GILA SHER

TRUTH, LOGICAL STRUCTURE, AND COMPOSITIONALITY

ABSTRACT. In this paper I examine a cluster of concepts relevant to the methodology of truth theories: 'informative definition', 'recursive method', 'semantic structure', 'logical form', 'compositionality', etc. The interrelations between these concepts, I will try to show, are more intricate and multi-dimensional than commonly assumed.

1. TRUTH AND RECURSION

In his 1933 paper, "The Concept of Truth in Formalized Languages", Tarski drew a tentative connection between the definition of truth (for a given language) and the recursive method:

> If the language investigated only contained a finite number of sentences fixed from the beginning, and if we could enumerate all these sentences, then the problem of the construction of a correct definition of truth would present no difficulties. For this purpose it would suffice to complete the following scheme: $x \in Tr$ *if and only if either* $x = x_1$ *and* p_1, *or* $x = x_2$ *and* $p_2, \ldots$ *or* $x = x_n$ *and* p_n, the symbols 'x_1', 'x_2', ..., 'x_n' being replaced by structural-descriptive names of all the sentences of the language investigated and 'p_1', 'p_2', ..., 'p_n' by the corresponding translation of these sentences into the metalanguage. But the situation is not like this. Whenever a language contains infinitely many sentences, the definition constructed automatically according to the above scheme would have to consist of infinitely many words, and such sentences cannot be formulated either in the metalanguage or in any other language. Our task is thus greatly complicated.
>
> The idea of using the recursive method suggests itself. [188–189]

Tarski's idea involves, however, an indirect use of the recursive method:

> Among the sentences of a language we find expressions of rather varied kinds from the point of view of logical structure, some quite elementary, others more or less complicated. It would thus be a question of first giving all the operations by which simple sentences are combined into composite ones and then determining the way in which the truth or falsity of composite sentences depends on the truth or falsity of the simpler ones contained in them. Moreover, certain elementary sentences could be selected, from which, with the help of the operations mentioned, all the sentences of the language could be constructed; these selected sentences could be explicitly divided into true and false, by means, for example, of partial definitions of the type described above. In attempting to realize this idea we are however confronted with a serious obstacle. Even a superficial analysis of Defs. 10–12 of

Synthese **126:** 195–219, 2001.

Section 2 shows that in general composite sentences are in no way compounds of simple *sentences*. Sentential functions do in fact arise in this way from elementary functions . . . ; sentences on the contrary are certain special cases of sentential functions. In view of this fact, no method can be given which would enable us to define the required concept directly by recursive means. The possibility suggests itself, however, of introducing a more general concept which is applicable to any sentential function, can be recursively defined, and, when applied to sentences, leads us directly to the concept of truth. These requirements are met by the notion of *satisfaction of a given sentential function by given objects* . . . [189]

Referring to the special language of the calculus of classes, Tarski specifies:

We shall use a recursive method in order to formulate a general definition of satisfaction of a sentential function by a sequence of classes, which will include as special cases all partial definitions of this notion that are obtained from the given scheme in the way described above. For this purpose it will suffice, bearing in mind the definition of sentential function, to indicate which sequences satisfy the inclusions '$x_k \subseteq x_l$' and then to specify how the notion we are defining behaves when the three fundamental operations of negation, disjunction, and universal quantification are performed on sentential functions. [192]

I will not repeat Tarski's formal account of the definition of truth for the language of the calculus of classes here, but assuming familiarity with this account, I will encapsulate it in the following way:

1.1. *Definition of Truth for* L_c

Let L_c be the formalized language of the calculus of classes whose primitive symbols are the individual variables 'x_1', 'x_2', . . . , the non-logical constant '$\subseteq$', the logical constants '$\sim$', 'V', '$\forall$', and the auxiliary symbols '(' and ')'.

Let the well-formed formulas and sentences of L_c be defined in the usual way, and let the meta-language of L_c, ML_c, be such that if s is a symbol of L_c, then $\underline{\underline{s}}$ is a canonical name of s in ML_c and $\underline{s}$ is a designated symbol of ML_c having the same meaning (or function) as s.[1]

1.1.1. *Recursive Definition of Satisfaction*

Let g be an assignment function for L_c, i.e., g assigns to each individual variable $\underline{\underline{x}}_i$ of L_c a class, $\underline{x}_i$, in the "universe" of classes. Then:

(a) If $\underline{\underline{x}}_i$ and $\underline{\underline{x}}_j$ are variables of L_c, then $\underline{\underline{x}}_i \underline{\underline{\subseteq}} \underline{\underline{x}}_j$ is satisfied by g iff $\underline{x}_i \underline{\subseteq} \underline{x}_j$.

(b) If Φ is a sentential function of L_c, then $\underline{\underline{\sim}}\Phi$ is satisfied by g iff $\underline{\sim}$ [Φ is satisfied by g].

(c) If Φ and Ψ are sentential functions of L_c, then $\Phi \underline{\underline{V}} \Psi$ is satisfied by g iff [Φ is satisfied by g] $\underline{V}$ [Ψ is satisfied by g].

(d) If Φ is a sentential function and $\underline{x}$ a variable of L_c, then $\underline{\forall x}\Phi$ is satisfied by g iff $\underline{\forall} g'[g'$ differs from g at most in its assignment to $\underline{x} \underline{\rightarrow} \Phi$ is satisfied by $g']$.

1.1.2. *Definition of Truth*
σ is a true sentence of L_c iff σ is a sentence of L_c and every assignment g for the variables of L_c satisfies σ.

2. RECURSION AND EXPLANATION

The recursive method makes two important contributions to the theory of truth: (A) It solves the technical problem of defining truth for a language with infinitely many sentences by finite means (once semantic values are assigned to the atomic elements). (B) It provides a template for an informative definition of truth. The informative, or explanatory, power of the recursive method was passed over by Tarski, but even a superficial comparison of his recursive definition of truth for L_c and a non-recursive alternative based on his suggestion in the first citation –

The sentence σ of L_c is true iff $[[\sigma = \underline{\forall x}\,(x \subseteq x)$ and $\underline{\forall x_1}(x_1 \subseteq x_1)]$ or $[\sigma = \sim \underline{\forall x}\,(x \subseteq x)$ and $\sim \underline{\forall x_1}(x_1 \subseteq x_1)]$ or $[\sigma = \underline{\forall x}\sim(x \subseteq x)$ and $\underline{\forall x_1}\sim(x_1 \subseteq x_1)]$ or $[\sigma = \underline{\forall x}\,\underline{\forall x}\,(x \subseteq x\ V x \subseteq x)$ and $\underline{\forall x_1}\underline{\forall x_2}(x_1 \subseteq x_2 V x_2 \subseteq x_1)]$ or $[\sigma = \underline{\forall x}\,\underline{\forall x}\,(x \subseteq x\ V \underline{\forall x}\,\underline{\forall x}\,(x \subseteq x\ V x \subseteq x))$ and $\underline{\forall x_1}\underline{\forall x_2}(x_1 \subseteq x_2 V \underline{\forall x_3}\underline{\forall x_4}(x_2 \subseteq x_1 V x_3 \subseteq x_4) V \underline{\forall x_5}(x_5 \subseteq x_2))$ or $\ldots]$ –

shows that the former is superior to the latter not just in being finite, but in revealing the *principles* underlying the truth and falsity of sentences of L_c.

