TOPOI LIBRARY

An Essay in Universal Semantics

Achille C. Varzi

KLUWER ACADEMIC PUBLISHERS

ACHILLE C. VARZI
Department of Philosophy,
Columbia University, New York

AN ESSAY IN UNIVERSAL SEMANTICS

KLUWER ACADEMIC PUBLISHERS
DORDRECHT / BOSTON / LONDON

A C.I.P. Catalogue record for this book is available from the Library of Congress.

ISBN 0-7923-5629-2

Published by Kluwer Academic Publishers,
P.O. Box 17, 3300 AA Dordrecht, The Netherlands.

Sold and distributed in North, Central and South America
by Kluwer Academic Publishers,
101 Philip Drive, Norwell, MA 02061, U.S.A.

In all other countries, sold and distributed
by Kluwer Academic Publishers,
P.O. Box 322, 3300 AH Dordrecht, The Netherlands.

Printed on acid-free paper

Printed in the Netherlands.

Acknowledgments

This essay originated as my doctoral dissertation, which was submitted in 1994 to the Department of Philosophy in the University of Toronto. Much that appears here appeared there, but the rewriting has been extensive and some parts have been substantially revised. I have also included new material drawn from recent work in the area, and I have made an effort to do away with outdated conjectures and speculations.

Some very special people helped me during this project, and I would like to express my deep gratitude to them: Hans Herzberger, my teacher, whose words and letters have been more supportive than I can say; Mark Vorobej, whose encouragement at crucial junctures proved invaluable; Piero Bonatti, Roberto Casati, Alex Simpson, and Jamie Tappenden, who helped me to clean up my ideas over the years; and Friederike Oursin, my wife, who shared every single stage of this work. I would also like to thank Peter Apostoli, John Bell, Hector Levesque, and Alasdair Urquhart for penetrating comments on my dissertation, some of which have helped to bring this monograph to its present form. During the final rewriting, I have benefited also from discussions with Horacio Àrlo Costa, John Collins, Haim Gaifman, Dominic Hyde, Isaac Levi, Graham Priest, and the participants in various talks that I have had the opportunity to give in various places; I am thankful to all for their remarks and provocations. Finally, I would like to thank Ermanno Bencivenga, both for his support and encouragement during the very final stage, and for the influence that some of his work has had on my own way of thinking about these matters from the beginning.

Contents

INTRODUCTION

Semantics, understood as a formal theory of the relationships between languages and their models, must involve some means of describing these domains: some notion of a language, and some notion of a model. To the extent that the account provides an accurate construal of both notions, to that extent it properly qualifies as "semantics". And to the extent that it does not depend on specific constraints upon either notion, to that extent the account is "universal", i.e., philosophically neutral and widely applicable.

It is this general approach that sets the background, and somehow the foreground, of the present essay. On the one hand, it seems to me that a broad outlook can still be beneficial to our understanding of a number of semantic issues, and this can only be achieved by working with a suitably wide notion of a language. A common limitation of much work in model-theoretic semantics—including "non-standard" theories—seems to me to lie in its relative idiosyncrasy. With few notable exceptions, proposals have been local rather than taking place within global linguistic frameworks. And although the resulting variety of theories has a corresponding variety of merits, how to assess them and to look for further progress is often subject to the possibility of providing more extensive accounts of the relevant linguistic structures. On the other hand, I am also convinced of the importance of working with a widest possible notion of a model, free of various conditions that are typically assumed in the context of formal semantics. A severe limitation of most theories— especially "standard" ones—is their commitment to some view or other as to what is to count as an "admissible" model for a given language. And although this way of proceeding has proven to be convenient for special purposes, all the same a certain embarrassment arises when one has to resort to artificial measures to regiment alternative intuitions or intractable phenomena.

