these parts are represented by *man* and $love_*(x_2, j)$. Then the meaning of the common noun phrase is given correctly by $\lambda x_2[man(x_2) \wedge love_*(x_2, j)]$. However, the translation rule yields incorrect results in case the translation of the common noun contains the occurrence of a variable that becomes bound by the λ-operator introduced in the translation rule. In order to avoid this, the editor of the collection of Montague's work on philosophy of language, R.H. Thomason, gave in a footnote a correction (Thomason 1974, p.261):

(7) To avoid collision of variables, the translation must be $\lambda x_m[man(x_m) \wedge \psi)]$, where ψ is the result of replacing all occurrences of x_n in β' by occurrences of x_m, where m is the least even number such that x_m has no occurrences in either α' or β'.

This rule introduces an operation on expressions: the replacement of a variable by one with a special index. However, finding the least even index that is not yet used is an operation that essentially depends on the form of the formulas. This is illustrated by the two formulas $x_1 = x_1$ and $x_2 = x_2$, which are logically equivalent (they are tautologies), but have a different least index that is not yet used. So Thomason's reformulation is an operation on representations, and not on meanings.

Nevertheless, (7) is correct in the sense that it does correspond with an operation on meanings. The operation on meanings can be represented in a much simpler way, using a polynomial, viz.:

(8) $\lambda P[\lambda x_n[P(x_n) \wedge \beta']](\alpha')$

This polynomial formulation avoids the binding of variables in α' by λx_n, so the complication of Montague's rule does not arise. Furthermore, it is much simpler than Thomason's correction of the rule.

5.5 Conclusion

These examples illustrate a method to find dangerous spots in a proposal: find the places where the translation into logic is not a polynomial. It is likely that compositionality is violated there. Either the proposal is incorrect in the sense that it makes unintended predictions, or it is correct, but can be improved (simplified) considerably by using a polynomial. The latter point, viz. that an operation on meanings can be expressed by means of a polynomial, (as illustrated in 5.3) can be given a mathematical basis (see section 8). These applications of compositionality exhibit the benefits of compositionality as a heuristic method.

6 Counterexamples to compositionality

6.1 Introduction

In the present section we consider some examples from natural language that are used in the literature as arguments against compositionality. Several other examples could be given, see Partee (1984). The selection here suits to illustrate the methods available to obtain compositionality. The presentation of the examples follows closely the original argumentation; proposals for a compositional treatment are given afterwards. In the last section the methods to obtain compositional solutions are considered from a general perspective.

6.2 Counterexamples

6.2.1 Would

The need for the introduction of the NOW-operator was based upon the classical example (Kamp 1971):

(1) A child was born that will become ruler of the world.

The following more complex variants are discussed by Saarinen (1979), who argues for other new tense operators.

(2) A child was born who would become ruler of the world.

(3) Joseph said that a child had been born who would become ruler of the world.

(4) Balthazar mentioned that Joseph said that a child was born who would become ruler of the world.

Sentence (2) is not ambiguous, the moment that the child becomes ruler of the world lies in the future of its birth. Sentence (3) is twofold ambiguous: the moment of becoming ruler can be in the future of the birth, but also in Joseph's future. And in (4) the child's becoming rules can even be in Balthazars future. So the number of ambiguities increases with the length of the sentence. Therefore Hintikka (1983, pp. 276-279) presents (2)-(4) as arguments against compositionality.

6.2.2 Unless

Higginbotham (1986) presents arguments against compositionality; we discuss variants of his examples (from Pelletier (1993*a*)). In (5) and (6) *unless* has the meaning of a (non-exclusive) disjunction.

(5) John will eat steak unless he eats lobster.

(6) Every person will eat steak unless he eats lobster.

However, in (7) the situation is different.

(7) No person will eat steak unless he eats lobster.

This sentence is to be represented as

(8) [No: person] (x eat steak $\wedge$ $\neg$ x eats lobster).

These examples show that the meaning of *unless* depends on the context of the sentence in which it occurs. Therefore compositionality does not hold.

6.2.3 Any

Hintikka (1983, pp. 266-267) presents several interesting sentences with *any* as challenges to compositionality. Consider

(9) Chris can win any match.

In this sentence it is expressed that for all matches it holds that Chris can win them, so *any* has the impact of a universal quantification. But in (10) it has the impact of an existential quantification.

(10) Jean doesn't believe that Chris can win any match.

Analogously for the pair (11) and (12), and for the pair (13) and (14):

(11) Anyone can beat Chris.

(12) I'd be greatly surprised if anyone can beat Chris.

(13) Chris will beat any opponent

(14) Chris will not beat any opponent.

All these examples show that the meaning of the English determiner *any* depends on its environment.

The most exciting example is the one given below. As preparation, recall that Tarski required a theory of truth to result in T-schemes for all sentences:

(15) 'ϕ' is true if and only if ϕ is the case.

A classical example of this scheme is:

(16) *Snow is white* is true if and only if snow is white

The next sentence is a counterexample against one half of the Tarskian T-scheme.

(17) *Anybody can become a millionaire* is true if anybody can become a millionaire.

This sentence happens to be false.

6.3 Compositional solutions

6.3.1 Would

A compositional analysis of (18) is indeed problematic if we assume that it has to be based on (19), because (18) is ambiguous and (19) is not.

(18) Joseph said that a child had been born who would become ruler of the world.

(19) A child was born who would become ruler of the world.

However, another approach is possible: there may be two derivations for (18). In the reading that 'becoming ruler' lies in the future of Joseph's saying it may have (20) as part.

(20) say that a child was born that will become ruler of the world

The rule assigning past tense to the main clause should then deal with the 'sequence of tense' in the embedded clause, transforming *will* into *would*. The reading in which the time of becoming ruler lies in the future of the birth could then be obtained by building (18) from:

(21) say that a child was born who would become ruler of the world.

The strategy to obtain compositionality will now be clear: account for the ambiguities by using different derivations. In this way the parts of (18) are not necessarily identical to substrings of the sentences under consideration (the involved tenses may be different). Such an approach is followed for other scope phenomena with tenses in Janssen (1983).

6.3.2 Unless

Pelletier (1993*a*) discusses the arguments of Higginbotham (1986) concerning *unless*, and presents two proposals for a compositional solution.

The first solution is to consider the meaning of *unless* to be one out of a set of two meanings. If it is combined with a positive subject (as in *every person will eat steak unless he eats lobster*) then the meaning 'disjunction' is selected, and when combined with negative subject (as in *no person eats steak unless he eats lobster*) the other meaning is selected. For details of the solution, see Pelletier (1993*a*). So *unless* is considered as a single word, with a single meaning, offering a choice between two alternatives. In the same way as in section 2 this can be defined by a function from contexts to values.

The second solution is to consider *unless* a homonym. So there are two words written as *unless*. The first one is $unless_{[-neg]}$, occurring only with subjects which bear (as is the case for *every person*) the syntactic feature [-neg], and having 'disjunction' as meaning. The second one is $unless_{[+neg]}$, which has the other meaning. Now *unless* is considered to be two words, each with its own meaning. The syntax determines which combinations are possible.

6.3.3 Any

Hintikka (1983, p.280) is explicit about the fact that his arguments concerning the non-compositionality of *any*-sentences are based upon specific ideas about their syntactic structure. In particular it is assumed that (22) is a 'component part' of (23)

(22) Anyone can beat Chris

(23) I'll be greatly surprised if anyone can beat Chris.

He claims that this analysis is in accordance with common sense, and in agreement with the best syntactic analysis. But, as he admits, other analyses cannot be excluded a priori; for instance that (24) is a component of (23).

(24) I'll be greatly surprised if — can beat Chris.

One might even be more radical in the syntax than Hintikka suggests, and introduce a rule that produces (23) from

(25) Someone can beat Chris.

Partee (1984) discusses the challenges of *any*. She shows that the situation is more complicated than suggested by the examples of Hintikka. Sentence (27) has two readings, only one of which can come from (26).

(26) Anyone can solve that problem.

(27) If anyone can solve that problem, I suppose John can.

Partee discusses the literature concerning the context-sensitivity of *any*, and concludes that here are strong arguments for two 'distinct' *any*'s: an *affective any* and a *free-choice any*. The two impose distinct (though overlapping) constraints on the contexts in which their semantic contributions 'make sense'. The constraints on affective *any* can be described in model-theoretic terms, whereas those of the free-choice *any* are less well understood. For references concerning this discussion see Partee (1984).

We conclude that the *any*-examples can be dealt with in a compositional way by distinguishing ambiguous *any*, with one or both readings eliminated when incompatible with the surrounding context.

6.4 General methods for compositionality

In this section we have encountered three methods to obtain compositionality:

1. New meanings.
   These are formed by the introduction of a new parameter, or alternatively, a function from such a parameter to old meanings. This was the first solution for *unless*

2. New basic parts
   Duplicate basic expressions, together with different meanings for the new expressions, or even new categories. This was the solution for *any*, and the second solution for *unless*.

3. New constructions
   Use unorthodox parts, together with new syntactic rules forming those parts and rules operating on those parts. This approach may result in abstract parts, new categories, and new methods to form compound expressions. This was the solution for the *would* sentences.

For most of counterexamples several of these methods are in principle possible, and a choice must be motivated. That is not an easy task because the methods are not just technical tools to obtain compositionality: they raise fundamental questions concerning the syntax and semantics interface. If meanings include a new parameter, then meanings have this parameter in the entire grammar, and it must be decided what role the parameter plays. If new basic parts are introduced, then each part should have meaning, and each part is available everywhere. If new constructions are introduced, they can be used everywhere. Other expressions may then be produced in new ways, and new ambiguities may arise. So adopting compositionality raises fundamental questions about what meanings are, what the basic building blocks are and what ways of construction are.

The real question is not whether a certain phenomenon can be analyzed compositionally, as enough methods are available, but what makes the overall theory (un)attractive or (un)acceptable. A case study which follows this line of argument is presented in appendix B: a study by Partee concerning genitives.

7 Fundamental arguments against compositionality

7.1 Introduction

In the present section we discuss some arguments against compositionality which are not based upon the challenge of finding a compositional solution for certain phenomena, but arguments which concern issues of a more fundamental nature. The examples present the original arguments, immediately followed by discussion.

7.2 Ambiguity

Pelletier presents arguments against compositionality based upon its consequences for the analysis of ambiguities (Pelletier (1993*b*),Pelletier (1994)). Some examples are:

(1) Every linguist knows two languages.

(2) John wondered when Alice said she would leave.

(3) The philosophers lifted the piano.

Sentence (1) is ambiguous regarding the total number of languages involved. In (2) the point is whether *when* asks for the time of departure, or the time of Alice's saying this, and in (3) the interpretation differs in whether they did it together or individually.

The above sentences contain no lexical ambiguity, and there are no syntactic arguments to assign them more than one constituent structure. Pelletier (1993*b*) says: 'In order to maintain the Compositionality Principle, theorists have resorted to a number of devices which are all more or less unmotivated (except to maintain the Principle): Montagovian "quantifying-in" rules, "traces", "gaps", "Quantifier Raising", ...features, and many more.'

The issue raised by Pelletier with respect to (1) is a old one, and arises as well for the classical *de dicto* - *de re* ambiguity of :

(4) John seeks a unicorn

Because the quantifying-in rules of Montague Grammar involve such a distortion from the surface form, various attempts have been made to avoid them. An influential proposal was to use Cooper storage (Cooper 1983): the sentence is interpreted compositionally, but the NPs (*every linguist* and *two languages*) are exempted. Their interpretations are put in a storage, and can be retrieved out of storage at a suitable moment. The order in which they are retrieved reflects their relative scope. So Cooper storage introduces an interpretation procedure and an intermediate stage in the model. Perhaps it is a compositional process, but it is questionable whether it constitutes a compositional semantics, because of the essential role of the storage mechanism.

Other approaches try to eliminate the ambiguity. Linguists have argued that the scope order is the surface order. This is known as 'Jackendoff's principle' (Jackendoff 1972). It has been said by semanticists that (1) has only one reading, viz. its weakest reading (*every* wide scope), and that the stronger reading is inferred, when additional information is available. Analogously for (4). These two approaches work well for simple sentences, but they are challenged by more complicated sentences in which the surface order is not a possible reading, or where the different scope readings are logically independent. The latest proposal for dealing with scope ambiguities is by means of 'lifting rules'. The meaning of a noun-phrase can, by means of rules, be 'lifted' to a more abstract level, and different levels yield different scope readings (Hendriks 1993, chapter 1).