We can distinguish two ways in which the recursive method contributes to an explanatory definition of truth: (A) The recursive method allows us to reduce the task of formulating an informative definition of truth for a given language L to that of formulating an informative definition of truth (or satisfaction) for the structurally simple sentences (or formulas) of L. (B) The recursive method allows us to give an informative account of the contribution of structural complexity to truth, i.e., of the way in which the structural features of sentences of L affect their truth value.

While the two types of explanation may work in tandem, in practice they have led to two different conceptions of a theory of truth: a *reductionist conception*, exemplified by Field (1972), and a *structuralist conception*, exemplified by Davidson (1965, 1967, 1970, etc.). On the reductionist approach the main task of an informative theory of truth is to explain the satisfaction conditions of the atomic formulas of a given language or, more

basically, the reference and application conditions of the primitive, nonlogical singular terms and predicates of that language. On the structuralist approach the main task is to explain the contribution of structural complexity to truth. On the first approach the burden of explanation falls on the basic, non-recursive entries in the definition of satisfaction; on the second on the structural, recursive entries. I have discussed this distinction and the difficulties involved in the first approach in Sher (1998/9 and 1999). Here I would like to investigate the nature of structural definitions of truth.

3. TRUTH AND STRUCTURE

What structural complexity is relevant to truth? Three answers that are sometimes conflated are: (a) *logical structure*, (b) *iterative structure*, and (c) *compositional structure*.[2] One source of this conflation may be Tarski's 1933 article, since it is a characteristic feature of the definition of truth presented in this article that the three answers listed above coincide for it. The set of complex structures taken into account by Tarski's definition of truth for L_c = the set of logically complex structures of L_c = the set of iteratively complex structures of L_c = the set of compositionally complex structures of L_c. Tarski's 1933 presentation, however, cannot serve as a basis for an answer to our question. One difficulty with Tarski's presentation is its restriction to a relatively narrow category of languages, namely that of the "formalized languages of the deductive sciences". This restriction is unmotivated from our perspective (Tarski's motivation for excluding other languages has nothing to do with the relevance or irrelevance of nonlogical structural features to truth), but it creates an unavoidable bias towards the first answer. Moreover, even with respect to logical languages the question cannot be adequately answered based on Tarski's presentation. The reason is simple: Tarski did not offer a general formulation of his theory, choosing to demonstrate its workings by a single, and exceedingly simple, example instead. This has led to numerous misunderstandings of his theory, including the claim that he did not develop a general theory of truth at all (even for logical languages). That Tarski did regard himself as developing a general theory of truth (for the designated range of languages), or a general method for defining truth (for this range of languages), is, however, clear from the following passage:

> For an extensive group of formalized languages it is possible to give a method by which a correct definition of truth can be constructed for each of them. The general abstract description of this method and of the languages to which it is applicable would be troublesome and not at all perspicuous. I prefer therefore to introduce the reader to this method in another way. I shall construct a definition of this kind in connection with a particular concrete

language and show some of its important consequences. The indications which I shall then give . . . will, I hope, be sufficient to show how the method illustrated by this example can be applied to other languages of similar logical construction. [Tarski 1933: 167–8]

By 'other languages of similar logical construction' Tarski means languages differing from the language of the calculus of classes in one respect only, namely, their order (i.e., having higher-order variables). If we think of *Tarski's method* as the collection of principles common to all the definitions of truth he would have given to "eligible" languages, then an adequate account of his method would require a non-trivial generalization of his example.[3] Moreover, Tarski's paper was written at a time when our understanding of logical structure was in its infancy. Since then new logical structures have been discovered (constructed, recognized) and these are not represented in Tarski's example. In fact, L_c is so simple that even syntactic structures widely known in the 1930's (e.g., complex singular terms) are not represented in it.

As for the three answers, it is clear that their coincidence in Tarski's example is accidental. Logical, iterative and compositional structures differ extensionally as well as intensionally: Intensionally, logicality is primarily a philosophical notion, while iterativity is a purely technical notion and compositionality a methodological notion. Extensionally, some logical structures are not iterative (e.g., the structure '. . . = —'), while some iterative structures are not logical (e.g., the structure 'father of (father of (. . .)'). The case of compositionality is more involved: not all compositional structures are logical, and not all iterative operators are compositional; but whether all logical structures are compositional is a matter of controversy (see below).

4. TRUTH AND LOGICAL STRUCTURE

The notion of logical structure is a philosophically loaded notion: logical structure gives rise to a special type of inference, playing a special role in all branches of knowledge, and our understanding of this structure is tied up with our understanding of that inference. *Inference*, in general, is a procedure for "moving" from statements (existent items of knowledge) to statements (new items of knowledge) without loss of *truth*; *logical* inference rests on a special connection between *logical structure* and truth. To explain the contribution of logical structure to truth is to explain this connection, and to develop an informative theory of the contribution of logical structure to truth is to delineate the general principles underlying this connection. Now, what distinguishes logical structure from other

types of structure is the type of term appearing in it. Logical structure is the pattern generated by "highlighting" the logical constants of a given well-formed expression and "dimming" its non-logical constants, and to understand the connection between logical structure and truth we must therefore understand (i) "who" the logical terms are, and (ii) how they behave semantically.

Logical terms are sometimes characterized syntactically, but the difference between logical and non-logical terms – as well as between different logical terms – is not syntactic: '*x* is identical to *y*' and '*x* is a sibling of *y*' are of the same syntactic category but the first is logical while the second is not; 'or' and 'and' are also syntactically alike, but they denote two altogether different logical operators. What distinguishes logical terms from non-logical terms – and logical terms from each other – is a feature of *content*: the relation denoted by '*x* is identical to *y*' is *formal* (in the sense of not distinguishing between formally identical, or *isomorphic*, argument-structures),[4] while that denoted by '*x* is a sibling of *y*' is *biological*. Similarly, the difference between 'John loves Mary **or** John loves Jane' and 'John loves Mary **and** John loves Jane' is a difference in *content*, not a difference in *syntax*. To determine what type of term is logical is to determine what type of content is logical, and such a determination is philosophical rather than a grammatical.

