Hyperintensions

Carl Pollard[1]

Abstract

Standard possible worlds semantics has been known from the start to have a problem with *granularity*: for a wide range of natural-language (NL) entailment patterns, not enough meaning distinctions are available to make predictions consistent with robust intuitions. Though numerous solutions have been proposed, often of great ingenuity and technical sophistication, none of these has gained widespread acceptance. As a result, most semanticists have made a practical decision to work in a framework known to have dubious foundations and leave the foundational problems to mathematical logicians. Here a new approach is proposed which may be simple enough and conservative enough to be practical for working empirical and computational semanticists. More specifically, I show how the use of a *higher-order logic with definable subtypes* leads to a novel and surprisingly straightforward solution of the granularity problem. I also call attention to a hitherto unnoticed problem in standard approaches to NL semantics having to do with *nonprincipal ultrafilters* and show why it does not arise under my proposal. The two main technical innovations that drive the proposal are (1) axiomatizing NL entailment as a preorder (as opposed to an order) on the set of (primitive) propositions, and (2) defining worlds as certain sets of propositions (viz. ultrafilters). These innovations provide just the tools we need to develop a formally explicit theory of *hyperintensions*[2], mathematical models of Fregean senses of a finer granularity than the familiar intensions (functions to extensions from worlds, where the worlds in turn are theoretical primitives) of mainstream Kripke/Montague-inspired NL semantics.

0. Introduction

Standard possible worlds semantics has been known from the start to have a problem with *granularity*: for a wide range of entailment patterns, not

[2]The proposal repairs a number of defects in an earlier effort to formulate a simply-typed hyperintensional semantic theory (Fox et al. 2002), Fox and Lappin 2005).

enough meaning distinctions are available to make predictions consistent with robust intuitions. Though a great many solutions have been proposed from the early 1940s on, often of great ingenuity and technical sophistication, none of these has gained widespread acceptance. As a result, most semanticists have made a practical decision to work in a framework known to have dubious foundations and leave the foundational problems to mathematical logicians. In this paper, a new approach is proposed which, I believe, is simple enough and conservative enough to be practical for working empirical and computational semanticists. More specifically, I show how the use of a higher-order logic with definable subtypes leads to a novel and surprisingly straightforward solution of the notorious granularity problem about natural-language (NL) meanings. I also call attention to a hitherto unnoticed problem in standard approaches to NL semantics having to do with nonprincipal ultrafilters and show why it does not arise under my proposal. The two main technical innovations that drive the proposal are (1) axiomatizing NL entailment as a preorder (as opposed to an order) on the set of (primitive) propositions, and (2) defining worlds as certain sets (viz. ultrafilters) of propositions.

To formalize my semantic theory, I work within a version of higher-order logic similar in its essentials (though not in the details of its presentation) to the boolean version of Lambek and Scott's (1986) **higher-order categorical logic**. This logic differs from the higher-order logics in the Church-Henkin-Montague tradition familiar to linguists in providing for **lambda-definable subtyping**, which plays a central role in my proposal. Set-theoretic models of theories in this kind of logic are very much like the familiar Henkin-style models, but augmented with cartesian products and lambda-definable subsets. The simplicity and familiarity of such models makes this kind of logic accessible and practical for working linguistic semanticists. However, there are more general categorical models (local boolean toposes), which make allowance for the possibility of uninhabited types (i.e. types other than the empty (counit) type for which there are no closed terms) should the need arise; and the boolean condition is easily dropped should one wish to experiment with intuitionistic theories of linguistic meaning.[3]

The paper is organized as follows. In section 1, I briefly review the main features of standard possible-worlds-based NL semantic theory, distinguishing those which I wish to retain from those that I will target for elimination. Section 2 reviews the well-known granularity problem, with special attention

---

[3]Hereafter, occasional categorical considerations will mostly be relegated to footnotes.

to its two most notorious subproblems, Frege's Hesperus-Phosphorus puzzle and the antisymmetry of entailment. Section 3 is an introduction to the general philosophical approach underlying my technical proposal, viz. that propositions are primitives and worlds constructed from them, not the other way around as is usually assumed. Section 4 introduces the second, and heretofore evidently unrecognized, problem of nonprincipal ultrafilters. In section 5, working in the metalanguage, I provide an algebraic theory of propositions that solves both the granularity problem and the nonprincipal ultrafilters problem. The remaining sections develop the logic within which I will formalize my theory, lay out the theory itself, and show by examples how it connects with—and serves as an adequate replacement for—standard posible-worlds semantics. Section 6 is an overview of the typed lambda calculus underlying the logic. Section 7 extends the typed lambda calculus to a higher-order logic. Section 8 develops the semantic theory and illustrates its application. And section 9 summarizes the main features of my proposal.

1 Trouble in Paradise

In NL semantics, at least in its static (as opposed to dynamic) aspects, there is a widely accepted, generally Fregean, story about the basics. It runs something like this:

(1) **The Peaceable Kingdom of NL Semantics**

    a. Meaning is a function from NL expressions[4] to things called **senses**.

    b. Declarative sentence meanings are called **propositions**.

    c. Meanings of names are called (after Carnap) **individual concepts**.

    d. A sense has an **extension**, and what that extension is in general depends on contingent facts (how things are).

    e. The extension of an expression's meaning is called the expression's **reference** or **denotation**.

    f. The things that can be the extension of a proposition (and therefore, the reference of a declarative sentence) are called **truth values**; and there are exactly two of them, called true and false.

    g. One proposition is said to **entail** another just in case, no matter how things are, if its extension is true, then so is the extension of the other.

---

[4]Here, as throughout, I write 'expression' as a shorthand for 'contextualized utterance of an expression', and likewise, *mutatis mutandis*, for 'declarative sentence', 'name', and other terms referring to categories of linguistic expressions.

h. It follows from the preceding that entailment is a preorder (reflexive transitive relation) on propositions, and so mutual entailment is an equivalence relation, also called **truth-conditional equivalence**.

i. One declarative sentence is said to **follow** from another iff the proposition it expresses is entailed by the proposition expressed by the other.

j. The things that can be extensions of individual concepts (and therefore, the references of names) are called **entities**.

k. The individual concepts typically expressed by names are **rigid**, in the sense that their extensions are independent of how things are.

Now the mainstream training in NL semantics includes an indoctrination into a certain classical higher-order formalization of this story, one which was mostly synthesized by Montague in the late 1960's out of ideas drawn from Carnap, Kripke, Church, and Henkin, and subsequently streamlined by Bennett, Gallin, Dowty and others in the 1970's and early 1980's. For expository purposes, I will present what I take to be the main components of this formalization in two groups: those which I do not wish to take issue with (at least not here), and those which I analyze as the source of the problems. First, those aspects of the Standard Formalization that will be preserved in my proposal:

(2) **The Standard Formalization: Aspects Worth Keeping**
One theorizes about senses and their extensions in a higher-order logic similar to Henkin's (1950) formulation of Church's (1940) simple theory of types:

a. A typed ($\beta\eta$-)lambda calculus with a type Bool for formulas and a basic type Ent for entities;

b. equality constants $=_A$ at all types;

c. the familiar lambda-calculus term equivalences (conversion) are formalized as object-language axioms about the $=_A$;

d. the usual logical constants are definable *à la* Tarski/Quine in terms of the $=_A$ and λ.

e. Following Henkin (1950), one adopts the axiom (explicitly rejected by Church) of **Boolean Extensionality**:
$$\forall_{x \in \text{Bool}} \forall_{y \in \text{Bool}}[(x \leftrightarrow y) \rightarrow (x = y)]$$

f. The resulting logic is (a) two-valued; and (b) sound and complete

4

with respect to (unrestricted[5]) Henkin models.

g. As in Gallin (1975), there is a type World (possible worlds). This improves on Montague's IL: e.g. there is a complete proof theory, and no up and down operators.

h. Meanings are assigned to NL expressions by translating them into the logic and then interpreting the logic into a model.

i. Thus meanings, their extensions, and worlds all live in the same model, and one can write nonlogical axioms (meaning postulates) about how these things are related to each other,

j. In any model, the set of propositions is equipped with a boolean structure in terms of which entailment and the meanings of NL "logical words" can be represented.

By contrast, I identify the following features of the Standard Formalization as the problematic ones to be weeded out:

(3) **The Standard Formalization: Aspects to Eliminate**

a. The type World is *basic*, i.e. worlds are primitives (cf. Kripke 1963).

b. Meanings are *intensions*, i.e. functions from the set of worlds[6].

   i. Name meanings (individual concepts):

   - are functions from worlds to entities; and so
   - if one assumes the rigidity of names (Kripke 1980), then co-referring names have the same meaning.

   ii. Declarative sentence meanings (propositions):

   - are (characteristic functions of) sets of worlds;
   - entailment is the subset-inclusion ordering on sets of worlds;
   - the meanings of *and*, *or*, and *if ... then* are, respectively, intersection, union, and relative complement.
   - In particular, entailment is *antisymmetric*. Thus:
   - truth-conditionally equivalent propositions are identical; and so
   - sentences that follow from each other have the same meaning.

---

[5]In the sense that the interpretations of the functional types only have to contain enough functions to interpret all closed terms.

[6]Or at least are equivalent to such functions, up to a permutation of their arguments. See, e.g. Carpenter 1997.

I will show that eliminating these undesirable aspects of mainstream semantics is not only easy, but also that it does no harm; nothing that linguists actually need semantic theory for depends on these features. To put it another way: they do not really model anything about linguistic meanings, but are mere artifacts of the formalization, and pernicious ones at that.