No matter which approach is taken to quantifier scope, the situation remains the same with respect to other examples (as (2) and (3)). They are semantically ambiguous, eventhough there are no arguments for more than one derivational structure.

The crucial assumption in Pelletier's arguments is that the derivation of a sentence describes its syntactic structure. But, as is explained in section 3, this is not correct. The derivation tree specifies which rules are combined in what order and this derivation tree constitutes the input to the meaning assignment function. One should not call something 'syntactic structure' which is not intended as such and then refute it, because the notion so defined does not have the desired properties. The syntactic structure (constituent structure) is determined by the output of the syntactic rules. Different derivational processes may generate one and the same constituent structure, and in this way account for semantic ambiguities.

The distinction between derivation and resulting constitutent structure is made in various grammatical theories. In section (3) is illustrated how the quantifying-in rules in Montague grammar derive the *de re* version of (4) and how the rules produce a syntactic structure that differs formally from the derivation tree. In Tree Adjoining Grammars (TAG's) the different scope readings of (1) differ in the order in which the noun-phrases are substituted in the basic tree for *know*. In transformational grammar the two readings of (2) differ in their derivation: in the reading where *when* asks for the time of leaving, is formed from

(5) John wondered Alice said she would leave *when*.

Another classical example is:

(6) The shooting of the hunters was bloody.

For this sentence transformational grammar derives the two readings from two different sources: one in which *the hunters* is in subject position and one in which it is in object position.

7.3 Ontology

In Hintikka (1983, chapter 10), an extended version of (Hintikka 1981), the issue of compositionality is discussed. Besides counterexamples to compositionality (most have been considered in section 6), he presents objections of a fundamental nature.

To illustrate Hintikka's arguments we consider an example involving branching quantifiers.

(7) Every villager has a friend and every townsman has a cousin who are members of the same party.

The meaning representation with branching quantifiers is:

(8) $$\begin{matrix}\forall x \exists y \\ \forall z \exists u\end{matrix}\Big\rangle M(x,y,z,u)$$

The representation indicates the dependency of the quantifiers: the choice of y depends only x, and of u only on z. Formula (8) is an example from a formal language that does not adhere to compositionality. The information about the dependencies of the quantifiers would be lost in a first-order representation.

As Hintikka says, it is easy to provide a linear representation with compositional interpretation when Skolem functions are used:

(9) $\exists f \exists g \forall x \forall z M(x, f(x), z, g(z))$

The connection with Hintikka's own (game-theoretical) treatment for (7) is that (9) can be interpreted as saying that Skolem functions exist which codify (partially) the winning strategy in the correlated game (op. cit. p.281). See chapter 6 of this Handbook, for more information on game theoretical semantics.

So compositionality can be maintained by replacing the first-order quantifiers by higher-order ones. About this, Hintikka (1983, p.20.) says 'It seems to me that this is the strategy employed by Montague Grammarians, who are in fact strongly committed to compositionality. However, the only way they can hope to abide by it is to make use of higher order conceptualizations. There is a price to be paid however. The higher order entities evoked in this "type theoretical ascent" are much less realistic philosophically and psycholinguistically than our original individuals. Hence the ascent is bound to detract from the psycholinguistic and methodological realism of one theory'. Furthermore (op. cit. p.283): 'On a more technical level, the unnaturalness of this procedure is illustrated by the uncertainties that are attached to the interpretation of such higher order variables [..]'. Finally, (op. cit. 285): 'Moreover, the first order formulations have other advantages over higher order ones. In first-order languages we can achieve an axiomatization of logical truths and of valid inferences'.

Hintikka is completely right in his description of the attitudes of Montague Grammarians: they use higher-order objects without hesitation if this turns out to be useful. His objection against compositionality is in a nutshell objecting to the higher-order ontology required by compositionality.

Some comments here are in order (the first two originate from Groenendijk & Stokhof, pers. comm.).

1. If first-order analysis is so natural and psychologically realistic, it would be extremely interesting to have an explanation why it took more than two thousand years since Aristotle before the notion 'first order' was introduced by Frege. And it was presented in a notation that differs considerably from our current notation, as it was not linear.

2. It is difficult to see why the first-order notation matters. If there are ontological commitments, then the notions used in the interpretation of the logic, in the metatheory, are crucial, and not the notation itself. It is, for instance difficult to understand why a winning strategy for a game is more natural than a function from objects to objects (cf. Hintikka's comment on (9)).

3. If it is a point of axiomatizability, it would be interesting to have an axiomatization of game theoretical semantics. As concerns intensional logic, one might use generalized models; with respect to these models there is an axiomatization even for the case of higher-order logic (Gallin 1975).

7.4 Synonymy

Pelletier discusses problems raised by the substitution of synonyms in belief-contexts ((Pelletier 1993*b*, Pelletier 1994)). Consider:

(10) Dentists usually need to hire an attorney.

(11) Tooth doctors commonly require the professional services of a lawyer.

Suppose that these two sentences are synonymous. If we assume that (12) and (13) are formed from respectively (10) and (11) by the same rules, then compositionality implies that (12) and (13) are synonymous.

(12) Kim believes that dentists usually need to hire an attorney.

(13) Kim believes that tooth doctors commonly require the professional services of a lawyer.

However, it easy to make up some story in which Kim believes the embedded sentence in (12), but not the one in (13). Pelletier formulates the following dilemma: either one has to state that (10) and(11) are not synonymous, and conclude that there are no synonymous sentences at all in natural language, or one has to give up compositionality.

Let us consider the situation in more detail. The standard model theoretic semantics says that the extension of *dentist* is a set of individuals; dependent on possible world and the time under consideration. So the meaning of *dentist* is a function from possible worlds and times. For most speakers the meaning of *tooth doctor* is the same function as for *dentist*. The source of the problem raised by Pelletier is that for Kim these meaning functions for *dentist* and *tooth doctor* might differ. This shows that the standard meaning notion is an abstraction that does not take into account that for someone the generally accepted synonymy might not hold. In order to account for this, the meaning function can be given the involved individual an additional argument. Then (10) and (11) are no longer synonymous, nor are (12) and (13). Thus there is no problem for compositionality: we have just found an additional factor.

Are we now claiming that, upon closer inspection, there are no synonymous sentences? The synonymy of belief-sentences is an old issue, and there is a lot of literature about it; for references see Partee (1982) and Salmon & Soames (1988). It seems that Mates (1950) already showed that almost any difference in the embedded clauses makes belief-sentences non-synonymous. But there are several cases of constructional (non-lexical) synonymy. Examples are (14) and (15), and (from Partee (1982)) sentences (16) and (17).

(14) Kim believes that John gives Mary a book

(15) Kim believes that John gives a book to Mary.

(16) Mary believes that for John to leave now would be a mistake

(17) Mary believes that it would be a mistake for John to leave now.

7.5 Psychology

An argument often put forward in defense of compositionality concerns its psychological motivation. The principle explains how a person can understand sentences he has never heard before (see also sections 1.2 and 1.3). This psychological explanation is an important ingredient of the Gricean theory of meaning. However, this motivation for compositionality is rejected by Schiffer (1987). On the one hand he argues that compositionality is not needed in order to give an explanation for that power. On the other hand, he argues that such a compositional approach does not work. We will restrict our attention to this aspect of his book.

A compositional semantic analysis of

(18) Tanya believes that Gustav is a dog.

assumes that belief is a relation between Tanya and some kind of proposition. There are several variants of the propositional theory of belief, some more representational, others more semantic. For all variants of these theories, Schiffer argues that they meet serious problems when they have to explain how Tanya might correctly come to the belief expressed in (18). As examples, we will consider two cases of semantic theories in which the proposition says that Gustav has the property of doghood (Schiffer 1987, pp. 56-57). One approach is that doghood is defined by more or less observable properties. Then the problem arises that these properties are neither separately necessary, nor jointly sufficient, for being a dog. We might learn, for instance, that under illusive circumstances dogs do not have a doggy appearance. As Schiffer remarks, this theory was already demolished by Kripke (1972), and replaced by a theory which says that doghood means being an element of a natural kind. This kind most reasonably is the species 'Canis familiaris'. Membership of this kind is determined by some complex genetic property and it is not something we are directly acquainted with. Now suppose that we encounter a race of dogs we do not recognize as such, and decide that 'shmog' stands for any creature of the same biological species as those creatures. Then (18) can be true, while (19) is false because Tanya may fail to believe that shmogs are dogs.

(19) Tanya believes that Gustav is a shmog

But in the explanation with natural kinds, the sentences have the same content.

Since none of the theories offer a plausible account of the role that *dog* plays in (18), there is no plausible account of the proposition that is supposed to be the content of Tanya's belief. Therefore there is nothing from which the meaning of (18) can be formed compositionally, so compositionality is not met.

Partee (1988) discusses Schiffer's arguments against compositionality, and I fully agree with her opinion that Schiffer does not make a sufficient distinction between semantic facts and psychological facts. There is a fundamental difference between semantic facts concerning belief contexts (as implication and synonymy), and questions that come closer to psychological processes (how can a person sincerely utter such a sentence). What Shiffer showed was that problems arise if one attempts to connect semantic theories with the relation between human beings and their language. Partee points out the analogy between these problems with belief and those with the semantics of proper names (how can one correctly use proper names without being acquainted with the referent). The latter is discussed and explained by Kripke (1972). Partee proposes to solve the problems of belief along the same lines. Her paper is followed by the reaction of Schiffer (Schiffer 1988). However, he does not react to this suggestion, nor to the main point: that a semantic theory is to be distinguished from a psychological theory.

7.6 Flexibility

Partee argues that a finite complete compositional semantics that really deals with natural language is not possible (Partee (1982), Partee (1988)). The reason is that compositional semantic theories are based upon certain simplifying assumptions concerning language, such as a closed language, closed world, a fixed set of semantic primitives and a fixed conceptual frame for the language users. The limitations of model theoretic semantics become clear when the relation is considered between the semantic theory and all the factors that play a role in the interpretation of natural language. The following cases can be distinguished.

1. For some parts of language the meaning can correctly be described as rigidly as just characterized. Examples are words like *and* and *rectangle*.
2. For other parts the semantics is jointly determined by the language users and the way the world is. The language users are only partially acquainted with the meanings. Examples are proper names and natural kinds.
3. There are parts of language where the speaker and hearer have to arrive at a mutually agreed interpretation. Examples are compounds like *boat train* and genitives like *John's team*, the resolution of demonstrative pronouns, and most lexical items.
4. For certain theory dependent terms, i.e. words like *socialism* or *semantics*, there is no expectation of the existence of a 'right' or 'best' interpretation. These terms constitute the main argument in Partee (1982).

Partee's position is the following. Compositional model-theoretic semantics is possible and important, but one should understand the limits of what it can do. In a system of compositional semantics the flexibility of language is abstracted away. Therefore it is too rigid to describe the real life process of communication, and limits the description of language users to creatures or machines whose minds are much more narrowly and rigidly circumscribed than those of human beings. This underscores the argument (mentioned above in the section 7.5) that a theory of natural language semantics should be distinguished from a theory of natural language understanding.

The arguments of Partee describe limitations of the compositional possible world semantics. But most limitations are, in my opinion, just temporary, and not essential. There are several methods to deal compositionally with factors such as personal differences, linguistic context, situational context or vagueness. One may use additional parameters (as in section 7.2 on ambiguity), context constants or variables (see appendix B on genitives), the influence from discourse can be treated compositionally (see section 4.4 on DRT), and vagueness by fuzzy logic. And if for some technical terms speaker and hearer have to come to agreement, and practically nothing can be said in general about their meaning, then we have not reached the limits of compositionality, but the limits of semantics (as is the title of Partee (1982)).

