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 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 substypes 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 make the proposal work are (1) axiomatization of NL entailment as a preorder (as opposed to an order) on the set of (primitive) propositions, and (2) definition of the set of worlds as a certain subset of the powerset of the set of propositions. 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 Carnap/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 enough meaning distinctions are available to make predictions consistent with robust intuitions. Though a great many solutions have been proposed,

[1]For advice and clarifying discussion, I am grateful to David Dowty, Chris Fox, Nissim Francez, Paul Gilmore, Howard Gregory, Jim Lambek, Shalom Lappin, Drew Moshier, Reinhard Muskens, Phil Scott, and Rich Thomason. Alas, it cannot be assumed that any of these people accept my conclusions.

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

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 substypes 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 make the proposal work are (1) axiomatization of NL entailment as a preorder (as opposed to an order) on the set of (primitive) propositions, and (2) definition of the set of worlds as a certain subset of the powerset of the set 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 more 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 to those that I will target for elimination. Section 2 reviews the well-known granularity problem, with special attention to its two most notorious subproblems, Frege's Hesperus-Phosphorus puzzle and the antisymmetry of entailment. Section 3 is an introduction to the

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

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 its **reference**.
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.
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 lambda-calculus term equivalences (conversion) are reinterpreted 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 with respect to (unrestricted[5]) Henkin models.

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

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

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 natural 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 (World $\Rightarrow$ Ent); and so
  - if one assumes the rigidity of names (Kripke 1972), then co-referring names have the same meaning.

  ii. Declarative sentence meanings (propositions):
  - are sets of worlds (World $\Rightarrow$ Bool);
  - entailment is the subset-inclusion ordering;
  - the meanings of *and*, *or*, and *if ... then* are, respectively, intersection, union, and relative complement.
  - In particular, entailment is *antisymmetric*. Thus:
  - 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 linguistics actually uses 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.

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

2.1 Hesperus and Phosphorus

Starting at the beginning (both historically and with respect to the type hierarchy), the Standard Formalization does not have enough individual concepts. 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. For example:

(4) **Hesperus and Phosphorus**

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

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

(5) **Frege's analysis of Hesperus and Phosphorus:**

a. The sense expressed by an expression depends on the senses expressed by its parts.[7]

b. Although they have the same reference, the names *Hesperus* and *Phosphorus* express different senses.

c. Hence the sentences in (4) express different propositions, so it is unsurprising that the ancients believed one but not the other.

In essence Frege's view was simply that the sense expressed by a sentence is determined by the senses expressed by its parts. Even though Hesperus and Phosphorus have the same extension—viz. the planet Venus—they are distinct senses, and therefore the propositions expressed by (3)a and (3)b are distinct as well. This was not a problem for Frege; it simply provided

[7] This is half of what is usually called Fregean Compositionality. The other half is the analogous statement about reference, which I take to be one of Frege's missteps. The proposal I am leading up to explicitly rejects it.

one motivation among many for him to posit the ontological category of senses.

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. So *Hesperus* and *Phosphorus* mean the same thing, a most unwelcome (and un-Fregean) consequence.

(6) **Hesperus and Phosphorus in the Standard Theory**

a. The meanings of *Hesperus* and *Phosphorus* are functions from worlds to entities.

b. Assuming rigidity of names (Kripke 1972), they are constant functions.

c. At at least one world, both functions take the value Venus, and so they are the *same* constant function.

d. So *Hesperus* and *Phosphorus* mean the same thing, and consequently (*pace* Frege) the sentences in (4) express *the same* proposition.

2.2 Equivalent Propositions

As we noted, in the Standard Formalization, entailment is antisymmetric and so equivalent propositions are identical. This is problematic because there is a naive, robust intuition that declarative sentences can follow from each other without meaning the same thing. This problem was noted at least as early as 1944 by C.I. Lewis. His response was to say that the meaning (in his terminology, *analytic meaning*) of an expression is not merely an intension but something more fine-grained, in more contemporary parlance essentially a phrase-structure tree of the expression with associated intensions for each constituent. Carnap's (1947) notion of *intensional isomorphism* is just identity of analytic meaning in Lewis's sense.[8].

