VARIABLE-BINDERS AS FUNCTORS

Achille C. Varzi
Istituto per la Ricerca Scientifica e Tecnologica (IRST)
I-38050 Povo (Trento), Italy

(Published in *Poznań Studies in the Philosophy of the Sciences and the Humanities* 40 (1995), 303-19)

INTRODUCTION

The second part of Ajdukiewicz's classic paper on *Syntactic Connexion* is devoted to a discussion of languages involving variable-binding operators such as quantifiers, integrals, and the like. Ajdukiewicz argues that such operators are not genuine functors, and consequently that those languages run afoul of the basic functor/argument schema. At most he suggests that all variable-binders might be restricted to one kind only, or to the combination of ordinary functors with a "circumflex" operator (the terminology is from Whitehead and Russell's *Principia Mathematica*). And that suggestion has ever since become quite popular, particularly under the impact of Church's *Lambda Calculus.*

This fact is of no little historical interest, for there is a misleading tendency to present lambda-equipped categorial languages as an extension—rather than an implementation—of Ajdukiewicz's original insights. More important, however, is the philosophical import of this account. For apart from any considerations of esthetic elegance and conceptual parsimony, the claim that functional application alone is not a sufficient paradigm for linguistic analysis ties in directly with a number of fundamental issues in semantics.

In this paper I argue that this is by no means a necessary course. I argue that a fairly general semantic framework can be developed where the only relevant distinction is indeed the functor/argument distinction, and where the only structural operation for generating expressions is functional application, with no need to resort to functional abstraction as well. I shall not prove any general results to the effect that such a framework is universally applicable. However, the overall apparatus is illustrated in connection with some concrete examples—notably quantificational and full categorial languages—which should suffice to support my point.

In the concluding section, I then offer some remarks to illustrate the philosophical import of this approach. Particularly, I briefly discuss a certain conception of logic that emerges from the proposed account: the view that logic is essentially a *theory* in the model-theoretic sense, i.e. the result of selecting a certain class of models as the only "admissible" interpretation structures (for a given language).

LANGUAGES

As I mentioned, my starting point is essentially Ajdukiewicz's notion of a "pure" categorial language. It is a very simple, rather abstract notion, and it is based primarily on the idea that the expressions of any language may be divided into two kinds of syntactic categories[1], or types: *individual* (or *primitive*) and *functional* (or *derived*).

Intuitively, the former correspond to those categories of expressions whose syntactic status need—or can—not be analyzed in terms of other categories. One of these could be, for instance, the category (Declarative) Sentence; another could be the category (Proper) Noun[2]. We need make little effort here to select such primitives with care. And to allow for the greatest generality, we may as well start off with an infinite number, bypassing the problem of specifying the intuitive status of each of them. For definiteness, I shall just identify the primitive types with the natural numbers, taking *0* and *1* to represent the traditionally fundamental categories of sentences and nouns respectively.

On this basis, an infinite set of derived types is obtained by introducing some operation on the set of primitive types, say the operation of pair formation[3]. More generally, we may assume that whenever t and t' are types, primitive or derived, a new derived type may be formed, which we may identify with the ordered pair t,t'. The intuitive idea is to think of such a type as corresponding to those expressions (functors, in the traditional terminology) which produce expressions of type t' when combined with expressions of type t. For instance, if *0* and *1* are interpreted as above, then $0,0$, $0,1$, $1,0$, and $1,1$ will be the types of such functors traditionally referred to as (monadic) Connectors, Subnectors, Predicators and Operators, respectively; $0,0$, $0,0$, $0,1$, $0,1$, etc. will be the types of the corresponding (monadic) Adverbial Modifiers; and so on[4].

In general, our set of types T can then be defined as the closure of ω under . And the idea I wish to consider is that the expressions of *any* declarative language L can be recursively specified on the basis of some type assignment to its symbols: for each t T, the L-expressions of type t 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 juxtaposition, but I shall leave that open) to pairs of expressions of type t',t and t' (for some t' T).

DEFINITION 1. A *language* is an ordered triple of the form $(\boldsymbol{s},\boldsymbol{g},\boldsymbol{E})$ satisfying the following conditions:

(a) $\boldsymbol{s}$ is a one-one function with $D\boldsymbol{s}:\alpha\rightarrow T$ (for some ordinal α)

(b) $\boldsymbol{g}$ is a one-one function with $R\boldsymbol{g}\cap R\boldsymbol{s}=0$ (and $D\boldsymbol{g}$ as below)

(c) $\boldsymbol{E}$ is the smallest system of sets closed under the clauses *(i)* for each pair β,t $D\boldsymbol{s}$: t $D\boldsymbol{E}$ and $\boldsymbol{s}(\beta,t)$ $\boldsymbol{E}_t$; and *(ii)* if t',t, t' $D\boldsymbol{E}$, x $\boldsymbol{E}_{t',t}$ and y $\boldsymbol{E}_{t'}$, then t $D\boldsymbol{E}$ and $\boldsymbol{g}(x,y)$ $\boldsymbol{E}_t$.

[Intuitively, where L=($\boldsymbol{s}$,$\boldsymbol{g}$,$\boldsymbol{E}$) is a language, the elements of $R\,\boldsymbol{s}$ are the *symbols* of L, and $\bigcup R\,\boldsymbol{E}$ is the class of all *expressions* generated from those symbols by repeated application of the *structural operation* $\boldsymbol{g}$ (the requirement that $\boldsymbol{s}$ and $\boldsymbol{g}$ be one-one functions with disjoint ranges is to secure that each expression of L can be uniquely represented as either a symbol or a compound of the form $\boldsymbol{g}(x,y)$). In particular, an L-symbol of type t is any x $R\,\boldsymbol{s}$ such that x=$\boldsymbol{s}$(β,t) for some β $DD\boldsymbol{s}$ (such a β may be called the ordinal index of x), while an L-expression of type t is any element of $\boldsymbol{E}_t$ [5]].