8 A mathematical model of compositionality

8.1 Introduction

In this section a mathematical model is developed that describes the essential aspects of compositional meaning assignment. The assumptions leading to this model have been discussed in section 3. The model is closely related to the one presented in 'Universal Grammar' (Montague 1970*b*). The mathematical tools used in this section are tools from *Universal Algebra*, a branch of mathematics that deals with general structures; a standard textbook is Graetzer (1979). For easy reference, the principle is repeated here:

> The meaning of a compound expression is a function of the meanings of its parts and of the syntactic rule by which they are combined.

8.2 Algebra

The first notion to be considered is *parts*. Since the information on how expressions are formed is given by the syntax of a language, the rules of the grammar determine what the parts of an expression are. The rules build new expressions from old expressions, so they are operators taking inputs and yielding an output. A syntax with this kind of rules is a specific example of what is called in mathematics an *algebra*. Informally stated, an algebra is a set with functions defined on that set. After the formal definitions some examples will be given.

Definitions 8.1. *An* **Algebra** $\mathcal{A}$, *consists of a set A called the* **carrier** *of the algebra, and a set F of functions defined on that set and yielding values in that set. So* $\mathcal{A} = \langle A, F \rangle$. *The elements of the carrier are called the* **elements** *of the algebra. Instead of the name function, often the name* **operator** *is used. If an operator is not defined on the whole carrier, it is called a* **partial operator**. *If* $E = F(E_1, E_2, .., E_n)$ *, then* $E_1, E_2, \ldots,$ *and* E_n *are called* **parts** *of E. If an operator takes n arguments, it is called an* **n-ary operator**.

The notion *set* is a very general notion, and so is the notion *algebra* which has a set as one of its basic ingredients. This abstractness makes algebras suitable models for compositionality, because it is abstracted from the particular grammatical theory. Three examples of a completely different nature will be considered.

1. The algebra $\langle \mathbf{N}, \{+, \times\} \rangle$ of natural numbers $\{1, 2, 3, \ldots\}$, with addition and multiplication as operators.
2. The set of trees (constituent structures) and the operation of making a new tree from two old ones by giving them a common root.
3. The carrier of the algebra consists of the words *boy, girl, apple, pear, likes, takes,the* and all possible strings that can be formed from them. There are two partial defined operations. R_{def} forms from a common noun a noun-phrase by adding the article *the*. R_S forms a sentence from two noun-phrases and a verb. Examples of sentences are *The boy likes the apple* and *The pear takes the girl*.

In order to avoid the misconception that anything is an algebra, finally a *non*-example. Take the third algebra (finite strings of words with concatenation), and add an operator that counts the length of a string. This not an algebra any more, since the lengths (natural numbers) are not elements of the algebra.

8.3 Generators

Next we will define a subclass of the algebras, viz. the finitely generated algebras. To give an example, consider the subset $\{1\}$ in the algebra $\langle \mathbf{N}, \{+\} \rangle$ of natural numbers. By application of the operator $+$ to elements in this subset, that is by calculating $1 + 1$, one gets 2. Then 3 can be produced (by $2 + 1$, or $1 + 2$), and in this way the whole carrier can be obtained. Therefore the subset $\{1\}$ is called *a generating set* for this algebra. Since this algebra has a finite generating set, it is called a finitely generated algebra. If we have in the same algebra the subset $\{2\}$, then only the even numbers can be formed. Therefore the subset $\{2\}$ is *not* a generating subset of the algebra of natural numbers. On the other hand, the even numbers form an algebra, and $\{2\}$ is a generating set for that algebra. More generally, any subset is generating set for some algebra. This can be seen as follows. If one starts with some set, and adds all elements that can be produced from the given set and from already produced elements, then one gets a set that is closed under the given operators. Hence it is an algebra.

Definitions 8.2. *Let $\mathcal{A} = \langle A, F\rangle$ be an algebra, and H be a subset of A. Then $\langle [H], F\rangle$ denotes the smallest algebra containing H, and is called the by H* **generated algebra**. *If $\langle [H], F\rangle = \langle A, F\rangle$, then H is called a generating set for $\mathcal{A}$. The elements of H are called* **generators**. *If H is finite, then $\mathcal{A}$ is called a* **finitely generated** *algebra.*

The first example in section 8.2 is a finitely generated algebra because

$$\langle \mathbf{N}, \{+, \times\}\rangle = \langle [\{1\}], \{+, \times\}\rangle.$$

The last example (with the set of strings over a lexicon) is finitely generated: the lexicon is the generating set. An algebra that is not finitely generated is $\langle \mathbf{N}, \{\times\}\rangle$, the natural numbers with multiplication (it is generated by the set of prime numbers).

A grammar that is suitable for a compositional meaning assignment has to be a generated algebra. Furthermore, some criterion is needed to select certain elements of the algebra as the generated language. For instance the expressions that are output of certain rules, or, (if the grammar generates tree like structures) the elements with root labeled S.

Definition 8.3. *A compositional grammar is a pair $\langle \mathcal{A}, S\rangle$, where $\mathcal{A}$ is a generated algebra $\langle A, F\rangle$, and S a selection predicate that selects a subset of A, so $S(A) \subseteq A$.*

8.4 Terms

In section 3 it was argued that *way of production* is crucial for the purpose of meaning assignment. Therefore it is useful to have a representation for such a production process or derivational history. In section 3 we represented such a derivation by means of a tree. That is not the standard format. Let us first consider the linguistic example given in section 8.2. By application of the operator R_{Def} to the noun *apple*, the noun phrase *the apple* is formed, and likewise *the boy* is formed by application of R_{Def} to *boy*. Next the operator R_S is applied to the just formed noun phrases and the verb *like*, yielding the sentence *the boy likes the apple*. This process is described by the following expression (sequence of symbols):

(1) $R_S\langle R_{Def}\langle \textit{boy}\rangle, R_{Def}\langle \textit{apple}\rangle, \textit{like}\rangle$

Such expressions are called **terms**. There is a simple relation of the terms to the elements in the original algebra. For instance, with the term $R_{Def}\langle \textit{apple}\rangle$ corresponds an element which is found by evaluating the term (i.e. executing the operator on its arguments), viz. the string *the apple*. In principle, different terms may evaluate to the same element, and the evaluation of a term usually is very different from the term itself. Terms can be combined to form new terms: the term (1) above, is formed from the terms $R_{Def}\langle \textit{apple},\rangle$, $R_{Def}\langle \textit{boy}\rangle$ and *like*. Thus the terms over an algebra form an algebra themselves.

Definition 8.4. *Let $\mathcal{B} = \langle [B], F\rangle$ be an algebra. The set of* **terms** *over $\mathcal{B} = \langle [B], F\rangle$, denoted as $T_{B,F}$, is defined as follows:*

1. *for each element in B a new symbol $b \in T_{B,F}$*
2. *For every operator in F there is a new symbol f. If f corresponds with a n-ary operator and $t_1, t_2, \ldots t_n \in T_{B,F}$, then $f\langle t_1, t_2, \ldots t_n\rangle \in T_{B,F}$.*

The terms over $\mathcal{B} = \langle [B], F\rangle$ form an algebra with as operators combinations of terms according to the operators of B. This algebra is called the **term algebra** *over $\langle [B], F\rangle$. This term algebra is denoted $\mathcal{T}_{B,F}$, or shortly $\mathcal{T}_{\mathcal{B}}$.*

In section 3 it was argued that, according to the principle of compositionality of meaning, the derivation of an expression determines its meaning. Hence the meaning assignment is a function defined on the term algebra.

8.5 Homomorphisms

The principle of compositionality does not only tell us on which objects the meaning is defined (terms), but also in which way this has to be done. Suppose we have an expression obtained by application of operation f to arguments $a_1, \ldots, a_n$. Then its translation in algebra $\mathcal{B}$ should be obtained from the translations of its parts, hence by application of an operator g (corresponding with f) to the translations of $a_1, \ldots, a_n$. So, if we let Tr denote the translation function, we have

$$Tr(f\langle a_1, \ldots, a_n\rangle) = g(Tr(a_1), ..Tr(a_n))$$

Such a mapping is called a *homomorphism*. Intuitively speaking, a homomorphism h from an algebra $\mathcal{A}$ to algebra $\mathcal{B}$ is a mapping which respects the structure of $\mathcal{A}$ in the following way. If in $\mathcal{A}$ an element a is obtained by means of application of an operator f, then the image of a is obtained in $\mathcal{B}$ by application of an operator corresponding with f. The structural difference that may arise between $\mathcal{A}$ and $\mathcal{B}$ is that two distinct elements of $\mathcal{A}$ may be mapped to the same element of $\mathcal{B}$, and that two distinct operators of $\mathcal{A}$ may correspond with the same operator in $\mathcal{B}$.

Definition 8.5. *Let $\mathcal{A} = \langle A, F\rangle$ and $\mathcal{B} = \langle B, G\rangle$ be algebras. A mapping $h : \mathcal{A} \to \mathcal{B}$ is called a* **homomorphism** *if there is a 1-1 mapping $h' : F \to G$ such that for all $f \in F$ and all $a_1, \ldots, a_n \in A$ holds $h(f(a_1, \ldots, a_n)) = h'(f)(h(a_1), \ldots, h(a_n))$*

Now that the notions 'terms' and' homomorphisms' are introduced, all ingredients are present needed to formalize 'compositional meaning assignment'.

> A compositional meaning assignment for a language A in a model B is obtained by designing an algebra $\langle [G], F\rangle$ as syntax for A, an algebra $\langle [H], F\rangle$ for B, and by letting the meaning assignment be a homomorphism from the term algebra T_A to $\langle [H], G\rangle$.

8.6 Polynomials

Usually the meaning assignment is not directly given, but indirectly via a translation into a logical language. In section 5 it is explained that the standard way to do this is by using polynomials. Here the algebraic background of this method will be investigated.

First the definition. A polynomial is term with variables, so

Definitions 8.6. *Let $\mathcal{B} = \langle [B], F\rangle$ be an algebra. The set $\mathbf{Pol}^{\mathbf{n}}_{\langle[\mathbf{B}],\mathbf{F}\rangle}$ – shortly Pol^n – of* **n-ary polynomial symbols***, or n-ary polynomials, over the algebra $\langle [B], F\rangle$ is defined as follows:*

1. *For every element in B there is a new symbol (a constant) $b \in Pol^n$.*
2. *For every i, with $1 \leq i \leq n$, there is a variable $x_i \in Pol^n$*
3. *For every operator in F there is a new symbol. If f corresponds with a n-ary operator, and $p_1, p_2, \ldots, p_n \in Pol^n$ then also $f(p_1, p_2, \ldots, p_n) \in Pol^n$.*

The set $Pol_{\langle [B],F\rangle}$ of **polynomial symbols** *over algebra $\langle [B], F\rangle$ is defined as the union for all n of the n-ary polynomial symbols, shortly $Pol = \bigcup_n Pol^n$.*

A polynomial symbol $p \in Pol^n$ defines an n-ary polynomial operator; its value for n given arguments is obtained by evaluating the term that is obtained by replacing x_1 by the first argument x_2 by the second, etc.

Given an algebra $\langle [B], F\rangle$ and a set P of polynomial over A, we obtain a new algebra $\langle [B], P\rangle$ by replacing the original set of operators by the polynomial operators. An algebra obtained in this way is a **polynomially derived algebra**.

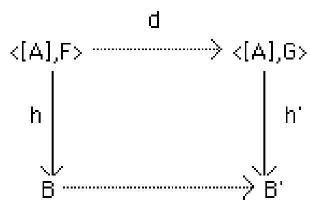

Figure 5: G is safe if for all $\mathcal{B}$ there is a unique $\mathcal{B}'$ such that h', the restriction of h, is a surjective homomorphism

If an operation is added to a given logic, it should be an operation on meanings. In other words, whatever the interpretation of the logic is, the new operator should have a unique semantic interpretation. This is expressed in the definition below, where h is a compositional meaning assignment to the original algebra, and h' describes the interpretation of new operators.

Definition 8.7. *Let $\langle [A], F \rangle$ be an algebra. A collection operators G is called* **safe** *if for all algebras $\mathcal{B}$ and all surjective homomorphisms h from $\mathcal{A}$ onto $\mathcal{B}$ holds that there is a unique algebra $\mathcal{B}'$ such that the restriction h' of h to the elements of $\langle [A], G \rangle$ is a surjective homomorphism.*

This definition is illustrated in figure 5.