With the scene set, we can now turn to the first of the two problems with the Standard Formalization that we have set our sights on: there are not enough intensions.

2   The Granularity Problem

By way of review, we briefly consider two manifestations of the granularity problem.

2.1   Hesperus and Phosphorus

As Frege (1892) realized, having the same reference is not a sufficient condition to allow replacement of one name for another in a sentence while preserving truth:

(4) **Hesperus and Phosphorus**

  a. (The ancients realized that) Hesperus was Hesperus.
  b. (The ancients realized that) Hesperus was Phosphorus.

Frege's view was that the names *Hesperus* and *Phosphorus*, while referring to the same planet (viz. Venus), express different senses and therefore there is no reason to expect *Hesperus is Hesperus* and *Hesperus is Phosphorus* to express the same proposition. And consequently, it is unsurprising that the ancients might well have believed the first but not the second.

But in the Standard Formalization, the senses of the two names *Hesperus* and *Phosphorus* are functions from worlds to entities, and at least one of these worlds, the two functions have the same value, namely the planet Venus. But if Kripke was right about the rigidity of names, then the two functions must both be the constant function that maps each world to Venus, i.e. *Hesperus* and *Phosphorus* mean the same thing. So by standard considerations of compositionality, *Hesperus is Hesperus* and *Hesperus is Phosphorus* must express the *same* proposition.

Of course there is a copious literature that seeks to justify the acceptance of this seemingly unattractive consequence. But on our proposal, which can

be seen as a technical implementation of Frege's view, there is no such consequence, and therefore no need for a justification.

2.2   Equivalent Propositions

As we noted, in the Standard Formalization, entailment is an order, and so truth-conditionally equivalent propositions are identical. The (elementary) proof depends crucially on the antisymmetry, and so does not generalize to preorders (transitive symmetric relations).

But there is a naive, robust intuition that declarative sentences can follow from each other without meaning the same thing. We illustrate with two examples.

(5) **Woodchucks and Groundhogs**

    a. Phil is a woodchuck.

    b. Phil is a groundhog.

On standard accounts, the equivalence of (5a) and (5b) would follow from a meaning postulate asserting that, necessarily, anything that is a woodchuck is a groundhog and conversely. Thus, we cannot accept that (say) Jim knows Phil is a groundhog without also accepting that Jim knows Phil is a woodchuck.

(6) **Paris Hilton and the Riemann Hypothesis**

    a. Paris Hilton is Paris Hilton.

    b. S. [Where S is either 'All nontrivial zeros of ζ have real part $1/2$' or 'Not all nontrivial zeros of ζ have real part $1/2$', whichever is true.]

In our second example of equivalent propositions, the propositions in question are both necessary truths. But according to the Standard Formalization, there is only one necessary truth (viz. the set of all worlds). And so it seems that the plausible premiss that Justin Timberlake knows that Paris Hilton is Paris Hilton forces the implausible conclusion that Justin Timberlake knows whether the Riemann Hypothesis is true.

Again, there are strenuous arguments (most notably by Stalnaker) on behalf of accepting these (naively) unsavory consequences. Rather than bother with constructing counterarguments, we will propose an account where there simply are no such consequences, and hence no need for justification.

3   Soft Actualism Recalled

In his defense of possible worlds, Stalnaker (1984) compares the standard view (that propositions are sets of possible worlds) with an alternative position, **soft actualism**, put forward by Robert Adams (1974). In Adams' terminology, this contrasts with **hard actualism**, which flatly denies the existence of nonactual possible worlds. Adams' position can be summarized as follows:

(7) **An Alternative: Robert Adams' (1974) "Soft Actualism"**

  a. Nonactual possible worlds exist in the sense of being logically constructed out of the actual world. Specifically:

  b. possible worlds are *maximal consistent sets of propositions*.

  c. Thus propositions are primitive and worlds are constructed, (not the other way around as per the Standard Formalization).

In fact, soft actualism was anticipated by Kripke's (1959) completeness theorem for S5, which implemented possible worlds as *complete assignments* of truth values to formulas, which are just characteristic functions of maximal consistent sets of formulas. Kripke's complete assignments in turn can be seen as a more precise rendering of Carnap's (1947) notion of a *state description*. And as Kripke (1963) noted, the essentials of his analysis of modal logic had also been anticipated in algebraic form by Jónsson and Tarski's (1951) representation theorem for boolean algebras with n-ary operators: the Kripke semantics is the case $n = 1$. Ultimately the roots of this approach lie in Stone's (1936) Representation Theorem (which the Jónsson-Tarski theory extends to the case of boolean algebras with operators). Stone's theorem implies (inter alia) that any boolean algebra B can be isomorphically embedded into a powerset $\wp(X)$, by taking X to be the cospectrum of B (i.e. the set of B's ultrafilters).[7] The connection is that if B is taken to be the set of propositions (in the sense of declarative sentence meanings) and the order induced by the boolean structure is taken to be entailment, then the ultrafilters correspond to exactly to the complete assignments.

But in 1963, for his more general completeness theorem for normal modal propositional calculi, Kripke abandoned this approach (of worlds as maximal consistent sets or ultrafilters) in favor of taking possible worlds as unanalyzed primitives. And in his influential papers on NL semantics, Montague

---

[7]The embedding maps each $b \in B$ to the set of ultrafilters containing it.

(1974) followed the lead of Kripke 1963, not Kripke 1959, in providing a possible-worlds semantics of English. Subsequently, it seems to have simply been taken for granted that Montague's way is *the* way, not because arguments were set forth somewhere or other that the Standard Formalization (worlds as primitives) provides a better framework for NL semantics than Soft Actualism (worlds as maximal consistent sets), but rather because the alternative was not even considered. Below we will consider the alternative, and argue that, as far as NL semantics is concerned, *Soft Actualism is preferable to the Standard Formalization*

In the sequel we will find it convenient to cast the essential content of the two approaches in algebraic terms, as follows:

(8) **Soft Actualism in Algebraic Form (Preliminary Version)**

    a. Propositions are elements of a boolean algebra[8] .

    b. Entailment is the order on that algebra,

    c. Possible worlds are the ultrafilters of that algebra.

    d. 'p is true at w' means $p \in w$.

(9) **The Standard Formalization in Algebraic Form**

    a. Propositions are sets of possible worlds.

    b. Entailment is the subset inclusion order on the powerset of the set of possible worlds (which is of course a boolean algebra).

    c. 'p is true at w' means $w \in p$.

How different are the two approaches? We focus on this question in the next two sections.

4 Nonprincipal Ultrafilters: an Overlooked Problem

To facilitate the comparison of (algebraicized) Soft Actualism and the Standard Formalization, it will be helpful to first lay out some of the basic facts about ultrafilters, starting with the following definition:

(10) **Definition (Ultrafilter of a Boolean Algebra)**
    Suppose B is a boolean algebra and $U \subseteq B$. Then U is an **ultrafilter** of B just in case the following three conditions are satisfied:

---

[8]Later we will weaken this to a boolean prealgebra.

a. it is closed under finite meets;

b. it is upper-closed relative to the order $\sqsubseteq$ on B induced by the boolean structure, i.e. for each $b \in U$, $\uparrow b \subseteq U$[9]; and

c. for every $b \in B$, exactly one of b and b' is in it.

For our purposes, the Stone Representation Theorem is most conveniently stated in the following form:[10]

(11) **Stone Representation Theorem**

a. Any boolean algebra B can be isomorphically embedded as a subalgebra of a powerset algebra $\wp(X)$.

b. X can be taken to be the set of ultrafilters of B, with the embedding mapping each $b \in B$ to the set of ultrafilters containing it.

(12) **Definition (Principal Ultrafilter)**
Suppose B is a boolean algebra and U an ultrafilter of B. Then U is called a **principal** ultrafilter provided it has a least element a. In that case a is said to be the **generator** of U, or to **generate** U.

(13) **Basic Facts about Principal Ultrafilters**

a. Suppose U is a subset of a boolean algebra B. Then U is a principal ultrafilter iff there is an atom of B, a, such that $U = \uparrow a$. In that case, a is the generator of U.

b. Hence there is a one-to-one correspondence between the atoms of B and its principal ultrafilters, mapping each atom to its upset.

c. If B is finite, every ultrafilter is principal. In that case the Stone embedding maps each $b \in B$ to the set of principal ultrafilters whose generators are the atoms a such that $a \sqsubseteq b$.

d. But if B is infinite, then (assuming the Axiom of Choice) not every ultrafilter is principal.

This last fact has a consequence for the Standard Formalization that seems to have gone unnoticed. To see why, suppose that the Standard Formalization is correct. In that case propositions are defined to be sets of possible worlds, and entailment is the subset inclusion order on the boolean

---

[9]For any member a of a boolean algebra (or, more generally, of a preorder) B, $\uparrow a =_{\text{def}}$ $\{b \in B \mid a \sqsubseteq b\}$. This is called the **upset** of a, or simply **up** of a.

[10]Stone's formulation was in terms of boolean rings (rather than boolean algebras) and their prime ideals, and had an important topological dimension that is ignored here.

algebra $\wp(W)$, where W is the set of all possible worlds. Then what does (13) say about the case $B = \wp(W)$? Well, in this case the atoms are just the singleton sets $\{w\}$, where w is a possible world. From (13b) it follows that there is a one-to-one correspondence between possible worlds and principal ultrafilters of B, with each world w corresponding to the set $\uparrow(\{w\})$ whose members are those sets of worlds of which w is a member. Note that this is just the set of all propositions true at w.