There is no doubt that this definition allows for some peculiarities which make little intuitive sense. For example, clause *(a)* allows for the possibility that a language L involve an arbitrary number of symbols for any type t T, but it is clear that unless some L-symbols are assigned derived categories, the structural operation of L would be empty, and no compound expression could be generated. Also, there is nothing in the above definition to secure that all symbols of a given language can actually be used to produce expressions of type *0,* i.e. sentences. Such oddities, however, are irrelevant in principle, and I shall refrain from introducing unnecessary complexities into the account.

Indeed, the generality of Definition 1 lies precisely in the fact that it allows one to single out a desired language or class of languages simply by implementing clauses *(a)-(c)* with the appropriate additional conditions. Thus, for instance, one may want to require that the functional expressions of a language L=($\boldsymbol{s}$,$\boldsymbol{g}$,$\boldsymbol{E}$) always yield individual expressions, in the sense that whenever $\boldsymbol{E}_{t',t}$ is defined (for t,t' T), $\boldsymbol{E}_{t'}$ is also defined, and hence so is $\boldsymbol{E}_t$. Further, among such languages one may want to pick up those languages only whose categories of expressions always cancel to a given type t, i.e. can be used to generate expressions of type t. As I mentioned, presumably most languages in use should be regarded as languages whose categories of expressions always cancel to *0*. Overall, however, the fundamental semantic notions must be outlined for the general case.

In this regard, it should also be noted that the above definition provides a purely combinatorial characterization of the notion of a language. In particular, the difference between symbols and compound expressions is a *relative* one: the latter are generated by application of $\boldsymbol{g}$, the former are not. This means that definition 1 does not imply that a language's symbols must necessarily lack internal structure, but only that this structure—if any—is irrelevant to the syntax of the language. Apart from the fact that this secures a certain neutrality with respect to the sort of entities languages are made of (which was already an attractive feature of Ajdukiewicz's original account), a derivative advantage of this account is precisely the possibility of treating variable-binding operators as ordinary functors, viz. as "structured" symbols consisting of the operator *together with* a corresponding bound variable.

A few concrete examples will help illustrate all of this. To this end, let L=($\boldsymbol{s}$,$\boldsymbol{g}$,$\boldsymbol{E}$)

be any language, and for all n ω and all t, t' T, let ‘ $t^{\langle n\rangle},t'$ ’ be an abbreviation for the type t_0 t_1, …, t_{n-1},t' … , where $t_i=t$ for each $i<n$ (for $n=0$, $t^{\langle n\rangle},t'$ reduces to t').

EXAMPLE 1.1. L is a *sentential language* iff RD**s** { $0^{\langle n\rangle},0$: $n \geq 0$} —i.e. iff L comprises a certain stock of *sentence symbols* (= symbols of type 0) along with a number of *n-ary connectors* (= symbols of type $0^{\langle n\rangle},0$) for various $n>0$.

To redeem some familiar notation, suppose L includes a binary connector ‘↓’ (of type 0, $0,0$). Then, for all A, B $\boldsymbol{E}_0$, we may set:

(a)	‘$(A\downarrow B)$’	for	‘$\boldsymbol{g}(\boldsymbol{g}(\downarrow, A), B)$’
(b)	‘$(\neg A)$’	for	‘$(A\downarrow A)$’
(c)	‘$(A \quad B)$’	for	‘$(\neg(A\downarrow B))$’
(d)	‘$(A \quad B)$’	for	‘$(\neg(\neg A) \quad (\neg B))$’
⋮			⋮

EXAMPLE 1.2. L is an *elementary language* iff RD**s** { $t^{\langle n\rangle},t'$: $n\geq 0$ & $t,t'\leq 1$} —i.e. iff L comprises a set $SENT_L$ of *sentence symbols* (of type 0) and a set $NOUN_L$ of *noun symbols* (of type 1) along with a set $CON_{L,n}$ of *n-ary connectors* (of type $0^{\langle n\rangle},0$), a set $SUB_{L,n}$ of *n-ary subnectors* (of type $0^{\langle n\rangle},1$), a set $PRED_{L,n}$ of *n-ary predicators* (of type $1^{\langle n\rangle},0$) and a set $OPER_{L,n}$ of *n-ary operators* (of type $1^{\langle n\rangle},1$) for each $n>0$. In particular, we may assume that if L is an elementary language, then there exist a denumerable set V_L $NOUN_L$ and various sequences $Q_{L,n}$ disjoint from R**s** (for $n>0$) such that $Q_{L,n}\times V_L=\{\ x,y\ :\ x,y\quad CON_{L,n}\}$. In that case we may speak of V_L as a set of *noun variables*, referring to a pair of the form $Q_{L,n,i}, v$ (where i $DQ_{L,n}$ and v V_L) as an *n-ary quantifier binding* the variable v.That is, from a purely syntactic perspective, we may treat quantifiers as a special kind of “structured” connectors[6].

Again, to introduce some familiar notation suppose L is an elementary language whose symbols comprise a binary quantifier ‘ $\downarrow, v$ ’ for each element of a set V_L of noun variables. Then, for all v V_L and all A, B $\boldsymbol{E}_0$ we may set:

(a)	‘$(A\overset{v}{\downarrow}B)$’	for	‘$\boldsymbol{g}(\boldsymbol{g}(\ \downarrow, v\ , A), B)$’
(b)	‘$(\neg A)$’	for	‘$(A\overset{u}{\downarrow}A)$’
(c)	‘$(A \quad B)$’	for	‘$(\neg(A\overset{w}{\downarrow}B))$’
(d)	‘$(A \quad B)$’	for	‘$(\neg(\neg A) \quad (\neg B))$’
⋮			⋮
(e)	‘$(\quad vA)$’	for	‘$(\neg(A\overset{v}{\downarrow}A))$’
(f)	‘$(\quad vA)$’	for	‘$(\neg(\quad v(\neg A)))$’

(where u is, as usual, the first variable foreign to A and w the first variable foreign to both A and B).