My anxiety about a general notion of language is chiefly methodological. However, the need for a general notion of model is more substantive and arguably central to any semantic project that aims at some universality. For instance, a major requirement of classical semantics is that every model be consistent (or coherent) as well as complete (or determinate) relative to the given language. Broadly speaking, this means that in classical semantics all models are assumed to provide a homomorphic interpretation of the language. They are made of well-defined, clear-cut entities, linked to one another and to the language's expressions in a univocal way: all singular terms (proper names and the like) are assumed to have a unique denotation; all general terms (predicates and the like) must represent properties or relations that are uniquely defined for each object in the domain; all sentences must receive a unique truth-value; and so on. Now, such an approach is perfectly defensible and it may well be advantageous for a number of special purposes. (It is certainly advantageous if we are interested in the semantics of mathematical theories, for instance.) However, when it comes to the general case the burden of the defense is philosophical—not semantical. There is no *a priori* semantic reason to rule out the possibility that (our representation of) what a language is about involves "gaps" and "gluts" of sorts. Indeed, since there is no general criterion for sieving out troublous forms of incompleteness from innocuous ones, there is no general and effective way of ruling out gaps without discarding many unproblematic cases as well. And since there is no general antidote against inconsistency, there is no guarantee that gluts can be ruled out without also rendering a great deal of perfectly innocent semantic thinking impossible.

This theoretical anxiety goes hand in hand with more practical considerations. A model represents the world, or one way in which the world could be. But a representation may be incomplete, or even inconsistent, and yet perfectly adequate for most purposes. Data bases and works of fiction are models of this sort. A data base need not be able to answer every possible query that can be formulated in the query language. And a novel need not be based on a thorough description of its imaginary world to make sense. We are only told a few things about the world of Sherlock Holmes, yet we can understand the Holmes stories and reason about them. As it turns out, we are also told inconsistent things: Watson's war wound is supposed to be in his shoulder, but also in his leg. Yet one discrepancy is no logical chaos. Lots of facts are perfectly clear about Holmes and Watson, in spite of the gaps and the gluts involved in the model of their world. These sorts of situation are quite normal and may arise in any model. It seems to me that a good semantic theory should be able to handle them.

There is, moreover, a general and widespread concern that a semantic theory based on the requirement of a perfect homomorphism between languages and models will be inapplicable to ordinary language. Vagueness, ambiguity, open texture, presupposition failure, sortal incorrectness—these are all common features of the language we ordinarily speak (but also the language of the exact sciences and the language of philosophy) that seem at odds with the requirements of standard semantics. For each of these "problems"—and for many others indeed—there is by now one or more corresponding "solutions" within some more or less standard framework. However, the resulting accounts are admittedly severe—sometimes *ad hoc*, sometimes ingenious, but rarely natural, and there is a growing consensus that a more liberal attitude towards those recalcitrant phenomena could yield a better overall theory. At any rate, it seems to me that even a successful regimentation of these phenomena would not diminish the value of semantic theories capable of taking them at face value—theories in which those "problems" are part of the data to be taken into account, rather than accidents to be explained away.

With all this, the last three decades have featured a great deal of work under the rubric of "going general" in model-theoretic semantics, sometimes with excellent results. Especially in the areas of natural language semantics and philosophical logic (but also in computer science and certain branches of mathematical logic), there has been a remarkable outgrowth of new semantic theories and frameworks. However, most of this work has arisen out of specific needs and for specific purposes—for instance, to account for one or another of the linguistic phenomena mentioned above, or to establish a completeness result for some logical calculus. Not much work has been done in the direction of a "general" generalization. Also, gaps have played a leading role in this trend: gluts have been ostracized for a much longer time, and the intrinsic duality between the two notions has not been given much consideration. This essay is an attempt to overcome these limitations. My approach will be top-down rather than bottom-up. (This is why I want to start with a reasonably abstract notion of language.) And the upshot will be a framework within which a large variety of semantic theories naturally fall and from which the semantics of a wide class of logics (standard and nonstandard) may be obtained as special cases. There is no claim to complete universality, of course. But my hope is that the approach, if not the framework as a whole, may shed new light on some important issues in formal semantics.