The Lewis-Carnap proposal was the starting point for the (ongoing) tradition of so-called "structured meaning" approaches to the granularity

[8]But Lewis's notion was limited in application to synthetic expressions, ones whose reference depends how things are; Carnap dropped this restriction

problem[9]. Almost as venerable is the tradition (initiated at least as early as Church (1950)) of exposing the inadeqacies of such approaches. In the intervening decades, a vast array of competing proposals have been made, involving, inter alia, partial possible worlds, impossible worlds, moving to an untyped lambda calculi, abandoning bivalence, abandoning one or more of Gentzen's structural rules, and abandoning Boolean Extensionality.

Some of these proposals are merely programmatic, with no explicit logic. Some provide a logic but don't quite manage to nail down the model theory. Some don't actually solve the problems they claim to solve. And the mathematical sophistication of some puts them beyond the reach of working semanticists who are not mathematical logicians but want a framework they can understand and use for linguistic analysis. Limitations of space preclude discussion of competing proposals here; for surveys, see Fox and Lappin (2005) and Pollard (in preparation). Here, I will just discuss briefly two examples of problems that equivalent propositions pose for the Standard Formalization, and then lead up to my own proposal.

(7) **Woodchucks and Groundhogs**

a. Phil is a woodchuck.

b. Phil is a groundhog.

(8) **Woodchucks and Groundhogs in the Standard Theory**

a. Standard-Theory Meaning Postulate[10]:

$$\forall_{w \in \mathrm{World}} \forall_{i \in \mathrm{Ind}} (\mathsf{woodchuck}(i)(w) \leftrightarrow \mathsf{groundhog}(i)(w))$$

b. By HOL, $\mathsf{woodchuck} = \mathsf{groundhog}$

c. Therefore (i) and (ii) express *the same* proposition:

i. Jim believes Phil is a groundhog.

ii. Jim believes Phil is a woodchuck.

[9] These in turn were a refinement of "inscriptional" approaches that treat objects of belief, knowledge, etc. not as propositions, but rather as linguistic expressions, or something syntactic associated with them, e.g. graphical representations or Gödel numbers.

[10] Under fairly standard assumptions, the verb phrase meanings here are functions from individual concepts to propositions (type Ind $\Rightarrow$ Prop = Ind $\Rightarrow$ (World $\Rightarrow$ Bool)). Uncurrying, permuting the arguments, and then currying shows this type to be equivalent (in terms of the Curry-Howard type logic) to the intensional type World $\Rightarrow$ (Ind $\Rightarrow$ Bool), i.e. the VP meanings amount to properties of individual concepts. In fact, they are moreover *extensional* properties, in the sense that whether an individual concept has one of them depends only on the extension of the individual concept; but this fact doesn't bear directly on the matter at hand.

The problem here of course is that the conclusion (c) seems wrong: Jim might well believe (i) without believing (ii), for example if he mistakenly thought that woodchcucks were those dam-building creatures with buck teeth.

The classic mainstream possible-worlds response to such examples, given its most careful articulation by Stalnaker (1984), is that Jim in fact *does* believe Phil is a woodchuck; he just wouldn't put it that way. I will return to Stalnaker's view of these matters in due course, but before that let's consider a different kind of example. In the mainstream possible-worlds account, there is exactly one necessarily true proposition. Now consider the propositions (expressed by the following sentences:

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

a. Paris Hilton is Paris Hilton.
b. All nontrivial zeros of ζ have real part 1/2.

(9b) is the Riemann Hypothesis, the most famous unresolved conjecture in all mathematics; here ζ is the Riemann ζ-function, a certain function of a complex variable.

