Functor–argument decomposition, compositionality and logical semantics

FUNCTOR–ARGUMENT DECOMPOSITION, COMPOSITIONALITY AND LOGICAL SEMANTICS

TAMÁS MIHÁLYDEÁK
Department of Computer Science
Faculty of Informatics
University of Debrecen

Contents

List of Figures

Acknowledgments

I am very grateful to the Fulbright Foundation for the opportunity to work on this book at Indiana University, Bloomington. I thank my colleagues for their valuable comments and remarks on previous versions of different chapters and for all kinds of other help.

Chapter 1

THE FREGE–HUSSERL TRIAD AND FUNCTOR–ARGUMENT DECOMPOSITION

Abstract The purpose of the present chapter is to outline the theoretical limits of the most general system based on functor–argument decomposition. The Frege–Husserl triad (the context principle, the principle of compositionality and the category principle) constitutes the most important theoretical (linguistic and logical–philosophical) presupposition pertaining to general type theoretical languages and their total or partial compositional semantics. We prove some theorems which make the consequences of the Frege–Husserl triad explicit, and after defining the notion of semantic categories in the spirit of Husserl we characterize Tarskian and Husserlian models both in total and partial semantics.

Keywords: The context principle, the principle of compositionality, the category principle

1. Background: The Frege–Husserl triad

More than 125 years ago a booklet written by a hardly thirty–year old logician/ philosopher was published. This proved to be a turning point in the history of logic and philosophy. Something began and brand new horizons opened up for thinkers. This booklet is Gottlob Frege's *Begriffsschrift, a formula language of pure thought modelled on that of arithmetic*[1], which occupies a special place in philosophical (and not only in philosophical) culture. 20th (and 21st)–century logic, philosophy of

[1]See Preface and Part I in Beaney, 1997.

logic and theoretical linguistics could not exist in their present style of the art without Frege's work.

In Frege's view, one of the most important inventions of Begriffsschrift is replacing traditional subject–predicate decomposition by functor–argument decomposition. He wrote the following:

> "The very invention of this Begriffsschrift, it seems to me, has advanced logic. ... [L]ogic hitherto has always followed ordinary language and grammar too closely. In particular, I believe that the replacement of the concept subject and predicate by argument and function will prove itself in the long run. It is easy to see how taking a content as a function of an argument gives rise to concept formation. ... The distinction between subject and predicate finds no place in my representation of a judgement."[2] Frege, 1997, pp. 51, 53

Are we aware of the significance of this step? How can we survey and evaluate its consequences? It had generally been accepted before Frege that grammatical structures constituted logical structures, moreover, all logical structures had to prove to be grammatical ones.

Having replaced (traditional) grammatical structures by structures relying on functor–argument decomposition, the essential question from the logical point of view is the following: Is there a theoretical limit of functor–argument decomposition. This limit can be found in the semantic mirror of functor–argument decomposition, i.e. in the semantic rules set by Frege and Husserl. Frege and Husserl formulated three principles which any model–theoretic approach to sense (meaning) has to be obey. The so called Frege–Husserl triad[3] consists of the following three principles:

(a) the context principle,

(b) the principle of compositionality,

(c) the category principle.

[2] I use the expression 'functor' instead of 'function' in order to differentiate an incomplete expression of a language from its semantic value.

[3] On the philosophical role of the Frege–Husserl triad see, for example, Werning, 2004.

The first two principles go back to Frege, the third one is due to Husserl. Generally speaking, the first one derives the sense (meaning) of an expression from the senses (meanings) of the expressions whose part it is, the second one proposes a connection between the sense (meaning) of a complex expression and the senses (meanings) of its ‘parts’, and the third one puts forward the requirement of resemblance between the systems of syntactic and semantic categories. It has to be noted here that there is no theoretical hierarchy concerning the principles, they exist side by side, each of them presupposes the others.

2. The first component of the triad: The context principle

From the semantic point of view, Frege’s context principle[4] or as W. Hodges puts it[5] Frege’s Dictum can be considered as a general leading idea. In *The Foundation of Arithmetic* Frege wrote the following, which is usually quoted as the context principle:

> “never to ask for the meaning of a word in isolation, but only in the context of a proposition;” Frege, 1980, p. x
>
> “It is enough if the proposition taken as a whole has sense; it is this that confers on its parts also their content.” Frege, 1980, p. 71

According to the context principle an expression has meaning (sense) only in the sentence in which it occurs. If a sentence is meaningful, then it makes its own parts meaningful, it gives senses (meanings) to them. The principle can only work if the existence of sentence senses (meanings) is presupposed. From a theoretical point of view this means that sentence senses (meanings) cannot be derived from ‘other senses (meanings)’, they constitute the core elements of the semantic system, and so they have to be introduced into the theory via definition. At the same time, in an abstract semantic theory sentence senses (meanings)

---

[4]Janssen remarks that many philosophers understand the principle of compositionality as the principle of contextuality. Janssen, 2001, p. 115 We try to differentiate the two principles clearly by the roles they play in the general theory.
[5]See Hodges, 2001a.

do not have to be characterized in advance at all, it is enough to pick out, to determine some entities as possible sentence senses (meanings).

3. The second component of the triad: The principle of compositionality

What is compositionality, what is meant by compositionality? Generally, it links the syntactic way in which an expression is composed ('the realm of expressions') to the semantic way in which its sense (meaning) is determined ('the realm of meanings'). Each of the following four formulations appears as the principle of compositionality. Different names (used in Szabo, 2000) for different versions are appropriate to indicate small differences among them.

- The Principle of Compositionality (in a wide sense): The meaning of a complex expression is determined by the meanings of its constituents and by its structure. Szabo, 2000, p. 475
- The Function Principle: The meaning of a complex expression is a function of the meaning of its constituents and of its structure. Szabo, 2000, p. 484
- The Building Principle: The meaning of a complex expression is built up from the meaning of its constituents. Szabo, 2000, p. 488
- The Substitutivity Principle: If two expressions have the same meaning, then substitution of one for the other in a third expression does not change the meaning of the third expression. Szabo, 2000, p. 490

First of all, it must be noted that the principle of compositionality (and each of its versions) is usually referred to as 'Frege's principle' (or 'Fregean principle'). Philosophers and logicians dealing with Frege have been discussing for many decades whether Frege accepted the principle of compositionality (or at least one of its versions), and if he did, to what extent. I do not want to go into the details here, hence I will mention only two attitudes, that of Hans Rott and Jeffrey Pelletier:

> "The principle is certainly very Fregean in spirit ... However, as Janssen, 1997, p. 421 points out, Frege does not seem to have stated compositionality as a principle in any of his writings." Rott, 2000, p. 625

> "Frege may have believed the principle of semantic compositionality, although there is no straightforward evidence for it and in any case it does not play any central role in any writing of his, not even in the 'argument form creativity/understandability' citations." Pelletier, 2001, p. 111

Secondly, it must be pointed out that there is a very extensive scientific discussion about the compositionality of natural language.[6] The place and role of the principle of compositionality in linguistics is quite controversial. It is regarded as a methodological principle, or a basic linguistic–philosophical law, or supervenience[7]. Nevertheless, structures which are relevant from the logical point of view (i.e. logical structures) must have been created in a compositional way.[8]

4. The third component of the triad: The category principle

In the second volume of *Logical investigation* Husserl theorized about the realm of meaning and set forth some very important rules connecting the spheres of meanings and forms.[9] The category principle plays a crucial role in logical semantics and in linguistics, for example, it considered to be a corner–stone of categorial grammar[10] and turns up in many contemporary linguistic theories.

Husserl's starting point was the following:

---

[6]See for example the thematic issue of *Journal of Logic, Language and Information* 10 (2001) on compositionality.

[7]See, for example, Partee, 1984 and Szabo, 2000.

[8]In Mihálydeák, 2006 I deal with the logical role of compositionality especially in two–component logical semantics.

[9]See Investigation IV, *The distinction between independent and non–independent meanings and the idea of pure grammar*, especially paragraph 10 Husserl, 1970, pp. 510–513. Some consequences of Husserl's view (the most important one for us is the category principle) can be applied with or without accepting his philosophy as a whole.

[10]See, for example, Bar-Hillel, 1950.

> "... one of the most fundamental facts in the realm of meaning: *that meanings are subject to a priori laws regulating their combination into new meanings*." Husserl, 1970, pp. 510

To show the generality of a priori laws he introduced a very important concept, the concept of semantic categories:

> "Meanings only fit together in antecedently definite ways, composing other significantly unified meanings, while other possibilities of combination are excluded by laws, and yield only a heap of meanings, never a single meaning. ... The impossibility attaches, to be more precise, not to what is singular in the meanings to be combined but to the essential *kinds*, the *semantic categories*, that they fall under." Husserl, 1970, pp. 510

With the help of the original notion of semantic categories (as essential kinds of meanings) one can grasp a very general feature of meanings, namely, the capacity of determining the possible ways of combining different meanings. Hence the rules of the game of how different meanings are to be combined can be set. Therefore, in its original sense, a semantic category is a set of meanings, not a set of expressions. We can generate a special classification in the realm of expressions corresponding to the system of semantic categories, and so we can speak about the semantic categories of expressions as well. Generally speaking, two expressions belong to the same semantic category if they form meaningful expressions when we combine them with the same expressions. In what follows I will use 'semantic category' in the latter sense, i.e. I will speak about the semantic category of expressions.

In the light of these considerations, *the category principle* can be formulated as follows:

Synonymous expressions belong to the same semantic category.

Having introduced the notion of semantic categories and proposed the category principle, Husserl puts forward a very important common research enterprise for logic and linguistics:

> "Hence arises the great task equally fundamental for logic and grammar, of setting forth the *a priori* constitution of the realm of meaning, of investigating the *a priori* system of the formal structures which leave open all material specificity of meaning in a 'form–theory of meanings'." Husserl, 1970, p. 513

Chapter 2

GENERAL FORMAL SYSTEMS

Abstract In this chapter functor–argument decomposition and compositional semantics, which relies on it, are considered, and some theorems are put forward which make explicit the theoretical (linguistic and logical philosophical) presuppositions of general type–theoretical languages and their total or partial semantics explicit.

Keywords: Compositionality, type theory, semantic category, partial semantics

One of the most general theoretical representations of functor–argument decomposition is the well–known type theory (or the different systems of type–theoretical language and/or logic). Type theory has a number of linguistic–philosophical, logical–philosophical and logical advantages[1], some of these are given below.

(a) Functor–argument decomposition constitutes its theoretical base, hence, it is suitable for representing the most general compositional formal system.

(b) The use of types provides conceptual clarity.

(c) The framework of types provides a rich, highly–structured system of semantic values that is extremely useful in formalization.

[1]See, for example, Thomason, 1999 and Thomason, 2001.

(d) The underlying logical architecture[2] is simple, and in special cases it can be considered as a natural generalization of first–order logic.

Generally, syntactic categories have to be distinguished from semantic ones. At the same time, our formal systems satisfy the following fundamental principle of formal type–theoretical semantics:

> *The mirror principle*: "Associated with every syntactic category C is a counterpart semantic category C^*, whose *mathematical type* 'mirrors' the *grammatical type* of C. And, in particular, every expression of syntactic category C is interpreted by an object of semantic domain C^*."Dunn and Hardegree, 2001, p. 142

On the basis of the mirror principle we will speak about types in what follows, and we will use types to define and denote different syntactic categories and the corresponding sets of possible semantic values.

1. General type–theoretical languages

First of all, the system of types has to be defined. The definition is usually an inductive one.[3] In its background, there are some linguistic (and logical) philosophical commitments. The system of types relies on primitive type(s). Therefore, we need to specify the set of primitive types.

Generally there is only one requirement: the symbol o must be a primitive type. From the philosophical point of view the main reason for this is that the symbol o is taken as the type of the most fundamental expressions of our formal language. Expressions of type o are called formulae. Formulae directly correspond to a special sort of conceptual content or information. More specifically formulae are structures of complete information or closed (and whole) conceptual content. In a given interpretation, formulae are intended to carry complete information which is usually called proposition in the literature. Considering natural language interpreted formulae can be linked to different classes

[2]It goes back to A. Church (see Church, 1940).
[3]For the nature of inductive definition see, for example, Ruzsa, 1997.

of declarative sentences of natural language that have the same sense, or express the same meaning in a given non–formal interpretation.

There is another, mainly semantic reason for declaring type o to be primitive. According to the context principle, an expression has sense (meaning) only in the sentence in which it occurs. Sometimes we need more than one primitive types (usually individual names constitute another primitive type). The main difference between primitive and non-primitive types is that the semantic domains of primitive types have to be given via definition, while the domains of non-primitive types are derived from those. Non-primitive types are usually called functor types.

Definition 2.1. Let PT be an arbitrary set of symbols, the set of primitive types, such that $o \in PT$. The set $TYPE_{PT}$ is defined inductively as follows:

1 $PT \subseteq TYPE_{PT}$;

2 $\alpha, \beta \in TYPE_{PT} \Rightarrow \langle \alpha, \beta \rangle \in TYPE_{PT}$.

Remark 2.2. From the syntactic point of view o is the type of formulae, and from the semantic point of view the type of their possible semantic. $\langle \alpha, \beta \rangle$ is the type of functors which, when they are filled in with an argument of type α, yield an expression of type β in syntax (in the formal language), and it stands for the type of functions from objects of type α to objects of type β in semantics.