The details of the framework form the core of the first chapter, *Foundations*. I first define languages and models, and then suggest a general technique for constructing semantic evaluations (of a language $\mathcal{L}$ on a model $\mathcal{M}$). Very roughly, the main idea is grounded on some structural properties of the class of all models and may be viewed as implementing a form of "supervaluationism": the value of an expression relative to a given model is a function of the values of that expression relative to various ways of "sharpening" the model (by filling in its gaps and weeding out its gluts, if any). The working of this account is illustrated in connection with some examples of increasing complexity. In the second chapter, *Developments*, I turn to the main properties of the resulting apparatus. Here again the emphasis is on the general picture; but I am also concerned with a study of the specific conditions under which some major results of standard semantics can be extended to our more general account. While this concern might at times appear scholastic, I hope it will help to provide a better understanding of the strengths and weaknesses of the approach pursued here, as well as offer new insights into why those results hold in the standard case. In particular, it is shown that the notion of logical validity as applied to single expressions is rather standard, and in fact reduces to the corresponding classical notion when this is available. Thus, gaps and gluts can be explained away as local phenomena that do not metastasize throughout the framework. However, when it comes to the general notion of logical entailment the picture is more intricate, and the notion of argument validity does not appear to be recursively specifiable except in very special cases. To shed light on this picture, a cost-benefit analysis of the framework—along with some general philosophical reflections—is offered in the *Concluding Remarks*.

Just about every technical notion in the following is construed, or assumed to be construed, in set-theoretic terms. The precise content of the underlying system of set theory will only rarely be of any importance, and I shall make little effort to exhibit the details of my discourse within it. At places, however, some explicit commitment on such matters is wanted, so I have included an *Appendix* which sorts these things out and also explains all necessary notational and terminological conventions. This should make the exposition self-contained. Certain passages, such as the proofs of the basic facts and examples in Section 1.3 [in square brackets], may be omitted by those who are prepared to take them on trust. In the main, however, the conceptual apparatus is intended as a whole and frequent use will be made at later stages of material introduced earlier.

1. FOUNDATIONS

I shall set up the basic framework in three steps. First, in Section 1.1, I shall introduce the notion of a *language*, regarded as a purely combinatorial, uninterpreted system. Next, in Section 1.2, I shall turn to the relative notion of a *model*, or interpretation system: this notion will be characterized in a parallel fashion and will allow for the kind of generality which I am aiming at. Finally, in Section 1.3, I shall turn to the task of bridging languages and models via the concept of a *valuation*: this is the most problematic and controversial part but also—I hope—the most interesting one.

1.1. LANGUAGES

I shall not be concerned with any specific language, but only with the bone structure of the concept of a language insofar as this matters for the purpose of clarifying basic semantic notions. In this sense my characterization will be minimalistic: a language is essentially a system of (well-formed) expressions generated from a (well-ordered) class of symbols by means of some (well-grounded) structural operation.

1.1.1. PRELIMINARIES

To spell out the details of the definition in a convenient form (avoiding all discussion concerning the exact nature of a language's symbols and structural operation), we can make use of the auxiliary concept of a *type*, or *category index*. That is, we can suppose that the symbols of any language form a family of objects (of some sort) indexed by a fixed set $\mathcal{T}$ of types, and we may refer to such an indexing to determine how those objects may combine with one another to produce what is to count as an expression of the language. Besides ontological neutrality and conceptual simplicity, this procedure will also have some derivative advantages, which will become clearer as the discussion proceeds.[1]

The set of all types, I shall assume, is built up recursively from an initial stock of *primitive* (or *individual*) *types*. Intuitively, such types are to be associated with those categories of expressions whose grammatical status need not—or cannot—be analyzed in terms of other categories. One of these could be, for example, the category Declarative Sentence; another could be the category Proper Name. In fact, for most purposes these two basic categories would probably suffice, although someone would perhaps insist on adding a third one, Common Noun, or on dispensing with Proper Name in favor of an alternative basic category Noun Phrase, or Verb Phrase.[2] In general, however, I shall make little effort here to select such primitives with care. What we need to do is simply to assume a certain number of primitive, undefined types. And to allow for the greatest generality, it seems convenient to start off with a potentially infinite number, bypassing the problem of specifying the intuitive status of each of them. Thus, for definiteness I stipulate to identify the primitive types with the natural numbers

$$0, 1, 2, \ldots$$

taking 0 and 1 to represent the traditionally fundamental categories of declarative sentences and proper names, respectively. (Nothing of what follows depends significantly on this specific choice, or on the fact that