Now it is standard amlong linguistic semanticists to assume that for a declarative sentence R, to know whether R s to know that R (if R is true), or to know the denial f R (if R is false). Now consider:

(10) **Paris Hilton and Riemann in the Standard Theory**

a. There is only one necessary truth, so whichever of (9b) and its denial is true expresses the same proposition as (9a).
b. Presumably, Paris Hilton knows that Paris Hilton is Paris Hilton.
c. So if (9b) is true, then Paris Hilton knows that all nontrivial zeros of ζ have real part 1/2.
d. And if (9b) is false, then Paris Hilton knows that not all nontrivial zeros of ζ have real part 1/2.
e. Hence, Paris Hilton knows whether all nontrivial zeros of ζ have real part 1/2.

To summarize, on the standard account, it seems that, as long as we are willing to concede that Paris Hilton knows that Paris Hilton is Paris Hilton, then we are forced to conclude that Paris Hilton knows whether the Riemann hypothesis is true. Of course it is possible, as Stalnaker has shown, to defend the drawing of such conclusions. My point here is not to argue with

Stalnaker's defense, but rather to point out that it is unnecessary. He *must* mount such a defense, because his semantic theory has as a consequence the identity of mutually entailing propositions; but on the proposal I will develop below, nothing forces entailment to be antisymmetric, so there is no need to come to terms with such a consequence.

3 Soft Actualism Recalled

In his defense of possible worlds, Stalnaker 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:

(11) **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 exactly the same thing as maximal consistent sets of formulas. But in 1963, Kripke abandoned this approach in favor of possible worlds as unanalyzed primitives, for his more general completeness theorem for normal modal propositional calculi, and that is where Montague got them from.

(12) **Kripke 1959: An Earlier Avatar of Soft Actualism**

a. 1959: Completness for S5. Worlds implemented as complete assignments of truth values to formulas (= maximal consistent sets).

b. 1963: Completness for normal modal PC: switched to primitive worlds.

c. Montague's IL followed Kripke 1963, not Kripke 1959 (alas).

Indeed, as Kripke acknowledged in a footnote to his 1963 paper, the essentials of his analysis of modal logic in turn had been anticipated in algebraic form even earlier by Jónsson and Tarski's (1950) representation theorem for boolean algebras with n-ary operators; the Kripke semantics is just the case $n = 1$.

(13) **And Earlier Still: Jónsson and Tarski (1950)**

a. They proved a general representation theorem for boolean algebras with n-ary operators (their Theorem 3.10).

b. The essence of Kripke's (1959) modal semantics is the case $n = 1$.

c. This theory is the extension to boolean algebras with operators of Stone's (1936) Representation Theorem for boolean algebras.

For our purposes, the essential content of the Stone Representation Theorem can be summarized as follows:

(14) **Stone Representation Theorem**

a. Any boolean algebra B is isomorphic to a subalgebra of a powerset algebra $\wp(X)$.

b. X can be taken to be the set of *ultrafilters* of B.

c. By definition, a subset U of B is an **ultrafilter** iff:

   i. it is closed under finite meets;

   ii. it is upper-closed; and

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

d. The Stone embedding maps each $b \in B$ to the set of ultrafilters containing it.

For example, if B is the set of (logical equivalence classes of) sentences of a classical logic , the ultrafilters are just the consistent complete theories, i.e. the maximal consistent sets of (equivalence classes of) sentences.

Based on these considerations, we can now cast Soft Actualism in algebraic form as follows:

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

a. Propositions are primitives;

b. they are the elements of a boolean algebra whose order is entailment;

c. possible worlds are just the ultrafilters; and

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

By comparison, the algebra of the Standard Formalization is as follows:

(16) **In the Standard Formalization:**

a. Worlds are primitives;
b. Propositions form a boolean algebra ordered by entailment;
c. The boolean algebra is the powerset of the set of worlds and the entailment order is subset inclusion.

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.