EXAMPLE 1.3. L is a *full categorial language* iff $RD\boldsymbol{s} = T$—i.e. iff L comprises a non-empty set $S_{L,t}$ of symbols for every type $t \quad T$. In particular, we may assume that if L is a full categorial language, then for each type $t \quad T$ there exist a sequence $A_{L,t}$ disjoint from $R\,\boldsymbol{s}$ and a denumerable set $V_{L,t} \quad S_{L,t}$ so that, for every $t' \quad T$, $\{\ x,y,t'\ :\ x,y,t' \quad S_{L,\ t',\ t,t'}\ \} = A_{L,t} \times V_{L,t} \times \{t'\}$. In that case we may speak of each $V_{L,t}$ $(t \quad T)$ as a set of *variables of type t*, referring to a triple of the form $A_{L,\ t,i}$ $,v,t'$ $(t,t' \quad T,\ i \quad DA_{L,t},\ v \quad V_{L,t})$ as an *abstractor of type* $t',\ t,t'$ binding the variable v.

To illustrate, suppose now L is a full categorial language with an abstractor λ,v,t' for each variable v and each type $t' \quad T$; then the usual lambda-combinatorial notation can easily be introduced—for instance:

(a)	'(λvA)'	for	'$\boldsymbol{g}(\ \lambda,v,t'\ ,A)$'
(b)	'$\mathbf{I}_t$'	for	'$(\lambda v_\xi v_\xi)$'
(c)	'$\mathbf{K}_{t,t'}$'	for	'$(\lambda v_\xi(\lambda v_\zeta v_\xi))$'
(d)	'$\mathbf{S}_{t,t',t''}$'	for	'$(\lambda v_\alpha(\lambda v_\beta\,(\lambda v_\xi \boldsymbol{g}(\boldsymbol{g}(v_\alpha,v_\xi),\boldsymbol{g}(v_\beta,v_\xi)))))$'

(provided of course $v \quad \bigcup\{V_{L,t} : t \quad T\}$, $t,t',t'' \quad T$, $A \quad \boldsymbol{E}_{t'}$, and $v_\xi,\ v_\zeta,\ v_\beta,\ v_\alpha$ are fixed variables of type $t,\ t',\ \ t,t'$ and $t,\ t',t'$ respectively).

MODELS

We can now turn to the notion of a model (of a language L). The general idea is simply that a model must act as a sort of semantic lexicon: it must determine what sort of things may be assigned to the basic components of the language as their contensive counterparts, and it must do so within the limits set by the relevant type distinctions. Thus, in general, a model for a given language L of the form $(\boldsymbol{s},\boldsymbol{g},\boldsymbol{E})$ will be a structure M of the form $(\boldsymbol{d},\boldsymbol{h},\boldsymbol{I})$, with $\boldsymbol{d}$, $\boldsymbol{h}$ and $\boldsymbol{I}$ a triple of functions satisfying essentially the same conditions as $\boldsymbol{s}$, $\boldsymbol{g}$ and $\boldsymbol{E}$ respectively: to each category of expressions $\boldsymbol{E}_t \quad R\boldsymbol{E}$ there will correspond some non-empty domain of interpretation $\boldsymbol{I}_t \quad R\boldsymbol{I}$; to each symbol $\boldsymbol{s}(\beta,t) \quad \boldsymbol{E}_t$ there will correspond a denotation $\boldsymbol{d}(\beta,t) \quad \boldsymbol{I}_t$; and to the structural operation $\boldsymbol{g} \quad (\bigcup R\boldsymbol{E})^3$ there will correspond an operation $\boldsymbol{h} \quad (\bigcup R\,\boldsymbol{I})^3$ subject to the same type restrictions, i.e. such that $x \quad \boldsymbol{I}_{t',t}$ and $y \quad \boldsymbol{I}_{t'}$ always imply $\boldsymbol{h}(x,y) \quad \boldsymbol{I}_t$[7].

DEFINITION 2. A *model* for a language $(\boldsymbol{s},\boldsymbol{g},\boldsymbol{E})$ is a structure $(\boldsymbol{d},\boldsymbol{h},\boldsymbol{I})$ satisfying the following conditions:

(a) $\boldsymbol{d}$ is a function on the domain of $\boldsymbol{s}$

(b) $\boldsymbol{h}$ is an operation distinct from $\boldsymbol{d}$

(c) $\boldsymbol{I}$ is a system of sets closed under the clauses *(i)* for each pair $\beta,t \quad D\boldsymbol{s}$: $t \quad D\boldsymbol{I}$ and $\boldsymbol{d}(\beta,t) \quad \boldsymbol{I}_t$; and *(ii)* if $t',t\ ,t' \quad D\boldsymbol{I}$, $x \quad \boldsymbol{I}_{t',t}$ and $y \quad \boldsymbol{I}_{t'}$, then $t \quad D\boldsymbol{I}$ and $\boldsymbol{h}(x,y) \quad \boldsymbol{I}_t$.

I think this is more or less the notion of a model that Ajdukiewicz had in mind, at least relative to the general notion of a language introduced above. Of course this is all quite abstract, and the intuitive appeal of the account depends on the actual make up of the functions $\boldsymbol{d}$, $\boldsymbol{h}$, $\boldsymbol{I}$. For example, if the sentence type 0 were in $D\boldsymbol{E}$, then $\boldsymbol{I}_0$ might be construed as a set of “truth-values”, or perhaps as a set of “propositions”. And if 1 or $1,0$ were in $D\boldsymbol{E}$, one might think of $\boldsymbol{I}_1$ as a set of “individuals” and $\boldsymbol{I}_{1,0}$ as a set of “properties”. However, in general I believe semantics ought to remain neutral with respect to the philosophical question of what kind of entities should be taken as interpretations of what expressions, particularly where commitment is not necessary (for the same reason, in the definition of a language the notions of “sentence”, “noun”, “predicator”, etc. are not defined independently, but only in relation to the auxiliary notion of a type).