[1] The syntactic framework developed here is in line with the general approach of categorial grammar, with some formal simplifications leaning on related insights in combinatory type theory. For the former, the classical references are Leśniewski [1929], Carnap [1934], and Ajdukiewicz [1935]. For the latter, the seminal contributions date back to Schönfinkel [1924] and Curry [1929, 1930], though the type hierarchy (the analogue of our $\mathcal{T}$) was not introduced until Curry [1953] (see Sanchis [1964]). The two theories, in turn, have been developed under the impact of Husserl [1900-01] and Russell [1908], respectively. For a general outline, see Casadio [1988] and Wood [1994]; for a survey of connections between the two disciplines, as they arise out of recent developments, see van Benthem [1990] and Steedman [1988, 1993]; for an overview of the use of types in semantic analysis, see Carpenter [1996] and Turner [1997].

[2] See, for instance, Lewis [1970] and Cresswell [1973].

the set ω of all natural numbers is denumerable: all that matters is that the class of primitive types be isomorphic to some fixed ordinal α. Even the proviso that $\alpha \geq 2$ is dictated merely by reasons of convenience and common practice.)

On this basis, an infinite set of *derived* (or *functional*) *types* is obtained by introducing some operation on the set of primitive types, say the operation $\langle\ \rangle$ of pair formation. More generally, I shall assume that whenever t_1 and t_2 are any types, primitive or derived, a new derived type may be formed, which we may identify with the ordered pair

$$\langle t_1, t_2 \rangle.$$

The intuitive idea is to think of such a type as corresponding to those categories of expressions (functors, in the traditional terminology) that produce expressions of type t_2 when combined with expressions of type t_1. Thus, for instance, if 0 and 1 are interpreted as above, then $\langle 0,0\rangle$, $\langle 0,1\rangle$, $\langle 1,0\rangle$, and $\langle 1,1\rangle$ will be the types of such categories of functors traditionally referred to as (monadic) Connectives, Subnectives, Predicates, and Operators, respectively; $\langle\langle 0,0\rangle,\langle 0,0\rangle\rangle$, $\langle\langle 0,1\rangle,\langle 0,1\rangle\rangle$, etc., will be the types of the corresponding (monadic) Modifiers; and so on. Here are some examples from English:

it is not the case that	connective	$\langle 0,0\rangle$
the claim that	subnective	$\langle 0,1\rangle$
the husband of	operator	$\langle 1,1\rangle$
cries	predicate	$\langle 1,0\rangle$
is smarter than	relational predicate	$\langle 1,\langle 1,0\rangle\rangle$
loudly	predicate modifier	$\langle\langle 1,0\rangle,\langle 1,0\rangle\rangle$

Note indeed that if we introduce derived types by means of a binary operation, $\langle\ \rangle$, we can initially speak of "monadic" functors only. However, this implies no loss of generality, for every functor can eventually be regarded as a monadic functor of a certain type. The relational predicate 'is smarter than' illustrates this point: 'is smarter than' is naturally regarded as a dyadic functor that makes a sentence out of two proper names (as in 'Ann is smarter than Bruce'); but it can equally be regarded as a monadic functor which, when applied to any given proper name (say 'Bruce'), produces an expression ('is smarter than Bruce') that behaves again as a monadic functor: a functor that makes a sentence out of a proper name ('Ann'). We can therefore treat such a relational predicate as having type ⟨1,⟨1,0⟩⟩. Likewise, dyadic connectives, subnectives, operators, etc. may be thought of as having the types ⟨0,⟨0,0⟩⟩, ⟨0,⟨0,1⟩⟩, ⟨1,⟨1,1⟩⟩, etc. More generally, whenever $n \geq 1$ and $t_1, \ldots, t_n, t_{n+1}$ are types, primitive or derived, the type