(17) **Basic Facts about Ultrafilters of Boolean Algebras**

a. A **principal** ultrafilter is one with a least element (its **generator**).
b. If $u \in B$ is an atom of the algebra, then

$$\uparrow u =_{\text{def}} \{b \in B \mid u \sqsubseteq b\}$$

is a principal ultrafilter with generator u, and every principal ultrafilter is of that form.
c. If B is finite, every ultrafilter is principal. In this case the Stone embedding maps each $b \in B$ to the set of principal ultrafilters whose generators are the atoms less than or equal to b.
d. **But if B is infinite, then (assuming the Axiom of Choice) B can be proven to have a nonprincipal ultrafilter.**

This last fact has a consequence for the Standard Formalization that seem to have gone unnoticed. Consider:

(18) **Standard Formalization meets Stone Representation**

a. Let B be the boolean algebra B of propositions (= sets of possible worlds).
b. Since B is a powerset algebra, the atoms are the singleton sets $\{w\}$ where w is a possible world.

c. So there is a one-to-one correspondence between possible worlds and principal ultrafilters, with w corresponding to $\uparrow \{w\}$; the generator $\{w\}$ is the conjunction of all the propositions true in w.

d. So, if there were no nonprincipal ultrafilters, the Standard Formalization and Soft Actualism would be notational variants.

e. But, uncontroversially, there are infinitely many propositions. So (assuming our ambient set theory has Choice) there is a nonprincipal ultrafilter; i.e. **there is a maximal consistent set of propositions which is not the set of propositions true in some fixed world**.

As far as I know this last point has not been noticed before, but it should have been. It points to something amiss about the Standard Formalization:

(19) **The Case of the Missing Worlds**

a. Intuitively, any maximal consistent set of propositions is a way the world might be,

b. But in the Standard Formalization, the ones which are nonprincipal ultrafilters are missing from the set of possible worlds!

What should a defender of the Standard Formalization say about the missing worlds? There seem to be only two ways open:

(20) **Options for Defending the Standard Formalization**

a. Give up Choice, just so that it can be consistently maintained that the algebra of propositions has no nonprincipal ultrafilters.

b. Try to argue that even though there are maximal consistent sets of propositions that the semantic theory is not taking into consideration, for some reason they just don't count.

Of course it is *imaginable* to go with one of these two options. But it seems odd, to say the least, that one's semantic theory, which is after all a collection of empirical hypotheses about the natural phenomenon of entailment (based on native speaker's judgments about when one sentence follows from another) could lead one to deprive oneself of Choice! And I don't even no where to begin in defending the thesis that certain maximal consistent sets of propositions shouldn't count as possible worlds, for the purposes of modelling Fregean senses as intensions. Why not just avoid the whole problem by just accepting Soft Actualism instead? Nobody, as far as I can tell, ever seems to have argued persuasively 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 Algebraicized

But what about Paris Hilton and the Riemann Hypothesis? As formulated algebraically in (13), 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 algebras, 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"; this is the Lindenbaum algebra construction. Why bohter 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. In short, what we want is something just 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*.[11] These were described, rather telegraphically, in Fox et al. (2002) under the name *boolean prelattices*[12]. Here I present them in a somewhat more leisurely fashion.

(21) **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.

[11] 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.

[12] They 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 fix what was wrong with FIL (see Pollard in preparation for discussion).

(22) **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 three binary operations and$'$, or$'$, and if$'$ ... then$'$ ..., such that, for all p, q, and r,

a. Truth: $p \models$ Truth

b. Falsity: Falsity $\models p$.

c. and$'$-elimination: (i) (p and$'$ q) $\models p$; and (ii) (p and$'$ q) $\models q$.

d. and$'$-introduction: If $p \models q$ and $p \models r$, then $p \models$ (q and$'$ r).

e. or$'$-introduction: (i) $p \models$ (p or$'$ q); and (ii) $q \models$ (p or$'$ q).

f. or$'$-elimination: If $p \models r$ and $q \models r$, then (p or$'$ q) $\models r$.

g. Modus Ponens: ((if$'$ p then$'$ q) and$'$ p) $\models q$.

h. Deduction: If (r and$'$ p) $\models q$, then $r \models$ (if$'$ p then$'$ q).

i. Negation: not$'$ $p \equiv$ (if$'$ p then$'$ Falsity)

j. Double Negation: (not$'$ (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 equivalence*; double negation is a case in point here. To put it another way: a boolean algebra is just a boolean prealgebra in which entailment is antisymmetric (i.e. $\equiv$ is equality of propositions).