Now suppose for a moment that the set of standard-formalization propositions were finite. Obviously this could be the case iff the set of possible worlds were finite. In that case all ultrafilters of B would be principal, so that the function mapping each world to the set of propositions true at that world would be a bijection from W to the set of ultrafilters of B. Specifically, each world w would be mapped to the principal ultrafilter whose generator is the singleton set whose only member is w (in other words, the ultrafilter whose members are those sets of worlds which have w as a member). From this it follows that Soft Actualism and the Standard Formalism would amount to the same thing, since a world w would belong to a proposition (set of worlds) p iff p belonged to the ultrafilter consisting of all those sets of worlds that have w as a member.

But the set of propositions is (uncontroversially) infinite, since it is easy to find a countably infinite set S of English sentences such that no semanticist would be willing to allow that there are two members of S that are even truth-conditionally equivalent, to say nothing of two members of S that express the same proposition. One such set consists of the sentences *Frege had a cat, Frege had two cats, Frege had three cats*, etc.; another consists of the sentences *Frege erred, Russell knew Frege erred, Frege knew Russell knew Frege erred*, etc.

As before, let B be the boolean algebra of Standard-Formalization propositions (sets of possible worlds), and remember that in B, conjunction, disjunction, implication, and negation are represented, respectively, by intersection, union, relative complement, and complement (of sets of worlds), and that entailment is represented by subset inclusion (of sets of worlds). Recall also that by definition, a subset U of B (i.e. a set of sets of worlds) is an ultrafilter iff (a) the intersection of any two of its members is in U (b) for any $p \in U$, every set of worlds with p as a subset is also in U; and (c) for every set of worlds p, exactly one of p and $W \setminus p$ is a member of U. In other words, (a) the conjunction of any two propositions in U is also in U; (b) any proposition entailed by a proposition in U is also in U; and (c) for any proposition p, exactly one of p and its negation is in U. This is exactly what we mean by a maximal consistent set of propositions. Intuitively, such a set

is a "possible way things might be": it tells, for every proposition, whether or not it is true; and it does so in a consistent way, in the sense that, even though it contains all the entailments of each of its members, it does not contain W (the set of all worlds, i.e. analytic falsehood).

Now of course many of these ultrafilters will be principal, generated by a singleton set $\{w\}$. Such an ultrafilter is the set of all Standard-Formalization propositions true at w. There is nothing surprising about this: of course we would expect the set of all propositions true at a given world w to be a maximal consistent set. But now assume—as we will—that our ambient set theory has Choice. Then at least one of the ultrafilters of B, call it N, is nonprincipal. Like any other ultrafilter of B, N is still a maximal consistent set of Standard-Formalization propositions. But since it is not principal, it has no least member, and there is no world w such that N is the set of all propositions true at w. So evidently there is at least one "possible way things might be" that does not correspond to any of the (antecedently given) possible worlds. Among ways things might be, N and its nonprincipal ilk are second-class citizens.

What should a defender of the Standard Formalization say about such second-class ways that things might be? I can think of two ways open. One way is to say that the ambient set theory does not have Choice. Then it could be consistently maintained that the algebra of propositions has no nonprincipal ultrafilters. But this seems a high price to pay, given the widely acknowledged utility of Choice in proving theorems. And anyway, why should semantic theory get to dictate what ambient set theory we use?

The other way open is to argue that some maximal consistent sets of propositions aren't *really* possible ways things could be, and therefore, when we are trying to tell two meanings apart, we don't care what extensions they have at them. I can't begin to imagine how such an argument would go, though the reader is of course welcome to try to develop one.

But why take on this burden? Why not just sidestep the whole problem by simply adopting Soft Actualism instead? I'm not aware of any persuasive arguments against it. Rather, the general acceptance of the Standard Formalization seems to have come about as a consequence of an accident of history, viz. that Montague happened to borrow Kripke's 1963 semantics for S5 instead of his 1959 one. In fact the proposal I am leading up to will be a form of Soft Actualism, so the existence of nonprincipal ultrafilters will not be a problem.

5 Soft Actualism Refined

But what about Paris Hilton and the Riemann Hypothesis? As formulated algebraically in (8), Soft Actualism shares with the Standard Formalization the problem that equivalent propositions are identical. Why? Simply because in both cases, entailment is being modelled by the order on a boolean algebra, and orders are antisymmetric. It's time to meet this problem head on.

In classical logics, the set of sentences does not form a boolean algebra under entailment. To get one you have to "divide out by logical equivalence", i.e. form the Lindenbaum algebra. Why bother to carry out this construction? Well, if you only care about sentences up to equivalence, it is a perfectly reasonable thing to do. But in our dealings with propositions, things are different. We still need boolean operations, in order to give meanings to the logical words like *and* and *or*, and we still want ultrafilters to do duty for possible worlds. What we definitely do *not* want is for entailment to be antisymmetric.[11] In short, what we want is something like a boolean algebra, but without the antisymmetry. Fortunately, there is just such a thing: a *boolean preordered algebra*, or (for short) a *boolean prealgebra*.[12] These were described, rather telegraphically, in Fox and Lappin 2001, Fox et al. 2002, and Fox and Lappin 2005 under the name *boolean prelattices*[13]. Here I present them in a more leisurely fashion.

(14) **Definition (Equivalence in a Preorder)**
Let $\sqsubseteq$ be a preorder on a set B. The **equivalence induced by $\sqsubseteq$**, written $\equiv_\sqsubseteq$, is defined by $a \equiv_\sqsubseteq b$ iff $a \sqsubseteq b$ and $b \sqsubseteq a$.

The subscript is omitted when no confusion can arise. It's easy to see that the equivalence induced by a preorder is indeed an equivalence relation.

(15) **Definition: Boolean Prealgebra**
A **boolean prealgebra** is a set equipped with a preorder $\models$; two nullary operations Truth and Falsity; one unary operation not$'$; and

---

[11]The central importance to semantic theory of avoiding the antisymmetry of entailment was pointed out to Shalom Lappin and the author by Howard Gregory in personal communication.

[12]Categorists call these *strict boolean categories*, and then dismiss them on the grounds that up to categorical equivalence they are the same thing as boolean algebras.

[13]These were used provide a model theory for a logic called FIL (fine-grained intensional logic). The present proposal can be seen as an attempt to overcome certain problematic aspects of FIL as a theory of natural language semantics (see Fox and Lappin 2005 and Pollard in preparation for discussion).

three binary operations and′, or′, and if′ ... then′ ..., such that, for all p, q, and r,

a. Truth: $p \models \text{Truth}$
b. Falsity: $\text{Falsity} \models p$.
c. and′-elimination: (i) $(p \text{ and}' q) \models p$; and (ii) $(p \text{ and}' q) \models q$.
d. and′-introduction: If $p \models q$ and $p \models r$, then $p \models (q \text{ and}' r)$.
e. or′-introduction: (i) $p \models (p \text{ or}' q)$; and (ii) $q \models (p \text{ or}' q)$.
f. or′-elimination: If $p \models r$ and $q \models r$, then $(p \text{ or}' q) \models r$.
g. Modus Ponens: $((\text{if}' p \text{ then}' q) \text{ and}' p) \models q$.
h. Deduction: If $(r \text{ and}' p) \models q$, then $r \models (\text{if}' p \text{ then}' q)$.
i. Negation: $\text{not}' p \equiv (\text{if}' p \text{ then}' \text{Falsity})$
j. Double Negation: $(\text{not}' (\text{not}' p)) \models p$

Later, the boolean prealgebra we care about is going to be used to model the entailment relation on propositions *qua* declarative sentence meanings; Truth is going to be some necessarily true proposition and Falsity some necessarily false one; the other boolean operations are going to be the meanings of the English logical words of the same spelling (less the prime).

The names given to the constraints on the boolean operations are chosen from logic rather than algebra as a gentle reminder of the origins of classical propositional logic as an attempt to codify the laws of valid natural-language argumentation. In algebraic terms: Truth is a top (greatest element); Falsity a bottom (least element); and′ a meet (greatest lower bound); or′ a join (least upper bound); if′ ... then′ a relative pseudocomplement; and not′ a pseudocomplement. Double negation makes the algebra (so far just a heyting prealgebra, i.e. a bicartesian closed preorder) boolean (and so we can drop the 'pseudo'-prefixes).

The fundamental fact about boolean prealgebras is that any equalities we expect to obtain in a boolean algebra obtain here too, but *only up to (induced) equivalence*; double negation is a case in point here. To put it another way: a boolean algebra is just a boolean prealgebra in which the preorder is antisymmetric (i.e. $\equiv$ is equality).

Boolean prealgebras are a special case of a still more general notion, viz. **preordered algebras**:

(16) **Definition: Preordered Algebra**

A **preordered algebra** is a set with both a preorder and an algebraic structure, such that the algebra operations are **tonic** (either monotonic or antitonic) on each of their arguments.