$$\langle t_1, \langle t_2, \langle \ldots, \langle t_n, t_{n+1} \rangle \ldots \rangle\rangle\rangle$$

may be used to represent the category of those "n-adic" functors that combine with n-tuples of expressions of type $t_1, t_2, \ldots, t_n$ (in this order) to produce expressions of type t_{n+1}. In this sense, as long as the second coordinate of a derived type is allowed to be a derived type itself, it will appear that our relying on ⟨ ⟩ imposes no significant restriction on the class of possible functors.[3] The only important thing is that each derived type be distinct from all primitive types, and that no derived type be generated in more than one way. (In other words, the set of all types must be freely generated from the set of primitive types.)

On the intended interpretation, the classification of all types into two main sorts, individual and functional, reflects of course the traditional tenet that the linguistic combination of constituents into constitutes should be regarded not as a concatenation *inter pares*, but rather as the result of the application of one constituent upon the other(s). Thus, in general, two expressions x and y may combine to produce a new expression z (of a certain type) if and only if x is a functor and y an expression whose type coincides with the first coordinate of the type of x. More specifically, the assumption is that the expressions of any lan-

[3] The idea goes back to Schönfinkel [1924]. Of course, we could equally well follow the opposite strategy, allowing for n+2-adic derived types for each $n \in \omega$ while requiring their second coordinate to be a basic type. On the intended interpretation, the equivalence of the two procedures corresponds to the set-theoretic isomorphism $A^{B_1 \times B_2 \times \ldots \times B_{n+1}} \approx (\ldots((A^{B_1})^{B_2})\ldots)^{B_{n+1}}$ (but see infra, note 18). Various authors, following in the footsteps of Bar-Hillel [1950], Lambek [1958, 1961], and Geach [1970], have also considered allowing for greater flexibility in the rules of type assignment or functional application (see Bach [1984] and Wood [1994] for overviews; see also Ranta [1994] for extensions obtained by basing the grammar on a constructive type theory, in the form developed by Martin-Löf [1984]): although such extensions are in the spirit of further generality, I believe the simpler framework adopted here is sufficiently powerful to induce no dramatic conceptual limitations.

guage $\mathcal{L}$ can be recursively specified on the basis of some assignment of types to the symbols of $\mathcal{L}$: for each type *t*, the expressions of $\mathcal{L}$ will include a certain (possibly empty) set of expressions of type *t*, and this will comprise all the symbols initially assigned to *t* plus all those expressions that result from applying a given structural operation (usually some form of concatenation or juxtaposition) to pairs of expressions of type $\langle t',t\rangle$ and t', respectively.

Note that, for the reasons explained above, one only needs a *binary* operation to generate expressions in this way.[4] Not so obvious perhaps is the fact that one would not achieve greater generality if one allowed expressions to be built up by means of many different modes of formation. For instance, languages involving formal variables or variable-binders (such as integrals, quantifiers, and the like) seem to run afoul of this simple apparatus, unless a structural operation of functional abstraction is assumed in addition to the operation of functional application described here.[5] More generally, it might be suggested that functional abstraction is needed in order to bring out certain important relations between different levels of linguistic analysis, for instance, between deep logical structure and surface realizations.[6] I don't know whether this is indeed so. But the examples given below will show that one can go a long way (including dealing with various forms of variable binding) without such extensions. Thus, also in this respect it will appear that the present approach—though obviously very simplistic and idealized—imposes no major restriction on the class of admissible languages.

[4] By contrast, if we followed the alternative procedure mentioned in note 3, we would need an n+2-ary structural operation whenever $\mathcal{L}$ involves "n+2-adic" functors, at the price of considerable complexities in the overall apparatus.

[5] This view can be traced back to Ajdukiewicz [1935], Part 2, and is partly supported by a result of Bar-Hillel, Gaifman & Shamir [1960] (which says that categorial grammars based exclusively on functional application are equivalent to context-free phrase-structure grammars). See for instance Marsh & Partee [1987]. For an overview of the issues, see Jacobson [1996], § 5. Some arguments in support of the simpler approach advocated here may be found in Varzi [1993].