Theorem 8.8 ((Montague 1970*b*))**.** *Polynomial operators are safe.*

Proof (sketch) Mimic the polynomial operators in the homomorphic image.

There are of course other methods to define operations on logic, but safeness is then not guaranteed. Examples are

- Replaces all occurrences of x by y
  There is no semantic interpretation for this operator because some of the new y's may become bound. So there is no algebra $\mathcal{B}'$ in the sense of the above theorem.

- Replace all existential quantifiers by universal ones
  For equivalent formulas (e.g. where one formula has $\forall$ and the other $\neg\exists\neg$) non-equivalent results are obtained.

- Recursion on the length of a formula
  In the model for logic the notion length has no interpretation, hence the recursion is not well-founded in the model.

In section 5 several examples were given which show that it is advisable to use only polynomial defined operators. This is not a restriction of the expressive power, as follows from the next theorem.

Theorem 8.9. *Let $\langle A, F \rangle$ be an algebra with infinitely many generators, and G a collection of safe operators over$\langle A, F \rangle$. Then all elements of G are polynomially definable.*

Proof A proof for this theorem is given by van Benthem (1979), and for many sorted algebras by F. Wiedijk in Janssen (1986*a*).

Theorem 8.9 is important for applications since it justifies the restriction to polynomially defined operators. Suppose one introduces a new operator, then either it is safe, and polynomially definable, or it is not safe, and consequently should not be used. In applications the requirement of infinitely many generators is not a real restriction, since the logic usually has indexed variables $x_1, x_2, x_3, \ldots$ Furthermore it is claimed (Wiedijk pers. comm.) that the theorem holds for any algebra with at least two generators.

We may summarize the section by giving the formalization of the principle of compositionality of meaning.

> Let L be some language. A compositional meaning assignment to L is obtained as follows. We design for L a compositional grammar $\mathcal{A} = \langle\langle A_L, F_L\rangle, S_L\rangle$, and a compositional grammar $\mathcal{B} = \langle\langle B, G\rangle, S_B\rangle$ to represent the meanings, where B has a homomorphic interpretation in some model M. The meaning assignment for L is defined by a homomorphism on from T_A to an algebra that is polynomially derived from $\mathcal{B}$.

8.7 Past and future of the model

The algebraic framework presented here is almost the same as the one developed by Montague in Universal Grammar (Montague 1970*b*). That article was written in a time that the mathematical theory of universal algebra was rather young (the first edition of the main textbook in the field (Graetzer 1979) originates from 1968). The notions used in this section are the notions that are standard nowadays, and differ at some cases from the ones used by Montague. For instance, he uses a 'disambiguated language', where we use a 'term algebra', notions which, although closely related, differ not only by name. The algebraic model developed by Montague turned out to be the same as the model used in computer science in the approach to semantics called *initial algebra semantics* (Adj 1978), as was noticed by Janssen & van Emde Boas (1981).

Universal algebra became an important tool in computer science, and there the notions from universal algebra were refined further. Since notions as coercion, overloading, subtyping and modularization play a role not only in computer science, but also in natural language semantics, the model presented in this section can be refined further. For instance, in linguistic applications the involved algebra always is a many sorted algebra (Adj 1977), and an order sorted algebra (Goguen & Diaconescu 1994) seems a very appropriate concept to cover the linguistic concept of 'subcategorization'. Of course, the algebras have to be computable (see e.g. Bergstra & Tucker (1987)). In section 9.5 a restriction will be proposed that reduces the compositional grammars to parsible ones. Further, one might consider the consequences of partial rules. An overview of developments concerning universal algebra in computer science is given in (Wirsing 1990). Montague's framework is redesigned using many sorted algebras in Janssen (1986*a*) and Janssen (1986*b*); that framework is developed further for dealing with flexibility in Hendriks (1993).

9 The formal power of compositionality

9.1 Introduction

In the present section the power of the framework with respect to the generated language and the assigned meanings will be investigated. It will be shown that on the one hand compositionality is restrictive in the sense that, in some circumstances, a compositional analysis is impossible. On the other hand it will be shown that compositionality does not restrict the class of languages that can be analyzed, nor the meanings that can be assigned. Finally a restriction will be considered that guarantees recursiveness.

9.2 Not every grammar can be used

In the preceding sections examples are given which illustrate that not every grammar is suitable for a compositional meaning assignment. The example below gives a formal underpinning of this. A grammar for a language is given, together with the meanings for its expressions. It is proven that it is not possible to assign the given meanings in a compositional way to the given grammar.

Example 9.1. *The basic expressions are the digits:* $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$. *There are two operations in the algebra. The first one makes from a digit a number name and is defined by* $G_1(d) = d$. *The second one makes from a digit and a number name a new number name by writing the digit in front of the number name:* $G_2(d, n) = dn$. *So* $G_2(2, G_1(3)) = G_2(2, 3) = 23$, *and* $G_2(0, G_1(6)) = G_2(0, 6) = 06$. *The meaning of an expression is the natural number it denotes, so* 007 *has the same meaning as* 7. *This meaning function is denoted by* M.

Fact 9.2. *There is no function* F *such that* $M(G_2(a, b)) = F(M(a), M(b))$.

Proof Suppose that there was such an operation F. Since $M(7) = M(007)$, we would have

$$M(27) = M(G_2(2, 7)) = F(M(2), M(7)) = F(M(2), M(007)) = M(G_2(2, 007)) = M(2007)$$

This is a contradiction. Hence no such operation F can exist.
End of Proof

This result is from Janssen (1986*a*); in Zadrozny (1994) a weaker result is proved, viz. that there does not exist a polynomial F with the required property.

A compositional treatment can be obtained by changing rule G_2. The digit should be written at the end of the already obtained number: $G_3(d, n) = nd$. Then there is a corresponding semantic operation F defined by $F(d, n) = 10 \times n + d$, for instance $M(07) = M(G_3(7, 0) = F(M(7), M(0)) = 10 \times M(0) + M(7)$. So a compositional assignment of the intended meaning is possible, but requires another syntax. This illustrates that compositionality becomes possible if semantic considerations influence the design of the syntactic rules.

9.3 Power from syntax

The next theme is the (generative) power of compositional grammars and of compositional meaning assignment. In this section we will consider the results of (Janssen 1986*a*), and in the next section those of (Zadrozny 1994).

In the theorem below it is proved that any recursively enumerable language can be generated by a compositional grammar. The recursively enumerable languages form the class of languages which can be generated by the most powerful kinds of grammars (unrestricted rewriting systems, transformational grammars, Turing machine languages etc.), or, more generally, by any kind of algorithm. Therefore, the theorem shows that if a language can be generated by any algorithm, it can be generated by a compositional grammar. The proof exploits the freedom of compositionality to choose some suitable grammar. The basic idea is that the rules of the grammar (operations of the algebra) can simulate a Turing Machine.

Theorem 9.3. *Any recursively enumerable language can be generated by a compositional grammar.*

Proof In order to prove the theorem, we will simulate a nondeterministic Turing machine of the following type. The machine operates on a tape that has a beginning but no end, and it starts on an empty tape with its read/write head placed on the initial blank. The machine acts on the basis of its memory state and of the symbol read by the head. It may move right (R), left (L) or print a symbol, together with a change of memory state. Two examples of instructions are

I_1 : q_1sq_2R (= if the Turing machine reads in state q_1 an s, then its state changes in q_2 and its head moves to the right)

I_2 : q_1sq_2t (= if the Turing machine reads in state q_1 an s, then its state changes in q_2 and it writes an t)

The machine halts when no instruction is applicable. Then the string of symbols on the tape (neglecting the blanks) is the generated string. The set of all the strings the nondeterministic machine can generate is the generated language.

A compositional grammar is of another nature than a Turing Machine. A grammar does not work with infinite tapes, and it has no memory. These features can be encoded by a finite string in the following way. In any stage of the calculations, the head of the Turing machine has passed only a finite number of positions on the tape. That finite string determines the whole tape, since the remainder is filled with blanks. The current memory state is inserted as an extra symbol in the string on a position to the left of the symbol that is currently scanned by the head. Such strings are elements of the algebra.

Each instruction of the Turing machine will be mimicked by an operation of the algebra. This will be shown below for the two examples mentioned before. Besides this, some additional operations are needed: operations that add additional blanks to the string if the head stands on the last symbol on the right and has to move to the right, and operations that remove at the end of the calculations the state symbol and the blanks from the string. These additional operations will not be described in further detail.

I_1 : The corresponding operator F_1 is defined for strings of the form w_1qsw_2 where w_1 and w_2 are arbitrary strings consisting of symbols from the alphabet and blanks. The effect of F_1 is defined by $F_1(w_1q_1sw_2) = w_1sq_2w_2$.

I_2 : The corresponding operator F_2 is defined for strings of the form $F_2(w_1q_1sw_2) = w_1q_2tw_2$.

Since the algebra imitates the Turing machine, the generated language is the same.
End of Proof The above result can be extended to meanings. The theorem below says that any meaning can be assigned to any language in a compositional way.

Theorem 9.4. *(Any language, any meaning)*
Let L be a recursively enumerable language, and $M : L \rightarrow D$ a computable function of the expressions of L into D. Then there are algebras for L and D with computable operations such that M is an homomorphism.

Proof In the proof of theorem 9.3 the existence is proven of an algebra $\mathcal{A}$ as syntax for the source language L. A variant $\mathcal{A}'$ of $\mathcal{A}$ is taken as grammar for L: the rules produce strings that end with a single #-sign, and an additional rule, say $R_\#$ removes that #. For the semantic algebra a copy of $\mathcal{A}'$ is taken, but instead of $R_\#$ there is a rule R_M that performs the meaning assignment M. Since M is computable, so is R_M. The syntactic rules of $\mathcal{A}'$ extended with $R_\#$ are in a one to one correspondence with the rules of $\mathcal{A}'$ extended with R_M. Hence the meaning assignment is an homomorphism.
End of Proof

9.4 Power from semantics

Zadrozny proves that any semantics can be dealt with in a compositional way. He takes a version of compositionality that is most intuitive: in the syntax only concatenation of strings is used. On the other hand, he exploits the freedom to use unorthodox meanings. Let us quote his theorem (Zadrozny 1994):

Theorem 9.5. *Let M be an arbitrary set. Let A be an arbitrary alphabet. Let "." be a binary operation, and let S be the set closure of A under ".". Let $m : S \rightarrow M$ be an arbitrary function. Then there is a set of functions M^* and a unique map $\mu : S \rightarrow M^*$ such that for all $s, t \in S$*

$$\mu(s.t) = \mu(s)(\mu(t)), \text{ and } \mu(s)(s) = m(s)$$

The first equality says that μ obeys compositionality, and the second equality says that from $\mu(s)$ the originally given meaning can be retrieved. The proof roughly proceeds as follows. The requirement of compositionality is formulated by an infinite set of equations concerning μ. Then a basic lemma from non-wellfounded set theory is evoked, the *solution lemma*. It guarantees that there is a unique solution for this set of equations – in non-wellfounded set theory. This non-wellfounded set theory is a recently developed model for set theory in which the axiom of foundation does not hold. Zadrozny claims that the result also holds if the involved functions are restricted to computable ones.

On the syntactic side this result is very attractive. It formalizes the intuitive version of compositionality: in the syntax there is concatenation of visible parts. However it remains to be investigated for which class of languages this result holds; with a partially defined computable concatenation operation only recursive languages can be generated.

Zadrozny claims that the result also holds if the language is not specified by a (partial) concatenation operation, but by a Turing Machine. However, then the attractiveness of the result disappears (the intuitive form of compositionality), and the same result is obtained as described in the previous section (older and with standard mathematics).

On the semantic side some doubts can be raised. The given original meanings are encoded using non wellfounded sets. It is strange that synonymous sentences get different meanings. Furthermore it is unclear, given two meanings, how to define a useful entailment relation among them.