(23) **Fundamental Facts about Boolean Prealgebras**

a. Equalities that hold for boolean algebras still hold **as equivalences** for boolean prealgebras.
b. A boolean algebra is just an antisymmetric boolean prealgebra.
c. Example: formulas of classical PL preordered by logical consequence. For this reason I usually refer to the members of a boolean prealgebra as its **propositions**.

A preordered algebra is of course both a preorder and an algebra, but there is more to it than that. Crucially, the algebra operations harmonize with the preorder in the sense of being **tonic** (either monotonic or antitonic) on each of their arguments. Explicitly:

(24) **Theorem (Tonicity of Boolean Operations)**
For all propositions p, q, r in 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 principle of substitutivity:

(25) **Corollary (Substitutivity with respect to Booleans)**
For all propositions p, q, r in 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. $(\text{not}' \ q) \equiv (\text{not}' \ p)$

From the Corollary, it is easy to see (inductively) that *if the only propositional operators are the booleans*, then substitution of equivalent propositions is always truth-preserving.

(26) **An Important Consequence of Boolean Substitivity**

a. **If the only propositional operators are the booleans,** then, by induction, substituting an equivalent proposition inside another proposition is always truth-preserving.

b. But there is no reason to expect this of propositional operators **in general**, e.g. *it is believed by Paris Hilton that* ...!

Looking at things from this perspective, it is perhaps not so surprising that an expectation came to prevail among people concerned with such matters that we should expect to *always* be able to substitute equivalents preserving truth. The Standard Formalization is constructed on the basis of this expectation. But this is an unreasonable expectation, of the same order of unreasonableness as expecting every function of a real variable to be either monotone increasing or monotone decreasing. The tonicity of the booleans makes them *special*; there is no reason to expect it to hold of propositional operators *in general*, e.g. *it is believed by Paris Hilton that* ...!

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

(27) **Definition (Ultrafilters in a Boolean Prealgebra)**
A subset w of a boolean prealgebra is called an **ultrafilter** iff, for all propositions p and q:

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:

(28) **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. However, the principal lemma Stone used to prove it does:

(29) **Stone's Lemma (There are Enough Ultrafilters)**
If p and q are propositions in a boolean prealgebra and $p \not\models q$, then there is an ultrafilter w such that $p \in w$ but $q \not\in w$.

This has the following important consequence:

(30) **Corollary (Propositional Equivalence and Ultrafilters)**
If p and q are propositions in a boolean prealgebra, then $p \equiv q$ iff for every ultrafilter w, $p \in w$ iff $q \in w$.

With these technical preliminaries behind us, we can now revise the algebraicization of Soft Actualism to the following form:

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

a. Propositions are primitives;
b. they form a boolean **pre**algebra preordered by entailment;
c. possible worlds are just the ultrafilters; and
d. 'p is true in w' just means $p \in w$.
e. Only change from (15): replace "algebra" by "prealgebra" in (b).

The only change from the preliminary version (15) are to the second clause, where the boolean algebra is replaced with a boolean prealgebra. 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 algbebra with entailment as subset inclusion, and there is just no getting around the fact that subset inclusion is antisymmetric.

(32) Algebraic Soft Actualism solves both:

a. **the problem with equivalent propositions** (they need not be equal), **and**
b. **the problem with nonprincipal ultrafilters** (they are included).
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 trhe 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 (1) and binary $(A \times B)$[13]

(33) **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.

(34) **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 $A \Rightarrow B$.

(35) **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 $\pi'_{A,B}(h) :: B$;

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

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:

(36) **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[14];
   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.