In fact, what I want to stress here is that the generality of this approach lies precisely in the fact it allows one to single out a desired model or class of models simply by implementing clauses *(a)*-*(c)* with the appropriate additional conditions, just as with the notion of a language. For instance, the definition allows the domains of interpretation associated with the basic categories of expressions of the given language L to be arbitrary sets, and hence to vary from model to model. That in practice one is often interested only in the study of models which agree in assigning the same, fixed domain of interpretation to certain basic categories of L (say, only models M such that $\boldsymbol{I}_0$ —if defined—is a fixed set of truth-values) is another matter, and may be accounted for by singling out that class of models whose elements satisfy the desired conditions. And of course the same criterion may be used to account for the fact that some symbols are often assumed to have a fixed “meaning” (i.e. denotation in the case of individual symbols, and denotation *plus* relative conditions on the structural operation in the case of functional symbols): in general, such assumptions can be regarded as determining a certain selection of models as the only admissible ones, and different assumptions will correspond to different selections. Thus, as I take it, to say e.g. that a language L has a classical conjunction connective ‘ ’ is to say that the only models to be considered are among those models where $\boldsymbol{I}_0$ is, say, the set $2=\{0,1\}$, and where ‘ ’ denotes that function $f: 2 \rightarrow 2^2$ such that $f(x)(y)=x \cap y$ for all x,y 2; to say that L has a standard multiplication operator ‘ ’ is to say that the only models to be considered are among those models where $\boldsymbol{I}_1$ is, say, the set ω, and where ‘ ’ denotes that function $f: \omega \rightarrow \omega^\omega$ such that $f(x)(y)=x \cdot y$ for all x,y ω; and so on.

In this regard, there is yet an important feature of our definition that marks a departure from (or rather, a generalization of) the standard Ajdukiewiczian approach: there is in fact no requirement that every domain of the form $\boldsymbol{I}_{t',t}$ be a “Fregean” set of functions $f: \boldsymbol{I}_{t'} \rightarrow \boldsymbol{I}_t$, hence no requirement that a model’s structural operation always yield the value $\boldsymbol{h}(x,y)=x(y)$ for x $\boldsymbol{I}_{t',t}$ and y $\boldsymbol{I}_{t'}$. Again, this is in the spirit of abstractness and philosophical neutrality. But we shall see in a moment that this generality is particularly important for the purposes stated in the Introduction. For surely, we cannot treat quantifiers as connectors *and* interpret all connectors as functions from truth-values to truth-values.

In fact, as I see it this general attitude with respect to the notion of a model has far reaching consequences, for it also seems to permit a uniform treatment of what are sometimes regarded as different *types* of semantical modeling. For example, the semantical analysis of so-called intensional languages is usually supposed to induce a conceptual departure from that of simpler case studies, for in that case the “intended meanings” of a language’s symbols seem to depend on various factors which are disregarded in the simpler accounts. Yet, within the present framework such cases can be dealt with like any other: to account for the relevant factors one only has to refer to the appropriate classes of models, requiring e.g. that the domains of interpretation be not just sets of entities (truth-values for $t=0$, individuals for $t=1$, etc.), but sets of functions ranging over those entities and taking as arguments indices from an appropriate set of *intensional features*. More precisely, for any language we can single out the class of its *indexical* models as consisting of all those models $M=(\boldsymbol{d},\boldsymbol{h},\boldsymbol{I})$ such that, for some set A of “possible features” and some system $\boldsymbol{U} = \boldsymbol{U}_t : t \quad D\boldsymbol{E}$ of universes of “entities”, $\boldsymbol{I}_t = \boldsymbol{U}_t{}^A$ for each $t \quad D\boldsymbol{E}$. And among such models, we can take the ones that best fit our purposes. (The nature of A is in fact liable to further qualifications depending on one’s specific needs. For instance, to account for a certain standard meaning of such modal connectors as ‘necessarily’, ‘possibly’, etc. one may construe A as a set of “possible worlds”; but if the language is also supposed to contain indexical locutions such as demonstratives, personal pronouns, etc., one might prefer thinking of A as a set of ordered pairs consisting of a “possible world” and a “context of use”; and in connection with tensed languages one would perhaps go even further, requiring A to be a set of ordered triples consisting of a “possible world”, a “context of use”, and a “moment of time”. All of this, as I see it, is just a matter of case-by-case specifications.)[8]

It is of course understood that if M is an indexical model relative to a given set of features A, the values of $\boldsymbol{h}$ should in general not depend on A unless the arguments do too. That is, one should have $\boldsymbol{h}(x,y)(a_i)= \boldsymbol{h}(x,y)(a_j)$ whenever $x(a_i)=x(a_j)$ and $y(a_i)=y(a_j)$ $(a_i, a_j \quad A)$. Also, such values should behave coherently with respect to the entities that make up the system of universes $\boldsymbol{U} = \boldsymbol{U}_t : t \quad D\boldsymbol{E}$. Thus, in general one should require that $\boldsymbol{h}(x_i,y)(a)= \boldsymbol{h}(x_j,y)(a)$ if $x_i(a)=x_j(a)$, and $\boldsymbol{h}(x,y_i)(a)=\boldsymbol{h}(x,y_j)(a)$ if $y_i(a)=y_j(a)$. Indexical models satisfying these additional conditions may be referred to as models *rooted* on the couple $(\boldsymbol{U}, A)$. As we now shall see, it is precisely models of this sort that can be employed to provide an interpretation of languages whose vocabulary includes variable-binders.

For concrete illustration, we can refer directly to the examples considered earlier.

EXAMPLE 2.1. If L is a sentential language, then M is a *bivalent sentential model* for L iff $\boldsymbol{I}_0=2$. In particular, suppose L contains only a binary connector '$\downarrow$' as in Example 1.1: then we may say that M is a *standard* sentential model for L iff M interprets '$\downarrow$' as the *joint-denial connector*:

(*) if $\boldsymbol{s}(\beta,t)=\downarrow$, then $\boldsymbol{h}(\boldsymbol{h}(\boldsymbol{d}(\beta,t),x),y)=1-(x \quad y)$ for every $x,y \quad \boldsymbol{I}_0$.

EXAMPLE 2.2. If L is an elementary language with a denumerable set V_L of noun variables, then M is a *bivalent elementary model* for L iff M is rooted on a couple ($\boldsymbol{U}_t: t \quad \boldsymbol{DE}$, A) such that $\boldsymbol{U}_0=2$ and $A=\boldsymbol{U}_1{}^{V_L}$.