It is common to draw a sharp distinction between form and content, identifying logic with form and other sciences with content, but the notion of logical form is induced by that of logical content. A theory of the contribution of logical structure to truth is a theory of the influence of a particular type of content on truth: logical content as distinct from modal, physical, biological, and other types of content. The three main tasks of an informative theory of the influence of logical content on truth are: (A) Formulate an informative criterion of logical constants (based on their content); (B) Give a systematic characterization of the satisfaction conditions of logical constants based on (A) and show how their influence on truth extends to complex logical structures; (C) Explain how the special connection between logical structure and truth gives rise to logical inference. Although Tarski's 1933 theory does not constitute, as it stands, an informative theory of the contribution of logical structure to truth in our sense, it does provide, together with his 1936 account of logical consequence and his 1966 criterion of logical constants, the basic building blocks for such a theory.[5]

5. TRUTH AND ITERATIVE STRUCTURE

Unlike the notion of logical structure, which is largely philosophical, the notion of iterative structure is purely technical. An iterative structure is generated by applying an iterative term to an eligible expression (n-tuple of expressions) in a permissible way, where an iterative term is a term whose application to a given expression may be reiterated indefinitely. We may define an iterative term as follows: A unary term t is *(strongly) iterative* iff for any well-formed expression $e = `t(e_1)'$, e is an eligible argument of t; an n-ary term t is (strongly) iterative iff for any well-formed expression $e = 't(e_1, \ldots, e_n)'$, at least one of the e_i's ($1 \leqslant i \leqslant n$) is of the same syntactic type as e, and for any eligible argument $\langle e_1, \ldots, e_n \rangle$ of t, the result of substituting e for any number of e'_i's sharing its syntactic type is an eligible argument of t.

The interest of iterative structure for a definition of truth lies in its productivity: every iterative operator generates infinitely many well-formed expressions, and the task of constructing a definition of truth for a language with iterative operators is therefore infinitistic. A natural device for dealing with iterative structures is the recursive method. The recursive method, as we have seen in Tarski's example, enables us to define truth, satisfaction and/or reference for languages with iterative terms (and a finite, or finitely definable, stock of primitive terms) in a finite number of steps, solving the problem of productivity. Not every language with iterative operators, however, is a suitable candidate for a recursive semantics. For example, a language which fails to satisfy the requirement of unique readability may not be.[6]

A definition of the contribution of *iterative* structure to truth cannot achieve the same level of informativeness as a definition of the contribution of *logical* structure to truth. Iterative terms possess no unity of content, and no philosophically significant type of inference is associated with them. How to handle such terms is a technically important question, but philosophically, a theory of the contribution of iterative structure of truth is of limited significance.

6. TRUTH AND COMPOSITIONALITY

Compositionality is a methodological notion: a constraint on the semantics of a given language and its relation to the syntax of that language. Many philosophers require that an adequate theory of truth be compositional, but some do not. The syntactic notion of compositionality is a relative notion: to say that a theory of truth is concerned with compositional structure is

to say that it is concerned with the kind of structure that is amenable to a compositional semantics.

What is compositional semantics? As of now, the notion of compositional semantics does not have a precise "canonical" definition, but its intuitive meaning is quite clear: *a compositional semantics uniquely determines the semantic value of a complex expression based on (a) the semantic values of its immediate parts, and (b) the way these parts are combined*. The notion of compositional semantics is often identified with that of recursive semantics, and a paradigmatic example of such a semantics is that of (classical) sentential logic. This semantics consists of a pair, $\langle Syn, Sem\rangle$, where *Syn* is an inductive, freely generated syntax, and *Sem* is a recursive definition of truth based on this syntax. Syntactically, each well-formed expression of a given sentential system is freely, i.e., uniquely, generated from a finite number of "atomic" expressions by finitely many (possibly zero) applications of functional iterative operators. Semantically, the definition of truth forms a homomorphic image of the syntactic definition: each atomic sentence is assigned a single truth value, T or F, and for each logical operator there is a rule which uniquely determines the truth value of a sentence formed by it, given the truth values of its immediate arguments. Thus, syntactically, the sentence '$\sim (P\&Q)$' is formed from the atomic sentences P and Q in two steps: (i) application of the operator '&' to the pair $\langle P, Q\rangle$, generating '$(P\&Q)$'; (ii) application of '$\sim$ to '$(P\&Q)$', generating '$\sim (P\&Q)$'. Semantically, the truth value of '$\sim (P\&Q)$' is formed from the truth values of P and Q in parallel steps: (i) application of the rule $V_{\&}$ to the pair $\langle V(P), V(Q)\rangle$, where $V(P/Q)$ is the truth value of P/Q and $V_{\&}$ is the semantic rule associated with '&': $V_{\&}(X, Y) = T$ if $X = Y = T$; otherwise $V_{\&}(X, Y) = F$; (ii) application of the rule $V_{\sim}$ to $V(P\&Q)$, where $V_{\sim}$ is the rule associated with '$\sim$': $V_{\sim}(X) = T$ if $X = F$; otherwise $V_{\sim}(X) = F$'. Other, more complex compositional definitions are Tarski's recursive definition of satisfaction for L_c and the recursive definitions of satisfaction-in-a-model found in standard textbooks of mathematical logic.[7]

7. MUST A DEFINITION OF TRUTH BE COMPOSITIONAL?

The main methodological advantages of a compositional semantics are those indicated in our discussion of a recursive definition of truth in Section I above: (i) a compositional semantics allows us to handle an infinite collection of target expressions by finite means; (ii) a compositional semantics is inherently informative. The first advantage has been widely discussed in the literature and I will not dwell on it at length here: an adequate se-

mantics must take into account the productivity, learnability, intelligibility and communicability of language, and compositionality is said to be either a necessary or a sufficient condition for satisfying this requirement.[8] The second, and less commonly discussed, advantage has to do with the virtues of a compositional theory as an informative theory: (a) A compositional semantics tells us what the basic units of the syntax and the semantics of a given language are, how the syntax and the semantics are related to each other, how complex syntactic/semantic units are related to basic syntactic/semantic units, etc. (Janssen, 1997); (b) A compositional semantics explains the "systematicity" of a given language (Fodor and LePore, 1992); (c) A compositional semantics creates a template for informative entries in the definition of reference, satisfaction and/or truth of a given language.

The last point can be explained by reference to Tarski's definition of truth for L_c. Consider Tarski's recursive entries for the logical connectives '$\sim$' and 'V'. We may read these entries in two ways, as providing informative or uninformative satisfaction conditions for formulas of L_c. The second reading is deflationist. On this reading, what Tarski is saying is that '**not** Φ' is satisfied by g iff Φ is **not** satisfied by g, and 'Φ **or** Ψ' is satisfied by g iff Φ is satisfied by g **or** Ψ is satisfied by g. If the meaning of 'not' or 'or' in the definiendum is unclear (if, for example, it is not clear whether 'not' is bivalent or trivalent, or whether 'or' is inclusive or exclusive), the satisfaction entries for 'not' and 'or' do not remove this unclarity. More importantly, the entries for 'not' and 'or' do not explain in what way the satisfaction conditions for formulas governed by these connectives are *logical*. On the first reading, Tarski's entries for 'not' and 'or' are shortcuts for a more elaborate treatment of the logical connectives (of the type described in Section IV above). This treatment includes (i) a general criterion for logical connectives, (ii) a philosophical explanation/justification of this criterion, (iii) a definition of logical connectives as Boolean truth-functions, (iv) a formulation of the satisfaction conditions of formulas governed by logical connectives by reference to the corresponding truth functions. An informative rendition of Tarski's entire definition of truth for L_c will require an analogous treatment of logical quantifiers.