In spite of these critical comments, the result is a valuable contribution to the discussion of compositionality. It shows that if we restrict the syntax considerably, but are very liberal in the semantics, a lot more is possible than expected. In this way the result is complementary to the results in the previous section. Together the results of Janssen and Zadrozny illustrate that without constraints on syntax and semantics, there are no counterexamples to compositionality. This gives the pleasant feeling that a compositional treatment is somehow always possible.

It has been suggested that restrictions should be proposed because compositionality is now a vacuous principle. That is not the opinion of this author. The challenge of compositional semantics is not to prove the existence of such a semantics, but to obtain one. The formal results do no help in this respect because the proofs of the theorems assume that some meaning assigning function is already given, and then turn it into a compositional one. Compositionality is not vacuous, because we have no recipe to obtain one, and because several proposals are ruled out by the principle. Restrictions should therefore have another motivation. The challenge of semantics is to design a function that assigns meanings, and the present paper argues that the best method is to do so in a compositional way.

9.5 Restriction to recursiveness

In this section a restriction will be discussed that reduces the generative capacity of compositional grammar to recursive sets. The idea is to use rules that are reversible. If a rule is used to generate an expression, the reverse rule can be used to parse that expression. Let us consider an example.

Suppose that there is a rule specified by $R_1(\alpha, \beta, \gamma) = \alpha\ \beta s\ \gamma$. So:

$$R_1(\textit{every man, love, a woman}) = \textit{every man loves a woman}$$

The idea is to introduce a rule R_1^{-1} such that

$$R_1^{-1}(\textit{every man loves a woman}) = \langle \textit{every man, love, a woman} \rangle$$

In a next stage other reverse rules might investigate whether the first element of this tuple is a possible noun phrase, whether the second element is a transitive verb, and whether the third element is a noun phrase. A specification of R_1^{-1} might be: find a word ending on an s, consider the expression before the verb as the first element, the verb (without the s) as the second, and the expression after the verb as the third element. Using reverse rules, a parsing procedure can easily be designed.

The following complications may arise with R_1^{-1} or with another rule:

- **Ill-formed input**
  The input of the parsing process might be a string that is not a correct sentence, e.g *John runs Mary*. Then the given specification of R_1^{-1} is applicable. It is not attractive to make the rule so restrictive that it cannot be applied to ill-formed sentences, because then rule R_1^{-1} would be as complicated as the whole grammar.

- **Applicable on several positions**
  An application of R_1^{-1} (with the given specification) to *The man who seeks Mary loves Suzy* can be applied both to *seeks*, and to *loves*. The information that *the man who* is not a noun-phrase can only be available when the rules for noun-phrase formation are considered. As in the previous case, it is not attractive to make the formulation of R_1^{-1} that restrictive that is is only applicable to wellformed sentence.

- **Infinitely many sources**
  A rule may remove information that is crucial for the reversion. Suppose that a rule deletes all words after the first word of the sentence. Then for a given output, there is an infinite collection of strings that has to be considered as possible inputs.

The above points illustrate that the reverse rule cannot be an inverse function in the mathematical sense. In order to account for the first two points, it is allowed that the reverse rule yields a set of expressions. In order to avoid the last point, it is required that it is a finite set.

Requiring that there is a reverse rule, is not sufficient to obtain a parsing algorithm. For instance, it may be the case the $y \in R_1^{-1}(y)$, and a loop arises. In order to avoid this, it is required that all the rules form expressions which are more complex (in some sense) than their inputs, and that the reverse rule yields expressions that are less complex than the input. Now there is a guarantee that the process of reversion terminates.

The above considerations lead to two restrictions on compositional grammars which together guarantee recursiveness of the generated language. The restrictions are a generalization of the ones in Landsbergen (1981), and provide the basis of the parsing algorithm of the machine translation system 'Rosetta' (see Rosetta (1994)) and of the parsing algorithm in Janssen (1989).

1. **Reversibility**
   For each rule R there is a reverse rule R^{-1} such that
   (a) for all y the set $R^{-1}(y)$ is finite
   (b) $y = R(x_1, x_2, \ldots, x_n)$ if and only if $\langle x_1, x_2, \ldots, x_n \rangle \in R^{-1}(y)$
2. **Measure condition**
   There is a computable function μ hat assigns to an expression a natural number: its measure. Furthermore
   (a) If $y = R(x_1, x_2, ...x_n)$, then $\mu(y) > \max(\mu(x_1), \mu(x_2,), \ldots, \mu(x_n))$
   (b) If $\langle x_1, x_2, \ldots, x_n \rangle \in R^{-1}(y)$ then $\mu(y) > \max(\mu(x_1), \mu(x_2,)...\mu(x_n))$

Assume a given grammar together with reverse rules and a computable measure condition. A parsing algorithm for M-grammars can be based upon the above two restrictions. Condition 1 makes it possible to find, given the output of a generative rule, potential inputs for the rule. Condition 2 guarantees termination of the recursive application of this search process. So the languages generated by grammars satisfying the requirements are decidable languages. Note that the grammar in the proof of theorem 9.3 does not satisfy the requirements, since there is no sense in which the complexity increases, if the head moves to the right or the left.

10 Other applications of compositionality

10.1 Semantics of programming languages

In this section some issues that emerge in semantics of computer science are addressed because they are interesting as regards compositionality.

Environments

In most programming languages names (identifiers) have to be declared: their type has to be stated, in some cases they have to be initialized. Such names can only be used within a certain range: the scope of their declaration. Identifiers with a certain declaration can be hidden temporarily by a new declaration for the same identifier. So the meaning of an identifier depends on the context in which it arises.

Denotational Semantics (Stoy 1977, de Bakker 1980) follows the methods of logic, and has compositionality therefore as a fundamental principle. In this approach an abstraction is used by which a compositional meaning assignment becomes possible. The notion 'environment' encodes which declarations are valid on a certain moment, and the meaning of an identifier depends on (is a function of) the environment. So the same statement can get another effect, depending on the environment with respect to which it is evaluated. Thus they practiced a strategy discussed in sections 2 and 7.

Jumps and continuations

Some programming languages have the instruction to jump to some other part of the program text. The effect of the jump instruction depends on what that other text means. Providing compositionally a meaning to the jumping instruction requires that it gets a meaning without having that other text of the program available. The solution provided in denotational semantics is to describe meanings with respect to possible 'continuations', i.e. with respect to all possible ways the computational process may continue.

Compositional proof systems

An important school is the Floyd-Hoare style of programming language semantics, which expresses meanings in terms of logical proofs (Floyd 1976, Hoare 1969). In doing so it makes use of another form of compositionality, viz. compositionality of proofs: proofs for subprograms can be combined into a proof for the whole program.

Parameter passing

There are several mechanisms for parameter passing; e.g. call by reference, call by value, and call by name. The last one is defined by means of syntactic substitution! In a compositional approach one would like to obtain the meaning of the entire construction by combining the meaning of the procedure name with the meaning of the parameter. Such a compositional analysis is given by Hung & Zucker (1991). They present a uniform semantic treatment for all those mechanisms.

Parallelism

In computer science the recent development of large networks of processors has focussed attention on the behavior of such large systems with communicating processors. New theoretical concepts are needed as the size of the networks produces new problems and the individual processors can themselves become quite complex. In the theory of such systems, compositionality is an important factor: a proof concerning the behavior of the system as a whole should be a function of the proofs for the separate processors. Significant in this respect is the title of de Roever (1985): 'The quest for compositionality- A survey of proof systems for concurrency'.

10.2 Other translations

A we have seen in section 5, a compositional meaning assignment is realized through compositional translation into logic. In other situations precisely the same happens - compositional translation - but the motivation is different. Below we consider translations between logics, between programming languages, and between natural languages.

Embedding logic

For many logical languages translations have been defined. The purpose is not to assign meanings, but to investigate the relation between the logics, for instance, their relative strength or their relative consistency. A famous example is Gödel's translation of intuitionistic logic into modal logic. It illustrates the method of using polynomially defined algebras.

In intuitionistic logic the connectives have a constructive interpretation. For instance $\phi \rightarrow \psi$ could be read as 'given a proof for ϕ, it can be transformed into a proof for ψ'. The disjunction $\phi \vee \psi$ is read as 'a proof for ϕ is available or a proof for ψ is available'. Since it may be the case that neither a proof for ϕ nor for $\neg\phi$ is available, it is explained why $\phi \vee \neg\phi$ is not a tautology in intuitionistic logic. These interpretations have a modal flavor, made explicit in the translation into modal logic.

Let us write Tr for the translation function. Then clauses of the translation are:

1. Tr(p) $= \Box p$, for p an atom
2. Tr($\phi \vee \psi$) $= Tr(\phi) \vee Tr(\psi)$
3. Tr($\phi \wedge \psi$) $= Tr(\phi) \wedge Tr(\psi)$

4. $\text{Tr}(\phi \to \psi) = \Box[Tr(\phi) \to Tr(\psi)]$.

Thus one sees that the disjunction and conjunction operator in intuitionistic logic correspond to the same operator of modal logic, whereas the implication corresponds to a polynomially defined operator. Since $\neg\phi$ is an abbreviation for $\phi \to \perp$, the translation of $p \vee \neg p$ is $\Box p \vee \Box\neg\Box p$ (which is not a tautology in modal logic).

The above example illustrates that the Gödel translation is an example of the method of compositional translation. A large number of translations between logics is collected in Epstein (1990, Chapter 10: 'Translations between Logic'. pp. 289-314). Almost all of them are compositional (there they are called 'grammatical translations'). The few that are not, are also in semantic respects deviant.

Compiler correctness

Compilation of a computer program can be viewed as a form of translation, viz. from a programming language to a more machine oriented language. The purpose is to instruct the machine how to execute the program. This translation has of course to respect the intended meaning of the programming language, an aim that is called 'compiler correctness'. It has been has been advocated that one can approach compiler correctness by using algebraic methods (Morris 1973, Thatcher, Wagner & Wright 1979), in other words, by working compositionally. Other arguments are given in Rus (1991).

Between natural languages

Translating from one natural language to another one is an action that should be meaning preserving. The machine translation project 'Rosetta' tries to reach this aim by following the principle of compositionality of translation. It reads (Rosetta 1994, p. 17)

> Two expressions are each others translation if they are built up from parts which are each others translation, by means of rules with the same meaning.

11 Conclusion

The principle of compositionality of meaning really means something. It is a restriction that rules out several proposals in the literature, and is certainly not vacuous. On the other hand it was shown that there are several methods to obtain a compositional meaning assignment; so it is not an impossible task. For counterexamples to compositionality solutions were proposed, and fundamental arguments were answered.

This practical experience was supported by mathematical proofs that the sentences of any language can be assigned any meaning in a compositional way. However, the formal results do not make it any easier to obtain a compositional semantics, so these results form no reason for restrictions.

Compositionality is not a formal restriction on what can be achieved, but a methodology on how to proceed. The discussions in this chapter have pointed to several advantages of this methodology, in particular its heuristic value. It suggests solutions to semantic problems. It helps to find weak spots in non-compositional proposals; such proposals have a risk of being defective. Cases where an initially non-compositional proposal was turned into a compositional one, the analysis improved considerably.

Compositionality requires a decision on what in a given approach the basic semantic units are: if one has to build meanings from them, it has to be decided what these units are. Compositionality also requires a decision on what the basic units in syntax are, and how they are combined. If a proposal is not compositional, it is an indication that the fundamental question what the basic units are, is not answered satisfactorily. If such an answer is provided, the situation under discussion is better understood. So the main reason to follow this methodology, is that compositionality guides research in the right direction!

Acknowledgments

I am indebted to Yuri Engelhardt, J. Goguen, Willem Groeneveld, Herman Hendriks, Lex Hendriks, Barbara Partee, F.J. Pelletier, the participants of the 'handbook-workshop' and especially to Ede Zimmermann for their comments on the earlier versions of this chapter. I am very grateful to Barbara Partee for the permission to include her work on genitives as an appendix. Originally it was written as part of her paper on compositionality (Partee 1984), but it was not included in the final version. It has, in mutual concert, slightly been edited and updated in order to fit in the present context. I thank D. Westerståhl and B. Pagin for their valuable help in obtaining an otherwise untraceable reference. For their stimulating guidance during the preparation of this chapter I thank the editors Johan van Benthem and Alice ter Meulen, who also turned my extraordinary English into intelligible prose.