This fact can be expressed in more explicit form as follows:

(17) **Theorem (Tonicity of Boolean Operations)**
  For all members p, q, r of a boolean prealgebra, if $p \models q$, then:

  a. (i) $(p \text{ and}' r) \models (q \text{ and}' r)$, and (ii) $(r \text{ and}' p) \models (r \text{ and}' q)$
  b. (i) $(p \text{ or}' r) \models (q \text{ or}' r)$, and (ii) $(r \text{ or}' p) \models (r \text{ or}' q)$
  c. $(\text{if}' \ q \ \text{then}' \ r) \models (\text{if}' \ p \ \text{then}' \ r)$
  d. $(\text{if}' \ r \ \text{then}' \ p) \models (\text{if}' \ r \ \text{then}' \ q)$
  e. $(\text{not}' \ q) \models (\text{not}' \ p)$

An immediate consequence of tonicity is the following highly restrictive form of substitutivity:

(18) **Corollary (Substitutivity with respect to Booleans)**
  For all members p, q, r of a boolean prealgebra, if $p \equiv q$, then:

  a. (i) $(p \text{ and}' r) \equiv (q \text{ and}' r)$, and (ii) $(r \text{ and}' p) \equiv (r \text{ and}' q)$
  b. (i) $(p \text{ or}' r) \equiv (q \text{ or}' r)$, and (ii) $(r \text{ or}' p) \equiv (r \text{ or}' q)$
  c. $(\text{if}' \ q \ \text{then}' \ r) \equiv (\text{if}' \ p \ \text{then}' \ r)$
  d. $(\text{if}' \ r \ \text{then}' \ p) \equiv (\text{if}' \ r \ \text{then}' \ q)$
  e. $(\text{not}' \ q) \equiv (\text{not}' \ p)$

Now in the case we will be concerned with, we use a boolean prealgebra to model the set of propositions. In that case, the preorder $\models$ models the entailment relation, and the algebra operations model the meanings of the corresponding English logic words. Thus replacing, e.g. a conjunct of a conjunctive English sentence, or the antecedent of a conditional English sentence, by a sentence with an equivalent meaning is predicted to preserve equivalence (mutual entailment) of the meaning of the whole sentence, and therefore truth as well. But it is unreasonable to expect substitutivity to hold of other propositional operations (e.g. the meaning of *Paris Hilton believes that* ...), because there is no reason to expect propositional operations in general to be tonic. To do so would be of the same order of unreasonableness as expecting every function of a real variable to be either nondecreasing or nonincreasing.

Now the notion of an ultrafilter generalizes straightforwardly from boolean algebras to boolean prealgebras:

(19) **Definition (Ultrafilter of a Boolean Prealgebra)**

A subset w of a boolean prealgebra B is called an **ultrafilter** iff, for all $p, q \in B$[14]:

   a. if $p, q \in w$ then $(p \text{ and}' q) \in w$;

   b. if $p \in w$ and $p \models q$, then $q \in w$; and

   c. either (exclusive disjunction) $p \in w$ or $(\text{not}' p) \in w$.

The following generalizes a standard result about boolean algebras:

(20) **Theorem (Ultrafilters and Boolean Homomorphisms)**

A subset of a boolean prealebra is an ultrafilter iff its characteristic function is a boolean homomorphism to the two-element boolean (pre)algebra.

It is obvious on a moment's reflection that the Stone Representation Theorem does *not* generalize to boolean prealgebras, since powerset algebras are antisymmetric, and therefore the function that maps each element to the set of ultrafilters containing it is not in general one-to-one.[15] However, the principal lemma Stone used to prove it does generalize:

(21) **Stone's Lemma (There are Enough Ultrafilters)**

If p and q are elements of a boolean prealgebra and $p \not\models q$, then there is an ultrafilter w such that $p \in w$ but $q \notin w$.

This has the following important consequence:

(22) **Corollary (Propositional Equivalence and Ultrafilters)**

If p and q are elements of a boolean prealgebra, then $p \equiv q$ iff for every ultrafilter w, $p \in w$ iff $q \in w$.

In particular, in the case of the prealgebra of propositions, if p and q are propositions, then they are equivalent (mutually entailing) iff they are true in the same worlds.

To summarize, we can now revise the algebraicization of Soft Actualism to the following form:

---

[14]Our choice of metavariables serves as a reminder that we are now thinking of the prealgebra elements as propositions and the ultrafilters as possible worlds.

[15]However, viewed categorically rather than algebraically, this function is a boolean category iso onto its image.

(23) **Soft Actualism in Algebraic Form (Revised Version)**

    a. Propositions are elements of a boolean **pre**algebra.

    b. Entailment is the **pre**order.

    c. Possible worlds are the ultrafilters.

    d. 'p is true in w' means $p \in w$.

The only changes in this formulation from the preliminary version (8) are to replace the boolean algebra by a boolean prealgebra and the order by the preorder. With this change, which will be incorporated as a central feature of my proposal, Algebraic Soft Actualism solves both the problem of equivalent propositions and the problem with nonprincipal ultrafilters. In particular, equivalent propositions, even though true in exactly the same possible worlds, need not be identical. An analogous move is not available for the Standard Formalization because there the propositions are a powerset algebra with entailment as subset inclusion, and there is just no getting around the fact that subset inclusion is antisymmetric.

(24) Algebraic Soft Actualism solves both:

    a. **the problem with equivalent propositions** (they need not be equal), **and**

    b. **the problem with nonprincipal ultrafilters** (they count as worlds).

    c. **No analog of this solution exists for the Standard Formalization**: there is just no getting around the fact that subset inclusion is antisymmetric!

    d. The remaining task is to incorporate Algebraic Soft Actualism into a formal theory of NL meaning.

The remainder of this paper is devoted to laying out a proposal incorporating this form of Soft Actualism into a logical theory that preserves the desirable features of the Standard Formalization (2) while excluding the problematic ones (3). We begin by describing the lambda calculus underlying the logic within which the theory will be expressed.

6   The Underlying Typed Lambda Calculus

Our point of departure is a (simply) typed lambda calculus (hereafter, TLC) along the lines of Henkin 1950 and Gallin 1975. The only difference is that we

follow Lambek and Scott(1986) in having finite product types, both nullary (the unit type 1) and binary $(A \times B)$[16].

(25) **TLC overview**

    a. Syntactically, a TLC consists of:

        i. types;

        ii. terms of each type; and

        iii. an equivalence relation on terms.

    b. In a (set-theoretic) interpretation:

        i. types denote sets;

        ii. a term denotes a member of the set denoted by its type; and

        iii. equivalent terms denote the same thing.

(26) **Types of the Underlying Typed Lambda Calculus**

    a. Each basic type is a type;

    b. 1 is a type;

    c. if A and B are types, so is $A \times B$; and

    d. if A and B are types, so is $A \Rightarrow B$.

(27) **Terms of the Underlying Typed Lambda Calculus**

    a. Each basic constant of type A is a term of type A;

    b. For each type A there is a countably infinite set of variables x_i^A $(i \in \omega)$ of type A ;

    c. $* :: 1$;

    d. For all $f :: A$ and $g :: B$, $(f, g) :: (A \times B)$;

    e. For all $h :: (A \times B)$, $\pi_{A,B}(h) :: A$ and $\pi'_{A,B}(h) :: B$;

    f. For all $f :: A \Rightarrow B$ and $a :: A$, $f(a) :: B$;

    g. For all $b :: B$, $\lambda_{x \in A} b :: A \Rightarrow B$.

In the preceding, '::' is to be read as 'is of type'.

In the following, '=' is used as a metalanguage name for the term equivalence relation:

---

[16]Thus the underlying type logic is positive intuitionistic propositional logic.

(28) **Term Equivalence for the Underlying Typed Lambda Calculus**

    a. (equivalence relation)

        i. $\vdash a = a$ (reflexivity);

        ii. $a = b \vdash b = a$ (symmetry);

        iii. $a = b, b = c \vdash a = c$ (transitivity);

    b. (congruence with respect to the term constructors)

        i. $a = c, b = d \vdash (a, b) = (c, d)$;

        ii. $f = g, a = b \vdash f(a) = g(b)$;

        iii. $a = b \vdash \lambda_x a = \lambda_x b$;

    c. (products)

        i. $\vdash a = *$ for all $a :: 1$;

        ii. $\vdash \pi(f, g) = f$;

        iii. $\vdash \pi'(f, g) = g$;

        iv. $\vdash (\pi(h), \pi'(h)) = h$;

    d. (conversion)

        i. $(\beta) \vdash [\lambda_{x \in A} \phi[x]](a) = \phi[a]$ if $a :: A$ is substitutable for x[17];

        ii. $(\eta) \vdash \lambda_{x \in A} f(x) = f$ for all $f :: A \Rightarrow B$ provided x does not occur freely in f; and

        iii. $(\alpha) \vdash \lambda_{x \in A} \phi[x] = \lambda_{y \in A} \phi[y]$ if y is substitutable for x.

(29) **Interpretation of the Underlying Typed Lambda Calculus**

A (**set-theoretic**) **interpretation** I[18] assigns to to each type A a set $I(A)$ and to each basic constant $a :: A$ a member $I(a)$ of $I(A)$, subject to the following constraints:

    a. $I(1) = \{0\}$;[19]

    b. $I(A \times B) = I(A) \times I(B)$;

    c. $I(A \Rightarrow B) \subseteq I(A) \Rightarrow I(B)$.[20]

---

[17]'Substitutable for x' means that no free variable occurrence in a or y becomes bound upon substitution for x.

[18]More generally, typed lambda calculi can be interpreted into (strict cartesian closed) categories which need not be set-theoretic. In the more general setting, $I(A)$ is an object of the category and for a term $\alpha :: A$, $I(\alpha)$ is an arrow from the terminal object $I(1)$ to $I(A)$. For expository simplicty, I speak as if the set-theoretic interpretations are the only ones, but there is no theoretical justification for this restriction.

[19]I do not distinguish notationally between the type 1 and its set-theoretic interpretation $1 = \{0\}$. Analogous remarks apply to the type constructors $\times$ and $\Rightarrow$.

[20]As in Henkin 1950, the set inclusion in clause (3) can be proper, as long as there are enough functions to interpret all functional terms.

(30) **Definition**

A **variable assignment** relative to an interpretation I is a function α that maps each variable to a member of the set that interprets its type, i.e. for each $x :: A$, $\alpha(x) \in I(A)$.