The question arises, however, whether every theory of truth which fulfills both the desideratum of *manageability* and that of *informativeness* is compositional. We can divide this question into two parts: (A) Can every such theory be given a compositional formulation? (B) Are all theories fulfilling these desiderata compositional? New results due to Janssen (1986) and Zadrozny (1994) establish a positive answer to the first question: For every reasonable semantic theory there is an equivalent compositional

semantics. But the answer to the second question is open. Hintikka and Sandu, for example, have argued that some non-compositional semantic theories are not only highly informative, but are in fact superior to the best available compositional alternatives.

Below I will present two examples of non-compositional definitions of truth. Both definitions account for the learnability of language, and each is highly informative relative to certain significant explanatory goals.

8. A PARTIALLY COMPOSITIONAL DEFINITION OF TRUTH

In the familiar logical languages of standard sentential, 1st-order and higher-order logic, each logical operator can be affixed to a given argument (or n-tuple of arguments) in one way.[9] This is also the case with languages of generalized 1st-order logic (see Mostowski 1957, Lindström 1966, Sher 1991, Chs. 2–4, and elsewhere). In these languages a logical structure is uniquely determined by specifying which logical operator generates it from which expressions. If, however, some of the operators can be combined with "eligible" expressions in more than one way, and if, moreover, different combinations of the same operator with the same expressions yield formulas with different truth conditions, a mode of combination has to be specified. This complication does not significantly affect the task of constructing a systematic semantics for a given language so long as there exists a finite upper bound on the number of modes of combination available to each operator. But if for some operators no such upper bound exists, the semantics is essentially affected. The languages of standard and generalized 1st-order logic with partially-ordered quantifications are of this kind.

The structure of a partially-ordered quantification was first introduced by Henkin (1959). Henkin constructed a language in which standard quantifier-expressions (i.e., expressions of the form 'Qx', where Q is either $\forall$ or $\exists$ and x is an individual variable) can be combined in any partial order to generate a well-formed quantifier-prefix. Given a quantifier-prefix Q and a formula Φ, 'QΦ' is a well-formed formula.[10] Now, we know from the familiar semantics of standard, linear logical languages that quantifier order affects the satisfaction conditions of formulas. Differences in quantifier-order represent differences in the government-dependence relations between quantifier-expressions, and differences involving the government-dependence relations of universal and existential quantifiers have significant semantic ramifications. Thus, a formula of the form

(1) $\quad \forall x \exists y \Phi x y$

is satisfied by an assignment g iff the extension of 'Φxy' under g is a relation with a universal (left) domain, but this condition is neither sufficient nor necessary for g to satisfy

(2) $\exists x \forall y \Phi xy$.

Since there is no finite upper bound to the size of quantifier-prefixes, there is no finite upper limit on the number of modes of combination available to each quantifier in languages of partial ordering. This, together with the fact that differences in quantifier-order are semantically material, imply that the semantics of languages of partial order is essentially different from that of languages of linear order (e.g., the familiar logical languages).

How shall we construct a semantics for a language of partial order? One method which naturally suggests itself is the *Translation Method*: If we can translate all well-formed formulas of the target language into an auxiliary language for which we have a familiar semantics, then we will be able to delineate the satisfaction conditions of formulas of the target language using that semantics. The resulting theory will not be the usual two-unit compositional system in which the semantic definition is a homomorphic image of the syntactic definition, but rather a three-unit system consisting of: (a) a syntactic definition of the target language; (b) an algorithm for translating well-formed formulas of that language to formulas of the auxiliary language; (c) a compositional truth definition for the auxiliary language.

Such a semantics was constructed by Henkin for the 1st-order language (or rather family of languages) of partial order with standard quantifiers, henceforth POL. Henkin's auxiliary language is the familiar language of 2nd-order logic with functional variables, and its semantics is the familiar compositional semantics of (linear) 2nd-order logic. The translation algorithm is, however, not compositional, and as a result the semantics is partially non-compositional. In spite of its partial non-compositionality, however, the theory satisfies the two methodological requirements of (i) a finite semantics for an infinite language, and (ii) a highly informative semantics (in some important respect). The first requirement is satisfied due to the algorithmic nature of the translation procedure. The second – due to the clear and transparent way in which the theory explains the influence of quantifier-ordering on truth.

I will not be able to present a full account of the semantics of POL here. (For a precise account of the syntax and translation algorithm see Walkoe 1970 or Sher 1997; the standard semantics of 2nd-order languages is described in many textbooks of mathematical logic.) But to demonstrate the way in which Henkin's semantics explains how quantifier-ordering

affects the satisfaction conditions of given formulas I will present two examples of the steps involved in the translation algorithm. As the reader will immediately recognize, the algorithm is the familiar Skolem algorithm extended to partially-ordered quantifications.

Consider the following two formulas of POL:

(3) $\forall x \exists y \forall z \exists w Rxyzw,$

and

(4) $\begin{matrix} \forall x \exists y \\ \\ \forall z \exists w \end{matrix} \quad Rxyzw.$

where R is a relational expression. (3) is a linear formula, i.e., a formula of standard (linear) 1st-order logic, while (4) is properly partially-ordered. The translation algorithm transforms (3) into

(5) $\exists f^1 \exists h^2 \forall x \forall z R(x, f(x), y, h(x, z)),$ [11]

and (4) into

(6) $\exists f^1 \exists h^1 \forall x \forall z R(x, f(x), y, h(z)).$

Each transformation proceeds in four steps:

Transformation of (3) to (5)

Step I: Each existential quantifier-expression (or, shortly, existential quantifier) is assigned a *rank*. The rank of an existential quantifier is the number of its "essential" dependencies, i.e., the number of universal quantifiers governing it.

Rank$(\exists y) = 1$; *Rank*$(\exists w) = 2$.

Step II: Each existential quantifier of rank n is assigned a new n-place functional variable, f^n.

$\exists y - f^1$; $\exists w - h^2$.

Step III: Each existential quantifier of rank n is assigned an n-place functional term "which traces its essential dependencies", $f(x_1, \ldots, x_n)$, where f is the functional variable associated with it and $x_1, \ldots, x_n$ are all the variables in the universal quantifiers governing it.

$\exists y - f(x)$; $\exists w - h(x, z)$.

Step IV: (3) is transformed into (5).

Transformation of (4) to (6)
Step I: *Rank*($\exists y$) = 1; *Rank*($\exists w$) = 1.
Step II: $\exists y - f^1$; $\exists w - h^1$.
Step III: $\exists y - f(x)$; $\exists w - h(z)$.
Step IV: (4) is transformed into (6).