Thus, to account for the fact that some L-expressions may involve *variable* symbols of type 1, we may just require M to be an "indexical" model relative to the set of "features" $\boldsymbol{U}_1{}^{V_L}$, i.e. expressions may be interpreted as functions of the relevant value assignments $a{:}V_L \rightarrow \boldsymbol{U}_1$. In particular, suppose L includes a binary quantifier ' $\downarrow,v$ ' for each variable $v \quad V_L$, as in Example 1.2: then we may say that M is a *standard* elementary model for L iff the following additional conditions hold for every $\beta,t \quad D\boldsymbol{s}$, every $v \quad V_L$ and every $x,y \quad \boldsymbol{I}_0$:

(a) if $\boldsymbol{s}(\beta,t) \quad V_L$, then $\boldsymbol{d}(\beta,t)(a)=\boldsymbol{d}(\beta,t)(b)$ for all $a,b \quad A$
(b) if $\boldsymbol{s}(\beta,t)=v$, then $\boldsymbol{d}(\beta,t)(a)=a(v)$ for all $a \quad A$
(c) if $\boldsymbol{s}(\beta,t)= \downarrow,v$, then $\boldsymbol{h}(\boldsymbol{h}(\boldsymbol{d}(\beta,t),x),y)(a)=\bigcap\{1-(x(a(v/u)) \quad y(a(v/u))): u \quad \boldsymbol{U}_1\}$ for all $a \quad A$

(i.e., iff M interprets ' $\downarrow,v$ ' as the *joint denial quantifier* binding the corresponding variable v, every constant symbol denoting *constant* functions of the right sort).

EXAMPLE 2.3. If L is a full categorial language with a denumerable set $V_{L\ ,\ t}$ of variables for each type $t \quad T$, then M is a *bivalent categorial model* for L iff M is rooted on a couple ($\boldsymbol{U}_t{:}t \quad T$, A) such that $\boldsymbol{U}_0=2$ and $A=\prod \ \boldsymbol{U}_t{}^{V_{L,t}}{:}\ t \quad T$.

Thus, again, since the expressions of L may involve *variable* symbols of various types, we require here that M be an "indexical" model based on the appropriate set of "features": sequences of value assignments $a_t{:}V_{L\ ,\ t} \rightarrow \boldsymbol{U}_t$ for each $t \quad T$. In particular, suppose L includes an abstractor λ,v,t' for each $v \quad \bigcup\{V_{L\ ,\ t}{:}\ t \quad T\}$ and each $t' \quad T$, as in Example 1.3: then we may say that M is a *standard* categorial model for L iff the following additional conditions hold for every $\beta,t \quad D\boldsymbol{s}$ and every $x \quad \boldsymbol{I}_{t'}$:

(a) if $\boldsymbol{s}(\beta,t) \quad \bigcup\{V_{L\ ,\ t}{:}t \quad T\}$, then $\boldsymbol{d}(\beta,t)(a)=\boldsymbol{d}(\beta,t)(b)$ for all $a,b \quad A$
(b) if $\boldsymbol{s}(\beta,t)=v$, then $\boldsymbol{d}(\beta,t)(a)=a_{\tau(v)}(v)$ for all $a \quad A$
(c) if $\boldsymbol{s}(\beta,t)= \lambda,v,t'$, then $\boldsymbol{h}(\boldsymbol{h}(\boldsymbol{d}(\beta,t),x),y)(a)=x(a(\tau(v)/a_{\tau(v)}(v/y(a))))$ for all $y \quad \boldsymbol{I}_{\tau(v)}$ and all $a \quad A$, where $\tau(v)$ is the type of v

(i.e. iff M interprets each ' λ,v,t' ' as the *functional abstractor* of type $t',\ t,t'$ binding the corresponding variable v, every constant symbol denoting *constant* functions of the right sort).

VALUATIONS

At this point we just need to build a bridge between languages and models in the obvious way, *via* the notion of a valuation. This is straightforward, for it suffices to note that every language is homomorphic to any of its models. That is, if $L=(\boldsymbol{s},\boldsymbol{g},\boldsymbol{E})$ is a language and $M=(\boldsymbol{d},\boldsymbol{h},\boldsymbol{I})$ a corresponding model, then there exists a unique map $f\colon \bigcup R\boldsymbol{E} \rightarrow \bigcup R\boldsymbol{I}$ such that (i) $f(\boldsymbol{s}(\beta,t))=\boldsymbol{d}(\beta,t)$ for each β,t $D\boldsymbol{s}$, and (ii) $f(\boldsymbol{g}(y,z))=\boldsymbol{h}(f(y),f(z))$ for each y,z $D\boldsymbol{g}$ [9]. This means that M is sure to provide *all* the information that is needed in order to assign a value to each expression of L: $\boldsymbol{d}$ assigns a denotation to each basic expression, and $\boldsymbol{h}$ tells us how to compute the denotation of a compound expression *given* the denotation of its component parts. Since all this information is perfectly reflected in the homomorphism that relates L and M, such a map —call it V—is the natural candidate for the role of a valuation of L on M: it yields the value $V(x)=\boldsymbol{d}(\beta,t)$ for $x=\boldsymbol{s}(\beta,t)$, and the value $V(x)=\boldsymbol{h}(V(y),V(z))$ for $x=\boldsymbol{g}(y,z)$.

DEFINITION 3. The *valuation* of a language L on a corresponding model M is the unique homomorphism V between L and M.

Just for the sake of completeness, one may refer to our previous examples to verify that V behaves normally when certain standard conditions are satisfied.