[6] Cresswell [1977] conjectures that all "semantically significant" transformational derivations might be seen as sequences of lambda-conversions, in the sense of Church [1940, 1941]. Alternatively, some authors have proposed supplementing functional application with a set of transformation-like rules: this was the gist of Lewis [1970] or Partee [1975, 1976].

Let then $\mathcal{T}$ be the set of all types, primitive or derived, built up in the indicated way:

$\mathcal{T}$ = the closure of ω under $\langle\,\rangle$.

Keeping in mind the intuitive roles described above, the general definition of a language goes as follows. (I shall first give the definition, and then illustrate its working with some examples.)

1.1.2. DEFINITION

A *language* is an ordered triple $(\mathbf{s}, \mathbf{g}, \mathbf{E})$ satisfying the following general conditions:

(a) $\mathbf{s}$ is a one-one function such that $\mathcal{D}\mathbf{s}\colon \alpha \to \mathcal{T}$ for some ordinal α;
(b) $\mathbf{g}$ is a one-one function such that $\mathcal{R}\mathbf{g} \cap \mathcal{R}\mathbf{s} = \varnothing$ (and $\mathcal{D}\mathbf{g}$ as below);
(c) $\mathbf{E}$ is the smallest system of sets closed under the properties: (i) if $\langle\beta,t\rangle \in \mathcal{D}\mathbf{s}$, then $t \in \mathcal{D}\mathbf{E}$ and $\mathbf{s}(\beta,t) \in \mathbf{E}_t$; and (ii) if $t,t' \in \mathcal{D}\mathbf{E}$, $x \in \mathbf{E}_t$, $y \in \mathbf{E}_{t'}$, and $t = \langle t',t''\rangle$, then $t'' \in \mathcal{D}\mathbf{E}$ and $\mathbf{g}(x,y) \in \mathbf{E}_{t''}$.

(For definiteness, I shall assume that $\mathbf{g}$ is the smallest function satisfying the stated conditions, i.e., $\mathbf{g}(x,y)$ is defined only if $\mathbf{g}(x,y) \in \bigcup\mathcal{R}\mathbf{E}$ by clause (c).)

It is understood that if $\mathcal{L} = (\mathbf{s}, \mathbf{g}, \mathbf{E})$ is any language, then the elements of $\mathcal{R}\mathbf{s}$ are the *symbols* of $\mathcal{L}$ and $\bigcup\mathcal{R}\mathbf{E}$ is the class of all *expressions* generated from those symbols by repeated application of the *structural operation* $\mathbf{g}$. The requirement that $\mathbf{s}$ and $\mathbf{g}$ be one-one functions is to avoid ambiguities; combined with the requirement that $\mathbf{g}$ be well-grounded on $\mathcal{R}\mathbf{s}$ (i.e., $\mathcal{R}\mathbf{g} \cap \mathcal{R}\mathbf{s} = \varnothing$), this will secure that every expression of $\mathcal{L}$ can be uniquely represented either as a symbol or as a compound of the form $\mathbf{g}(x,y)$. In particular, by an *$\mathcal{L}$-symbol of type t* (where $t \in \mathcal{R}\mathcal{D}\mathbf{s}$) I shall understand any element $x \in \mathcal{R}\mathbf{s}$ such that $x = \mathbf{s}(\beta,t)$ for some $\beta \in \mathcal{D}\mathcal{D}\mathbf{s}$ (such a β will simply be called the *alphabetic* or *ordinal index* of x); and an *$\mathcal{L}$-expression of type t* (where $t \in \mathcal{D}\mathbf{E}$) will be any element of $\mathbf{E}_t$. Since $\mathcal{D}\mathbf{s}$ is a function, this assignment of types to symbols and therefore to expressions is sure to be unique: the sets in the system $\mathbf{E}$ are all pairwise disjoint. (This follows immediately by induction on the number of applications of $\mathbf{g}$.)