(31) **Extending an Interpretation Relative to an Assignment**

Given a variable assignment α relative to an interpretation I, there is a unique extension of I, denoted by I_α, that assigns interpretations to all terms, such that:

a. For each variable x, $I_\alpha(x) = \alpha(x)$;

b. for each basic constant a, $I_\alpha(a) = I(a)$;

c. $I_\alpha(*) = 0$;

d. for each $f :: A$ and $g :: B$, $I_\alpha((f, g))$ is $\langle I_\alpha(f), I_\alpha(g) \rangle$;

e. for each $h :: (A \times B)$, $I_\alpha(\pi(h))$ is the first component (= projection onto $I(A)$) of $I_\alpha(h)$; and $I_\alpha(\pi'(h))$ is the second component (= projection onto $I(B)$) of $I_\alpha(h)$;

f. for each $f :: A \Rightarrow B$ and $a :: A$, $I_\alpha(f(a)) = (I_\alpha(f))(I_\alpha(a))$; and

g. for each $b :: B$, $I_\alpha(\lambda_{x \in A} b)$ is the function from $I(A)$ to $I(B)$ that maps each $a \in I(A)$ to $I_\beta(b)$, where β is the variable assignment that coincides with α except that $\beta(x) = a$.

Note that for any term a, $I_\alpha(a)$ depends only on the restriction of α to the free variables of a. In particular, if a is a constant (i.e. a closed term), then $I_\alpha(a)$ is independent of α so we can simply write $I(a)$. Thus, an interpretation for the basic types and basic constants extends uniquely to all types and all constants. Moreover, in any such interpretation, the interpetations of equivalent terms are always identical.

7   From Typed Lambda Calculus to Higher-Order Logic

In typed lambda calculi such as the one just introduced, the equality symbol denoting term equivalence is a metalanguage symbol, not a symbol of the calculus; and correspondingly, an "equation" between two terms is not itself a term: the equivalence of two terms can only be asserted in the metalanguage, not in the calculus itself.

Following Henkin (1950) and Lambek and Scott (1986), we now turn our typed lambda calculus into a higher-order predicate logic as follows:

20

(32) **From TLC to HOL**

    a. Assume a basic type Bool of **truth values**.

    b. For each type A, add a basic constant $=_A :: (A \times A) \Rightarrow$ Bool.

    c. The equations (28) are no longer taken as defining an equivalence relation on terms, but rather as object-language axioms about equality (of whatever the terms denote).

Now all the usual (intuitionistic) connectives and quantifiers are definable:[21]

(33) **Definitions of Logical Constants in HOL**

    a. $\mathsf{true} =_{\mathrm{def}} * = *$

    b. $\forall_{x \in A}\phi =_{\mathrm{def}} \lambda_{x \in A}\phi = \lambda_{y \in A}\mathsf{true} \quad$ for $\phi \in$ Bool

    c. $\mathsf{false} =_{\mathrm{def}} \forall_{x \in \mathrm{Bool}}x$

    d. $\wedge =_{\mathrm{def}} \lambda_{(x,y) \in \mathrm{Bool} \times \mathrm{Bool}}(x, y) = (\mathsf{true}, \mathsf{true})$

    e. $\rightarrow \; =_{\mathrm{def}} \lambda_{(x,y) \in \mathrm{Bool} \times \mathrm{Bool}}(x = x \wedge y)$

    f. $\leftrightarrow \; =_{\mathrm{def}} \lambda_{(x,y) \in \mathrm{Bool} \times \mathrm{Bool}}[(x \rightarrow y) \wedge (y \rightarrow x)]$

    g. $\neg =_{\mathrm{def}} \lambda_{x \in \mathrm{Bool}}(x \rightarrow \mathsf{false})$

    h. $\vee =_{\mathrm{def}} \lambda_{(x,y) \in \mathrm{Bool} \times \mathrm{Bool}}\forall_{t \in \mathrm{Bool}}(((x \Rightarrow t) \wedge (y \Rightarrow t)) \Rightarrow t)$

    i. $\exists_{x \in A}\phi =_{\mathrm{def}} \forall_{t \in \mathrm{Bool}}(\forall_{x \in A}(\phi \Rightarrow t) \Rightarrow t)$

In spite of the suggestive name Bool, so far this higher-order logic is only intuitionistic.[22] To make it classical, we add (again following Lambek and Scott) the axiom[23]

(34) **Axiom of Excluded Middle**
    $\vdash \forall_{t \in \mathrm{Bool}}(t \vee \neg t)$

We also need the following axiom, explicitly rejected by Church but added by Henkin (for completeness relative to Henkin models):

---

[21]Church went in the other direction, introducing negation, disjunction, and universal quantification as basic constants and then defining equality via Leibniz's Law:

$$a =_A b =_{\mathrm{def}} \forall_{f \in A \Rightarrow \mathrm{Bool}}[f(a) \rightarrow f(b)]$$

[22]This is reflected by the definitions of false, $\vee$, and $\exists$. In the presence of (34), these reduce to the familiar definitions as DeMorgan duals of true, $\wedge$, and $\forall$, respectively.

[23]Caution: This axiom looks as if it makes the logic not only classical but also bivalent. In fact it does give bivalence for set-theoretic models, but not for general categorical ones.

(35) **Axiom of Boolean Extensionality**

$$\vdash \forall_{(x,y)\in\mathrm{Bool}\times\mathrm{Bool}}[(x \leftrightarrow y) \rightarrow (x = y)]$$

This axiom equates bi-implication with boolean equality. Church deliberately omitted this axiom because he had a more intensional notion of the boolean type: for him it was a type of propositions, not just truth values. But for us, this axiom is not problematic, because in our semantic theory we will add another basic type Prop for propositions. For our purposes, two truth values (i.e. members of $I(\mathrm{Bool})$) will be just fine.

The next ingredient of our HOL, again borrowing from Lambek and Scott, provides for (separation) subtypes:

(36) **Subtypes and Characteristic Functions**

    a. Besides (26), we have one more way of forming types: if $a :: A \Rightarrow$ Bool is closed, then A_a is a type (intuitively, the subtype of A whose members satisfy the predicate a);

    b. Besides (27), we have one more way of forming terms: if $a :: A \Rightarrow$ Bool is closed, then $\mathsf{emb}_a :: A_a \Rightarrow A$; and

    c. we have two additional axiom schemas

        i. $\vdash \forall_{(y,z)\in A_a \times A_a}[(\mathsf{emb}(y) = \mathsf{emb}(z)) \rightarrow y = z]$

        ii. $\vdash \forall_{(x,a)\in A\times(A\Rightarrow\mathrm{Bool})}(a(x) \leftrightarrow \exists_{y\in A_a} x = \mathsf{emb}_a(y))$

Intuitively: emb_a is the subset embedding of A_a into A; the axiom schemas say that emb_a is injective and that a is the characteristic function of A_a. More carefully put: for any set in the model that interprets a type A, any subset of that set whose characteristic function is lambda-definable (i.e. which interprets a closed term of type $A \Rightarrow$ Bool) is also in the model.[24]

We require the following axiom to ensure that there are really two truth values (not just one):

(37) **Nondegeneracy**

$$\vdash \neg(\mathsf{true} = \mathsf{false})$$

Finally, we need to ensure that we will be able to prove an object-language analog of Stone's Lemma for the specific (internal) boolean prealgebra we are using to model the set of propositions. There are numerous options here. For example, we can impose one of the standard higher-order versions of Choice, from which Stone's Lemma (for *all* boolean prealgebras) is known to follow. Alternatively, we can just directly impose

---

[24]Together with the rules and axioms already given, these axioms says of a categorical model that it is a *boolean topos* with $I(\mathsf{true}) : I(1) \rightarrow I(\mathrm{Bool})$ as its subobject classifier.

Stone's Lemma.[25] Or weaker still, we can just impose Stone's Lemma for the boolean prealgebra of propositions, i.e.

$$\vdash \forall_{p,q \in \text{Prop}}[(p \not\models q) \rightarrow \exists_{s \in (\text{Prop} \Rightarrow \text{Bool})}(u(s) \wedge s(p) \wedge \neg s(q))]$$

where Prop is the type of propositions, $\models$ is the constant of type (Prop $\times$ Prop) $\Rightarrow$ Bool that denotes the entailment relation, and u is a certain term (see following section) of type (Prop $\Rightarrow$ Bool) $\Rightarrow$ Bool) which encodes the property (of sets of propositions) of being an ultrafilter.

8   A Hyperintensional Semantic Theory

8.1   First Steps

Now that we have a suitable logic, we can use it to precisely formalize a Soft Actualist semantic theory that retains the desirable characteristics of standard possible-worlds semantics while eliminating the problematic aspects discussed earlier. We start by choosing our basic nonlogical types. Instead of one (Henkin) or two (Gallin), we have three: Ind (individual concepts), Ent (entities, the things that can be extensions of individual concepts), and Prop (propositions). The type Bool of things (truth values) that can be extensions of propositions has already been supplied by the HOL.[26] Crucially, there is no basic type World.

(38) **Basic Nonlogical Types for Hyperintensional Semantics**

    a. Ent (entities)

    b. Ind (individual concepts, the hyperintensions that have entities as their extensions)

    c. Prop (propositions, the hyperintensions that have truth values as their extensions)

Although we will be able to construct Carnap/Montague-style intensions in our theory, we will not use them to model meanings (Fregean senses). Instead, we use **hyperintensions**, which are of the following types:

---

[25]This is known to be equivalent to the Boolean Prime Ideal Theorem and weaker than Choice in toposes.