EXAMPLE 3.1. If L is a sentential language with a binary connector $\downarrow$ (as in Example 1.1) and M a standard sentential model for L (as in Example 2.1), then the usual conditions hold for all A,B $\boldsymbol{E}_0$:

(a)	$V(A\downarrow B)=1$	iff	$V(A)=0$ and $V(B)=0$
(b)	$V(\neg A)=1$	iff	$V(A)=0$
(c)	$V(A\quad B)=1$	iff	$V(A)=1$ or $V(B)=1$
(d)	$V(A\quad B)=1$	iff	$V(A)=1$ and $V(B)=1$
$\vdots$	$\vdots$		

EXAMPLE 3.2. If L is an elementary language with a binary quantifier ‘ $\downarrow,v$ ’ for each variable v V_L (as in Example 1.2) and M is a standard elementary model for L rooted on a pair ($\boldsymbol{U}_t : t$ $D\boldsymbol{E}$, $\boldsymbol{U}_1{}^{V_L}$) (as in Example 2.2) then the following usual conditions hold for all A,B $\boldsymbol{E}_0$, all v V_L, and all a $\boldsymbol{U}_1{}^{V_L}$:

(a)	$V(A\overset{v}{\downarrow}B)(a)=1$	iff	$V(A)(a(v/u))=0$ and $V(B)(a(v/u))=0$ for all u $\boldsymbol{U}_1$
(b)	$V(\neg A)(a)=1$	iff	$V(A)(a)=0$
(c)	$V(A\quad B)(a)=1$	iff	$V(A)(a)=1$ or $V(B)(a)=1$
(d)	$V(A\quad B)(a)=1$	iff	$V(A)(a)=1$ and $V(B)(a)=1$
$\vdots$	$\vdots$		
(e)	$V(\quad vA)(a)=1$	iff	$V(A)(a(v/u))=1$ for some u $\boldsymbol{U}_1$
(f)	$V(\quad vA)(a)=1$	iff	$V(A)(a(v/u))=1$ for every u $\boldsymbol{U}_1$

EXAMPLE 3.3. If L is a full categorial language with a denumerable set $V_{L,t}$ of variables for each $t \quad T$ and an abstractor λ, v, t' for each variable v and each type t' (as in Example 1.3) and M is standard categorial model for L rooted on a pair ($\boldsymbol{U}_t$: $t \quad T$, $\prod \boldsymbol{U}_t^{V_{L,t}}$: $t \quad T$) (as in Example 2.3), then the following conditions hold for all $t, t', t'' \quad T$, all $v \quad \bigcup\{V_{L,t} : t \quad T\}$, all $A \quad \boldsymbol{E}_{t'}$, all $x \quad \boldsymbol{E}_t$, all $y \quad \boldsymbol{E}_{t,t',t''}$, and all $z \quad \boldsymbol{E}_{t,t'}$:

(a) $V(\lambda v A(v)) = V(A)$
(b) $V(\mathbf{I}_t(x)) = V(x)$
(c) $V(\mathbf{K}_{t,t'}(x, A)) = V(x)$
(d) $V(\mathbf{S}_{t,t',t''}(y, z, x)) = V(y(x, z(x)))$ [10].

DISCUSSION

With Definition 3 our semantic framework is complete. It is in fact a semantic framework, not a semantic theory—a framework wherein each of a variety of semantic theories falls rather naturally: as the particular form of the language and the specific conditions on the relevant models are varied, the theory varies.

Now I believe this is rather close to what Ajdukiewicz had in mind. What is attractive in the conception of semantics originated with his work is precisely that it leads to a general framework fixed in advance, rather than a theory to be worked out anew in each case: we only have to specify a type assignment for the language's symbols (eventually along with a suitable structural operation) and then specify which, among the indefinitely many structures that give a homomorphic interpretation of the language, are to count as "admissible" models. The account outlined here is a bit more abstract than Ajdukiewicz's, hence more general. But it is precisely this trade-off between abstractness and generality that shows the enormous potentiality of Ajdukiewicz's original approach to semantics. I have on purpose refrained from considering extending the basic notion of a categorial language, as suggested in more recent literature [11], to emphasize this aspect.

I would now like to conclude with a couple of remarks on the resulting overall picture, both of which tie in with this general perspective.

The first remark concerns specifically the fact that also languages involving variable-binding operators seem to fit in naturally with the proposed account. I don't know whether this could by itself justify a claim of generality [12]. Certainly the fact that we can analyze a language with quantifiers by the simple functor/argument schema is not unattractive, but that is only one way of putting it: there are several alternatives available that do the job as well. What I think is more interesting is the fact that also a language with, say, a lambda operator can be treated in the same fashion. There is a rather widespread belief that within a purely categorial framework one cannot go very far in the analysis of complex linguistic structures, though much can be done with the additional help of the functional abstraction operator: arguably there is a strong connection between abstraction and certain types of syntactic/semantic transformations that allow one to come near to the surface of most natural language sentences (probably as close as we need for most purposes) [13]. From this point of view, then, the present account becomes fairly attractive and justifies a different, more positive attitude towards categorial investigations.

The second remark is also related to the above, but it bears on a more general issue in the philosophy of semantics. Very often, the necessity of using variables and variable-binding operators is taken to have important consequences with respect to the role of semantic theorizing. In particular, it is often taken to imply that certain crucial aspects of our language lie "outside", or perhaps "above", the realm of semantics: insofar as such symbols *must* receive a fixed interpretation, there must be a corresponding set of *logical* principles that we need "keep in mind" when we come to spell out the semantic framework we want to use. This is rather typical, for instance, in the customary model-theoretic way of presenting first-order logic, where the "meaning" of bound variables and quantifiers is characterized only indirectly through the recursive definition of the truth-value of a sentence. And the same applies to lambda-equipped categorial languages, where the "meaning" of the lambda operator is usually fixed during a recursive definition of the value of an expression. To a certain extent this way of proceeding reflects neither less nor more than a natural need for short cuts: if we are not going to consider other ways of interpreting those symbols, there is no need to do otherwise. But the question is how—and somehow even whether—one could do otherwise. The question is whether it is possible in principle to do semantics without also doing logic.