These transformations explain how quantifier-dependencies affect the truth conditions of a given sentence by correlating "essential" dependencies with functional operations. Every dependence of the form '$\forall x_1 \ldots x_n \exists y$' is represented by an n-place function, and its exact constitution is represented by the choice of variables in the corresponding functional expression: the dependence of '$\exists w$' on the single quantifier '$\forall z$' in (4) is reflected in the fact that in (6) 'w' is replaced by a 1-place functional expression whose argument is 'z', namely '$h(z)$', and its dependence on the two quantifiers '$\forall x$' and '$\forall z$' in (3) is reflected in the fact that in (5) 'w' is replaced by '$h(x, z)$'.

Intuitively, the difference in satisfaction conditions between (3) and (4) is reflected in the difference in meaning between:

(7) Every villager has a relative who hates-and-is-hated-by some relative of every townsman,

understood as

(8) Every villager V has a relative R_v such that for every townsman T, there is a townsman relative R_t, such that R_v hates-and-is-hated-by R_t,

and

(9) Every villager has a relative *and* every townsman has a relative who *all* hate each other,

understood as

(10) Every villager V has a relative R_v and every townsman T has a relative R_t such that all the R_v's hate-and-are-hated-by all the R_t's,

i.e., as

(11) There is a group with relatives of all the villagers and a group with relatives of all the townsman, such that each villager-relative in the first group hates-and-is-hated by all the townsman-relatives in the second group.

(These sentences are variations on an example due to Hintikka 1973).

Whereas linear dependencies induce (multiple) tree-like truth conditions, non-linear dependencies induce a more complex type of truth conditions. Linear quantifications say that there are so many things x *for each of which* there are thus many things y, *for each of which* . . . , . . . there are that many things z such that x, y, . . ., and z stand in the quantified relation R. In contrast, non-linear or "branching" quantifications of the type of (4) say that there is a group with so many elements and a group with thus many elements such that *each* element of the first group stands to *all* elements of the second group in the relation R. The satisfaction conditions associated with linear quantifications are of the type 'for each there are'; those associated with branching quantifications are of the type 'each-all'. (See Sher 1990).

Recently, POL was also given a direct and fully compositional semantics (Hodges, 1997). This semantics, however, does not offer as clear an account of the relation between quantifier-ordering and satisfaction-conditions, and therefore is not (as it stands) a viable alternative to Henkin's semantics. Moreover, the translational method allows us to deal with richer languages than POL, while the compositional method has yet to be extended to such languages. Thus, a translational semantics exists for the 1st-order language of partially-ordered *generalized* quantifiers (of the syntactic type of '$\forall$' and '$\exists$'), henceforth GPOL, but not (yet) a compositional semantics. I will not present here a full account of the semantics of GPOL (for details see Sher 1997), but as in the case of Henkin's semantics, I will demonstrate the workings of its translation algorithm by a few examples.[12]

The main difference between the translation algorithms of POL and GPOL has to do with the fact that certain features of $\forall$ and $\exists$ are not shared by all GPOL quantifiers. Thus, in POL only dependencies of existential on universal quantifiers are "essential", and "essential" dependencies are all *functional*, but this is not the case in GPOL. For that reason, Henkin's algorithm associates 2nd-order terms only with existential quantifiers (in a given prefix) and these terms are all functional, but the translation algorithm of GPOL associates 2nd-order terms with *every* quantifier (in a prefix) and these terms are all *relational* (rather than functional). Let us take two very simple examples:

(12) $\quad Q_1 x\, Q_2 y\, R x y$

and

(13) $\begin{matrix} Q_1x \\ \\ Q_2y \end{matrix}$ Rxy,

where Q_1 and Q_2 are any generalized quantifiers (of the designated syntactic type). The translation algorithm of GPOL transforms (12) into

(14) $\exists X^1 \exists Y^2 (Q_1 x X x$ & X is a maximal set such that $\forall x (Xx \rightarrow Q_2 y Y x y)$ & Y is a maximal relation such that $\forall x \forall y (Y x y \rightarrow Rxy))$

and (13) into

(15) $\exists X^1 \exists Y^1 (Q_1 x X x$ & $Q_2 y Y y$ & $\langle X, Y \rangle$ is a maximal pair such that $\forall x \forall y (Xx$ & $Yy \rightarrow Rxy))$.[13]

Each transformation proceeds in six steps:

Transformation of (12) to (14)

Step I: Each quantifier (expression) is assigned a *rank*. The rank of a quantifier is the number of its dependencies (i.e., the number of quantifiers governing it) +1.

Rank$(Q_1x) = 1$; *Rank*$(Q_2y) = 2$.

Step II: Each quantifier of rank n is assigned a new n-place relational variable, X^n.

$Q_1x - X^1$; $Q_2y - Y^2$.

Step III: Each quantifier of rank n is assigned an n-place relational formula "which traces its dependencies", $Xx_1 \ldots x_n$, where X is the relational variable associated with it and $x_1, \ldots, x_n$ are the variables in it and in all the quantifiers governing it.

$Q_1x - Xx$; $Q_2y - Yxy$.

Step IV: Each quantifier is assigned a formula expressing its satisfaction conditions relative to the quantifiers governing it, if any.

$Q_1x - Q_1xXx$; $Q_2y -$ X is a maximal set such that $\forall x (Xx \rightarrow Q_2 y Y x y)$.[14]

Step V: The quantified formula is assigned a formula that expresses its satisfaction conditions relative to the innermost quantifiers binding it.

$Rxy - Y$ is a maximal relation such that $\forall x \forall y(Yxy \rightarrow Rxy)$.[15]

Step VI: (12) is transformed into (14).

Transformation of (13) to (15)

Step I: $Rank(Q_1x) = 1$; $Rank(Q_2y) = 1$.
Step II: $Q_1x - X^1$; $Q_2y - Y^1$.
Step III: $Q_1x - Xx$; $Q_2y - Yy$.
Step IV: $Q_1x - Q_1xXx$; $Q_2y - Q_2yYy$.
Step V: $Rxy - \langle X, Y\rangle$ is a maximal pair such that $\forall x \forall y(Xx \& Yy \rightarrow Rxy)$.[16]
Step VI: (13) is transformed into (15).

Intuitively, the difference in truth conditions between (12) and (13) can be explained by the following examples:

(16) Six boys have (each) dated six girls,

and

(17) (Exactly) six boys and (exactly) six girls have all dated each other.

(16) says that there is a maximal set with exactly six boys each one of which has dated exactly six girls (possibly not the same girls). (17) says that there is a set with exactly six boys and a set with exactly 6 girls, and this pair of sets is a maximal pair such that each boy in the first set has dated all the girls in the second set.[17] As in the case of (3) and (4), linear quantification is (multiple-) tree-like, while genuine branching quantification requires the existence of a "massive nucleus" of objects standing in a given relation; linear quantifiers stand in the "for each ... there are" relation, while (genuine) branching quantifiers stand in the "each ... all" relation.