Type–theoretical languages are the most general with respect to functor–argument decomposition. In type–theoretical languages there are only two syntactic operations: filling in a functor with an argument and lambda abstraction. The latter provides a way of creating a functor from an expression.

Definition 2.3. A *type–theoretical language* is an ordered quadruple

$$L = \langle LC, Var, Con, Cat \rangle$$

satisfying the following conditions:

1 LC is the set of theoretical constants.[4] $LC = \{\lambda, (,)\}$

2 $Var = \cup_{\alpha \in TYPE_{PT}} Var(\alpha)$, where $Var(\alpha)$ is a denumerably infinite sets of symbols[5].

3 $Con = \cup_{\alpha \in TYPE_{PT}} Con(\alpha)$, where $Con(\alpha)$ is a denumerably set of symbols.[6]

4 All mentioned sets of symbols are assumed to be pairwise disjoint sets.

5 $Cat = \cup_{\alpha \in TYPE_{PT}} Cat(\alpha)$, where the sets $Cat(\alpha)$ are defined by the inductive rules (a)... (c) as follows[7]:

(a) $Var(\alpha) \cup Con(\alpha) \subseteq Cat(\alpha)$;

(b) $C \in Cat(\langle\alpha, \beta\rangle)$, $B \in Cat(\alpha) \Rightarrow$ '$C(B)$' $\in Cat(\beta)$;

(c) $A \in Cat(\beta)$, $\tau \in Var(\alpha) \Rightarrow$ '$(\lambda\tau A)$' $\in Cat(\langle\alpha, \beta\rangle)$;

Definition 2.4.

(a) Let L be a type-theoretical language and $A \in Cat$. Then a *subterm* of A is defined inductively as follows:

i if $A \in Cat$, then A is a subterm of A;

ii if $A =$ '$C(D)$', then C and D are subterms of A;

iii if $A =$ '$(\lambda\tau C)$', then C is a subterm of A;

iv if B is a subterm of A and C is a subterm of B, then C is a subterm of A.

---

[4]In this definition "λ" is the usual lambda operator that goes back to A. Church (Church, 1940) and represents lambda abstraction. A theoretical constant has the same semantic value (or sense) in every interpretation as a logical constant in a logical system.

[5]$Var(\alpha)$ is the set of variables of type α.

[6]Con is the set of non–theoretical symbols of L. The semantic value of an expression belonging to the set Con is given by an interpretation. (In a logical system Con is the set of nonlogical constants.)

[7]Cat is the set of all well–formed expressions of L. A given set $Cat(\alpha)$ is the α–category of L ($\alpha \in TYPE_{PT}$).

(b) A variable τ is a *free variable* of $A \in Cat$ if there is an occurrence of τ in A which is not in a subterm $(\lambda\tau C)$ of A.

(c) $V(A) = \{\tau : \tau \in Var$ and τ is a free variable of $A\}$. If $V(A) = \emptyset$, then the expression A is *closed*. A is *open* if it is not closed.

(d) B $(\in Cat(\gamma))$ is *substitutable for the variable* τ $(\in Var(\gamma))$ *in* A if no free variables of B become bound by the substitution. Let A^B_τ denote the term obtained from A by replacing all free occurrences of τ by B.

(e) If $A \in Cat$ and $B, C \in Cat(\gamma)$, then $A[C \downarrow B]$ $(\in Cat)$ is obtained by replacing a subterm occurrence (i.e. an occurrence which is not preceded immediately by λ) of B by C.

2. General type–theoretical semantics

The functor–argument frame is the compositional mirror of type–theoretical languages. (From the mathematical point of view a type–theoretical language is homomorph to the functor–argument frame.) It can be said that the functor–argument frame gives *possible* semantic values.

Definition 2.5. A *total functor–argument frame* F is the system of sets

$$\langle Dom_F(\gamma)\rangle_{\gamma \in TYPE_{PT}}$$

such that

(a) If $\gamma \in PT$, then $Dom_F(\gamma)$ is an arbitrary nonempty set.

(b) $Dom_F(\langle\alpha,\beta\rangle) = Dom_F(\beta)^{Dom_F(\alpha)}$ for all $\langle\alpha,\beta\rangle \in TYPE_{PT}$.[8]

Definition 2.6. A *partial functor–argument frame* PF is the system of sets

$$\langle Dom_{PF}(\gamma)\rangle_{\gamma \in TYPE_{PT}}$$

such that

[8]If D_1, D_2 are sets, then $D_2^{D_1}$ is a function set, i.e. $D_2^{D_1} =_{def} \{f : D_1 \mapsto D_2\}$

(a) if $\gamma \in PT$, then $Dom_{PF}(\gamma)$ is an arbitrary set with a distinguished member Θ_γ, which is called the null entity of type γ, such that $Dom_{PF}(\gamma) \setminus \{\Theta_\gamma\} \neq \emptyset$;

(b) $Dom_{PF}(\langle\alpha,\beta\rangle) = Dom_{PF}(\beta)^{Dom_{PF}(\alpha)}$ for all $\langle\alpha,\beta\rangle \in TYPE_{PT}$ and $\Theta_{\langle\alpha,\beta\rangle} = g$ where $g \in Dom_{PF}(\langle\alpha,\beta\rangle)$ and $g(u) = \Theta_\beta$ for all $u \in Dom_{PF}(\alpha)$.

Interpretive functions and assignments associate the constants and variables of type–theoretical languages with their semantic values. In a model, which consists of a frame, an interpretive function and an assignment, semantic rules can be defined which determine the semantic values of compound expressions with respect to the given model.

Definition 2.7. A *(total or partial) model* M on G is an ordered triple

$$\langle G, \varrho, v\rangle$$

where

(a) G is a (total or partial) functor–argument frame;

(b) ϱ, v are functions with domains Con and Var, respectively[9] such that

i if $a \in Con(\alpha)$, then $\varrho(a) \in Dom_G(\alpha)$;

ii if $\tau \in Var(\alpha)$, then $v(\tau) \in Dom_G(\alpha)$.

Remark 2.8.

(a) A model M on G is total or partial if G is a total or partial functor–argument frame, respectively.

(b) If $M = \langle F, \varrho, v\rangle$ is a total model on F, then

$$Dom_M(\alpha) = Dom_F(\alpha).$$

[9] ϱ is an interpretive function, v is an assignment.

(c) If $PM = \langle PF, \varrho, v\rangle$ is a partial model on PF, then

$$Dom_{PM}(\alpha) = Dom_{PF}(\alpha) \setminus \{\Theta_\alpha\}.$$

(d) If M $(= \langle G, \varrho, v\rangle)$ is a total or partial model, $\xi \in Var(\gamma)$ and $u \in Dom_G(\gamma)$, then the model M^u_ξ $(= \langle G, \varrho, v[\xi : u]\rangle)$ is like M except that $v[\xi : u](\xi) = u$.

Definition 2.9. A total or partial model M $(= \langle G, \varrho, v\rangle)$ assigns each expression A of type α a *semantic value* $[\![A]\!]_M$ on the basis of the semantic rules:

(a) if $a \in Con(\gamma)$, then $[\![a]\!]_M = \varrho(a)$;

(b) if $\xi \in Var(\gamma)$, then $[\![\xi]\!]_M = v(\xi)$;

(c) if $A \in Cat(\langle\alpha, \beta\rangle)$ and $B \in Cat(\alpha)$, then

$$[\![A(B)]\!]_M = [\![A]\!]_M([\![B]\!]_M);$$

(d) if A is an expression of type β and $\xi \in Var(\alpha)$, then $[\![\lambda\xi A]\!]_M = g$, where g is a function from $Dom_G(\alpha)$ to $Dom_G(\beta)$ such that $g(u) = [\![A]\!]_{M^u_\tau}$ for all $u \in Dom_G(\alpha)$.

Proposition 2.10. If M is a total model and $A \in Cat(\alpha)$, then $[\![A]\!]_M \in Dom_M(\alpha)$. If M is a partial model, then $[\![A]\!]_M \in Dom_M(\alpha) \cup \{\Theta_\alpha\}$.

Definition 2.11. If M is a total or partial model, then A is *meaningful* with respect to M, in symbols $A \in Cat^M_{mf}$, if $A \in Cat(\alpha)$ for some type α and $[\![A]\!]_M \in Dom_M(\alpha)$.

Remark 2.12. If M is a total model, then all $A \in Cat$ are meaningful, i.e. there is no difference between the notion of well–formedness and that of meaningfulness. Only in the case of partial models is there a real difference between the two notions.

Theorem 2.13. *If $A \in Cat$, $M_1 = \langle G, \varrho, v_1\rangle$ and $M_2 = \langle G, \varrho, v_2\rangle$ are two (total or partial) models of L with the same frame G and interpretive function ϱ, and $v_1(\tau) = v_2(\tau)$ for all $\tau \in V(A)$, then $[\![A]\!]_{M_1} = [\![A]\!]_{M_2}$.*

Proof 2.14. The proof can be obtained by structural induction based on Definition 2.3. It is trivial in the case of 5a and 5b. Turning to 5c, since $v_1[\tau : u](\xi) = v_2[\tau : u](\xi)$ for all $\xi \in V(\lambda\tau A) \cup \{\tau\}$ $(\supseteq V(A))$, we get that $[\![(\lambda\tau A)]\!]_{M_1}(u) = [\![A]\!]_{{M_1}^u_\tau} = [\![A]\!]_{{M_2}^u_\tau} = [\![(\lambda\tau A)]\!]_{M_2}(u)$ for all u.

Proposition 2.15. If $A \in Cat$ is a closed expression, then $[\![A]\!]_M$ is independent from v i.e $[\![A]\!]_M = [\![A]\!]_{M^u_\tau}$ for all $\tau \in Var(\gamma)$ and $u \in Dom_F(\gamma)$.[10]

In order to prove the law of lambda–conversion 2.21 first we have to consider the law of replacement 2.17 and Lemma 2.20. The first one says that in semantics semantic values are only taken into consideration and no attention is payed to the expression itself — apart from its type — whose semantic value is given. It does not matter how a semantic value is determined, or which form of the compound expression gets the semantic value. This property is expressed in the law of replacement by means of universal replacement of expressions belonging to the same type with the same semantic value. From the logical–philosophical point of view, the law of replacement is a special type–theoretical formulation of a version of the principle of compositionality which is called the substitutivity principle:

> The Substitutivity Principle: "If two expressions have the same meaning, then substitution of one for the other in a third expression does not change the meaning of the third expression." Szabo, 2000, p. 490

It must be emphasized here that the law of replacement can only be considered as a restricted version of the substitutivity principle, the unrestricted form of the substitutivity principle holds only in Husserlian models, which have been discussed in Section 2. The next definition introduces the notion of 1–compositionality. 1–compositional systems fulfil a restricted version of the substitutivity principle, and Corollary 2.19 of

[10]In the case of closed expressions we can handle models as ordered pairs of frames and interpretive functions.

law of replacement 2.17 says that our general system is compositional in the sense of 1–compositionality.

Definition 2.16. Let M be a model of L. We say that M is *1–compositional* if for all well–formed expressions A, B, C ($A, B, C \in Cat$) and variable τ ($\tau \in Var$) such that $(\lambda\tau C)(A), (\lambda\tau C)(B) \in Cat^M_{mf}$ the following holds:

$$[\![A]\!]_M = [\![B]\!]_M \Rightarrow [\![(\lambda\tau C)(A)]\!]_M = [\![(\lambda\tau C)(B)]\!]_M$$

Theorem 2.17 (Law of replacement)**.** *If $A \in Cat$, $B, C \in Cat(\gamma)$, and M is a (total or partial) model of L, then*

$$[\![B]\!]_M = [\![C]\!]_M \Rightarrow [\![A]\!]_M = [\![A[C\downarrow B]]\!]_M.$$

Proof 2.18. The proof can be obtained by structural induction based on Definition 2.3. It is trivial in the case of 5a.

Considering 5b: If $A =$ '$A_1(A_2)$', then

$$A[C\downarrow B] = \text{`}A_1[C\downarrow B](A_2)\text{'} \text{ or } A[C\downarrow B] = \text{`}A_1(A_2[C\downarrow B])\text{'}.$$

Since $[\![A_1]\!]_M = [\![A_1[C\downarrow B]]\!]_M$ and $[\![A_2]\!]_M = [\![A_2[C\downarrow B]]\!]_M$, we can get in the first case that

$$\begin{aligned}[\![A]\!]_M = [\![A_1]\!]_M([\![A_2]\!]_M) = [\![A_1[C\downarrow B]]\!]_M([\![A_2]\!]_M) = \\ = [\![A_1[C\downarrow B](A_2)]\!]_M = [\![A[C\downarrow B]]\!]_M\end{aligned}$$

and in the second case that

$$\begin{aligned}[\![A]\!]_M = [\![A_1]\!]_M([\![A_2]\!]_M) = [\![A_1]\!]_M([\![A_2[C\downarrow B]]\!]_M) = \\ = [\![A_1(A_2[C\downarrow B])]\!]_M = [\![A[C\downarrow B]]\!]_M.\end{aligned}$$

Turning to 5c: If $A =$ '$(\lambda\tau A')$', $\tau \in Var(\gamma)$, then $A[C\downarrow B] = \lambda\tau A'[C\downarrow B]$.