As I see it, there has never been much clarity on this point [14]. Nevertheless my suggestion is that within the approach outlined here the question seems to admit a simple answer. Insofar as quantifiers, functional abstractors, and the like can be treated as ordinary functors, one has good grounds to regard these symbols as being *on a par* with all the others—hence good grounds to treat logics as true *theories* in the usual semantic sense. To treat something as a "logical" word is to select a certain class of models as the only "logically" admissible ones, just as to treat something as, say, a "mathematical" symbol is to select a certain class of models as the only "mathematically" admissible ones. Earlier I exemplified this point with reference to the usual interpretation of the conjunction connective and the multiplication operator. Now the suggestion is that a similar argument could be given for any other logical word, witness the examples given in the previous sections. Classical sentential logic can be obtained by selecting a sentential language along with the class of its standard bivalent models; classical elementary logic can be obtained by selecting an elementary language along with the class of its standard bivalent models; the classical typed lambda-calculus can be obtained by selecting a full categorial language along with the class of all of its standard bivalent models; and so on. Including, as I mentioned, logics usually classified as "intensional" (e.g. modal or tense logics).

Further generalizations can also be considered. For instance, as it is the semantic framework outlined here embeds the requirement that every model provide a *homomorphic* interpretation of the corresponding language, which reflects the standard assumption that a model must be made of well-defined, sharp-cut entities, neatly linked to one another and to the language's expressions in a univocal way. Philosophically—as well as for practical reasons—one could find this a serious limitation in the scope of a semantic theory, particularly in the spirit of the above considerations. There is no *a priori* semantic reason to rule out the possibility that (our representation of) what we talk about may involve "gaps" and/or "gluts" of various sorts. Indeed, since there is no general linguistic criterion for incompleteness, there is no general way that incompletenesses can be ruled out without weeding out a variety of unproblematic cases as well. And since there is no general decision procedure for inconsistency, there is no general and effective way that inconsistencies can be ruled out without rendering a great deal of perfectly innocent thinking impossible. Without going into any details here, it is therefore worth remarking that the approach described above can rather easily be generalized so as to cover such cases as well. This would of course make things a bit more complicated. For there is no homomorphism between a language and an incomplete model, while there can be more than one between a language and an inconsistent model. Still, the point remains that as long as we can define a general notion of a valuation linking arbitrary languages and models thereof, be it or not a "pure" homomorphic valuation[15], a general semantic framework is available relative to which a variety of cases can be treated in a uniform fashion. And I think this is definitely in the spirit (methodologically, if not philosophically) of Ajdukiewicz's original insights.

NOTES

[1]Ajdukiewicz (1935) spoke of *semantic* categories, following the terminology of Husserl's *Untersuchungen* (1900-1901) and Leśniewski's *Grundzüge* (1929). Presumably this is because he was already thinking of their semantic role, but I find that a bit misleading. In the following I shall mainly speak of types.

[2]These are indeed the two main primitive categories Ajdukiewicz had in mind, though he did not rule out the possibility of different choices. Some authors, for instance Lewis (1970), have insisted on a third one, Common Noun, and others have considered dispensing with Proper Noun in favour of Noun Phrase, or even Verb Phrase.

[3]I shall use standard set-theoretic notation and terminology: note only that if f is a function, I write '$\mathcal{D}f$' and '$\mathcal{R}f$' for the domain and the range of f, respectively, while '$f(x/y)$' denotes the function exactly like f except that its value at x is y.

[4]Since we define derived types by means of a binary operation, we can of course speak of

"monadic" functors only. However, it is a well-known fact that this implies no loss of generality: for whenever $n > 1$ and $t_1, \dots, t_n, t_{n+1}$ are types, the derived type $t_1, t_2, \dots, t_n, t_{n+1} \dots$ may be used to represent the category of those "*n*-adic" functors that combine with *n*-tuples of expressions of type $t_1, t_2, \dots, t_n$ (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, our relying on imposes no significant restriction on the class of possible functors. (To illustrate, the English word '*and*' is a dyadic functor that makes a sentence out of two sentences; but it can equally be regarded as a monadic functor that, when applied to a sentence, produces an expression that behaves again as a monadic functor: a functor that makes a sentence out of a sentence). The idea goes back to Schönfinkel (1924) and reflects the set-theoretic isomorphism $A^{B_1 \times B_2 \times \dots B_{n+1}} \approx (\dots((A^{B_1})^{B_2})\dots)^{B_{n+1}}$.

[5]Although we speak of *the* type of a symbol—and consequently of an expression—it could be argued that this is only an apparent limitation: it is not difficult to imagine words belonging to more than one syntactic category, but we can always deal with such cases by treating them as distinct symbols with a common "surface realization". (For instance, the word '*light*' in English may seem to function as either a noun or an adjective, but we can also deal with this peculiarity the way common dictionaries do: by distinguishing between a word '$light_1$' and a word '$light_2$', to be classified into two distinct syntactic categories). Apparently the point was made by Ajdukiewicz himself, though several authors have later proposed alternative accounts.

[6]The requirement that $Q_{L,n} \times V_L = \{ x,y : x,y \quad CON_{L,n}\}$ is of course but one way of securing syntactic precision. Note also that in a similar fashion one could for instance treat descriptors as suitable "structured" subnectors: given sequences $D_{L,n}$ disjoint from $R\boldsymbol{s}$ so that $D_{L,n} \times V_L = \{ x,y : x,y \quad SUB_{L,n}\}$, one could speak of a pair of the form $D_{L,n,i}, v$ as a *n-ary descriptor binding* the variable v. In the present context, I shall not go into languages with descriptors to avoid certain well-known complications in their semantics.

[7]The requirements that each domain of interpretation $\boldsymbol{I}_t$ be *non-empty*, that $\boldsymbol{d}$ be a *total function* on $D\boldsymbol{s}$, and that $\boldsymbol{h}$ be a *total operation* on $\bigcup\{\boldsymbol{I}_{t',t} \times \boldsymbol{I}_{t'} : t', t',t \quad D\boldsymbol{I}\}$ reflect a major assumption of standard semantics, namely that each model must provide a *complete* and *consistent* interpretation of the language. It is of course possible to give up such requirements so as to admit incomplete and/or inconsistent structures as models *bona fide*. For instance, allowing $\boldsymbol{d}$ to be a *partial* function would correspond to a more liberal attitude with respect to the possibility of non-denoting symbols; allowing it to be a *relation* would leave room for symbols with more than one denotations; etc. Although this is in the spirit of greater generality, in the present context such a departure from standard semantics would take us too far. A fuller account may be found in a forthcoming work entitled *Universal Semantics*.