Based on the procedure delineated above we transform

(18)
$$\begin{matrix} Q_1x\,Q_2y \\ \\ Q_3z\,Q_4w \end{matrix} \qquad Rxyzw$$

into

(19) $\exists X^1 \exists Y^2 \exists Z^1 \exists W^2 [Q_1xXx \ \& \ X$ is a maximal set such that $\forall x(Xx \rightarrow Q_2yYxy) \ \& \ Q_3zZz \ \& \ Z$ is a maximal set such that $\forall x(Xz \rightarrow Q_4wWzw) \ \& \ \langle Y, W\rangle$ is a maximal pair such that $\forall x \forall y \forall z \forall w(Yxy \ \& \ Wzw \rightarrow Rxyzw)$],

and similar translations exist for other well-formed formulas of GPOL.[18]

Like Henkin's semantics for POL, the translational semantics of GPOL provides a clear account of the way quantifier ordering affects the truth conditions of a given sentence. Each quantifier is assigned (i) a relational formula tracing its dependencies, (ii) a formula expressing its satisfaction condition relative to its governors, and (iii) a maximality condition constraining the satisfaction conditions of its dependents. Together, these explain its share in the truth conditions of the quantified sentence. The truth-conditions of the sentence as a whole are given through its translation, and due to the transparent nature of the latter, the role played by its specific quantifiers and their specific ordering is clarified. We may liken the translational semantics of partially-ordered quantifiers to Russell's semantics of definite descriptions: in both cases an unwieldy structure is treated by a *contextual definition*, i.e., a method for translating any well-formed formula in which this structure occurs into another formula in which it does not occur and which has a familiar – in fact, a compositional – semantics.

My second example of a non-compositional yet informative definition of truth belongs to an interesting and important category of definitions which I will call "Conceptual bridge" definitions.

9. "CONCEPT-BRIDGING" DEFINITIONS

We have encountered two definitions of truth: A compositional definition centering on the contribution of logical structure (as determined by logical constants) to truth, and a partially-compositional definition centering on the influence of quantifier-order on truth. The first definition aims at explicating the "logical component" in truth, the second – the "quantifier-ordering component". Both definitions aim at capturing the central principles underlying the influence of the target component on truth, regardless of choice of terminology: Any term can be used in the definition, so long as the goals of informativeness, precision and accuracy are well-served. This, however, is not the case with all definitions of truth or, for that matter, of other notions. Some definitions aim at constructing a *bridge* between two sets of concepts: a set of concepts associated with the definiendum, and a set of concepts used in, and associated with, the definiens. I will call this type of definition a "synthesizing" or "concept-bridging" definition. Synthesizing definitions are the antithesis of "analytic" or "conventional" definitions. They create new, at times unexpected connections between two areas of thought, two methods of inquiry. By forging these connections they open the way to a merger of two sci-

ences or to a fertilization of one science by the conceptual tools of the other. A paradigm example of a *concept-bridging* definition, or a cluster of such definitions, is the cluster of Cartesian definitions of geometric notions in algebraic terms. These definitions bring algebraic methods to geometry and geometric intuitions to algebra, with fruitful results for both fields. Another example of concept-bridging definitions is the family of *reductive* definitions. Reductive definitions set out to explain the concepts, practices and phenomena of one science in terms of those of another, explaining chemical notions in physical terms (Physicalism), mathematical notions in logical terms (Logicism), etc. Such definitions play an important methodological role, unifying, simplifying and minimizing the cognitive enterprise. (Logicism, for example, reduces the two tasks of explaining logic as well as mathematics to the one task of explaining logic.)

While some definitions are originally designed as concept- or field-bridging definitions, not all concept-bridging definitions are constructed with that aim in mind. By identifying linguistic with logical complexity Tarski had (either intentionally or fortuitously) oriented his 1933 definition of truth towards logic, laying the ground for a fruitful synthesis of logic and semantics. His definition, with its special entries for logical constants (generators of logical structures), provided the resources for a semantic definition of *logical consequence* (Tarski 1936), and this, in turn, opened the way to a systematic development of logical semantics (model theory).

Tarski's "bridge building" definition is compositional, but the idea of forging a fruitful bridge between two sets of concepts is not inherently connected with compositionality. Hintikka's game-theoretic definition of truth for languages with partially-ordered standard quantifiers (POL) is a case in point.

10. A NON-COMPOSITIONAL DEFINITION OF TRUTH

In numerous writings Hintikka and Sandu have developed a game-theoretic semantics for languages of partial order with standard quantifiers. (See, for example, Hintikka 1996 and Hintikka & Sandu 1997.) This semantics has two parts: (i) a syntax for languages of partial order with standard quantifiers and (ii) a non-compositional, game-theoretic definition of truth for sentences of that language. The basic idea is described by Hintikka and Sandu (1997) as follows:

> The leading ideas of game-theoretical semantics . . . can be seen best from . . . the semantics of quantifiers. In using quantifiers and in theorizing about them, it is hard not to use game-laden terms, especially if one thinks of seeking and finding as a game. In traditional informal mathematical jargon, quantifiers are routinely expressed by such phrases as

"given any value of x, one can find a value of y such that". In several natural languages, existence is expressed by phrases translatable as "one can find".

…

[A] game-theoretical treatment of quantification theory …is …[a] codification of [such] natural and time-honored ways of thinking and speaking ….(363)

The central notion of game-theoretic semantics is the notion of *strategy*. Each sentence is associated with a zero-sum game between two characters, one trying to verify it and the other trying to falsify it. Hintikka named the two characters 'Myself' and 'Nature'; I will use 'Humanity' and 'Nature'. The game proceeds according to fixed rules reflecting the satisfaction conditions of the logical constants appearing in the sentence. (For details see above references.) A *strategy* is a set of rules – or functions – which tell a player how to proceed in the game; a *winning strategy* ensures a Win. A sentence is true iff there is a *winning strategy* for verifying it, i.e., iff Humanity has a strategy which enables her to win regardless of Nature's moves. A sentence is false iff there is a winning strategy for falsifying it, i.e., iff Nature has a strategy which enables him to win regardless of Humanity's moves. (These definitions assume a non-bivalent semantics.)

The game-theoretic method is especially suitable for languages with standard quantifiers due to the functional nature of their essential dependencies. A strategy for a partially-ordered sentence corresponds to a string of Skolem functions instantiating the functional variables of its Henkin translation, and a *winning strategy* (for Humanity) corresponds to a string of functions "verifying" this translation. Henkin, indeed, offered his translational semantics as a systematization of an intuitive game-theoretic interpretation, and Hintikka resystematized these intuitions using the resources of the mathematical theory of games.