[26]The basic types Ent and Prop (but not Ind) should be reminiscent of Thomason's (1980) Intentional Logic. This and other points of comparison with Thomason's system are discussed in Pollard (in preparation).

(39) **The set of hyperintensional types is defined as follows:**

    a. 1 is a hyperintensional type[27];

    b. Ind and Prop are hyperintensional types;

    c. If A and B are hyperintensional types, so are $A \times B$ and $A \Rightarrow B$;

    d. If $a :: A \Rightarrow \text{Bool}$ is closed and A is a hyperintensional type, so is A_a.

    e. Nothing else is a hyperintensional type.

In short, the hyperintensional types are obtained by closing the set of basic hyperintensional types under the TLC type constructors and subtyping.

For simplicity, let us assume that the syntactic part of our linguistic theory provides the basic syntactic types NP, It, There[28], N, and S, and that $\times$ and $\Rightarrow$ are the only syntactic type constructors.[29] Then at the level of types, the mapping from linguistic expressions to their senses (hyperintensions) is defined recursively as follows:

(40) **The Mapping from NL Syntactic Types to Meaning Types**

    a. $\mathsf{Sem}(\text{It}) = \mathsf{Sem}(\text{There}) = 1$;

    b. $\mathsf{Sem}(\text{NP}) = \text{Ind}$;

    c. $\mathsf{Sem}(\text{S}) = \text{Prop}$;

    d. $\mathsf{Sem}(\text{N}) = \text{Ind} \Rightarrow \text{Prop}$

    e. $\mathsf{Sem}(X \times Y) = \mathsf{Sem}(X) \times \mathsf{Sem}(Y)$; and

    f. $\mathsf{Sem}(X \Rightarrow Y) = \mathsf{Sem}(X) \Rightarrow \mathsf{Sem}(Y)$.

    g. $\mathsf{Sem}(X_a) = \mathsf{Sem}(X)$.

Thus dummy pronouns are semantically vacuous; NPs (for simplicity, limited to names) express individual concepts; sentences express propositions; the TLC type constructors are preserved; and embedding a linguistic expression of a certain type into a supertype does not affect its meaning.[30]

---

[27] 1 is used as the meaning type for "semantically vacuous" saturated expressions, e.g. dummy pronouns.

[28] It and There are syntactic types for the dummy pronouns *it* and *there* respectively.

[29] In particular, we eschew Lambek's directional (/ and \) implications. Instead, we follow Curry (1961), de Groote (2001), Ranta (2002), Muskens (2003), and Pollard (2004a,b) in assuming that abstract syntactic combinatorics (or *tectogrammar*, to use Curry's term) is nondirectional, with word order determined by the interface between tectogrammar and *phenogrammar* (roughly, phonology broadly construed to include word order).

[30] The term-level syntax-to-meaning mapping sem is also 'structure-preserving' in the sense of being a cartesian-closed functor, i.e. it preserves pairing, projections, function application, and lambda-abstraction. See Pollard 2004a,b for discussion.

Again at the level of types, the mapping from hyperintensions to extensions is defined recursively as follows:[31]

(41) **Extensional types corresponding to hyperintensional types**

    a. $\mathrm{Ext}(1) =_{\mathrm{def}} 1$;

    b. $\mathrm{Ext}(\mathrm{Ind}) =_{\mathrm{def}} \mathrm{Ent}$;

    c. $\mathrm{Ext}(\mathrm{Prop}) =_{\mathrm{def}} \mathrm{Bool}$;

    d. $\mathrm{Ext}(A \times B) =_{\mathrm{def}} \mathrm{Ext}(A) \times \mathrm{Ext}(B)$;

    e. $\mathrm{Ext}(A \Rightarrow B) =_{\mathrm{def}} A \Rightarrow \mathrm{Ext}(B)$; and

    f. $\mathrm{Ext}(A_a) =_{\mathrm{def}} \mathrm{Ext}(A)$

The first clause has the effect that semantically vacuous linguistic expressions also have vacuous reference. The next two clauses have the effect that names refer to entities and sentences to truth values. Clause (d) has the effect that a pair of expressions (e.g. the two complements of a ditransitive verb taken as a unit) denotes the pair of references of the two expressions. The last clause says that the reference of an expression of a given type remains unchanged if the expression is embedded into a supertype.

The interesting clause is (e). Two examples suffice to motivate it.[32] First, consider an intransitive verb (type $\mathrm{NP} \Rightarrow \mathrm{S}$) such as barks, and let bark' = sem(barks) :: $\mathrm{Ind} \Rightarrow \mathrm{Prop}$. Then for any given world w (just what we mean by that will be spelled out presently), the reference of barks at w will be the extension at w of bark', which has type $\mathrm{Ind} \Rightarrow \mathrm{Bool}$ (essentially a set of individual concepts, viz. the ones which 'bark at w').[33] Second, consider a predicate such as it-is-obvious-that (type $\mathrm{S} \Rightarrow \mathrm{S}$)[34], and let obvious' = sem(it-is-obvious-that) :: $\mathrm{Prop} \Rightarrow \mathrm{Prop}$. Then for any world w, the reference

---

[31]Here we make the usual, but unjustified, simplifying assumption that every meaning has an extension at every world. In a refinement of the theory discussed in Pollard (in preparation), partial function types are used to account for the fact that some meanings (e.g. meanings of names of fictional characters) may lack extensions at some worlds.

[32]Here and henceforth, we engage in a systematic abuse of notation whereby an object-language term is used as a metalanguage name of the interpretation of the term into a set theoretic model I. This is to reduce the notational clutter that would result from constantly writing '$I(\alpha)$' in place of 'α'. For example, we say 'the verb barks expresses the property of individual concepts bark'' in place of the technically correct 'the verb $I(\text{barks})$ expresses the property of individual concepts $I(\text{bark'})$'.

[33]The fact that whether or not an individual concept i barks at w depends only on the extension of i at w, i.e. that bark' is an extensional property, will be captured by a meaning postulate.

[34]For simplicity we ignore the distinction between uncomplementized and complementized sentences.

of it-is-obvious-that at w will be the extension at w of obvious′, which has type Prop $\Rightarrow$ Bool (essentially a set of propositions, viz. the ones which are obvious at w).

What about the extensions themselves? Since the extension of a given hyperintension varies from world to world, it might appear that the lack of a basic type World is going to pose a problem. In fact it won't; we will return to this point too in the following subsection.

The time has come to deal with the relation that forms the central subject matter of NL semantics, viz. entailment. In a model of our theory, entailment is the interpretation of the basic object-language constant

$$\models :: (\text{Prop} \times \text{Prop}) \Rightarrow \text{Bool}$$

and equivalence of propositions is defined as mutual entailment:

$$\equiv \; =_{\text{def}} \; \lambda_{(p,q)}((p \models q) \wedge (q \models p))$$

For readers reared on Montague's IL or on Ty2, where entailment is modelled by the subset inclusion relation on propositions (qua sets of possible worlds), it might appear puzzling that there is a basic object-language constant ($\models$) which is interpreted in a model as the entailment preorder on the set of propositions. By comparison, in Ty2 (with types s and t renamed to World and Bool respectively, and World $\Rightarrow$ Bool the type for propositions qua sets of worlds), the constant (i.e. closed term) that is interpreted as entailment is $\lambda_{p \in \text{World} \Rightarrow \text{Bool}} \lambda_{q \in \text{World} \Rightarrow \text{Bool}} \forall_{w \in \text{World}}[p(w) \rightarrow q(w)]$, which is of course not a basic constant. This difference is directly reflective of the fundamental philosophical difference between the Standard Formalization, (where worlds are theoretical primitives, while propositions and the entailment relation are constructed) and our algebraic version of Soft Actualism, where the entailment relation on propositions is a theoretical primitive.

Another possible source of confusion is that there is another sense of the word *entailment* as model-theoretic semantic consequence: if ϕ and ψ are two sentences (in the sense of closed boolean terms) of our object language, then ϕ entails ψ in this sense provided, for every interpretation I, if $I(\phi) = \text{true}$, then $I(\psi) = \text{true}$. Assuming our proof theory is complete this is equivalent to $\vdash \phi \rightarrow \psi$. We will reserve the term *entailment* for natural language entailment, the empirical relation between declarative sentence meanings whose investigation is widely considered to be the central task of natural language semantics.

We now introduce nonlogical axioms which say of entailment that it is a preorder:

26

(42) **Preorder Axioms for Entailment**

    a. $\vdash \forall_p (p \models p)$

    b. $\vdash \forall_{p,q,r} (p \models q) \rightarrow ((q \models r) \rightarrow (p \models r)))$

Crucially, entailment is *not* antisymmetric; $\equiv$ **cannot** be proven equal to $=_{\text{Prop}}$.[35]

    Next we introduce the constants used to translate English 'logic words':

(43) **Translations of English "Logic Words"**

    a. truth :: Prop abbreviates the translation of an arbitrarily chosen necessarily true English sentence.

    b. falsity :: Prop abbreviates the translation of an arbitrarily chosen necessarily false English sentence.

    c. not' :: Prop $\Rightarrow$ Prop translates *it is not the case that*.

    d. and', or' :: (Prop $\times$ Prop) $\Rightarrow$ Prop are the respective translations of (the sentential conjunctions) *and* and *or*.

    e. if' ... then' translates *if ... then*.

and suitable nonlogical (!) axioms (meaning postulates) for them which ensure that in a model of the semantic theory, the interpretation of the type Prop forms a boolean prealgebra with the meanings of the logic words as the boolean operations (cf. 15):

(44) **Meaning Postulates for the Translations of English Logic Words**

    a. $\vdash \forall_p (p \models \mathsf{truth})$

    b. $\vdash \forall_p (\mathsf{falsity} \models p)$

    c. $\vdash \forall_{(p,q)} ((p \text{ and' } q) \models p)$
       $\vdash \forall_{(p,q)} ((p \text{ and' } q) \models q)$

    d. $\vdash \forall_{(p,q)} [((p \models q) \wedge (p \models r)) \rightarrow (p \models (q \text{ and' } r))]$

    e. $\vdash \forall_{(p,q)} (p \models (p \text{ or' } q))$
       $\vdash \forall_{(p,q)} (q \models (p \text{ or' } q))$

    f. $\vdash \forall_{(p,q,r)} [((p \models r) \wedge (q \models r)) \rightarrow ((p \text{ or' } q) \models r)]$

    g. $\vdash [((\text{if' } p \text{ then' } q) \text{ and' } p) \models q]$

    h. $\vdash \forall_{(p,q,r)} [((r \text{ and' } p) \models q) \rightarrow (r \models (\text{if' } p \text{ then' } q))]$

    i. $\vdash \forall_p ((\text{not' } p) \equiv (\text{if' } p \text{ then' falsity}))$

    j. $\vdash \forall_p [(\text{not' } (\text{not' } p)) \models p]$

---

[35] Cf. (35), which says $\leftrightarrow$ **is** equal to $=_{\text{Bool}}$.