[8]I think this is a rather natural exploitation of some basic ideas of Montague Grammars, particularly of the treatment in Montague (1970). Note of course that one could also take the notion of an indexical model as fundamental, regarding all other models as models indexed by a feature's singleton.

[9]That f is unique is easily seen, provided $R\boldsymbol{g} \cap R\boldsymbol{s} = 0$. Obviously $f(x)$ is defined if x is a symbol. And if $f(y)$ and $f(z)$ are both defined, then $f(\boldsymbol{g}(y,z)) = \boldsymbol{h}(f(y), f(z))$ is also defined, since $\boldsymbol{h}$ is a function.

[10] For readability, I use here a notation of the form '$(w(w_1, \dots, w_n))$' as an abbreviation for '$\boldsymbol{g}(\dots(\boldsymbol{g}(\boldsymbol{g}(w, w_1), w_2),\dots), w_n)$'.

[11] The first substantial refinements or extensions of the pure categorial paradigm go back to the work of Bar-Hillel (1960) and Geach (1970). Some indications of the current trends may be found in Bach (1984), Oehrle, Bach and Wheeler, eds. (1988) and Buszkowski, Marciszewski and van Benthem, eds. (1988).

[12] Certain aspects of the proposed account might still sound unpalatable. For instance, as Peter Simons has pointed out to me, there is no way one can single out an elementary language where "vacuous" quantification is forbidden. However, I am inclined to regard such limitations as symptomatic of a more general problem concerning the generative power of Definition 1, typical of virtually every categorial grammar and independent of how one feels about variable-binding through functors. For instance, there is no way one can single out a sentential or elementary language where "useless" conjunctions such as $A \quad A$ are forbidden. From this point of view, the corresponding general solution would simply involve redefining the structural operation $\boldsymbol{g}$ as required: rather than a *total* function on $\bigcup\{\boldsymbol{E}_{t',t} \times \boldsymbol{E}_{t'} : t',\ t',t \quad D\boldsymbol{E}\}$, it should be allowed to be a *partial* function (of some sort). Thus, for instance, an elementary language where $\boldsymbol{g}(\quad \downarrow v\ , A)$ is defined only if the noun variable v is a constituent symbol of the sentence A would be a language with no "vacuous" quantification.

[13] The work done in the tradition of Cresswell (1973) is most indicative of this approach.

[14] The question is not whether one can live without any logic, as it were. For of course, one could always say that semantics is set theory, and *that* surely embeds quite a bit of logic.

[15] In my *Universal Semantics*, I propose a general solution based on an extension of Van Fraassen's notion of a supervaluation. An alternative approach has been proposed in Muskens (1989).

REFERENCES

Ajdukiewicz K. (1935), 'Die syntaktische Konnexität', *Studia Philosophica* 1, 1-27 (Eng. trans. by H. Weber: 'Syntactic Connexion', in S. McCall (ed.), *Polish Logic 1920–1939*, Oxford: Clarendon Press, 1967, pp. 207-231).

Bach E. (1984), 'Some Generalizations of Categorial Grammars', in F. Landman and F. Veltman (eds.) *Varieties of Formal Semantics*, Dordrecht/Cinnaminson: Foris, pp. 1-23.

Bar-Hillel Y. (1960), 'On Categorial and Phrase-Structure Grammars', *Bulletin of the Research Council of Israel* 3, 1-16.

Buszkowski W., Marciszewski W., van Benthem J., eds. (1988), *Categorial Grammar*, Amsterdam/Philadelphia: John Benjamin.

Church A. (1940), 'A Formulation of the Simple Theory of Types', *The Journal of Symbolic Logic* 5, 55-68.

Cresswell M. J. (1973), *Logics and Languages*, London: Methuen.

Geach P. T. (1970), 'A Program for Syntax', *Synthese* 22, 3-17.

Husserl E. (1900-1901), *Logische Untersuchungen*, Halle: Max Niemeyer (2nd ed. 1913/21; Eng. trans. by J. N. Findlay, *Logical Investigations*, London: Routledge and Kegan Paul, 1970).

Leśniewski S. (1929), 'Grundzüge eines neuen Systems der Grundlagen der Mathematik', *Fundamenta Mathematicae* 14, 1-81 (Eng. trans. by M. P. O'Neil: 'Fundamentals of a New System of the Foundations of Mathematics', in S. Leśniewski, *Collected Works*, ed. by S. J. Surma, J. T. Srzednicki, D. I. Barnett, and V. F. Rickey, Dordrecht/Boston/London: Kluwer/Nijhoff, 1992, Vol. 1, pp. 129-173).

Lewis D. K. (1970), 'General Semantics', *Synthese* 22, 18-67.

Montague R. (1970), 'Universal Grammar', *Theoria* 36, 373-398.

Muskens, R. (1989), *Meaning and Partiality*, Ph.D. Thesis, Katholieke Universiteit Brabant.

Oehrle R., Bach E., Wheeler D., eds. (1988), *Categorial Grammars and Natural Language*, Dordrecht/Boston: Reidel.

Schönfinkel M. (1924), ‘Über die Bausteine der mathematischen Logik’, *Mathematische Annalen* 92, 305-316 (Eng. trans. by S. Bauer-Mengelberg, ‘On the Building Blocks of Mathematical Logic’, in J. van Heijenoort (ed.), *From Frege to Gödel: A Sourcebook in Mathematical Logic, 1879-1931*, Cambridge, Mass.: Harvard University Press, 1967, pp. 355-366.

Van Fraassen B. C. (1966), ‘Singular Terms, Truth-Value Gaps, and Free Logic’, *The Journal of Philosophy* 63, 481-95.

Whitehead A. N, Russell B. A. W. (1910-1913), *Principia Mathematica*, Cambridge: Cambridge University Press.