Due to structural induction we have $[\![A']\!]_{M^u_\tau} = [\![A'[C\downarrow B]]\!]_{M^u_\tau}$ for all $u \in Dom(\gamma)$, so

$$\begin{aligned}[\![A]\!]_M(u) = [\![\lambda\tau A']\!](u) = [\![A']\!]_{M^u_\tau} = [\![A'[C\downarrow B]]\!]_{M^u_\tau} = \\ = [\![\lambda\tau A'[C\downarrow B]]\!]_M(u) = [\![A[C\downarrow B]]\!]_M(u)\end{aligned}$$

for all $u \in Dom(\gamma)$.

Corollary 2.19. If M is a (total or partial) model of L, then M is 1–compositional.

Lemma 2.20. If B is substitutable for variable τ in A, M is a (total or partial) model, and $[\![B]\!]_M = u$, then $[\![A^B_\tau]\!]_M = [\![A]\!]_{M^u_\tau}$.

Theorem 2.21 (Lambda–conversion law)**.** *If $A \in Cat$, $\tau \in Var(\beta)$, $B \in Cat(\beta)$ and B is substitutable for τ in A, then $[\![(\lambda\tau A)(B)]\!]_M = [\![A^B_\tau]\!]_M$ for all (total or partial) models M.*

Proof 2.22. $[\![(\lambda\tau A)(B)]\!]_M = [\![(\lambda\tau A)]\!]_M([\![B]\!]_M) = [\![(\lambda\tau A)]\!]_M(u) = = [\![A]\!]_{M^u_\tau} = [\![A^B_\tau]\!]_M$ if $[\![B]\!]_M = u$.

3. Properties of total and partial models

Let us turn now to the characterization of different, total or partial models. The most important question is the following: What is the relation between semantics relying on different (partial or total) frames? In order to compare and combine different models we have to introduce some notion first. In the following definitions let L ($= \langle LC, Var, Con, Cat\rangle$) be a type–theoretical language and M ($= \langle G, \varrho, v\rangle$) its total or partial model.

Definition 2.23.

(a) If $\approx$ is an equivalence relation on the set Cat' ($\subseteq Cat$), then $\approx$ is a *synonymy* for L. The set Cat' is the field of synonymy $\approx$.

(b) *Syntactic synonymy* for L is the synonymy $\cong_L$ generated by the syntax of L, i.e. $A \cong_L B$ if and only if there is a type γ such that $A, B \in Cat(\gamma)$.

(c) *Synonymy* generated by a model M is a synonymy $\approx_M$ for L with the field Cat^M_{mf} such that $A \approx_M B \Leftrightarrow [\![A]\!]_M = [\![B]\!]_M$.

(d) *Closed synonymy* (or *c–synonymy*) generated by a model M is a synonymy $\approx_{Mc}$ for L with the field $\{A : A \in Cat, A \text{ is closed}\} \cap Cat^M_{mf}$ such that $A \approx_{Mc} B \Leftrightarrow [\![A]\!]_M = [\![B]\!]_M$.

(e) A synonymy $\approx$ for L is *semantic* if there is a model M of L such that $\approx_M$ equals $\approx$.

The next proposition shows that in a general type–theoretical compositional framework syntactic synonymy can be treated as a degenerate semantic synonymy.

Proposition 2.24. The syntactic synonymy for L is semantic (in a degenerate sense).

Proof 2.25. Let M be a total model of L such that every semantic domain of M has only one member and these domains are pairwise disjoint sets. In this case M is degenerate in the sense that it makes no semantic difference between expressions belonging to the same type. $A \cong_L B \Leftrightarrow$ there is a $\gamma \in TYPE_{PT}$ such that $A, B \in Cat(\gamma) \Leftrightarrow [\![A]\!]_M, [\![B]\!]_M \in Dom_M(\gamma) \Leftrightarrow [\![A]\!]_M = [\![B]\!]_M \Leftrightarrow A \approx_M B$.

Remark 2.26. In what follows a model of L which generates the synonymy $\cong_L$ will be denoted by M_L and will be called 'syntactic' model.

Definition 2.27.

(a) Two models M_1, M_2 of a language L are said to be *equivalent* (closed equivalent, *c–equivalent*) if $\approx_{M_1}$ equals $\approx_{M_2}$ ($\approx_{M_1c}$ equals $\approx_{M_2c}$), i.e. their generated synonymies (c–synonymies) are equivalent.

(b) Given two synonymies $\approx$ and $\approx'$ for L we say that $\approx$ is *compatible with* $\approx'$ if for all expressions A, B ($\in Cat$) in the field of both synonymies $A \approx B \Leftrightarrow A \approx' B$

(c) Given two synonymies $\approx$ and $\approx'$ for L we say that $\approx$ is *closed compatible with* (or *c–compatible with*) $\approx'$ if for all closed expressions $A, B (\in Cat)$ in the field of both synonymies $A \approx B \Leftrightarrow A \approx' B$

(d) We say that two models M_1, M_2 of L are *compatible* (*closed compatible*) if their generated synonymies $\approx_{M_1}$, $\approx_{M_2}$ are compatible (c–compatible).

Proposition 2.28. If M_1, M_2 are equivalent models of L, then M_1 and M_2 are compatible and c–compatible.

Proposition 2.29. If M_1, M_2 are equivalent models of L, then M_1 and M_2 are c–equivalent.

Proposition 2.30. If M $(= \langle G, \varrho, v\rangle)$ is a model of L, $\tau \in Var(\gamma)$ and $u \in Dom_G$, then the models M and M_τ^u are c–equivalent.

Proof 2.31. The proof is immediate from Proposition 2.15.

Proposition 2.32. If two models of L, M_1, M_2, are compatible, then M_1 and M_2 are c–compatible.

Proposition 2.33. Let two models of L, M_1 and M_2 be total. M_1 and M_2 are compatible if and only if M_1 and M_2 are equivalent.

Proof 2.34. To verify Proposition 2.28 we have to prove only that if the total models M_1 and M_2 are compatible, then they are equivalent. Since M_1 and M_2 are total models, $Cat_{mf}^{M_1} = Cat = Cat_{mf}^{M_2}$ and so for all $A, B \in Cat \quad A \approx_{M_1} B \Leftrightarrow A \approx_{M_2} B$.

In order to investigate the relation between total and partial semantic systems, we need a 'total' or 'pseudo partial' part of a partial frame PF, which will be denoted by PF^t.

Definition 2.35. Let PF be a partial frame. The *total part* PF^t *of the partial frame* PF is the system of sets

$$\langle Dom_{PF}^t(\gamma)\rangle_{\gamma \in TYPE_{PT}}$$

for which the following hold:

(a) if $\gamma \in PT$, then $Dom_{PF}^t(\gamma) = Dom_{PF}(\gamma) \setminus \{\Theta_\gamma\}$;

(b) if $\gamma = \langle\alpha, \beta\rangle$ then $Dom_{PF}^t(\gamma) \subseteq Dom_{PF}(\gamma)$ such that for all $f \in Dom_{PF}^t(\langle\alpha, \beta\rangle)$, $f(u) \in Dom_{PF}^t(\beta)$ if $u \in Dom_{PF}^t(\alpha)$; $f(u) = \Theta_\beta$ otherwise.

Remark 2.36.

(a) For the sake of brevity we will use the notation 'Dom_F^t' in the case of total frame F. Obviously, in this case $Dom_F^t(\gamma) = Dom_F(\gamma)$ for all $\gamma \in TYPE_{PT}$.

(b) If M $(= \langle G, \varrho, v\rangle)$ is a total or partial model, then $Dom^t_M(\gamma) = Dom^t_G(\gamma)$ for all $\gamma \in TYPE_{PT}$.

Definition 2.37. An expression A of type γ is *total* with respect to M if $[\![A]\!]_M \in Dom^t_M(\gamma)$.

Proposition 2.38.

(a) If a non–logical constant A of a primitive type is meaningful with respect to a model M of L, then A is total, i.e. if $A \in Con(\gamma)$ where $\gamma \in PT$, and $A \in Cat^M_{mf}$, then $[\![A]\!]_M \in Dom^t_M(\gamma)$.

(b) If $A \in Cat(\langle\alpha, \beta\rangle)$ and $B \in Cat(\alpha)$ are total with respect to M, then $A(B)$ is total with respect to M.

Definition 2.39.

(a) If $\approx, \approx'$ are synonymies for L, we say that $\approx'$ *extends* $\approx$ (or it is an *extension* of $\approx$) if the field of $\approx'$ includes that of $\approx$ and the two synonymies are compatible.

(b) If M_1, M_2 are models of L, we say that M_2 *extends* M_1 (or that it is an *extension* of M_1) if $[\![A]\!]_{M_2} = [\![A]\!]_{M_1}$ for all $A \in Cat^{M_1}_{mf}$.

(c) If M_1, M_2 are models of L, the notation $M_2 \geq M_1$ will indicate that $\approx_{M_2} \supseteq \approx_{M_1}$.

Remark 2.40. If $M_2 \geq M_1$, then the domain of M_2 includes that of M_1, but within the latter domain M_1 may make more distinctions than M_2.

Proposition 2.41. The models M_1 and M_2 of L are equivalent if and only if both $M_2 \geq M_1$ and $M_1 \geq M_2$ hold.

Proposition 2.42. If M_2 extends M_1, then $M_2 \geq M_1$. (In this case M_2 makes exactly the same distinctions in the field of M_1 as M_1 does.)

Proposition 2.43. A total model is maximal in the sense that all of its extensions are equivalent to it.

Proof 2.44. Let M be a total model of L and M' an extension of M. In that case M, M' are compatible and on the basis Proposition 2.33 it follows that they are also equivalent.

Proposition 2.45. A total model M of L is minimal in the sense that there is no total model M' such that M extends M' and M and M' are not equivalent.

Corollary 2.46. If a total model M extends M' in a way that M and M' are not equivalent, then M' is a partial model of L.

Chapter 3

TARSKIAN AND HUSSERLIAN MODELS IN GENERAL TYPE–THEORETICAL SEMANTICS

Abstract After introducing the notion of semantic categories in the spirit of Husserl we characterize Tarskian and Husserlian models both in total and partial semantics. Characteristic theorems (stating the necessary and sufficient conditions) for these will also be proved.

Keywords: Semantic category, Tarskian model, Husserlian model, partial semantics

1. Tarskian models

In Section 3 of Chapter 2 the properties of models were investigated by means of their generated synonymies. Syntactic synonymy was defined, and it was shown that it is a degenerate semantic synonymy. Both sorts of synonymy (syntactic and semantic) are generated by formal systems of type-theoretical languages and its semantics. In his well–known paper (Tarski, 1983) Tarski introduced a new classification, which plays a crucial role in the formal models of natural language. The classification and therefore the associated synonymy is located — at least in some cases — between syntactic synonymy on the one hand and synonymies generated by non–degenerate models of our language on the other hand.

Definition 3.1. If L is a type–theoretical language, M is a model of L and A, B are well–formed expressions (or grammatical terms, i.e. $A, B \in Cat$), then we say that A, B *belong to the same semantic category with* *respect to* M (they *have the same* M*–category*), in symbols $A \sim_M B$, if for every expression C ($\in Cat$) and a variable τ ($\in Var$)

$$(\lambda\tau C)(A) \in Cat^M_{mf} \Leftrightarrow (\lambda\tau C)(B) \in Cat^M_{mf}.$$

In a very general sense the next proposition has been mentioned by Tarski. In the present framework it is as follows:

Proposition 3.2. If M is a (total or partial) model of L, then $\sim_M$ is a synonymy with the field of Cat.

Proof 3.3. The proof immediately follows from Definition 3.1.

Theorem 3.4. *$A \sim_M B \Rightarrow A \cong_L B$ (and so $\cong_L \supseteq \sim_M$), where M is a (total or partial) model of L.*

Proof 3.5. The proof is indirect. Suppose that $A \sim_M B$ and $A \ncong_L B$. Then there are $\alpha, \beta \in TYPE_{PT}$ such that $\alpha \neq \beta$ and $A \in Cat(\alpha)$, $B \in Cat(\beta)$. If $\tau \in Var(o)$, then $(\lambda\tau\tau) \in Cat^M_{mf}$. If $\xi \in Var(\alpha)$, then $(\lambda\xi(\lambda\tau\tau))(A) \in Cat(\langle o, o\rangle)$ and by means of Lambda–conversion law 2.21 $[\![(\lambda\xi(\lambda\tau\tau))(A)]\!]_M = [\![(\lambda\tau\tau)^A_\xi]\!]_M = [\![(\lambda\tau\tau)]\!]_M$ and so $(\lambda\xi(\lambda\tau\tau))(A) \in Cat^M_{mf}$. At the same time $(\lambda\xi(\lambda\tau\tau))(B) \notin Cat$ and so $(\lambda\xi(\lambda\tau\tau))(B) \notin Cat^M_{mf}$. Therefore $A \nsim_M B$.

Corollary 3.6. If A, B are well–formed but not meaningful expressions with respect to a partial model M, i.e. $A, B \in Cat \setminus Cat^M_{mf}$, then

$$A \sim_M B \Leftrightarrow A \cong_L B$$

With the hel of the notion of semantic category Tarski laid down a very important principle called the first principle of the theory of semantic categories[1], which, as he says, is very natural "from the standpoint of ordinary usage of language" Tarski, 1983, p. 216. In our terminology the informal version of the principle is as follows:

> The first principle of the theory of semantic categories: Two expression of our language have the same semantic category if there is an expression of our

[1]Its original version can be found in Tarski, 1983, p. 216.

language such that it produces meaningful expressions when combined with them.[2]

The following definition specifies the formal version of the first principle of the theory of semantic categories, and introduces the notion of a Tarskian model:

Definition 3.7. We say that a model M of L is *Tarskian* if it is the case that if there is a meaningful expression C and a variable τ such that $(\lambda\tau C)(A)$ and $(\lambda\tau C)(B)$ are both meaningful, then A and B have the same M–category.