A Appendix: related principles

In this section we present shortly several principles which arise in discussions in the literature concerning compositionality. Some are variants of compositionality, others are alternatives, or are independent of compositionality.

Compositionality of meaning

The version one mostly finds in the literature is: *The meaning of a compound expression is built from the meanings of its parts.* A more precise version is (Partee et al. 1990, p. 318.): *The meaning of a compound expression is a function of the meanings of its parts and of the syntactic rule by which they are combined.* This principle is main theme of this chapter.

Compositionality of translation

The translation of a compound expression is built from the translations of its parts. This principle was a guideline in the design of a variant of Eurotra (Arnold 1985, Arnold & des Tombes 1987). A symmetric and more precise version (see also section 10.2) is given in Rosetta (1994, p. 17): *Two expressions are each other's translation if they are built up from parts which are each other's translation, by means of rules with the same meaning.* This principle is analogous to the compositionality principle.

Context independence thesis

The meaning of an expression should not depend on the context in which it occurs (Hintikka 1983, p. 262). Closely related with the 'inside outside principle'. This thesis follows from the compositionality principle.

Contextuality principle

A word has a meaning only in the context of a sentence, not in separation. (Frege 1884, p. xii). This principle seems to be the opposite of compositionality, see the discussion in section 1.3.

Determinacy thesis

The meaning of E must completely be determined by the meanings of the expressions $E_1, E_2, \ldots, E_n$ *from which it constructed* (Hintikka 1983, p. 264). This thesis follows from the compositionality principle.

Frege's principle

'Frege's Principle' is another name for the principle of compositionality. Whether the ascription to Frege is accurate, is discussed in section 1.3.

Initial algebra semantics

In computer science a well known approach to semantics (Adj 1977, Adj 1978). It states that *the syntax is an initial algebra, the meanings form an algebra of the same type, and meaning assignment is a homomorphism.* Intuitively the notion 'Initial' says that two elements in an algebra are different unless it is explicitly said that they are the same. A standard example of an initial algebra is a term algebra, hence compositionality of meaning is an example of initial algebra semantics.

Inside outside principle

The proper direction of a semantic analysis is from the inside out.

(Hintikka 1983, p. 262). This principle follows from the compositionality principle.

Leibniz' principle

A well known principle concerning semantics of the philosopher Leibniz (Gerhardt 1890, p. 228) *Eadem sunt, quorum substitui alteri, salva veritate.* (Those-the-same are, of-which is-substitutable for-the-other, with truth).

The principle is understood as saying that two expressions refer to the same object if in all contexts they can be interchanged without changing the truth value. We may generalize it to all kinds of expressions, stating that two expressions have the same meaning if in all contexts the expressions can be interchanged without changing the truth value. This is the reverse of the consequences for meanings of the compositionality principle. Note that, according to our standards, this principle is sloppy formulated, because it confuses the things themselves, with the expressions to referring to them (see Church (1956, p. 300) or Quine (1960, p. 116)).

Rule to rule hypothesis

For each syntactic rule, which tells how a complex expression E is formed from simpler ones say $E_1, E_2, \ldots, E_n$*, there is a corresponding semantic rule which tells how the meaning of E depends on the meanings of* $E_1, E_2, \ldots, E_n$. The term 'rule to rule hypothesis' originates from (Bach 1976), it is called **parallelism thesis** in Hintikka (1983). This hypothesis is the kernel of the compositionality principle.

Semantic groundedness

An alternative for compositionality proposed by Pelletier (1994). It is, like compositionality, based on an inductively defined meaning assignment, The difference is that here the induction does not follow the syntactic definition, but can be based on any other grounded ordering. An example is a definition of propositional logic in which the syntax forms the biimplication $\phi \leftrightarrow \psi$ from ϕ and ψ, but in which the meaning is defined by means of the two implications $\phi \rightarrow \psi$ and $\psi \rightarrow \phi$.

Surface compositionality

If expression E is built from expressions $E_1, E_1 \ldots E_n$, then these parts are actual parts of the resulting expression, they occur unchanged as subexpressions of E. A further refinement of the principle and a grammar obeying the principle is given in Hausser (1984). It is called the **invariance thesis** by Hintikka (1983, p. 263). This is a very restricted version of the compositionality principle.

B Appendix: Genitives - A case study (by B. Partee)

B.1 Introduction

In this appendix we will consider a difficult case for compositionality: the variety of meanings of genitives. It will turn out that the problems can be solved compositionally by methods discussed before. The aim of this section is to illustrate that this is not the end of the story. Designing a compositional solution for a given phenomenon may implicate decisions that have consequences in other parts of the grammar, and these consequences have to be taken into account as well. It is possible that the new insights give an improvement of the grammar as a whole, but it may also be the case the system becomes unnecessarily complicated. If certain decisions can be given no other argumentation than to preserve compositionality, then we may have chosen the wrong solution, or we may be working with a too narrow conception of compositionality.

B.2 The problem

Here are some initial data:

(1) (a) John's team
(b) A team of John's
(c) That team is John's

(2) (a) John's brother
(b) A brother of John's
(c) * That brother is John's

(3) (a) John's favorite movie
(b) A favorite movie of John's
(c) * That favorite movie is John's

Informally, we can give a unified description of the interpretation of the genitive phrase *John's* that applies to all these cases if we say that the genitive always expresses one argument of a relation (in intensional logic, something like ${}^{\vee}R(j)$). But the relation can come from any of the three sources:

1. The context. In (1), the most salient relevant relation might be ”plays for”, ”owns”, ”has bet on”, ”writes about for the local newspaper”, or any of an essentially open range of possibilities (henceforth the ”free R” reading).)

2. An inherently relational noun, like *brother* in (2)

3. A relational adjective, like *favorite* in (3).

I'll refer to the last two cases as the ”inherent R” readings.

Compositionality asks for a uniform semantics of the genitive construction in syntax. Since not all examples contain a relational noun or adjective, the best hope for a unified analysis would clearly seem to be to try to assimilate all cases to the ”free R case”. This is in fact the strategy carried out by Hellan (1980). Simplifying his approach, we may say that he points out that an inherently relational noun can be presumed to make its associated relation salient in the context (while still being analyzed as a simple CN syntactically and a one-place predicate semantically).

Maybe this approach works for the given examples, but serious problems emerge if we consider the contrasts between NP-internal and predicative uses of genitives. In addition to the contrast among the (a) and (b) cases in (1)-(3) above, an interesting pattern of interpretations can be found in Stockwell, Schachter & Partee (1973). They give the following examples (the section on genitives was primarily done by Schachter and Frank Heny):

(4) (a) John's portrait. *(ambiguous)*
(b) (i) A portrait of John's. *(free R only)*
(ii) A portrait of John. *(inherent R only)*
(c) That portrait is John's. *(free R only)*

What emerges from these examples is that while predicative genitives (the (c) cases in (1)-(4)) are easily interpreted in terms of a free relation variable which can get its value from the context, they do not seem able to pick up a relation inherent within the subject-NP as a value for that variable. The postnominal genitive and non-genitive *of* - complements in (4b) seem to offer a minimal contrast which is neutralized in the prenominal genitive (4a), providing further evidence that the ”free R” and the ”inherent R” readings should be represented distinctly at some level within the grammar.

A caveat should be added concerning the predicative genitives. In some cases they appear to get an inherent relational reading, as in:

(5) I knew there were three grandmothers behind the curtain, but I didn't know one of them was mine.

We can understand *mine* in (5) as *my grandmother*; but I believe the complicating factor is a result of the phenomenon described in transformational terms as *one(s)* - deletion (Stockwell et al. 1973). It seems that whenever a genuinely t/e-type genitive appears, it must be interpreted with a free R variable. In the present section the full-NP reading of bare genitives (of which (5) is an example) are omitted from further consideration.

B.3 A compositional analysis

Above we we have seen that the genitive construction seems to have two basic meanings. A strategy described in previous sections can be applied here: eliminating the ambiguity by introducing new parts. This is done by enriching the syntax to include a category TCN of ”transitive common noun phrases”, thus making the inherently relational nature overt in their syntactic category and semantic type. The basic idea is that there are two basic genitive constructions, a predicative one with a free R variable (context-dependent), and an adnominal one which applies to transitive common nouns and fills in an argument place, yielding an ordinary one-place CN as result. The predicative one also has a postnominal counterpart, but of category CN/CN, and both have determiner counterparts of categories NP/TCN and NP/CN respectively.

Below a grammar for the genitive is presented; this grammar will be extended in the next section. Details of the analysis not immediately relevant to the genitive issue are not to be taken too seriously.

1. **Predicative Genitives** ((1c)-(4c))
   - Syntax: $[NP\text{'}s]_{t/e}$
   - Semantics: $\lambda x[{}^{\vee}R_i(NP')(x)]$ or equivalently ${}^{\vee}R_i(NP')$
   - Notes: The R_i in this interpretation is free; if context dependency should rather be treated by special constants, this would be one of those.

2. **Postnominal genitives** ((1b)-(4b))

   (a) **Free R Type**
   - Syntax: $[\textit{of } NP\text{'}s]_{CN/CN}$
   - Semantics: $\lambda P \lambda x[{}^{\vee}P(x) \wedge^{\vee} R_i(NP')(x)]$
     or in the notation of Partee & Rooth (1983): $\lambda P[{}^{\vee}P \sqcap {}^{\vee}R_i(NP')]$
   - Notes: This is exactly parallel to the conversion of t/e adjectives to CN/CN adjectives

   (b) **Inherent R type**
   - Syntax: $[\textit{of } NP\text{'}s]_{CN/TCN_{[+gen]}}$
   - Semantics: $\lambda R \lambda x[{}^{\vee}R_i(NP')(x)]$
   - Notes: The symbol $\mathrm{TCN}_{[+\mathrm{gen}]}$ is used to mark the subcategory of relational nouns which can take postnominal *of* + genitive (*brother, employee, enemy*, but not *portrait, description, height*); some relational nouns take *of* + accusative, some can take both. The data are messy; "heaviness" of the NP plays a role. Note that the agentive "by John" reading of (4b) counts as a free R reading; only the "of John" reading is blocked in (4b) and (4c).

3. **Prenominal genitives** ((1a)-(4a))

   (a) **free R type**
   - Syntax: $[NP\text{'}s]_{NP/CN}$
   - Semantics: Tantamount roughly to the *the* + [*of* NP's]$_{\mathrm{CN/CN}}$, but see Notes below. Using Montague's treatment of *the*, this is:

     $$\lambda Q \lambda P[NP'({}^{\wedge}\lambda z[\exists x[\forall y[[{}^{\vee}Q(y) \wedge^{\vee} R_i(y)(z) \leftrightarrow y = x] \wedge^{\vee} P(x)]])]$$

   - Notes: A quantifier in a prenominal genitive always has wide scope, while those in postnominal genitives seem to be ambiguous. The uniqueness condition this analysis imputes to *John's brother* is disputable, especially when the whole noun phrase occurs in predicate position.

   (b) **Inherent R type**
   - Syntax: $[NP\text{'}s]_{NP/TCN}$

- Semantics: Similarly tantamount to the *the* + [*of* NP's]$_{\text{CN/TCN}}$:

$$\lambda R \lambda P[NP'(^\wedge \lambda z[\exists x[\forall y[[^\vee R(z)(y) \leftrightarrow y = x] \wedge^\vee P(x)]]])]$$

- Notes: The order of the arguments of R are reversed in the two determiners; this reflects the intuitive difference in natural paraphrases using e.g.*owns* for the free R in *John's team* and *(is a) sister of* for *John's sister*. But this difference is not predicted or explained here, and to be fully consistent the arguments in the two other 'free R' genitives should be reversed as well.

B.4 Consequences for adjectives

In the previous section a compositional analysis is given for the genitive construction by distinguishing two types of common nouns. But having more types of common nouns, implicates more types of prenominal adjectives, viz. CN/CN, TCN/TCN and TCN/CN. We consider examples of adjectives of the new types.