The game-theoretic semantics of POL can be viewed as a mere alternative to the translational semantics delineated above, but by bringing together game-theoretical and semantic notions Hintikka and Sandu set the ground for a new and potentially fruitful synthesis: a synthesis of a cluster of notions revolving around *information*: *informational dependence* and *independence*, *choice under conditions of perfect* or *imperfect information*, etc., and a cluster of notions revolving around *scope*: *quantifier scope*, *partial-ordering*, *government* and *dependence*, etc.[19]

The notion of information is central to games: at each step in a game each player is given either full, or partial, or null information about earlier moves in the game, and this information (or lack thereof) is pertinent to his/her choice of action at that step:

In games like chess, each player has access to the entire earlier history of the game, but in many others a player's knowledge of what has happened earlier is incomplete. In this case we are dealing with a game with *imperfect information*. A move made in ignorance of another one is said to be *informationally independent* of the latter. For instance, in many card games one does not know which cards one's opponent has picked up earlier. (Hintikka & Sandu 1989: 571–572, my italics)

In semantic games informational dependence corresponds to *quantifier-dependence* and informational independence to *quantifier-independence*. Quantifier dependence and independence are, in turn, a matter of *scope*. Thus, in

(1) $\forall x \exists y \forall z \exists w R(x, y, z, w)$,

'$\exists w'$ is in the scope of both '$\forall z$' and '$\forall x$'; therefore it is dependent on both; while in

(2) $\begin{matrix} \forall x \exists y \\ \\ \forall z \exists w \end{matrix} \quad R(x, y, z, w)$

'$\exists w$' is in the scope of '$\forall z$' but not '$\forall x$'; therefore it is dependent on the first but not on the second. In linear quantifications each quantifier either governs or is dependent on any other quantifier; in non-linear quantifications this is not the case. Accordingly, the semantic games associated with linear quantifications are games of *perfect information*, while those associated with non-linear quantifications are games of *imperfect information*. To see how scope differences have to do with access to information, compare the game associated with (1) with that associated with (2). The first game proceeds as follows:

Game (1):

Step 1 [$\forall x$]: Nature (initial falsifier) chooses an individual *a* in the universe as a value for x (with the aim of falsifying the sentence).

Step 2 [$\exists y$]: Humanity (initial verifier) *is given information on Nature's choice in Step 1,* and based on this information chooses an individual *b* in the universe as a value for y (with the aim of countering Nature's move).

Step 3 [$\forall z$]: Nature chooses an individual *c* in the universe as a value for z.

Step 4 [$\exists w$]: Humanity *is given information on Nature's choices in Steps 1 and 3* and based on this information chooses an individual *d* in the universe as a value for w.

In this game Humanity has *perfect information* about Nature's moves: in Step 2 Humanity knows what move Nature made in Step 1, and in Step 4 she knows what moves he made in Steps 1 and 3. But in the second game Humanity has only partial information about his moves:

Game (2):

Step 1 [$\forall x$]: Nature chooses an individual a as a value for x.

Step 2 [$\exists y$]: Humanity *is given information on Nature's choice in Step 1*, and based on this information chooses an individual b as a value for y.

Step 3 [$\forall z$]: Nature chooses an individual c as a value for z.

Step 4 [$\exists w$]: Humanity *is given information on Nature's choice in Step 3 (but not Step 1)*, and based on this (partial) information chooses an individual d as a value for w.

Here, in Step 4 Humanity has no information on Nature's move in Step 1; the information given her in Step 2 is taken away. In both games Humanity wins iff she chooses two individuals, c and d, such that given Nature's choice of a and b, a, b, c and d stand in the relation R. But in game (1) Humanity acts under conditions of full information, while in Game (2) she acts under conditions of partial information.

Thinking of semantic games as played by Humanity against Nature suggests that truth is a "game of knowledge". Another metaphor for players in such games is 'Us against Them'. In game (1) Us is a unified team. Each action is taken by the team as a whole, and the team has full (informational) access to past moves. In game (2) Us is divided into two teams which have no contact with each other. Each team is given information on one of Them's moves and acts in ignorance of its other move (and the other team's action). Us win iff the two teams make the "right" choice in spite of having no communication. Under this metaphor (2) is true iff Us have an initial winning strategy that does not require sharing of information between its teams. Other suggestive metaphors include a game between isolated prisoners and an interrogator, a game in which two "independent" particles proceed without "knowledge" of each other's moves, yet their movement is (or has the similitude of being) "coordinated", etc. (All these metaphors suggest further uses of the semantics).

Under what conditions is there a winning strategy for a sentence under conditions of imperfect information? Game theoretic semantics is *truth-conditional* (Hintikka, 1987). Whether a verifying strategy exists depends, therefore, on how things are in the world, i.e., on the pattern of individuals

standing in the quantified relation (in our example, R). Our analysis in the last section suggests that the pattern satisfying (1) is that of a "multiple tree" while the pattern satisfying (2) is that of a "massive nucleus" (of objects standing in R). This special feature of the "branching" situation explains why each team of Us can make the right choice in ignorance of the other's: when Reality is especially cooperative (when the relation contains a "massive nucleus") knowledge can be obtained under conditions of less than perfect information.

11. CONCLUSION

In this paper I have offered a few observations on the methodology of structural theories of truth. Truth is a multi-dimensional concept, and a methodology of truth-theories must recognize this feature. The notion of structure is partly topic-related: there is logical structure – structure induced by highlighting the logical constituents of sentences - modal structure, epistemic structure, and even physical and biological structure. To understand the role different types of structure play in determining the truth value of sentences is to understand the principles governing the semantic behavior of the respective "distinguished" constants (the principles governing logical constants being different from those governing modal, epistemic and other types of constant.) Unlike the notion of logical structure, the notion of iterative structure is content-neutral, but (perhaps just for this reason) its philosophical significance is more limited. The notion of compositional structure is rather involved, having to do with the interrelation between semantics and syntax. While compositionality is conducive to clarity and informativeness, some explanatory goals are better served by non-compositional, or partially compositional, theories.

NOTES

[1] The same convention applies to defined symbols of L_C, e.g., '$\rightarrow$'. Below, parentheses will be omitted when convenient.

[2] See, for example, Davidson (1973: 71) and LePore and Ludwig (1999). By saying that in these places the authors conflate the three answers I don't mean that they are unaware of, say, uses 'logicality' that differ from those of 'compositionality', but that in the context of a theory of truth (or a truth-conditional theory of meaning) they see no reason for distinguishing the two concepts.

[3] For a more detailed discussion of this point see Sher (1999).

[4] An argument-structure for a 2-place relation between individuals is a structure of the form $\langle A, \langle a, b\rangle\rangle$, where A is a non-empty set and a, b are members of A.