Remark 3.8. A model M of L is Tarskian if and only if it fulfills Tarski's first principle of the theory of semantic categories.

Theorem 3.9 (Characteristic theorem of Tarskian models)**.** *The model M of L is Tarskian, if and only if the synonymies $\sim_M$ and $\cong_L$ are equivalent, i.e. $\sim_M$ equals $\cong_L$.*

Proof 3.10. First we prove that if M is Tarskian, then the synonymies $\sim_M$ and $\cong_L$ are equivalent.
Relying on Theorem 3.4, since the field of $\sim_M$ equals the field of $\cong_L$, we have to prove only that $A \cong_L B \Rightarrow A \sim_M B$. If $A \cong_L B$, then there is a type α such that $A, B \in Cat(\alpha)$. If $\xi \in Var(\alpha)$ and $\tau \in Var(o)$, then $(\lambda\xi(\lambda\tau\tau))(A),\ (\lambda\xi(\lambda\tau\tau))(B) \in Cat^M_{mf}$. Since M is Tarskian, $A \sim_M B$.

Secondly we prove that if the synonymies $\sim_M$ and $\cong_L$ are equivalent then M is Tarskian.
Let A, B be arbitrary expressions, and suppose that there is an expression C and a variable τ, such that $(\lambda\tau C)(A) \in Cat^M_{mf}$ and $(\lambda\tau C)(B) \in Cat^M_{mf}$. Since $Cat^M_{mf} \subseteq Cat$, $A \cong_L B$ and so $A \sim_M B$.

Remark 3.11. The Characteristic theorem of Tarskian models 3.9 says that all Tarskian models of L have the same system

[2] A version of the principle is quoted by Hodges Hodges, 2001b, p. 11.

of semantic categories and this system is equivalent to the system of syntactic categories.

Proposition 3.12. If a model M of L is total, then the synonymies $\sim_M$ and $\cong_L$ are equivalent, i.e. $\sim_M$ equals $\cong_L$.

Proof 3.13. Making use of Theorem 3.4, since the field of $\sim_M$ equals the field of $\cong_L$, we have to prove only that $A \cong_L B \Rightarrow A \sim_M B$. If $A \cong_L B$, then there is a type α such that $A, B \in Cat(\alpha)$. Let $C \in Cat$ and $\tau \in Var$. Since M is a total model, $Cat = Cat^M_{mf}$ and so $(\lambda\tau C)(A) \in Cat^M_{mf} \Leftrightarrow \tau \in Var(\alpha) \Leftrightarrow (\lambda\tau C)(B) \in Cat^M_{mf}$. Therefore $A \sim_M B$.

Theorem 3.14. *If M is a total model of L, then M is Tarskian.*

Proof 3.15. The proof directly follows from Proposition 3.12 and Theorem 3.9.

Corollary 3.16. Non–Tarskian models are partial.

Proof 3.17. The proof immediately follows from Theorem 3.14.

2. Husserlian models

In Section 1 we dealt with the relation between syntactic and semantic categories. The next step is to investigate the bridge between the system of semantic categories and the classification generated by the equivalence relation $\approx_M$.

Definition 3.18.

(a) Let M_1, M_2 be models. We say that M_1 and its generated synonymy $\approx_{M_1}$ are M_2–*Husserlian* if $A \approx_{M_1} B \Rightarrow A \sim_{M_2} B$ for all $A, B \in Cat$.

(b) We say that a model M of L is *Husserlian* if it is M–Husserlian. (That is $A \approx_M B \Rightarrow A \sim_M B$ for all $A, B \in Cat$.)

(c) We say that a model M $(= \langle G, \varrho, v\rangle)$ of L is *strictly Husserlian* if M' $(= \langle G, \varrho, v'\rangle)$ is Husserlian for all assignments v'.

(d) We say that the generated synonymy $\approx_M$ of a model M is *Husserlian* (*strictly Husserlian*) if the model M is Husserlian (strictly Husserlian).

The notion of Husserlian models creates a connection between generated synonymy and M–category. It requires that two expressions with the same semantic value with respect to M have to belong to the same M–category, hence, by means of Theorem 3.4 they have to have the same type. More precisely:

Proposition 3.19. If a model M of L is Husserlian and $A \approx_M B$ for some $(A, B \in Cat)$, then $A \cong_L B$, i.e. there is a $\gamma \in TYPE_{PT}$ such that $A, B \in Cat(\gamma)$.

Proof 3.20. The proof immediately follows from Definition 3.18 and Theorem 3.4.

Corollary 3.21. If a model M of L is Husserlian, then $\cong_L \supseteq \approx_M$, i.e. $M_L \geq M$.

Corollary 3.22. Let M_1, M_2 be models of L. If M_1 is M_2–Husserlian, then it is M_L–Husserlian.

Theorem 3.23. *Let M be a Tarskian model of L. The model M is Husserlian if and only if $\cong_L \supseteq \approx_M$, i.e. $M_L \geq M$.*

Proof 3.24. The proof directly follows from Corollary 3.21 and Proposition 3.12.

Corollary 3.25. Let M be a total model of L. M is Husserlian if and only if $\cong_L \supseteq \approx_M$, i.e. $M_L \geq M$.

Law of replacement 2.17 says that an expression can substitute for another one without changing the semantic value of the compound expression if the semantic value of the first expression equals that of the second one. In the law there is a special condition which is usually regarded as not too important. This condition requires that the two expressions in question have to belong to the same syntactic category. Without this supposition, the law of replacement holds only in Husserlian models. That is the reason why I said earlier that Law of replacement 2.17 is only a restricted version of the substitutivity principle [see in Chapter 2], a version of the principle of compositionality. Its unrestricted type–theoretical formulation is the following Husserlian law of replacement.

Theorem 3.26 (Husserlian law of replacement)**.** *If* $A, B, C \in Cat$ *and* M *is a Husserlian model of* L*, then*

$$[\![B]\!]_M = [\![C]\!]_M \Rightarrow [\![A]\!]_M = [\![A[C \downarrow B]]\!]_M$$

Proof 3.27. The proof is immediate from Proposition 3.19 and Law of replacement 2.17.

Theorem 3.28 (Conversion of Husserlian law of replacement)**.** *If for all* $A, B, C \in Cat$

$$[\![B]\!]_M = [\![C]\!]_M \Rightarrow [\![A]\!]_M = [\![A[C \downarrow B]]\!]_M,$$

then M *is a Husserlian model of* L*.*

Proof 3.29. The proof is indirect. Suppose that the model M is not Husserlian. Then there are $B, C \in Cat$ such that $B \approx_M C$ ($[\![B]\!]_M = [\![C]\!]_M$) and $B \not\sim_M C$. Therefore there is some $D \in Cat, \tau \in Var$, such that $(\lambda\tau D)(B) \in Cat^M_{mf}$ and $(\lambda\tau D)(C) \notin Cat^M_{mf}$. In consequence of Law of replacement 2.17, it is impossible that $B,\ C \in Cat(\gamma)$ for some $\gamma \in TYPE_{PT}$, since on the contrary $[\![(\lambda\tau D)(B)]\!]_M = [\![(\lambda\tau D)(C)]\!]_M$ holds. Therefore, there are $\alpha, \beta \in TYPE_{PT}$ such that $\alpha \neq \beta$ and $B \in Cat(\alpha)$, $C \in Cat(\beta)$. Let $A =$ '$(\lambda\xi\xi)(B)$', where $\xi \in Var(\alpha)$. $A \in Cat$ and $A[C \downarrow B] \notin Cat$, hence $[\![A]\!]_M \neq [\![A[C \downarrow B]]\!]_M$.

Definition 3.30. A model M of L *fulfils the substitutivity principle* if for all $A, B, C \in Cat$

$$[\![B]\!]_M = [\![C]\!]_M \Rightarrow [\![A]\!]_M = [\![A[C \downarrow B]]\!]_M.$$

The next theorem shows that the substitutivity principle is a strong version of the principle of compositionality. In our theoretical framework all models are compositional, but a model fulfils the substitutivity principle if and only if it is Husserlian.

Theorem 3.31 (Characteristic theorem of Husserlian models)**.** *A model M of L is Husserlian if and only if it fulfils the substitutivity principle.*

Proof 3.32. The proof derives from Husserlian law of replacement 3.26, and Conversion of Husserlian law of replacement 3.28.

Definition 3.33. A model M of L is *strongly compositional* if it fulfils the substitutivity principle.

Remark 3.34. Characteristic theorem of Husserlian models 3.31 says that the property of being strongly compositional is equivalent to being Husserlian.

Corollary 3.35. If M is a Tarskian model of L and $\cong_L \supseteq \approx_M$, then it fulfills the substitutivity principle.

Theorem 3.36. *A model M of L is strictly Husserlian if and only if the sets $Dom_M(\gamma)$ $(\gamma \in PT)$ are pairwise disjoint sets.*

Proof 3.37. I have to point out that the sets $Dom_M(\gamma)$ $(\gamma \in PT)$ are pairwise disjoint sets if and only if the sets $Dom_M(\gamma)$ $(\gamma \in TYPE_{PT})$ are pairwise disjoint sets.

First we prove that if M $(= \langle G, \varrho, v\rangle)$ is strictly Husserlian, then the sets $Dom_M(\gamma)$ $(\gamma \in TYPE_{PT})$ are pairwise disjoint sets. The proof is indirect. Suppose that M is strictly Husserlian and there is a semantic value u such that $u \in Dom_M(\alpha) \cap Dom_M(\beta)$ where $\alpha \neq \beta$. Let $\tau_1 \in Var(\alpha)$, $\tau_2 \in Var(\beta)$ and v' be an assignment such that $v'(\tau_1) = u = v'(\tau_2)$. If $M' = \langle G, \varrho, v'\rangle$, then $[\![\tau_1]\!]_{M'} = [\![\tau_2]\!]_{M'}$ but $\tau_1 \not\cong_L \tau_2$ and as a result of Proposition 3.19, M' is not Husserlian. So M is not strictly Husserlian.

Second, it is enough to prove that if M is a model of L and the sets $Dom_M(\gamma)$ ($\gamma \in TYPE_{PT}$) are pairwise disjoint sets, then M is Husserlian. The proof is indirect. Suppose that $A \approx_M B$ and $A \not\sim_M B$ where $A, B \in Cat^M_{mf}$. Then $A \not\cong_L B$, and so there are $\alpha, \beta \in TYPE_{PT}$ such that $A \in Cat(\alpha)$, $B \in Cat(\beta)$ and $\alpha \neq \beta$. By means of Proposition 2.10 $[\![A]\!]_M \in Dom_M(\alpha)$, $[\![B]\!]_M \in Dom_M(\beta)$. Since $[\![A]\!]_M = [\![B]\!]_M$, $Dom_M(\alpha) \cap Dom_M(\beta) \neq \emptyset$.

Definition 3.38. A (total or partial) frame G is *strictly Husserlian* if the sets $Dom_G(\gamma)$ ($\gamma \in PT$) are pairwise disjoint sets.

Corollary 3.39. If M is a model on a strictly Husserlian frame then the model M of L is strictly Husserlian.

Corollary 3.40. The degenerate model M_L, which generates the syntactic synonymy $\cong_L$, is strictly Husserlian, hence the synonymy $\cong_L$ is also strictly Husserlian.

Theorem 3.41. *A model M is Husserlian if and only if there is a strictly Husserlian model M' such that $M' \geq M$.*

Proof 3.42. Deriving from Corollary 3.40 the model M_L is strictly Husserlian. If M is Husserlian then by Corollary 3.21 $M_L \geq M$.

Let M' be a strictly Husserlian model such that $M' \geq M$. Then by Corollary 3.21 $M_L \geq M'$ and so $M_L \geq M$. This means that if $A \approx_M B$, then $A \cong_L B$, i.e. there is $\gamma \in TYPE_{PT}$ such that $A, B \in Cat(\gamma)$. Therefore, $(\lambda\tau C)(A) \in Cat$ if and only if $(\lambda\tau C)(B) \in Cat$ for any $C \in Cat$ and $\tau \in Var$. By means of Law of replacement 2.17 $[\![(\lambda\tau C)(A)]\!]_M = [\![(\lambda\tau C)(B)]\!]_M$, hence $(\lambda\tau C)(A) \in Cat^M_{mf} \Leftrightarrow (\lambda\tau C)(B) \in Cat^M_{mf}$, i.e. $A \sim_M B$.

3. Extensions of models

Definition 3.43. Let M and M' be models of L and suppose that M' is an extension of M.

(a) We say that M' is a *cofinal extension* of M if every expression in $Cat_{mf}^{M'}$ is a subterm of an expression in Cat_{mf}^{M}.

(b) We say that M' is an *end–extension* of M if every expression in $Cat_{mf}^{M'}$ that is a subterm of an expression in Cat_{mf}^{M} is already incldedin Cat_{mf}^{M}.

Definition 3.44. Let M and M' be models of L. M is *partially 1–compositional over* M' if for all $A, B, C \in Cat$ and $\tau \in Var$

$A \approx_M B$ and $(\lambda\tau C)(A), (\lambda\tau C)(B) \in Cat_{mf}^{M'} \Rightarrow (\lambda\tau C)(A) \approx_{M'} (\lambda\tau C)(B)$

Proposition 3.45. Every model M of L is partially 1–compositional over M_L.