1. TCN/CN: $favorite_1$, as in *John's favorite movie.*
   - Syntax: $[favorite]_{TCN/CN}$
   - Semantics: Lexical; roughly

   $$favorite'_1 = \lambda P[\lambda y[\lambda x[^\vee P(x) \textit{ and y likes x best out of } {}^\vee P]]]$$

2. TCN/TCN: $favorite_2$, as in *John's favorite brother*
   - Syntax: $[favorite]_{TCN/TCN}$, probably derivable by lexical rule from $favorite_1$.
   - Semantics: lexical, but derivative; roughly

   $$favorite'_2 = \lambda R[\lambda y[\lambda x[^\vee R(y)(x) \wedge favorite'_1(^\wedge(^\vee R(y)))(x)]]]$$

This analysis of inherently relational adjectives creates non-basic TCN's which act just like basic TCN's with respect to genitives. Once these categories are admitted, it appears that a number of traditionally CN/CN adjectives like *new* also fit here as well; we can distinguish four separate (but related) *new*'s as follows:

1. $[new_1]_{t/e}$ "hasn't existed long" *(a new movie)*
2. $[new_2]_{CN/CN}$ "hasn't been a CN long" *(a new movie star)*
3. $[new_3]_{TCN/TCN}$ "hasn't been a TCN-of long" *(my new friend)*
4. $[new_4]_{TCN/CN}$ "hasn't been in the (free) R_i-relation too long"
   (John's new car is an old car)

New_4 is definable in terms of new_3 and a free R as is shown in:

(6) $new'_4 = \lambda P[\lambda y[\lambda x[^\vee P(x) \wedge new'_3 R(y)(x)]]]$

Note the difference between $[favorite]_{\text{TCN/CN}}$ with an "inherent" R built into its meaning, and $[new]_{\text{TCN/CN}}$ which introduces a 'free R' , which in turn acts as "inherent" for the genitive.

Thus the analysis of genitives has stimulated a more refined analysis of adjectives. The above treatment gives a reasonable account of the data: the distribution of 'inherent' and 'free' R readings is explained by treating the 'inherent R' genitive as something which must be in construction with a TCN, which can only happen within the NP, while the 'free R' genitive is basically a predicate. The fact that TCN's can almost always be used as plain CN's would be attributed to the existence of highly productive lexical rules which "detransitivize' TCN's, interpreting the missing argument as existentially quantified or as an indexical or variable.

B.5 Doubts about the introduction of TCN's

Although the grammar from the previous two sections deals with the phenomena, and gives interesting insights, there can be serious reservations about introducing the category TCN into the syntax along with the associated distinctions in the categories of adjectives and determiners. The distinction between transitive and intransitive verbs has clear syntactic and morphological as well semantic motivation in many languages, while with nouns the motivation is almost entirely semantic. I believe that the analysis given above incorporates ingredients of a good explanation, but puts too much of it in the syntax.

Besides these general considerations, there are also phenomena which raise doubts. Consequences emerge when we consider what explanation to give of the semantics of *have* in sentences like (7)-(9).

(7) John has a car

(8) John has a sister

(9) John has three sisters and two brothers.

We could account for (7) and (8) by positing two *have*'s, one ordinary transitive verb (IV/NP) $have_1$ interpreted as a free variable R (with typical values such as 'own', but highly context dependent), plus a $have_2$ of category IV/TCN interpreted as in:

(10) $have_2' = \lambda R \lambda x[\exists y R(x)(y)]$

This requires us to treat *a sister* in (8) as not an NP, but a TCN, and similarly for even more complex indefinite noun phrases, as in (9). We could defend such a departure from apparent surface syntax, with arguments about the inadequacy of Montague's treatment of predicate nominals as ordinary NP's and with appeals to the diversity and interrelatedness across languages of constructions expressing possession, existence, and location, justifying the singling out of *have* for special treatment along with *be*. But putting this in terms of categorial distinctions in the syntax would predict the impossibility of sentences like:

(11) John has piles of money and no living relatives

(12) John has a tutor, a textbook, and a set of papers

(13) John has a good job, a nice house, a beautiful wife, clever children, and plenty of money (and an ulcer).

Conjoinability is a very strong test of sameness of syntactic and semantic category, and in this case it supports the traditional assumption that these are all NP's, and not a mixture of NP's and TCN's. This suggests that the interaction of the interpretation of *have* with relational nouns should not be dealt with by multiplying syntactic categories. And while the conjunction test does not give similarly clear evidence in the genitive construction, I expect that if we can find a way to treat the *have* data without TCN's in the syntax, we will be able to extend it to a treatment of the genitives (probably still recognizing two genitives, but without invoking TCN's to explain the difference).

B.6 Genitives and compositionality

There are several points at which the problems raised by the genitive construction relate to general issues concerning compositionality

1. If we were not committed to local and deterministic compositionality, we could extract a uniform core meaning that all the genitives described above share: [NP's] means ${}^{\vee}R(NP')$. And we could, I think, describe general principles that dictate what more must be "filled in" for the postnominal and determiner uses, and whether the variable is to be left free or bound by a λR operator. This approach would couple a uniform interpretation of the genitive with a not totally implausible interpretation strategy that could be caricatured as "try to understand" (according to Bach a term originating from Philip Gouch). Arguments for such an interpretation strategy for semantically open-ended expressions are given in Partee (1988).

2. Montague's strategy for maintaining uniformity in the face of apparent diversity might be characterized as "generalize to the worst case". I don't think that will work for the analysis of the genitives, since trying to assimilate all genitives to the "free R" case gives the wrong result for the distribution of "inherent" readings. The only way I can see to give a *uniform* treatment of all genitives in English is to leave part of the meaning out of the grammar as sketched in paragraph 1) above. Perhaps a type-shifting along the lines of Partee (1987) could be explored.

3. If we do maintain the compositionality principle by building in the kind of multiple categorization described above, we simplify the process of determining semantic information from syntactic form, but complicate the task of parsing and ambiguity resolution, since we have simultaneously increased lexical and syntactic ambiguity.

4. The motivation for the introduction of TCN's was a desire to make explicit the role of the implicit second argument of relational nouns in the interpretation of genitives. In quantificational genitives like *every woman's husband* and in similar cases with *have*, the implicit argument becomes a bound variable (for other examples of this phenomenon, see section 4 in Partee (1984)). This seems to give an obstacle to a treatment which would absorb these implicit arguments into meanings of the predicates, namely the absence of any way to describe "variable binding" phenomena without an overt variable to bind. Since syntactic evidence goes rather strongly against introducing transitive common nouns, this adds to the motivation for seeking an alternative that would allow variable-like meanings as parts of predicate meanings, as argued in Partee (1989).

5. Although most of the above points suggest that the given treatment is not completely satisfactory, one aspect should be mentioned. For the compositional solution it is clear that it deals with the phenomena, how it would work out in a grammar, and how it would interact with other rules. For the suggested alternatives (interpretation strategy, partially unspecified meanings, new variable mechanisms) this is unclear.

References

Adj (1977), 'Initial algebra semantics and continuous algebras', *Journal of the ACM (Association for Computing Machinery)* **24**, 68–95. Adj ={J.A. Goguen, J.W. Thatcher, E.G. Wagner, J.B. Wright}.

Adj (1978), An initial algebra approach to the specification, correctness and implementation of abstract data types, *in* R. Yeh, ed., 'Current Trends in Programming Methodology', Prentice Hall, pp. 80–149. Adj = {J.A. Goguen, J.W. Thatcher, E.G. Wagner}.

Arnold, D. & des Tombes, L. (1987), Basic theory and methodology in EUROTRA, *in* S. Nirenburg, ed., 'Machine Translation. Theoretical and Methodological Issues', Cambridge University Press, Cambridge, pp. 114–134.

Arnold, D. J. *et al.* (1985), A MU1 view of the CAT framework in EUROTRA, *in* 'Proceedings of the Conference on Theoretical and Methodological Issues in Machine Translation of Natural Languages', Colgate University, Hamilton NY, pp. 1–14. auhors = {D.J. Arnold, L. Jaspaert, R.L. Johnson, S.Krauwer, M. Rosner, L. des Tombes, G.B. Varile and S. Warwick}.

Bach, E. (1976), An extension of classical transformational grammar, *in* 'Problems in Linguistic metatheory', Michigan State University. Proceedings of the 1976 conference.

de Bakker, J.W. (1980), *Mathematical theory of program correctness*, Series in Computer Science, Prentice Hall, London.

Barwise, J. & Perry, J. (1983), *Situations and attitudes*, Bradford books, Cambridge, Mass.

van Benthem, J.F.A.K (1979), Universal algebra and model theory. Two excursions on the border, Technical Report ZW-7908, Dept. of mathematics, Groningen University.

Bergstra, J.A. & Tucker, J.V. (1987), 'Algebraic specifications of computable and semicomputable data types', *Theoretical Computer Science* **50**, 137–181.

Church, A. (1956), *Introduction to mathematical logic*, Vol. 1, Princeton Univ. Press, Princeton N.J.

Cooper, R. (1983), *Quantification ans Syntactic Theory*, number 21 *in* 'Synthese Language Library', Reidel, Dordrecht.

Davidson, D. & Harman, G., eds (1972), *Semantics of natural language*, number 40 *in* 'Synthese library', Reidel, Dordrecht.

Dowty, D.R. (1976), Montague grammar and the lexical decomposition of causative verbs, *in* Partee (1976), pp. 201–245.

Dummett, M. (1973), *Frege. Philosophy of language*, Duckworth, London.

Epstein, R.L. (1990), *The semantic foundation of logic. vol 1. Propositional logic.*, number 35 *in* 'Nijhof international Philosophy Series', M. Nijhof.

Floyd, R.W. (1976), Assigning meanings to program, *in* J.T. Schwartz, ed., 'Mathematical aspects of computer science', number 19 *in* 'Proc. Symp. in Applied Mathematics', American Mathematical Society, Providence (R.I.), pp. 19–32.

Frege, G. (1884), *Die Grundlagen der Arithmetik. Eine logisch-mathematische Untersuchung über den Begriff der Zahl*, W. Koebner, Breslau. Reprint published by: Georg Olms, Hildesheim, 1961.

Frege, G. (1923), Logische untersuchungen. dritter teil: Gedankenfüge, *in* 'Beiträge zur Philosophie des Deutschen Idealismus', Vol. III, pp. 36–51. Reprinted in I. Angelelli (ed.), *Gottlob Frege. Kleine Schriften*, Georg Olms, Hildeheim, 1967, pp. 378-394. Translated as *Compound thoughts* in P.T.Geach & R.H. Stoothoff (transl.), *Logical investigations. Gottlob Frege*, Basil Blackwell, Oxford, 1977, pp. 55-78.

Gallin, D. (1975), *Intensional and higher-order modal logic*, number 17 *in* 'Mathematics studies', North Holland, Amsterdam.

Gerhardt, C.I., ed. (1890), *Die philosphischen Schriften von Gottfried Wilhelm Leibniz*, Vol. 7, Weidmannsche Buchhandlung, Berlin.

Goguen, J. & Diaconescu, R. (1994), 'An Oxford survey of order sorted algebra', *Mathematical Structures in Computer Science* **4**, 363–392.

Graetzer, G (1979), *Universal Algebra. Second Edition*, Springer, New York. First edition published by van Nostrand, Princeton, 1968.

Groenendijk, J. & Stokhof, M. (1991), 'Dynamic predicate logic', *Linguistics and Philosophy* **14**, 39–100.

Hausser, R.R. (1984), *Surface compositional grammar*, number 4 *in* 'Studies in theoretical linguistics', Fink Verlag, Müenchen.

Hellan, L. (1980), Toward an integrated theory of noun phrases, Technical report, University of Trondheim.

Hendriks, H. (1993), Studied Flexibility. Categories and types in Syntax and Semantics, PhD thesis, University of Amsterdam. in ILLC dissertation Series 1993-5.

Henkin, L., Monk, J.D. & Tarski, A. (1971), *Cylindric algebras. Part I*, number 64 *in* 'Studies in logic and the foundations of mathematics', North Holland, Amsterdam.

Higginbotham, J. (1986), Linguistic theory and Davidson's program in semantics, *in* E. le Pore, ed., 'Truth and Interpretation. Perspectives on the Philosophy of Donald Davidson', Blasil Blackwell, Oxford, pp. 29–48.