8.2 Constructed Worlds

Now we have meanings, but how can we have any notion of meanings having extensions at worlds if we don't have worlds? In order to conduct the usual semantic business with worlds (modality, counterfactuals, the taking of extensions at worlds, etc.), we need to *have* worlds in the theory. This might seem problematic, since we have no basic type for them. However, the existence of lambda-definable subtypes comes to our rescue. The fact of the matter is: we *do* have worlds:

(45) **Without Worlds, how can Meanings have Extensions?**

    a. We *do* have worlds, but they are hiding. Where are they hiding?

    b. Well, worlds are certain sets of propositions, so they are a subset of the set that interprets Prop $\Rightarrow$ Bool. Which subset?

    c. Answer: the subset whose members are ultrafilters of the boolean prealgebra that interprets Prop.

    d. But this just a set-theoretic construction on models, isn't it? Don't we really need a *type* of worlds in the logical theory?

    e. Yes, but we have such a type: World is the type

$$(\text{Prop} \Rightarrow \text{Bool})_u$$

where $u :: (\text{Prop} \Rightarrow \text{Bool}) \Rightarrow \text{Bool}$ is the predicate on sets of propositions such that $u(s)$ says of s that it is an ultrafilter!

This is possible because ultrafilterhood is a definable predicate of sets of propositions:

(46) **Being an Ultrafilter is a Lambda-Definable Predicate:**

    a. u is $\lambda_s[a(s) \wedge b(s) \wedge c(s)]$ where

        i. $a(s)$ says s is closed under entailment;

        ii. $b(s)$ says s is closed under and'; and

        iii. $c(s)$ says that for each proposition p, exactly one of p and $(\mathsf{not'}p)$ is in s.

    b. To be explicit:

        i. $a(s)$ is $\forall_{(p,q)}[(s(p) \wedge p \models q) \rightarrow s(q)]$;

        ii. $b(s)$ is $\forall_{(p,q)}[(s(p) \wedge s(q)) \rightarrow s(p \text{ and' } q)]$; and

        iii. $c(s)$ is $\neg s(\mathsf{falsity'}) \wedge \forall_p(s(p) \vee s(\mathsf{not' } p))$.

So we really had worlds all along. This means we are in a position to say what it means for a proposition to be true at one of them.

(47) **How to Say "p is True at w"**

a. In the Standard Formalization: $p(w)$.

b. Under our proposal: the first guess would be $w(p)$, but this is ill-typed since $w :: \text{World}$, not $w :: \text{Prop} \Rightarrow \text{Bool}$.

c. But $\text{World} = [\text{Prop} \Rightarrow \text{Bool}]_u$ where u is defined as in (46), so $\mathsf{emb}_u :: \text{World} \Rightarrow (\text{Prop} \Rightarrow \text{Bool})$ denotes the embedding of the set of worlds into the set of sets of propositions.

d. So the right way to say 'p is true at w' is $\mathsf{emb}_u(w)(p)$.

e. For this reason, I will usually abbreviate $\mathsf{emb}_u(w)(p)$ to $p@w$.

8.3 Extensions of Hyperintensions at Worlds

Now that we know what worlds are and what it means for a proposition to be true at one of them, the time has come to make sense of the notion of a meaning having an extension at a world. Remember: we can't just "evaluate" the meaning at the world, since meanings are hyperintensions, not intensions! Instead, we treat the notion of extension as a family of functions (parametrized by the set of hyperintensional types)

$$\mathsf{ext}_A :: A \Rightarrow (\text{World} \Rightarrow \text{Ext}(A))$$

that take hyperintensions to intensions (functions from worlds to extensions of the appropriate type). We will get specific about this presently, but first it is necessary to consider a fundamental asymmetry between the two basic hyperintensional types Prop and Ind. It is generally agreed that the extension of a meaning (Fregean sense) is determined by how things are, or to put it another way, that a meaning and a world jointly determine an extension. For us, meanings are hyperintensions, and for the nonbasic hyperintensional types, we will be able to recursively define the extension determined by a hyperintension and a world in terms of hyperintensions of lower types whose extensions are were determined at an earlier stages of the recursion. So what about the basic hyperintensional types? Well, for propositions, we have already tipped our hand: the extension of p at w will be $p@w =_{\text{def}} \mathsf{emb}_u(w)(p)$. But what about individual concepts? Ind is a basic type, so if i is an individual concept and w a world, the extension of i at w cannot be calculated from a recursive definition. Rather, what the extension of i is at w must be something about w. To get at this intuition,

we assume there is a basic constant has-as-extension $:: (\text{Ind} \times \text{Ent}) \Rightarrow \text{Prop}$; i.e. has-as-extension$(i, e)$ is the proposition that i has e as its extension. So for every world w, this proposition should be true at w just in case $\text{ext}_{\text{Ind}}(i)(w)$ is e. We express this as a nonlogical axiom:

(48) **The Determination of Extensions of Individual Concepts at Worlds**

$\vdash \forall_{i \in \text{Ind}, e \in \text{Ent}, w \in \text{World}}[(\text{ext}_{\text{Ind}}(i)(w) = e) \leftrightarrow \text{ has-as-extension}(i, e)@w]$

From this it follows that for hyperintensions of both of the basic hyperintensional types, the extension at any world is directly given by which propositions are true at that world.

Now we can explain how extensions at worlds are determined for hyperintensions of higher types.

(49) **Axioms for Extensions of Hyperintensions of Nonbasic Types**

a. $\vdash \forall_{w,p}(\text{ext}_{\text{Prop}}(p)(w) = p@w)$
b. $\vdash \forall_{w,i}[(\text{ext}_{\text{Ind}}(i)(w) = e) \leftrightarrow \text{ has-as-extension}(i, e)@w]$
c. $\vdash \text{ext}_1(w)(*) = *$
d. $\vdash \forall_{w,c}(\text{ext}_{A \times B}(c)(w) = (\text{ext}_A(\pi(c))(w), \text{ext}_B(\pi'(c))(w)))$
e. $\vdash \forall_{w,f}(\text{ext}_{A \Rightarrow B}(f)(w) = \lambda_{x \in A}\text{ext}_B(f(x))(w))$

The first two clauses, the base of the recursion, just repeat (47d) and (48) respectively. Clause (c) just says vacuous meanings have vacuous extensions. Clause (d) says extensions of pair-meanings are just the pairs of the corresponding extensions.

The last clause (49e) is the interesting one, because it makes explicit an important respect in which the hyperintensional notion of compositionality differs from that of the Standard Formalization (and of Frege). On the standard view, reference is compositional: it is possible to determine the reference of an expression from the references of its immediate constituents and how they are put together. But there is something about this view that is radically at odds with a fundamental fact about how language works: We can figure out what an expression means without knowing what the contingent facts of the world are! If we hear someone say that Paris Hilton believes snow is white, we don't have to look out the window to figure out what proposition was expressed. In short, *meaning* is compositional. Given that reference is jointly determined by meaning and the world, there is simply no basis for claiming *reference* is compositional: we figure out compositionally what the *meaning* of *Paris Hilton believes snow is white*

first, and *then*, if we want to determine the truth value, we try to determine from the world whether the proposition that snow is white is the sort of thing Paris Hilton believes. The *extension* (truth value) of that proposition simply does not come into it.[36]

8.4 Extensional Properties

For A a hyperintensional type, we will call a closed term of type $A \Rightarrow \mathsf{Prop}$ an A-**predicate** and its interpretation an A-**property**. For many properties that serve as NL meanings, for any world w, whether a meaning has that property at w depends only on the meaning's extension at w. Such properties (and by extension, predicates whose interpretations are such properties) are called **extensional**. For example *barks* expresses an extensional property of individual concepts, but *was believed by the ancients to be Phosphorus* does not; *is false* expresses an extensional property of propositions, but *it is obvious that* does not. We can characterize this notion of extensionality by the meaning postulate template in (50a).[37]

(50) **Extensionality for Predicates**

    a. We define an A-predicate f to be **extensional** iff

$$\vdash \forall_{w,a,a'}[(\mathsf{ext}(a)(w) = \mathsf{ext}(a')(w)) \rightarrow (f(a)@w = f(a')@w)]$$

    b. More generally, a closed hyperintensional term $f :: A \Rightarrow B$ is called **extensional** iff:

$$\vdash \forall_{w,a,a'}[(\mathsf{ext}(a)(w) = \mathsf{ext}(a')(w)) \rightarrow (\mathsf{ext}(f(a)(w) = \mathsf{ext}(f(a')(w))$$

    To take another kind of example, NL determiners are (A-parametrized) families of extensional $(A \Rightarrow \mathsf{Prop}) \times (A \Rightarrow \mathsf{Prop})$-predicates: for two A-predicates P and Q and a world w, whether every P is a Q at w depends only on the extensions of P and Q at w. In our theory, this fact is derivable from the following nonlogical axiom scheme:

---

[36] In order to square examples of this kind with his notion of compositionality, Frege had to resort to claiming that utterances of sentences in certain contexts had the customary sense of the sentence as the reference. Our account has no need for this sleight of hand.