Definition 3.46. Let M and M' be models of L. M is *fully abstract over* M' if for all $A, B \in Cat$ such that $A \not\approx_M B$ there is an expression C ($\in Cat$) and a variable τ ($\in Var$) such that either exactly one of $(\lambda\tau C)(A)$ and $(\lambda\tau C)(B)$ is meaningful with respect to M' (i.e. in the field of $\approx_{M'}$, in the set $Cat_{mf}^{M'}$), or both are and $(\lambda\tau C)(A) \not\approx_{M'} (\lambda\tau C)(B)$.

Remark 3.47. Definition 3.46 requires that for all non–synonym pairs of well-formed expressions A, B (with respect to M) there is a well–formed functor of the form $(\lambda\tau C)$ that enables us to distinguish A from B in the model M'.

The next proposition shows that the system of types and thus the syntax of type–theoretical languages play a specific role in the formal construction of models, at least in the following sense: the differences built into the types of expressions can be expressed semantically (i.e. they appear in every model).

Proposition 3.48. The syntactic model M_L is fully abstract over every model M.

Proof 3.49. If $A \not\approx_{M_L} B$, where $A, B \in Cat$, then A and B have different types, say α and β, respectively. Let C be the expression '$(\lambda\xi\xi)$', where

$\xi \in Var(o)$. Since $(\lambda\xi\xi) \in Cat^M_{mf}$, we have $(\lambda\tau(\lambda\xi\xi))(A) \in Cat^M_{mf}$, where $\tau \in Var(\alpha)$, but $(\lambda\tau(\lambda\xi\xi))(B) \notin Cat$ and so $(\lambda\tau(\lambda\xi\xi))(B) \notin Cat^M_{mf}$.

Chapter 4

A CRUCIAL STEP FROM GENERAL SEMANTICS TO LOGICAL SEMANTICS: IDENTITY ON OBJECT LEVEL

Abstract In functor–argument language identity does not appear to be a theoretical constant. Nevertheless the use of identity is unavoidable on meta level in order to express the properties of defined semantics. The main question of this chapter is that how the theoretical constant of identity can be introduced into functor–argument language and its semantics.

Keywords: Identity, classical connectives, conservative generalizations of classical conectives

In Chapter 3 the relation between a type–theoretical language and its semantics was investigated. First we defined the most general version of type–theoretical languages, and afterwards we introduced the notion of total and partial model. The mirror principle played a fundamental role in our approach. The most important question was to investigate the way type–theoretical syntax marks the semantic system, i.e. whether the semantic system is independent to some extent or whether it can only only a deformed and dim mirror of syntactic structure. The general investigation showed that the mirror is really dim, it may even deform the shapes.

In general type–theoretical semantics we used the meta–level identity of semantic values order to define semantic synonymy, semantic category, Tarskian and Husserlian semantics. Without the meta–level identity of semantic values we cannot say really anything about the behaviour of the system. Identity sentences (sentences containing identity as the main theoretical constant) express very important facts about the semantic system: if an identity sentence holds (on the meta–level), then the two expressions appearing in the components cannot be distinguished in the semantic system in the sense that if something holds for one of them, it will hold for the other one as well.

Now we have to face the following theoretical question: is there any possibility to embed meta–level identity into general type–theoretical languages? With the help of object–level identity the properties of our general system can be expressed in our object language. Since we are interested in the most general theoretical presuppositions of functor–argument decomposition and semantic compositionality, we have to be very careful and take only short steps. The general construction provides us a separate way to introduce identity (which will express that the semantic values of two expressions coincide).

1. Type–theoretical languages with identity

First we have to define the system of types. We have to give only the set of primitive types: let $PT_=$ be an arbitrary set of symbols such that $o \in PT$ and $o_= \in PT$. By Definition 2.1, we obtain the set $TYPE_=$, the set of types of type–theoretical languages with identity. $o_=(\in PT)$ is the type of identity sentences.

Type–theoretical languages with identity can be defined easily. We have to modify the definition of the set LC, the set of theoretical constants, and introduce a new rule for identity:

Definition 4.1. A *type–theoretical language with identity* is a type–theoretical language ($L_= = \langle LC_=, Var, Con, Cat \rangle$) satisfying the following conditions:

(a) $LC_= = \{\lambda, =, (,)\}$ $\quad$ $(= LC \cup \{=\})$;

(b) $Con(o_=) = \emptyset$

(c) When defining the set Cat we need to add the following rule to the original definition 2.3:
$A, B \in Cat \Rightarrow$ '$(A = B)$' $\in Cat(o_=)$.

Remark 4.2. $Cat(o_=)$ is the set of identity sentences. $Con(o_=) = \emptyset$, ie. primitive identity sentences are superfluous.

2. General type–theoretical semantics for languages with identity sentences

To define total and partial functor–argument frame we have to specify the set of possible values of identity sentences. From a theoretical point of view this is the initial stage where truth values appear. In *Über Sinn und Bedeutung* Frege emphasized that according to *Begriffsschrift* identity sentences belong to our language, identity is in–language, i.e. identity sentences are about our language, our signs and not about the world outside of our language, the objects of the world.[1] In our reconstruction we have not said anything about truth and falsity, true and false do not appear as possible semantic values of sentences. At this stage we cannot avoid true and false as semantic values, but they will only be the possible semantic values of identity sentences. Generally, sentences (members of the set $Cat(o)$) have only undetermined possible semantic values (possible senses).

A total functor–argument frame $F_=$ for type–theoretical languages with identity $L_=$ is a total functor–argument frame such that

$$Dom_{F_=}(o_=) = \{0, 1\}.$$

A partial functor–argument frame $PF_=$ for type–theoretical languages with identity $L_=$ is a partial functor–argument frame such that

$$Dom_{PF_=}(o_=) = \{0, 1, 2\},$$

and $\Theta(o_=) = 2$.

[1]Problems concerning this approach are dealt with in Chapter 5.

Interpretive functions and assignments are as in the general case. In a model, which consists of a frame, an interpretive function and an assignment, only one new semantic rule is required, but total and partial cases need to be distinguished:

- Let M $(= \langle F_{=}, \varrho, v\rangle)$ be a total model.
  - If $A, B \in Cat$, then $[\![(A = B)]\!]_M = \begin{cases} 1, & \text{ha } [\![A]\!]_M = [\![B]\!]_M; \\ 0, & \text{otherwise.} \end{cases}$
- Let M $(= \langle PF_{=}, \varrho, v\rangle)$ be a partial model.
  - If $A \in Cat(\alpha), B \in Cat(\beta)$, then

$$[\![(A = B)]\!]_M = \begin{cases} 1, & \text{if } [\![A]\!]_M \neq \Theta(\alpha),\ [\![B]\!]_M \neq \Theta(\beta) \\ & \text{and } [\![A]\!]_M = [\![B]\!]_M; \\ 2, & \text{if } [\![A]\!]_M = \Theta(\alpha) \text{ or } [\![B]\!]_M = \Theta(\beta); \\ 0, & \text{otherwise.} \end{cases}$$

In the most important case (in the case of identity sentences) the semantic rule of identity is the following:

If $A, B \in Cat(o_{=})$, then

$$[\![(A = B)]\!]_M = \begin{cases} 1, & \text{if } [\![A]\!]_M = [\![B]\!]_M \neq 2; \\ 2, & \text{if } [\![A]\!]_M = 2 \text{ or } [\![B]\!]_M = 2; \\ 0, & \text{otherwise.} \end{cases}$$

3. ‘Classical’ logical connectives for identity sentences

The classical logical connectives for identity sentences can be introduced into type–theoretical languages with identity via definitions. There is no need to give different definitions for total and partial cases, classical logical connectives and their partial versions can be derived in the same way. In partial models the connectives inherit the semantic value gap (see for example Bochvar’s internal connectives), and so they are conservative generalizations of classical connectives.

The symbols ‘↑’ (Verum), ‘↓’ (Falsum), and ‘¬’ (Negation) are to be introduced as follows:

(a) $\uparrow =_{def}$ "$(\lambda pp) = (\lambda pp)$";

(b) $\downarrow =_{def}$ "$(\lambda pp) = (\lambda p \uparrow)$";

(c) $\neg =_{def}$ "$\lambda p(p = \downarrow)$".

where $p \in Var(o_=)$.

Remark 4.3. It is obvious that if M is a total model and PM is a partial model, then

(a) $[\![\uparrow]\!]_M = [\![\uparrow]\!]_{PM} = 1$;

(b) $[\![\downarrow]\!]_M = [\![\downarrow]\!]_{PM} = 0$;

(c) if $A \in Cat(o_=)$, then

$$[\![\neg A]\!]_M = 1 - [\![A]\!]_M;$$
$$[\![\neg A]\!]_{PM} = \begin{cases} 1 - [\![A]\!]_{PM}, & \text{if } [\![A]\!]_{PM} \neq 2; \\ 2, & \text{otherwise.} \end{cases}$$

To get the definition of conjunction of identity sentences we need universal quantification at least over type $o_=(o_=)$, but we cannot use the customary definition (that works over any type)

$$\forall =_{def} \text{“}\lambda P(P = \lambda x \uparrow)\text{”} \;\; (P \in Var(o_=(\alpha)),\; x \in Var(\alpha))$$

since it yields 'too strong' universal quantification in partial case.

In both (total and partial) cases the following definitions can be used:

If $P \in Var(o_=(\alpha))$, $x \in Var(\alpha)$, then

$$\forall =_{def} \text{“}\lambda P(P = \lambda x(P(x) = P(x)))\text{”}$$

We have got the following semantic rules:

- If M is a total model, and $F \in Cat(o_=(\alpha))$, then

$$[\![\forall(F)]\!]_M = \begin{cases} 0, & \text{if there is a } u \in Dom_M(\alpha) \text{ such that } [\![F]\!]_M(u) = 0; \\ 1, & \text{otherwise.} \end{cases}$$

- If PM is a partial model, and $F \in Cat(o_=(\alpha))$, then

$$[\![\forall(F)]\!]_M = \begin{cases} 2, & \text{if } [\![F]\!]_{PM} = \Theta(\alpha) \\ 0, & \text{if there is a } u \in Dom_M(\alpha) \text{ such that } [\![F]\!]_M(u) = 0; \\ 1, & \text{otherwise.} \end{cases}$$

The definition of conjunction can be the following in both cases:

- If $p, q \in Var(o_=)$, $f \in Var(o_=(o_=))$, then

$$\wedge =_{def} \text{“}(\lambda p(\lambda q \forall f(p = ((p = q) = (f(p) = f(q))))\text{”}$$

- If $A, B \in Cat(o_=), f \in Var(o_=(o_=))$ and f has no free occurrences in A and B, we have the following contextual definition:

$$(A \wedge B) =_{def} \wedge(A)(B) = \text{“}\forall f(A = ((A = B) = (f(A) = f(B))))\text{”}$$

The semantic rule of conjunction is as below:

- If M is a total model, and $A, B \in Cat(o_=)$, then

$$[\![(A \wedge B)]\!]_{PM} = \begin{cases} 1, & \text{if } [\![A]\!]_{PM} = [\![B]\!]_{PM} = 1; \\ 0, & \text{otherwise.} \end{cases}$$

- If PM is a partial model, and $A, B \in Cat(o_=)$, then

$$[\![(A \wedge B)]\!]_{PM} = \begin{cases} 2 & \text{if } [\![A]\!]_{PM} = 2 \text{ or } [\![B]\!]_{PM} = 2; \\ 1 & \text{if } [\![A]\!]_{PM} = [\![B]\!]_{PM} = 1; \\ 0 & \text{otherwise.} \end{cases}$$

By means of negation and conjunction other connectives can be defined:

- disjunction: $(A \vee B) =_{def} \neg(\neg A \wedge \neg B)$;
- implication: $(A \supset B) =_{def} \neg(A \wedge \neg B)$.

The corresponding semantic rules are interesting only in the partial case:

- If PM is a partial model, and $A, B \in Cat(o_=)$, then

$$[\![(A \vee B)]\!]_{PM} = \begin{cases} 2, & \text{if } [\![A]\!]_{PM} = 2 \text{ or } [\![B]\!]_{PM} = 2; \\ 0, & \text{if } [\![A]\!]_{PM} = [\![B]\!]_{PM} = 0; \\ 1, & \text{otherwise.} \end{cases}$$

$$[\![(A \supset B)]\!]_{PM} = \begin{cases} 2, & \text{if } [\![A]\!]_{PM} = 2 \text{ or } [\![B]\!]_{PM} = 2; \\ 0, & \text{if } [\![A]\!]_{PM} = 1 \text{ and } [\![B]\!]_{PM} = 0; \\ 1, & \text{otherwise.} \end{cases}$$

Chapter 5

COMPOSITIONALITY FROM THE LOGICAL–PHILOSOPHICAL POINT OF VIEW

Abstract The most general questions to be adressed are the following: 1. what is the role of compositionality in modern logic? 2. how does it work in two–component logical semantics? We analyze the possible appearances of the principle of compositionality in two–component logical semantics. Finally, some of the most fundamental notions of intensional logical semantics are given while maintaining the priority of compositionality concerning sense.

Keywords: Compositionality, logical semantics, two–component semantics, extensionality, intensionality.