[5] For further elaboration of the points made in this Section see Sher (1996, 1998/9, 2000).
[6] See, e.g., Enderton (1972): 26.
[7] For an excellent account of compositionality, see Janssen (1997).
[8] Davidson was especially influential in introducing this theme. For the view that a learnable language need not be compositional see, e.g., Schiffer (1987).
[9] By a 'standard' logical language I mean a language with the standard logical constants. I think of the standard quantifiers ($\forall$ and $\exists$) as generating a well-formed expression from a *pair* of expressions – a variable and a well-formed formula.
[10] Henkin allows formulas with infinitely many occurrences of primitive expressions, but here I will limit myself to finitely long formulas.
[11] Whenever the arity of a functional or a relational symbol is clear from the context, I remove its superscript.
[12] This translation algorithm extends pioneering work by Barwise (1979) on the semantics of branching monotone-increasing generalized quantifiers, and it applies to any partially-ordered generalized quantifiers (of the syntactic type of '$\forall$' and '$\exists$') regardless of monotonicity. Some examples of such quantifiers are 'At least/Exactly/At Most α', where α is any cardinal number, 'Most', 'Finitely/Infinitely/Countably/Uncountably Many', and 'An Even/Odd Number of'. (Q is monotone-increasing iff '$Qx\ \Phi\ \&\ \forall x(\Phi \rightarrow \Psi)$' implies '$Qx\ \Psi$'. 'At least α' is monotone-increasing, while 'exactly α' and 'at most α' are not. Both $\forall$ and $\exists$ are monotone-increasing.) For another approach to the extension of Barwise's semantics see Westerståhl (1987).
[13] For simplifications of (14) and (15) when the quantifiers involved are monotone-increasing, see footnotes 14 and 15.
[14] If all the quantifiers governing Q_2 are monotone-increasing, the maximality requirement may be omitted. In that case the quantifier condition associated with Q_2y would be '$\forall x(Xx \rightarrow Q_2yYxy)$'.
[15] If all the quantifiers binding 'Rxy' are monotone-increasing, the maximality requirement may be omitted.
[16] See footnote 15.
[17] I have used 'dated *each other*' in (17) to make the English formulation smoother; this has nothing to do with the branching structure as a quantifier structure.
[18] It is easy to see that when (18) is instantiated by (4), the corresponding instantiation of (19) is equivalent to (6).
[19] This game-theoretic semantics has other goals and results, but I will limit my discussion here to the general connection it forges between information and quantifier-order.

REFERENCES

Barwise, J.: 1979, 'On Branching Quantifiers in English', *Journal of Philosophical Logic* **8**, 47–80.
Davidson, D.: 1965, 'Theories of Meaning and Learnable Languages', in Davidson 1984, pp. 3–15.
Davidson, D.: 1967, 'Truth and Meaning', in Davidson 1984, pp. 17–36.
Davidson, D.: 1970, 'Semantics for Natural languages', in Davidson 1984, pp. 55–64.
Davidson, D.: 1973, 'In Defence of Convention T', in Davidson 1984, pp. 65–75.

Davidson, D.: 1984, *Inquiries into Truth & Interpretation*, Oxford University Press, Oxford.
Enderton, H. B.: 1972, *A Mathematical Introduction to Logic*, Academic Press, San Diego.
Field, H.: 1972, 'Tarski's Theory of Truth', *The Journal of Philosophy* **69**, 347–375.
Henkin, L.: 1959, 'Some Remarks on Infinitely Long Formulas', *Finitistic Methods: Proceedings of the Symposium on Foundation of Mathematics*, Warsaw, 1961, pp. 167–183.
Hintikka, J.: 1973, 'Quantifiers vs. Quantification Theory', *Dialectica* **27**, 329–335.
Hintikka, J.: 1987, 'Game-Theoretical Semantics as a Synthesis of Verificationist and Truth-Conditional Meaning Theories', in E. LePore (ed.), *New Directions in Semantics*, Academic Press, London, pp. 235–258.
Hintikka, J.: 1996, *The Principles of Mathematics Revisited*, Cambridge University Press, Cambridge.
Hintikka, J. and G. Sandu: 1989, 'Informational Independence as a Semantical Phenomenon', *Logic, Methodology and Philosophy of Science* VIII, North Holland, Amsterdam, pp. 571–589.
Hintikka, J. and G. Sandu: 1997, 'Game-Theoretical Semantics', in Van Benthem & Ter Meulen 1997, pp. 361–410.
Hodges, W.: 1997, 'Compositional Semantics for a Language of Imperfect Information', *Logic Journal of the IGPL* **5**, 539–563.
Janssen, T. M. V.: 1986, *Foundations and Applications of Montague Grammar, Part I: Philosophy, Framework, Computer Science*, CWI Tracts # 19, Center for Mathematics and Computer Science, Amsterdam.
Janssen, T. M. V.: 1997, 'Compositionality', in Van Benthem and Ter Meulen 1997, pp. 417–473.
LePore, E. and K. Ludwig: forthcoming, 'What is Logical Form?', *Protosociology*.
Lindström, P.: 1966, 'First Order Predicate Logic with Generalized Quantifiers', *Theoria* **32**, 186–195.
Mostowski, A.: 1957, 'On a Generalization of Quantifiers', *Fundamenta Mathematicae* **44**, 12–36.
Schiffer, S.: 1987, *Remnants of Meaning*, MIT, Cambridge, MA.
Sher, G.: 1990, 'Ways of Branching Quantifiers', *Linguistics and Philosophy* **13**, 393–422.
Sher, G.: 1991, *The Bounds of Logic: A Generalized Viewpoint*, MIT, Cambridge, MA.
Sher, G.: 1996, 'Did Tarski Commit 'Tarski's Fallacy'?', *Journal of Symbolic Logic* **61**, 653–686.
Sher, G.: 1997, 'Partially-Ordered (Branching) Generalized Quantifiers: A General Definition', *Journal of Philosophical Logic* **26**, 1–43.
Sher, G.: 1998/99, 'On the Possibility of a Substantive Theory of Truth', *Synthese* **117**, 133–172.
Sher, G.: 1999, 'What Is Tarski's Theory of Truth?', *Topoi* **18**, 149–166.
Sher, G.: 2000, 'The Formal-Structural View of Logical Consequence', manuscript.
Tarski, A.: 1933, 'The Concept of Truth in Formalized Languages', in Tarski (1983), pp. 152–278.
Tarski, A.: 1936, 'On the Concept of Logical Consequence', in Tarski 1983, pp. 409–420.
Tarski, A.: 1966, 'What Are Logical Notions', *History and Philosophy of Logic* **7**, (1986), pp. 143–154.
Tarski, A.: 1983, *Logic, Semantics, Metamathematics*, 2nd edn., Hackett, Indianapolis, IN.
Van Benthem, J. and A. Ter Meulen (eds): 1997, *Handbook of Logic & Language*, MIT, Cambridge, MA.

Walkoe, W. J. Jr.: 1970, 'Finite Partially-Ordered Quantification', *Journal of Symbolic Logic* **35**, 535–555.
Westerståhl, D.: 1987, 'Branching Generalized Quantifiers and Natural Language', in P. Gärdenfors (ed.), *Generalized quantifiers*, D. Reidel, Dordrecht, pp. 269–298.
Zadrozny, W.: 1994, 'From Compositional to Systematic Semantics', *Linguistics and Philosophy* **17**, 329–342.

Department of Philosophy
University of California
San Diego
La Jolla, CA
U.S.A.
E-mail: gsher@ucsd.edu