Hintikka, J. (1981), Theories of truth and learnable languages, *in* S. Kanger & S. Ohman, eds, ''Philosphy and Grammar: papers on the Occasion of the Quincentennial of Uppsala University', Reidel, Dordrecht, pp. 37–57.

Hintikka, J. (1983), *The Game of Language. Studies in Game-Theoretical Semantics and Its Applications*, number 22 *in* 'Synthese Language Library', Reidel, Dordrecht. In collaboration with J. Kulas.

Hoare, C.A.R. (1969), 'An axiomatic base for computer programming', *Communications of the ACM (Association for Computing Machinery)* **12**, 576–580.

Hung, H.-K. & Zucker, J. I. (1991), Semantics of pointers, referencing and dereferencing with intensional logic, *in* 'Proc. 6th annual IEEE symposium on Logic in Computer Science', IEEE Computer Society Press, Los Almolitos, California, pp. 127–136.

Jackendoff, R.S. (1972), *Semantic Interpretation in Generative Grammar*, MIT Press, Cambridge, Mass.

Janssen, T.M.V. (1983), Scope ambiguities of tense, aspect and negation, *in* F. Heny & B. Richards, eds, 'Syntactic categories: Auxiliaries and related puzzles', number 20 *in* 'Synthese Language Library', Reidel, Dordrecht, pp. 55–99.

Janssen, T.M.V. (1986*a*), *Foundations and Applications of Montague Grammar: part 1, Philosophy, Framework, Computer Science*, number 19 *in* 'CWI tract', Centre for Mathematics and Computer Science, Amsterdam.

Janssen, T.M.V. (1986*b*), *Foundations and Applications of Montague Grammar: part 2, Applications to Natural Language*, number 28 *in* 'CWI tracts', Centre for Mathematics and Computer Science, Amsterdam.

Janssen, T.M.V. (1989), Towards a universal parsing algoritm for functional grammar, *in* J.H. Conally & S.C.Dik, eds, 'Functional grammar and the computer', Foris, Dordrecht, pp. 65–75.

Janssen, T.M.V. & van Emde Boas, P. (1981), Some remarks on compositional semantics, *in* D. Kozen, ed., 'Logic of Programs', number 131 *in* 'Springer Lecture Notes in Computer Science', Springer, Berlin, pp. 137–149.

Janssen, T.M.V. & van Emde Boas, P. (1977), On the proper treatment of referencing, dereferencing and assignment, *in* A. Salomaa & M. Steinby, eds, 'Automata, languages and programming (Proc. 4th. coll. Turku)', number 52 *in* 'Lecture notes in computer science', Springer, Berlin, pp. 282–300.

Kamp, H. (1971), 'Formal properties of "now"', *Theoria* **37**, 227–273.

Kamp, H. (1981), A theory of truth and semantic representation, *in* T. Janssen Groenendijk, J. & M. Stokhof, eds, 'A theory of truth and semantic representation', CWI, Amsterdam. reprinted in J. Groenendijk, T. Janssen, and M. Stokhof, eds. *Truth, Interpretation and Information*, Foris, Dordrecht, 1984.

Kamp, H & Reyle, U. (1993), *From Discourse to Logic. Introduction to the Modeltheoretic Semantics of natural language*, number 42 *in* 'Studies in Linguistics and Philosphy', Reidel, Dordrecht.

Katz, J.J. (1966), *The philosophy of language*, Harper and Row, London.

Katz, J.J. & Fodor, J.A. (1963), 'The structure of a semantic theory', *Language* **39**, 170–210.

Kreisel, G. & Krivine, J.L. (1976), *Elements of mathematical logic. Model Theory*, number 2 *in* 'Studies in logic and the foundations of mathematics', North Holland, Amsterdam.

Kripke, S. (1972), Naming and necessity, *in* Davidson & Harman (1972), pp. 253–355. Also Basil Blackwell, 1980.

Kripke, S. (1976), Is there a problem about substitutional quantification?, *in* G. Evans & J.H. McDowell, eds, 'Truth and meaning. Essays in semantics', Clarendon Press, Oxford, pp. 325–419.

Landsbergen, J. (1981), Adaption of Montague grammar to the requirements of parsing, *in* Groenendijk J.A.G., T.M.V. Janssen & M.B.J. Stokhof, eds, 'Formal methods in the study of language. Proceedings of the third Amsterdam colloquium', number 135 & 136 *in* 'CWI Tracts', Centre for Mathematics and Computer Science, Amsterdam, pp. 399–420. 2 volumes.

Lewis, D. (1970), 'General semantics', *Synthese* **22**, 18–67. Reprinted in Davidson & Harman (1972, 169-248) and in Partee (1976, 1-50).

MacCawley, J.D. (1986), 'Concerning the base component in a transformational grammar', *Foundations of Language* **4**, 55–81.

Marcus, R.B. (1962), 'Interpreting quantification', *Inquiry* **5**, 252–259.

Mates, B. (1950), Synonomy, *in* 'Meaning and Interpretation', number 25 *in* 'Publications in Philosophy', Univ. of California, pp. 201–226. Reprinted in L.Linsky *Semantics and the Philosophy of Language*, Univ. of Illinois Press, Urbana, 1952.

Monk, J.D. (1976), *Mathematical logic*, number 37 *in* 'Graduate texts in mathematics', Springer, Berlin.

Montague, R. (1970*a*), English as a formal language, *in* Visentini et. al., ed., 'Linguaggi nella societa et nella technica', Edizioni di communita, Milan, pp. 188–221. distributed by the Olivetti Corporation, Milan, reprinted in Thomason (1974, pp. 188-221).

Montague, R. (1970*b*), 'Universal grammar', *Theoria* **36**, 373–398. Reprinted in Thomason (1974, pp. 222-246).

Montague, R. (1973), The proper treatment of quantification in ordinary English, *in* K.J.J. Hintikka, J.M.E. Moravcsik & P. Suppes, eds, 'Approaches to natural language, Synthese Library 49', Reidel, Dordrecht, pp. 221–242. Reprinted in Thomason (1974, pp. 247-270).

Morris, F.L (1973), Advice on structuring compilers and proving them correct, *in* 'Proc. ACM, Symp on principles of programming languages, Boston, 1973', ACM, pp. 144–152.

Muskens, R. (1993), A compositional discourse representation theory, *in* p. Dekker & M. Stokhof, eds, 'Proceedings 9th Amsterdsam Colloquium. Part II', Institute for Language, Logic and Computation, University of Amsterdam, pp. 467–486.

Partee, B. (1973), 'Some transformational extensions of Montague grammar', *Journal of Philosophical Logic* **2**, 509–534. Reprinted in (Partee 1976, 51-76).

Partee, B. (1982), Believe-sentences and the limits of semantics, *in* S. Peters & E. Saarinen, eds, 'Processes, Beliefs, and Questions', Synthese Language Library, Reidel, Dodrecht, pp. 87–106.

Partee, B. (1984), Compositionality, *in* F. Landman & F. Veltman, eds, 'Varieties of Formal Semantics', number 3 *in* 'GRASS', Foris, Dordrecht, pp. 281–311.

Partee, B. (1987), Noun phrase interpretation and type-shifting principles, *in* J. Groenendijk, D. de Jongh & M. Stokhof, eds, 'Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers', number 8 *in* 'GRASS', Foris, Dodrecht, pp. 115–143.

Partee, B. (1988), 'Semantic facts and psychological facts', *Mind & Language* **3**, 43–52. Symposium on *Remnants of Meaning* - contrib. 4.

Partee, B. (1989), Binding implicit variables in quantified contexts, *in* C. Wiltshire, B. Music & R. Graczyk, eds, 'Papers from CLS 25', Chicago Linguistic Society, Chicago, pp. 342–365.

Partee, B. & Rooth, M. (1983), Generalized conjunction and type ambiguity, *in* R. Bauerle, C. Schwartz & A. von Stechow, eds, 'Meaning, Use and Interpretation of Language', Walter de Gruyter, Berlin, pp. 361–383.

Partee, B., ed. (1976), *Montague Grammar*, Academic Press, New York.

Partee, B., ter Meulen, A. & Wall, R.E. (1990), *Mathematical Methods in Linguistics*, number 30 *in* 'Studies in Linguistics and Philosophy', Kluwer, Dordrecht.

Pelletier, F.J. (1993*a*), On an argument against semantic compositionality, *in* D. Westerståhl, ed., 'Logic, Methodology and Philosophy of Science', Kluwer, Dordrecht.

Pelletier, F.J. (1993*b*), Some issues involving internal and external semantics, *in* J. Macnamara & G. Reges, eds, 'The logical foundations of cognition', Oxford Univ. Press, Oxford.

Pelletier, F.J. (1994), 'The principle of semantic compositonality', *Topoi* **13**, 11–24.

Popper, K. (1976), *Unended quest. An intellectual autobiography*, Fontana.

Pratt, V.R. (1979), Dynamic logic, *in* de Bakker & van Leeuwen, eds, 'Foundations of computer science III, part 2, Languages, logic, semantics', number 100 *in* 'CWI Tracts', Centre for Mathematics and Computer science, Amsterdam, pp. 53–82.

Quine, W.V.O. (1960), *Word and Object*, The MIT Press, Cambridge (Mass.).

de Roever, W.P. (1985), The quest for compositionality - a survey of proof systems for concurrency, part 1, *in* E.J. Neuhold, ed., 'Proc. IFIP working group 'The role of abstract models in Computer Science', North-Holland, Amsterdam.

Rosetta, M.T. (1994), *Compositional Translation*, The Kluwer International Series in Engineering and Computer Science 230, Kluwer, Dordrecht. (M.T. Rosetta = L. Appelo, T. Janssen, F. de Jong, J. Landsbergen (eds.)).

Rus, T. (1991), 'Algebraic construction of compilers', *Theoretical Computer Science* pp. 271–308.

Saarinen, E. (1979), Backwards-looking operators in tense logic and in natural language, *in* J. Hintikka, I. Niiniluotov & E. Saarinen, eds, 'Essays on mathematical and philosophical logic. Proceedings of the 4th. Scandinavian logic symposium', number 122 *in* 'Synthese Library', Kluwer, Dordrecht, pp. 341–367.

Salmon, N. & Soames, S. (1988), *Propositions and Attitudes*, Oxford Readings in Philosophy, Oxford Univ. Press.

Schiffer, S. (1987), *Remnants of Meaning*, MIT Press, Cambrige, Mass.

Schiffer, S. (1988), 'Reply to comments', *Mind & Language* **3**, 53–63. Symposium on *Remnants of Meaning*. contrib. - 5.

Schütte, K. (1977), *Proof theory*, number 225 *in* 'Grundlehren der Mathematische Wissenschaften', Springer, Berlin.

Stockwell, R., Schachter, P. & Partee, B.H. (1973), *The major Syntactic Structures of English*, Holt, Rinehart, and Winston, New York.

Stoy, J.E. (1977), *Denotational semantics: the Scott-Strachey approach to programming language theory*, The MIT Press, Cambridge (Mass.).

Thatcher, J.W., Wagner, E.G. & Wright, J.B. (1979), More on advice on structuring compilers and proving them correct, *in* H.A. Maurer, ed., 'Automata, languages and programming. (Proc. 6th. coll. Graz)', number 71 *in* 'Lecture Notes in Computer Science', Springer, Berlin.

Thomason, R.H. (1974), *Formal Philosophy. Selected Papers of Richard Montague*, Yale University Press, New Haven.

Wirsing, M. (1990), Algebraic specification, *in* J. van Leeuwen, ed., ‘Handbook of Theoretical Computer Science’, Vol. B. ’Formal Models and Semantics’, Elsevier, chapter 13, pp. 675–780.

Wittgenstein, L. (1921), Tractatus logico-philosphicus.Logisch-philosphische Abhandlung, *in* Ostwald, ed., ‘Annalen der Naturphilosphie’. Reprint published by Basil Blackwell, Oxford, 1959.

Zadrozny, W. (1994), ‘From compositional to systematic semantics’, *Linguistics and Philosophy* **17**, 329–342.

Zeevat, H. (1989), ‘A compositional approach to Kamp’s DRT’, *Linguistics and Philosophy* **12**(1), 95–131.