In the previous chapters we surveyed several theoretical consequences of functor–argument decomposition. Functor–argument decomposition is the only way in which structures can be constructed. Relying on the principle of compositionality, type–theoretical semantics gives a set-theoretical representation of the possible system of semantic values, but this constitutes only the bare bones of the scheme. In Chapter 4 a small step was taken towards logical systems. Before turning to two–component logical semantics we have to return to the principle of compositionality. We have expressions (identity sentences) with semantic values which look like truth values, but for example we do not know anything precise about the possible values of sentences. Senses (or meanings) are fundamental and primary semantic values, but can we characterize them in a more detailed fashion? The following straightforward questions arise here:

- Can the possible semantic values of identity sentences be considered to be a specific type of senses or do they constitute a new type of semantic values?
- Is there any relation between the possible semantic values of sentences and that of identity sentences?

In Chapter 1 we gave a short overview of the principle of compositionality (primarily taking into account natural languages). Now we focus on the logical role of the principle of compositionality.

1. Compositionality and two–component semantics

Before Frege the grammatical structure of a natural language expression coincided with its logical structure. It was Frege who put an end to this practice. According to Frege the grammatical structure may only appear as one of the possible logical structures, but an expression (or in a more sophisticated way, the expressed conceptual content) may have, and usually does have logical structures that are different from the grammatical structure of the expression.

In figure 5.1 the afore-mentioned situation is represented, where no distinction is made between grammatical structure and logical structure. Here, the principle of compositionality holds and works on the level of natural language. Logical structures originate from natural language directly, based only on the compositionality of natural language. Received logical laws get meaning through associated structures and compositionally joined natural language meanings. Therefore, there is no separate room for the (either informal/pretheoretic or formal) principle of compositionality concerning the logical features of natural language expressions. Natural language compositionality acts not only on the

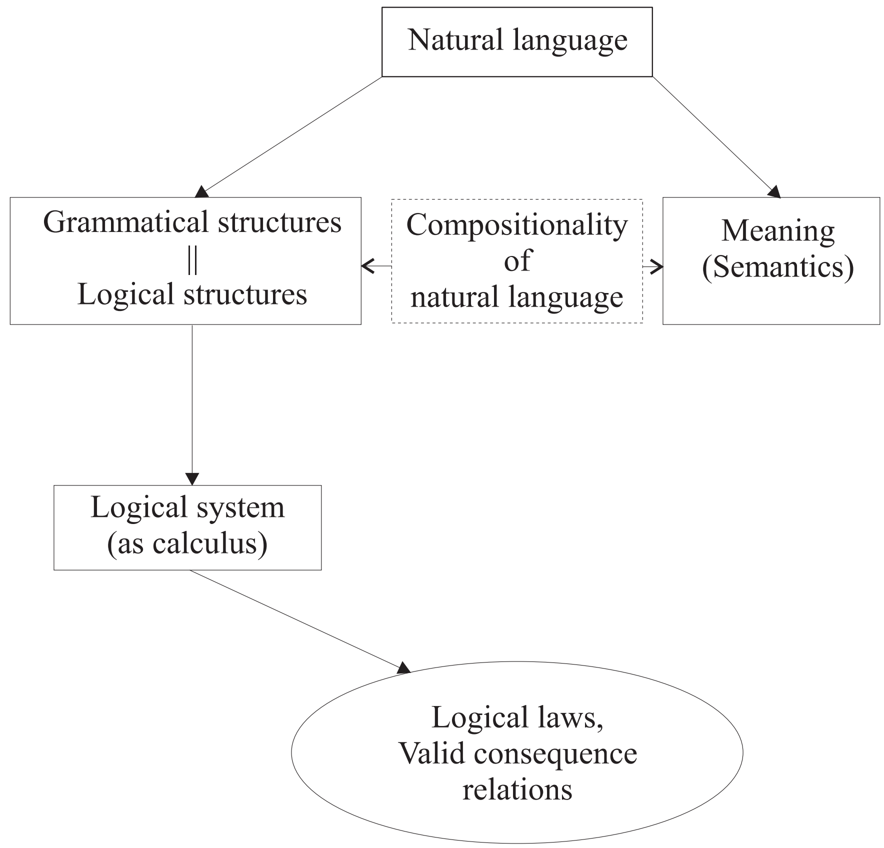

Figure 5.1. Situation before Frege

level of natural language but on the level of logical investigation as well, as logical compositionality. It produces the main patterns of logical structures. The logical system is imprisoned, at least in some sense, in natural language.

The situation after Frege's very famous words in Begriffsschrift changed significantly. In figure 5.2 there a special box appears for informal compositionality (of information/conceptual content). What does that mean exactly? In order to answer this question we have to define the notion semantic value, and specify what kinds of semantic values we have. Obviously, truth values play a crucial role in the system of

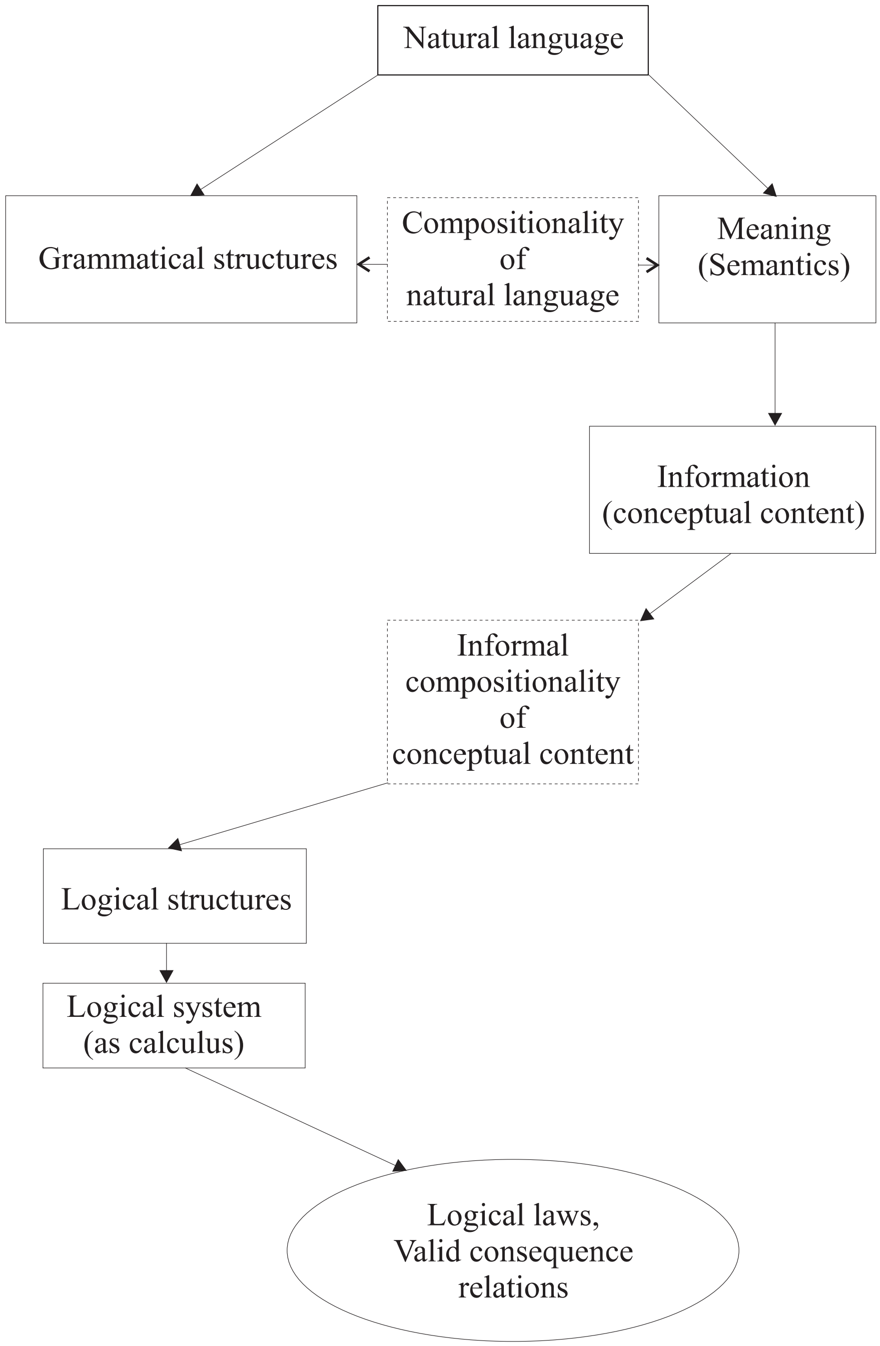

Figure 5.2. The autonomy of grammatical and logical structures

logically relevant semantic values. It can be said — according to Frege's Begriffsschrift —, that the system of semantic values, which is in the informal background, is dominated by truth values in the sense that we derive the semantic values of functors from the set of truth values and the set of objects.[1]

In his semantic writings[2] Frege recognized that a more flexible system of semantic values was needed to explain, for example, the origin of information content of identity statements. In his most famous semantic paper, *Über Sinn und Bedeutung* (Frege, 1952c) he introduced an extensive version of two–component semantics, he differentiated sense and reference (or Sinn and Bedeutung).[3]

Taking the principle of compositionality seriously two questions may arise:

1 How should we modify the principle of compositionality (of the informal level)?

2 How does functor–argument decomposition which yields the main logical structures cooperate with the system of semantic values of two–component semantics?

If we try to answer the first question, we have to take into consideration the fact that many philosophers duplicate the principle and attribute both of these principles (which concern the reference and the sense of compound or complex expressions, respectively) to Frege. "Crucial to Frege's theory are a pair of principles concerning the referent and

[1] As it is well–known, Frege considers sentences as a special type of names, and he puts the possible semantic values of sentences, i.e. truth values, into the set of possible semantic values of names, i.e. the set of objects. This unification proved problematic later, and therefore the development of logical semantics has not followed Frege in that aspect.

[2] For example: Frege, 1952a; Frege, 1952c; Frege, 1952b

[3] There is no standard terminology for different semantic values. In the literature many pairs appear: sense–reference, meaning–reference, sense–meaning, sense–nominatum, sense–denotatum, meaning–denotatum, intension–extension, intension–factual value. While on the informal level I use sense and reference, on the formal level I will use intension and extension.

sense of complex expressions. These are the Principle of Compositionality (Interchange) of Reference and the analogous Principle of Compositionality (Interchange) of Sense. They hold that the referent or sense of a complex is a function only of the referents or senses, respectively, of the constituent expression."[4] Carnap was the first to attribute both versions of the principle explicitly to Frege. He wrote the following in his fundamental semantic book, *Meaning and Necessity*:

"Frege Principles of Interchangeability:

. . . First principle . . . the nominatum of the whole expression is a function of the nominata of the names occurring in it.

. . . Second principle . . . the sense of the whole expression is a function of the senses of the names occurring in it."[5]

Now let us turn to the second question, i.e. the behaviour of functor–argument decomposition in the case of the two principles of compositionality. One may think that there is no problem at all, there are two different principles of compositionality, and we can use them to determine the logically relevant semantic values. But which of these principles do we have to take into consideration, and how? In logical investigations we are interested in the truth value of a sentence, i.e. in its reference. The reference can be determined by means of the first principle concerning reference, and the function which occurs in it asks for the reference of the arguments. Frege recognized that in some cases the reference of the whole expression cannot be determined by means of the references of its parts. Sometimes we need to take into consideration not only the reference of an argument but also its sense. However, at first glance this contradicts the principle of compositionality concerning reference. How did Frege try to get rid of the problematic situation? As it is well–known he differentiated between direct occurrence form indi-

[4]Salmon, 1994, p. 112, quoted by Pelletier, 2001, p. 88.

[5]Carnap, 1947, p. 121, quoted by Pelletier, 2001, p. 89. The principles are similar to the function principle, which is the second version of the principle of compositionality.

rect occurrence, ordinary (as he called it direct or customary) reference from indirect (oblique) reference and he said that if the occurrence of an argument is indirect, then its reference is its ordinary sense.

Let us focus now on functors. The output of the function included in the principle of compositionality concerning reference is the reference of the whole expression. We have seen that the reference of an expression might depend on either the reference or the sense of its arguments. Thus two types of functors can be differentiated:

1 If a given expression occurs directly in an expression, then the given expression can be considered to be an argument and the remaining part of the whole expression, is the functor. In that case the functor affects the ordinary reference of the given expression.

2 If a given expression occurs indirectly in an other expression, then the functor (the remaining part of the whole expression) affects the indirect reference (i.e. the ordinary sense) of the given expression.

We can say that the latter type of functors affect the (ordinary) sense of their arguments. Since reference is attached to reference, everything seems to be governed by the first principle. However, indirect reference is ordinary sense, and in order to get ordinary sense the second principle should be applied. Thus, the second principle is needed to determine the reference of the whole expression since in certain cases indirect reference is crucial to determine the reference of the whole expression. So we need the second principle not only in those cases when we are interested in the sense of the whole expression, but also when we want to determine its ordinary reference. Frege could not avoid maling use of both principles, but he did not mention how to apply the second one, how to derive the sense of the whole expression and what the connection is between the two principles.

At this point we can differentiate two main types of functors and introduce the notions of Fregean intensional and Fregean extensional functors, which will prove very useful later on. Let a functor be extensional or intensional in the Fregean sense if the occurrence of its argument is direct or indirect, respectively. Obviously, a functor is extensional in the Fregean sense if and only if it is not intensional in the Fregean sense.[6] We have to note that all functors in Frege's semantic theory (called Fregean functors) affect the reference of their argument and their outputs are usually of a given type of reference.