[37] We call (50a) a template rather than a schema because we don't want *every* substitution instance to be a nonlogical axiom, just the ones resulting from the replacement of f by predicates whose extensionality we want the theory to assert.

(51) **Meaning Postulate for every**

$$\vdash \forall_{w,P,Q}[\text{every'}(P,Q)@w \leftrightarrow \forall_x(\text{ext}(P)(w)(x) \rightarrow \text{ext}(Q)(w)(x))]$$

8.5 Equivalence Revisited

It is noteworthy that even though meanings are not intensions according to our theory, there is still a place for intensions, because for any hyperintensional term $a :: A$, $\text{ext}(a)$ is of type World $\Rightarrow \text{Ext}(A)$. In other words, Ext is interpreted as a (type-parametrized) function from hyperintensions to intensions. It might seem paradoxical for the extension of a meaning to be an intension, but from the hyperintensional perspective, intensions are nothing more than the result of gluing together extensions across all worlds.

We call two hyperintensional terms a and b of the same type **equivalent** iff

$$\vdash \text{ext}(a) = \text{ext}(b)$$

Correspondingly, we call two hyperintensions **equivalent** if, at every world w, they have the same extension at w. Note that truth-conditional equivalence (mutual entailment) of propositions, which as we have seen is provably the same thing as as belonging to the same worlds (ultrafilters), is a special case of equivalence in this sense. Representative examples of equivalent hyperintensions are the meanings of:

1. *Hesperus* and *Phosphorus*
2. *woodchuck* and *groundhog*
3. *Paris Hilton is Paris Hilton* and whichever is true, the Riemann Hypothesis or its denial.

Of course nothing forces equivalent hyperintensions to be the same. This being the case, within the framework of hyperintensional semantics it becomes possible to raise a question which does not even make sense in intensional semantics: are there any properties which, though not extensional, are nevertheless **intensional** in the sense that, at any world and for any hyperintension a of type A, whether a has the property at w depends only on $\text{Ext}(a)$?

(52) **Intensional Hyperintensions**

Call a closed hyperintensional term $f :: A \Rightarrow B$ **intensional** iff

$$\vdash \forall_{a,b}(\text{ext}(a) = \text{ext}(b) \rightarrow \ \text{ext}(f(a)) = \text{ext}(f(b)))$$

Consider, for example, an S5-style neccessity operator as follows[38]:

(53) **S5 Necessity**

    a. Introduce a constant $\text{nec} :: \text{Prop} \Rightarrow \text{Prop}$

    b. Meaning Postulate: $\vdash \forall_{w,p}((\text{nec}(p))@w \leftrightarrow \ (p \equiv \text{truth}))$

Clearly, nec is an intensional property of propositions; if a proposition has it at a world (and therefore at any world), then so does any equivalent proposition. As expected, all necessary truths are equivalent. By contrast, the propositional property of being obvious to Paris Hilton isn't intensional: presumably, that Paris Hilton is Paris Hilton is obvious to her, but whichever of the Riemann Hypothesis and its denial is true surely is not. Indeed, we might *define* a **modal operator** to be an intensional property of propositions.[39]

9   Conclusion

For over 60 years, it has been known that there are not enough intensions to model NL meanings in a natural way. And the hitherto unremarked yet perplexing problem of nonprincipal ultrafilters (that some maximal consistent sets of propositions don't count as possible worlds) suggests that the idea of taking worlds as a primitive of semantic theory is a serious misstep. In this paper, I proposed an axiomatic theory of NL meaning that straightforwardly solves both of these problems, seemingly at no penalty.

The theory is expressed in a simple extension of classical higher-order predicate logic, which in turn is based on an altogether mainstream typed lambda calculus; the only essential difference between the logic used here and the familiar Henkin/Gallin-style logic is the addition of an analog of the set-theoretic axiom scheme of separation. The set-theoretic models are just the familiar Henkin models, augmented with cartesian products and subsets with lambda-definable characteristic functions; and a more general,

---

[38] This illustrates another difference between the present proposal and Thomason (1980): there is no need to reintroduce a basic world type to handle modality

[39] Another question suggested by this definition: besides modal operators, are there other classes of intensional hyperintensions of semantic interest?

categorical model theory is also available should one care to explore the consequences of setting aside familar assumptions, such as wellpointedness (the assumption that there are no uninhabited types aside from ones isomorphic to the counit type 0), or even Excluded Middle. Worlds and intensions are still available, for whatever semantic uses one might choose to put them to; but the worlds are constructed rather than primitive, and the intensions are not meanings but rather what equivalent meanings have in common.

There are two key ideas that make the theory work:

1. Entailment is not assumed to be antisymmetric.
2. Worlds are constructed from propositions (as in Kripke 1959), not the other way around (as in Kripke 1963).

The theory makes no recourse to untyped lambda calculus, polymorphic typing, partial possible worlds, impossible worlds, giving up one or more of Gentzen's structural rules, or even giving up possible worlds. The math is no harder than the math in PTQ, just a little different and considerably less idiosyncratic. As far as I can tell, we can still do everything we wanted to do in mainstream semantics, without having to accept (as mainstream semantics requires us to do) that Paris Hilton knows whether the Riemann hypothesis is true.

References

[1] R. Adams. Theories of actuality. *Noûs*, 8:211–231, 1974.

[2] Carpenter, B. *Type-Logical Semantics.* MIT Press, Cambridge, MA, 1997.

[3] Carnap, R. *Meaning and Necessity.* University of Chicago Press, Chicago, 1947.

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

[5] Curry, H. Some logical aspects of grammatical structure. In R. Jakobson, ed., *Structure of Language and its Mathematical Aspects*, pages 56–68. American Mathematical Society, Providence, 1961.

[6] P. de Groote. Toward abstract categorial grammars. In *Proceedings of the 39th Annual Meeting and 10th Conference of the European Chapter of the Association for Computational Linguistics*, pages 148–155, 2001.

[7] C. Fox and S. Lappin. A framework for the hyperintensional semantics of natural language with two implementations. In P. de Groote, G. Morrill, and C. Retoré, eds., *Logical Aspects of Computational Linguistics*, Springer Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin and New York, pages 175–192, 2001.

[8] C. Fox and S. Lappin. *Foundations of Intensional Semantics*. Blackwell, Oxford, 2005.

[9] C. Fox, S. Lappin, and C. Pollard. A higher-order fine-grained logic for intensional semantics. In *Proceedings of the Seventh International Symposium on Logic and Language*, pages 37–46, Pécs, Hungary, 2002.

[10] G. Frege. On sense and reference. In P. Geach and M. Black, eds., *Translations from the Philosophical Writings of Gottlob Frege*, 3rd edition, pages 56–78. Basil Blackwell, Oxford, 1980.

[11] D. Gallin. *Intensional and Higher Order Modal Logic*. North-Holland, Amsterdam, 1975.

[12] R. Goldblatt. *Topoi: the categorial analysis of logic*. North-Holland, Amsterdam, 1983.

[13] L. Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.

[14] B. Jónsson and A. Tarski. Boolean algebras with operators, part I. *American Journal of Mathematics*, 73:891–939, 1951.

[15] S. Kripke. A completeness theorem in modal logic. *Journal of Symbolic Logic*, 24:1–14, 1959.

[16] S. Kripke. Semantic analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[17] S. Kripke. *Naming and Necessity*. Harvard University Press, Cambridge, Massachusetts, 1980.

[18] J. Lambek and P. Scott. *Introduction to higher-order categorical logic*. Cambridge University Press, Cambridge, 1986.

[19] R. Montague. The proper treatment of quantification in ordinary English. In R. Thomason, editor, *Formal Philosophy: Selected Papers of*

Richard Montague, pages 247–270. Yale University Press, New Haven, 1974.

[20] R. Muskens. Languages, lambdas, and logic. In G.-J. Kruiff and R. Oehrle, editors, *Resource Sensitivity in Binding and Anaphora*, pages 23–54. Kluwer, 2003.

[21] C. Pollard. *Hyperintensions*. Monograph in preparation.

[22] C. Pollard. Higher order categorical grammar. In *Proceedings of Categorial Grammar 2004*, pages 340–361, 2004. Revised version to appear in a special issue of *Journal of Applied Logic*.

[23] C. Pollard. Type-logical HPSG. In *Proceedings of Formal Grammar 2004*, pages 107–124, 2004.

[24] A. Ranta. Grammatical framework. *Journal of Functional Programming*, 14:145–189, 2002.

[25] R. Stalnaker. *Inquiry*. Bradford Books/MIT Press, Cambridge, MA, 1984.

[26] M. Stone. The theory of representation for boolean algebras. *Transactions of the American Mathematical Society*, 40:37–111, 1936.

[27] R. Thomason. A model theory for propositional attitudes. *Linguistics and Philosophy*, 4:47–70, 1980.