This subtle distinction between direct and indirect occurrence has a very problematic consequence: the reference (and therefore the sense) of an expression depends on the context in which it occurs and, obviously, we have to determine not only the indirect reference of an argument, but also its indirect sense. In the Fregean approach reference cannot be identified with sense, thus we have to speak about the sense of an expression occurring in an indirect context and this can be taken to be the sense of the expression's ordinary sense (etc.).

The next question to be adressed is how the problem mentioned above, the one connected with context–dependence (or more precisely occurrence–dependence) of the type of reference can be avoided. It must be emphasized here that the problem itself is not the context–dependence of reference. The real problem is that when we deal with a typical fixed situation (where the ordinary reference and the ordinary sense are given) in some cases the ordinary reference is the reference, while in other cases the indirect reference i.e. the ordinary sense is the reference.

Carnap recognized this problem in Frege's approach, and tried to follow another method. He characterized the differences between his and Frege's approach as follows:

---

[6]That type of definition of the Fregean intensional and extensional functors is not usual. Generally the notion of extensional functor is defined first, and non–extensional functors are considered to be intensional ones; the defined functors are not always Fregean.

> "A decisive difference between our method and Frege's consists in the fact that our concepts, in distinction to Frege's, are independent of the context. An expression in a well–construed language system always has the same extension and the same intension: but [in Frege's theory] in some context it has its ordinary nominatum and its ordinary sense, in other contexts its oblique nominatum and its oblique sense."[7]

The definitions of Fregean intensional and extensional functors apparently need to be modified only slightly to get definitions applicable to Carnap's approach. We only have to transfer the sensitivity of the type of semantic value from occurrences to functors. The result is that the functors which affect the reference of their arguments in order to get the reference of the output (of the whole expression) are extensional, and the functors which are not extensional and affect the senses of their arguments are intensional. Now it seems to be the case that the problem of context–dependence (or occurrence–dependence) is solved. We may also realize that the notion of extensional functor corresponds to the first principle of compositionality concerning reference; however, that of intensional functors does not correspond to the second principle concerning sense, since an intensional functor produces not the intension but the reference of its output.

Is there any way to embed the second principle in this picture? Ii my opinion this is desirable, since the second principle is more fundamental than the first one. Sense is the most fundamental semantic value. In order to belong to a natural language, in order to be an expression of a given language, the expression has to be meaningful. Adopting Kripke's treatment of names, only proper names can form exceptions, namely proper names may be expressions of a given natural language without having sense. (Usually nobody wants to 'understand' a proper name, everybody wants to use it for referring to something.) It can be said that except for proper names there is no expression in any natural language which has reference but no sense, since meaningfulness is the crucial

[7] Carnap, 1947, p. 125

characteristic of an expression that belongs to a given language. In two–component semantics an expression may have sense without having reference, but we cannot understand an expression which has reference but no sense.

Relying on the two principles of compositionality we can say that the reference (nominatum) of a functor is the function which derives us the reference (nominatum) of the whole expression from the references (nominata) of different arguments, and that the sense of a functor is the function which provides the sense of the whole expression from the senses of different arguments. Thus we can conclude that every expression has sense, but what about its reference? The reference of an extensional functor is given by the first principle of compositionality directly, but we have to suppose that the reference of the argument is defined. However, in the case of intensional functors the notion of reference cannot be defined, since there is no function which would give the reference of the output from the reference of the input. What happens when an argument of an extensional functor is an intensional one? To solve this problem we have to permit semantic value gaps, hence the sense of a functor will be not a total but a partial function. Therefore, the sense of an extensional functor would be a partial function on the possible senses of its arguments that is not defined when the argument is the sense of intensional functor[8].

2. Conclusion

From the logical philosophical point of view, introducing functor–argument decomposition and accepting its dominance result in informal compositionality of conceptual content which differs from compositionality in natural language. In two–component logical semantics two different principles have to be represented. We have showed that the first

[8]Consider for instance the functor '*Peter believes ...*' and ' *Peter does not believe ...*'.

principle of compositionality concerning reference may work in the case of extensional functors. The second one concerning sense is more general than the first one, it holds for all functors. In intensional logical semantics both principles are needed. It is clear that for maintaining the priority of the second principle we have to admit semantic partiality into our system to introduce the extensional–intensional differentiation.

In figure 5.3 a new box appears (at least in comparison with figure 5.2), the box of logical semantics. In order to embed the two principles of compositionality in our system, to represent them formally, and to differentiate extensionality and intensionality we have to create a whole system of logically relevant semantic values. As it is well–known, possible word semantics has great potential to treat intensionality (and extensionality) in logical semantics. Nevertheless, there are some theoretical differences with respect to the role of the two principles of compositionality, which appear not only in the logical–philosophical background but in the formal system as well.

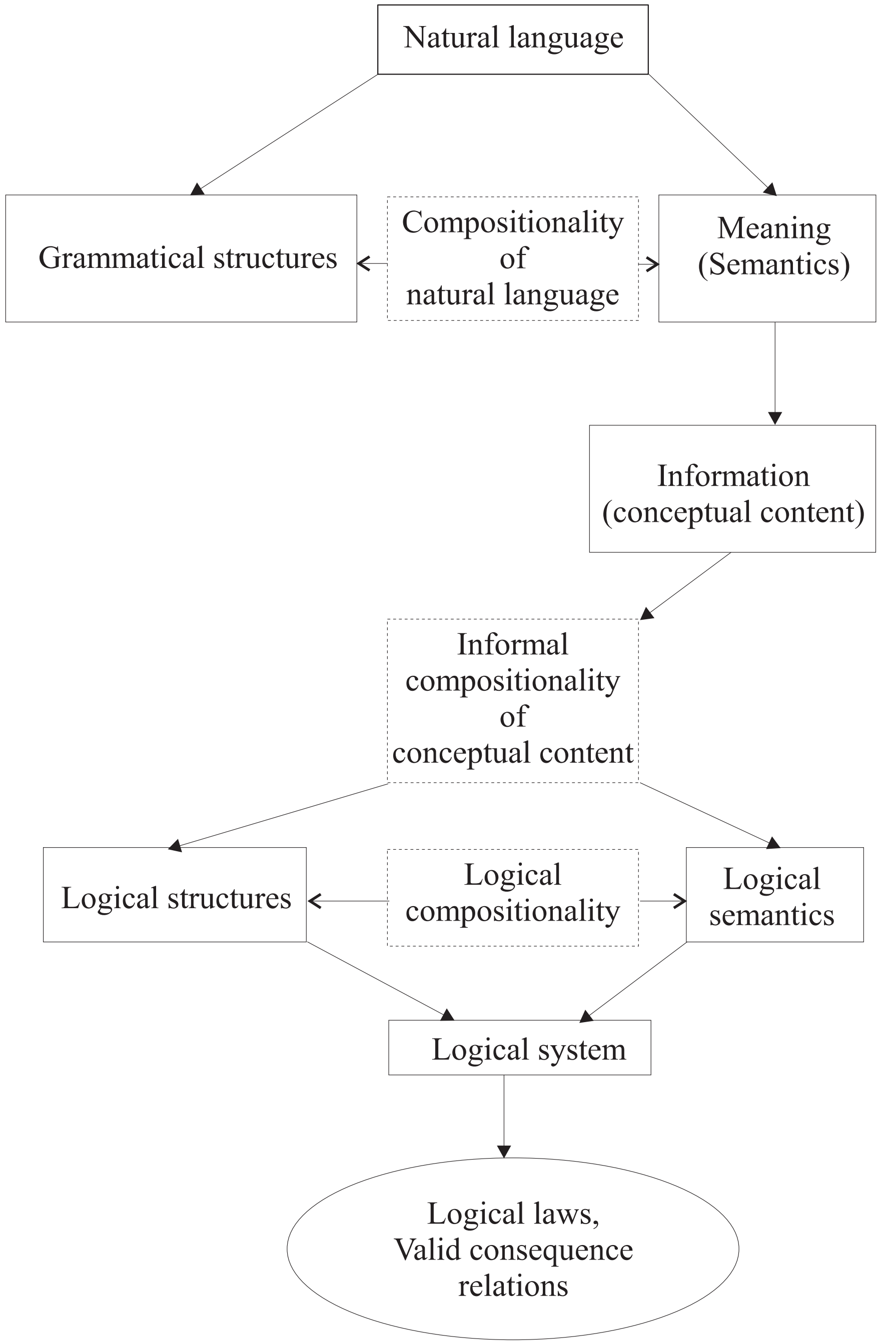

Figure 5.3. The place of logical (formal) semantics

Chapter 6

GENERAL LOGICAL SYSTEMS OF FUNCTOR–ARGUMENT DECOMPOSITION

Abstract We consider general logical systems of functor–argument decomposition. The defined notion of contexts as introduced here plays a crucial role in defining central logical notions such as satisfiability, consequence relations and validity. We outline the most important possibilities which in turn lead to different logical systems.

Keywords: Context, logically relevant frame, extensionality, intensionality, intensional logic, partial logic

It is needless to say that a type theoretical language with its possible models does not constitute a logical system, since the notion of functor–argument frame is too universal, logically relevant semantic values cannot appear in it. Therefore, there is no real opportunity to give the notion of logically valid inferences, consequence relations. However, at the same time there are many different possible ways to modify functor–argument frames in order to get logical systems. In the construction of logical systems I will show one of the most general such ways which is especially relevant from the logical–philosophical point of view.

1. The most elementary cases

A type–theoretical language with identity is closer to constituting a logical system than a type-theoretical language without identity since in the former some semantic values (of identity sentences) appear which look like logically relevant semantic values. From a theoretical point of view it is not problematic to define a very simple logical system for *identity sentences*. The received system would be very similar to classical propositional logic (in the total case), and to propositional logic allowing truth value gaps (in the partial case). These cases are so simple that we will not deal with them.

As it was mentioned in Chapter 4 the proper question is what can be said about the semantic values of sentences in the light of the possible semantic values of identity sentences. If we try to follow a very simple method, we can embed identity sentences into the set of sentences (i.e. we can suppose that $Cat(o_=) \subset Cat(o)$). If we focus on the total case, then in the semantic definition of total frames we may introduce a stronger condition: $D(o) = D(o_=)$, (i.e. for example in this case there are only two possible semantic values for a sentence: 0 or 1, in other words it may be true or false). On the one hand this means that in syntax there is no need to differentiate sentences from identity sentences, we can avoid introducing $o_=$, the type of identity sentences, and on the other hand, that in semantics the senses (meanings) of sentences can appear only in a very restricted manner: formulae may have 0 or 1 as semantic values. It is obvious that the received systems will be different versions of the logical system which is usually called extensional (type–theoretical) logic. The decision concerning the possible semantic values of sentences outlined above dominates the whole system and it has serious consequences. More specifically 'real' senses of sentences disappear and only one aspect of sense can be handled: whether a sentence with a given sense is true or false in a fixed context. The received system can be used to represent well-known extensional properties.

2. Context as a bridge between different sorts of semantic values

The next step is to introduce the notion of context. This step is very important, since it provide a real possibility to represent sentential sense which differs significantly from the 'extensional sense'. (At first we will only deal with sentences.) As it was emphasized many times in previous chapters senses are the primary semantic values. They have many different roles, but one of these is especially important: If a sentence has any other semantic value besides sense, then its sense has to determine the other semantic value. Semantic values of identity sentences are appropriate candidates for other semantic values of sentences. Therefore, if a sentence in general may have the same semantic value as an identity sentence, then the sense of the sentence has to determine it.

How can the sense of a sentence determine its other semantic value? The answer for this question can be found in those situations when we need these values or when we use them, and so it is very straightforward: only in the case of uttering the given sentence are we interested this other value of the sentence. A sentence utterance can only be grasped in connection with utterances of other sentences. From a theoretical point of view, usually a set of utterances of given sentences is considered to be a simple representation of context.

How can we represent a context? The natural (and usual) way is to provide those sentences which are true in the context in question (or to specify which sentences are true, which are false and which are irrelevant). Obviously, this depends on the senses of the given sentences, hence the precise formulation is the following: a context is a special representation which is based on a set of senses of the relevant sentences, and which also includes specifying the truth values of those sentences.

In a formal model, the main component of a context is a function from the set of senses of sentences to the set of semantic values of identity sentences (i.e. to the set of possible truth values). Up to this point we have dealt with sentences only, but, as expected, there is no theoretical difference with respect to the behaviour of expressions of other primitive types. Expressions of primitive types other than the type of sentences play a similar and crucial role in constructing the notion of context as sentences. (Almost the same can be said about these expression as about sentences, however, there is no such aid that could be compared to the one provided by identity sentences.) In order to define the general notion of context we have to specify the sets of 'secondary' semantic values of expressions in the case of every primitive type. We will call 'secondary' semantic values extensions (or factual values using Ruzsa's original terminology). The set of extensions of a given primitive type γ will be denoted by $D_{ext}(\gamma)$.

Definition 6.1. A *system of extensions of primitive type(s)* is the system of sets

$$\langle D_{ext}(\gamma)\rangle_{\gamma\in PT}$$

such that

(a) $D_{ext}(o) = \{0, 1, 2\}$, $\Theta_o^{ext} = 2$ (Θ_o^{ext} the extensional null entity of type o);

(b) if $\gamma \in PT$ and $\gamma \neq o$, then $D_{ext}(\gamma)$ is an arbitrary set with a distinguished member Θ_γ^{ext}, which is called the extensional null entity of type γ;

Definition 6.2. Let G ($= \langle Dom(\gamma)\rangle_{\gamma\in TYPE_{PT}}$) be a (total or partial) frame, and SE ($= \langle D_{ext}(\gamma)\rangle_{\gamma\in PT}$) be a system of extensions of primitive type(s). A *contextual function for the frame G* relying on the system of extensions SE is a function C_G such that

(a) the domain of the function C_G is $\cup_{\gamma\in PT} Dom(\gamma)$;

(b) if $u \in Dom(\gamma)$, then $C_G(u) \in D_{ext}(\gamma)$ ($\gamma \in PT$).

In logically relevant cases not only a frame is needed, but also a context for the frame. The the next definition introduces the notion of a context for a frame.

Definition 6.3. A (total or partial) *context for a frame* G is an ordered triple

$$\langle G, SE, C_G \rangle$$

where

(a) G $(= \langle Dom(\gamma) \rangle_{\gamma \in TYPE_{PT}})$ is a (total or partial) frame;

(b) SE $(= \langle D_{ext}(\gamma) \rangle_{\gamma \in PT})$ is a system of extensions of primitive type(s);

(c) C_G is a contextual function for the frame G relying on the system of extensions SE.

In Definition 2.7 we introduced a general notion of models. By means of the notion of a context for a frame the notion of logically relevant models can be introduced.

Definition 6.4. A *logically relevant (total or partial) model* $[M_C]$ is an ordered triple

$$\langle CFF, \varrho, v \rangle$$

where

(a) CFF $(= \langle G, SE, C_G \rangle)$ is a (total or partial) context for the frame G;

(b) ϱ, v are functions as in Definition 2.7.

If we define central logical notions as satisfiability, unsatisfiability, consequence relation and validity by means of context sensitive frames and logically relevant (total or partial) models, we get very strong notions.

Definition 6.5. Let Γ be a set of formulae, i.e. $\Gamma \subset Cat(o)$ and A a formula, i.e. $A \in Cat(o)$.

(a) Γ is satisfiable if there is a logically relevant model M_C such that $C_G(\llbracket A \rrbracket_{M_C}) = 1$ for all $A \in \Gamma$, where $M_C = \langle CFF, \varrho, v \rangle$ and $CFF = \langle G, SE, C_G \rangle$

(b) The set Γ is unsatisfiable if it is not satisfiable.

(c) A is a logical consequence of Γ ($\Gamma \vDash A$) if the set, $\Gamma \cup \{\neg A\}$ is unsatisfiable.

(d) A is valid ($\vDash A$) if $\emptyset \vDash A$.

(e) A is irrefutable if there is no logically relevant model M_C such that $C_G(\llbracket A \rrbracket_{M_C}) = 0$.

Logically relevant intensional models provide a new level where different features of sense can be represented. The main idea is that the possible context can be determined in logically relevant intensional models.

Definition 6.6. Let $\langle G, \varrho, v \rangle$ be a (total or partial) model, and SE be a system of extensions, C_G^i be a contextual function from G to SE for $i \in I$, where I is an arbitrary nonempty set. A *logically relevant intensional (total or partial) model* $[M_C^{int}]$ is the set of ordered triples

$$\{\langle CFF_i, \varrho, v \rangle : i \in I\}$$

where $CFF_i = \langle G, SE, C_G^i \rangle$ is a (total or partial) context for the frame G.

By means of logically relevant intensional models a great number of ‘classical’ intensional features can be represented. For example, extensionality can be represented on two different levels: as extensionality in a context, and as extensionality in a logically relevant intensional model.

3. An example: 'classical' intensional logic

In this section we will show how to reconsider 'classical' intensional logic in the light of our general investigation. At the same time one can recognize the theoretical sources of our notion of context, and one can imagine its various theoretical role.

Following the traditional method, we may suppose that only two symbols belong to the set of primitive types, type o, i.e. the type of formulae as it appears in Definition 2.1, and type ι, the type of individual names. The system of types generated by o and ι as primitive types will be denoted by $TYPE_{Fr}$. In what follows we can suppose that our language is a type–theoretical language based on $TYPE_{Fr}$.

The next question is how to define frames relying on the standard method, which proceeds from extensions to intensions. From a general point of view, sense is the primary semantic value, hence we have to define the frame of logically relevant senses, i.e. the frame of intensions. Following the method of possible world semantics we can say that the intension of a formula is the rule that determines whether the formula expresses a true or a false statement in a given situation (world). This rule can represent the truth conditions of a formula. The intension of an individual name is the rule which determines its reference in a given situation (world).

An intensional functor–argument frame is a functor–argument frame such that

- The set of primitive types contains type ι, the type of individual names, and type o, the type of formulae.

- The rules mentioned above, which serve as intensions, are functions from the set of indices to the set of objects or truth values in the case of primitive types, and from the semantic domain of the input to the semantic domain of the output otherwise.

Definition 6.7. By an *intensional functor–argument frame* F_{int} let us mean an ordered triple

$$F_{int} = \langle U, I, D_{int} \rangle$$

satisfying the following conditions:

1 $D_{int}(o) = \{0, 1\}^I$;

2 $D_{int}(\iota) = U^I$;

3 $D_{int}(\langle \alpha, \beta \rangle) = D_{int}(\beta)^{D_{int}(\alpha)}$ for all $\langle \alpha, \beta \rangle \in TYPE_{Fr}$

$M \vDash_i A$ means that the function, which is the semantic value of formula A ($A \in Cat(o)$) with respect to M, is 1 at i, i.e. $[\![A]\!]_M(i) = 1$. Using intensional functor–argument frames we can introduce one of the simplest notions of logical consequence. Obviously it has to be presupposed that 0 and 1 have special logical roles or logical "meanings". 1 indicates that a sentence has the property preserved by the intended notion of consequence relation. For the sake of simplicity we can say that 1 and 0 correspond to truth and falsity, respectively. I have to emphasize that there is no need to say anything about the nature of truth values here.

Definition 6.8. $\langle F, \varrho, v, i \rangle$ is said to be a *true intensional representation* of Γ ($\subseteq Cat(o)$) if

1 F ($= \langle U, I, D_{int} \rangle$) is an intensional functor–argument frame;

2 $\langle F, \varrho, v \rangle$ ($= M$) is a model on F;

3 $i \in I$;

4 $M \vDash_i A$ for all $A \in \Gamma$.

Definition 6.9. Suppose that $\Gamma \subseteq Cat(o)$ and $A \in Cat(o)$. A is a strong semantic consequence of Γ ($\Gamma \Vdash A$) if A is true, i.e. $M \vDash_i A$ in every true intensional representation of Γ.

In the framework outlined above the semantic value of any formula is a sentence intension, and we can speak about truth and falsity, since sentence intensions are functions from indices to truth values. Therefore, sentences (and individual names) have two different sorts of semantic value. In the first place they have intensions (corresponding to their informal senses) and in the seocnd place formulae have truth values (and individual names have reference) at a given index. However, only intensions of compound type expressions are present. A natural question arises here: is there any connection between the truth values of two formulae if one of them involves the other as a subformula? From a general point of view the answer is 'no' or at least 'it depends'. However, in special cases we may recognize some deterministic connection between the semantic values in question. In order to get the whole picture we will use the well–known family of extensional semantic values.

Definition 6.10. By an *extensional functor–argument frame* F_{ext} let us mean an ordered pair

$$F_{ext} = \langle U, D_{ext} \rangle$$

satisfying the following conditions:

1 U is an arbitrary non–empty set;

2 $D_{ext}(\iota) = U$;

3 $D_{ext}(o) = \{0, 1\}$;

4 $D_{ext}(\langle \alpha, \beta \rangle) = D_{ext}(\beta)^{D_{ext}(\alpha)}$ for all $\langle \alpha, \beta \rangle \in TYPE_{Fr}$

Remark 6.11. The difference between intensional and extensional functor–argument frames is manifested only in the definitions of domains of primitive types. In extensional cases, where M is a model on an extensional functor–argument frame, if $A \in Cat(o)$, then $M \vDash A$ means that $[\![A]\!]_M = 1$.

Definition 6.12. A model $M = \langle F, \varrho, v \rangle$ on F is said to be a *true extensional representation* of Γ ($\subseteq Cat(o)$) if

1 F is an extensional functor–argument frame;

2 $M \vDash A$ for all $A \in \Gamma$.

Definition 6.13. Suppose that $\Gamma \subseteq Cat(o)$ and $A \in Cat(o)$. A is a strong semantic consequence of Γ ($\Gamma \Vdash A$) if A is true with respect to M i.e. $M \vDash A$ in every true extensional representation M of Γ.

We have a type–theoretical language, and two different notions of frames, intensional and extensional. Both contain logically relevant semantic values at least for sentences and individual names. The semantic values of compound type expressions are generated from the semantic values of primitive type expression by the principle of contextuality.

References

Bar-Hillel, Y. (1950). On syntactical categories. *Journal of Symbolic Logic*, 15:1–16.

Beaney, M., editor (1997). *The Frege Reader*. Blackwell, Oxford.

Carnap, R. (1947). *Meaning and Necessity: A study in semantics and modal logic*. University of Chicago Press, Chicago, IL.

Church, A. (1940). A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68.

Dunn, J. M. and Hardegree, G. M. (2001). *Algebraic methods in philosophical logic*, volume 41 of *Oxford logic guide*. Oxford University Press, New York.

Frege, G. (1952a). Function and concept. In Geach and Black, 1952, pages 21–41. Translated by P. T. Geach form 'Funktion und Begriff', H.Pohle, Jena, 1891.

Frege, G. (1952b). On concept and object. In Geach and Black, 1952, pages 42–55. Translated by P. T. Geach from 'Über Begriff und Gegenstand', Vierteljahrsschrift für wissenschaftliche Philosophie 16, 1892, pp. 192-206.

Frege, G. (1952c). On sense and reference. In Geach and Black, 1952, pages 56–78. Translated by M. Black from 'Über Sinn und Bedeutung', Zeitschrift für Philosophy und philosophische Kritik 100, 1892, pp. 25-50.

Frege, G. (1980). *The Foundation of Arithmetic. A logic-mathematical enquiry into the concept of number*. Basil Blackwell, Oxford, second revised edition. Translated by J. L. Austin, from 'Grundlagen der Arithmetik. Eine logisch-matematisch Untersuchung Über den Begriff der Zahl', W. Koebner, Breslau, 1884.

Frege, G. (1997). Begriffsschrift, a formula language of pure thought modelled on that of arithmetic. In Beaney, 1997, pages 47–78. Selections (Preface and part I). Translated by M. Beaney from 'Begriffsschrift, eine der arithmetischen nachgebildete Formelsprachen des reinen Denkens', L. Nebert, Halle, 1879.

Geach, P. T. and Black, M., editors (1952). *Translations from the philosophical writing of Gottlob Frege*. Basil Blackwell, Oxford.

Hodges, W. (2001a). A context principle. Ms.

Hodges, W. (2001b). Formal features of compositionality. *Journal of Logic, Language and Information*, 10:7–28.

Husserl, E. (1970). *Logical investigation*, volume II. Routledge & Kegan Paul.

Janssen, T. M. V. (1997). Compositionality. In Benthem, J. v. and Ter Meulen, A., editors, *Handbook of Logic and Language*, pages 417–473. Elsevier, MIT Press, Amsterdam, The Netherlands, Cambridge, MA.

Janssen, T. M. V. (2001). Frege, contextuality and compositionality. *Journal of Logic, Language and Information*, 10:115–136.

Mihálydeák, T. (2006). The logical-philosophical basis of logical systems. In Dietz, K., editor, *My Fulbright Experience*, pages 111–120. Budapest.

Partee, B. (1984). Compositionality. In Landman, F. and Veltman, F., editors, *Varieties of Formal Semantics*, pages 281–312. Foris, Dordrecht.

Pelletier, F. J. (2001). Did Frege believe Frege's principle? *Journal of Logic, Language and Information*, 10:87–114.

Rott, H. (2000). Words in contexts: Fregean elucidations. *Linguistics and Philosophy*, 23:621–641.

Ruzsa, I. (1997). *Introduction to metalogic*. Áron Publishers, Budapest.

Salmon, N. (1994). Sense and reference: Introduction. In Harnish, M., editor, *Basic Topics in the Philosophy of Language*, pages 99–129. Prentice-Hall, Englewood Cliffs, NJ.

Szabo, Z. G. (2000). Compositionality as supervenience. *Linguistics and Philosophy*, 23:475–505.

Tarski, A. (1983). The concept of truth in formalized language. In Corcoran, J., editor, *Logic, Semantics, Metamathematics*, pages 152–278. Hackett Publishing, Indianapolis, second edition.

Thomason, R. H. (1999). Type theoretic foundations of context, part 1: Contexts as complex type-theoretic objects. In Bouquet, P., Serafini, L., Brézillon, P., Benerecetti, M., and Castellani, F., editors, *Modeling and Using Contexts: Proceedings of the Second International and Interdisciplinary Conference, CONTEXT'99*, pages 352–374. Springer–Verlag, Berlin.

Thomason, R. H. (2001). Contextual intensional logic: Type–theoretic and dynamic considerations. Ms.

Werning, M. (2004). Compositionality, context, categories and the indeterminacy of translation. *Erkenntnis*, 60:145–178.