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

(37) **Interpretation of the Underlying Typed Lambda Calculus**
A **(set-theoretic) interpretation** I[15] 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. $I(1) = \{0\}$;[16]
b. $I(A \times B) = I(A) \times I(B)$;
c. $I(A \Rightarrow B) \subseteq I(A) \Rightarrow I(B)$.[17]

(38) **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 $x :: A$, $\alpha(x) \in I(A)$.

(39) **Extending an Interpretation Relative to an Assignment**
Given a variable assignment α relative to and 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 $\beta(x) = a$.

---

[15] 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 $I(A)$. To simplify the exposition, I speak as if the set-theoretic interpretations are the only ones, but there is no theoretical justification for this restriction, and it may not even be desirable.

[16] Since $\{0\} = 1$, this means that I am not distinguishing notationally between the type 1 and its set-theoretic interpretation.

[17] I am not distinguishing notationally between $\times$ and $\Rightarrow$ as lambda-calculus type constructors and as set operations (or categorical bifunctors). Note that, 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.

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:

(40) **From TLC to HOL**

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

b. For each type A, add an equality basic constant $=_A :: (A \times A) \Rightarrow \text{Bool}$.

c. The equations (36) are no longer taken as defining an equivalence relation on terms terms but rather as object-language axioms about equality (of whatever the terms denote).

Now all the usual (intuitionistic) connectives and quantifiers are definable:[18]

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

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

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

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

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

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

d. $\wedge =_{\text{def}} \lambda_{(x,y)\in\text{Bool}\times\text{Bool}}(x,y) = (\mathsf{true},\mathsf{true})$
e. $\rightarrow\ =_{\text{def}} \lambda_{(x,y)\in\text{Bool}\times\text{Bool}}(x = x \wedge y)$
f. $\leftrightarrow\ =_{\text{def}} \lambda_{(x,y)\in\text{Bool}\times\text{Bool}}[(x \rightarrow\ y) \wedge (y \rightarrow\ x)]$
g. $\neg =_{\text{def}} \lambda_{x\in\text{Bool}} x \rightarrow\ \mathsf{false}$
h. $\vee =_{\text{def}} \lambda_{(x,y)\in\text{Bool}\times\text{Bool}} \forall_{t\in\text{Bool}}(((x \Rightarrow t) \wedge (y \Rightarrow t)) \Rightarrow t)$
i. $\exists_{x\in A}\phi =_{\text{def}} \forall_{t\in\text{Bool}}(\forall_{x\in A}(\phi \Rightarrow t) \Rightarrow t)$

In spite of the suggestive name Bool, so far this higher-order logic is only intuitionistic.[19] To make it classical, we add (again following Lambek and Scott) the axiom[20]

(42) **Axiom of Excluded Middle**

$$\vdash \forall_{t\in\text{Bool}}(t \vee \neg t)$$

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

(43) **Axiom of Boolean Extensionality**

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

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 $I(\text{Bool})$) will be just fine.

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

(44) **Subtypes and Characteristic Functions**

a. Besides (34), we have one more way of forming types: if $a :: A \Rightarrow \text{Bool}$ is closed, then A_a is a type (intuitively: the subtype of A whose members satisfy the predicate a);

[19] This is reflected by the definitions of false, $\vee$, and $\exists$. In the presence of (42), these reduce to the familiar definitions as DeMorgan duals of true, $\wedge$, and $\forall$, respectively.

[20] 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; in the general case other machinery (see (ii) below) is needed to enforce bivalence.

b. Besides (35), we have one more way of forming terms: if $a :: A \Rightarrow \text{Bool}$ is closed, then $\mathsf{ker}_a :: A_a \Rightarrow A$ (intuitively: the embedding of A_a into A); and

c. we have one further axiom schema

$$\vdash \forall_{(x,a)\in A\times(A\Rightarrow\text{Bool})}(a(x) \leftrightarrow \exists_{y\in A_a} x = \mathsf{ker}_a(y))$$

(Intuitively: a is the characteristic function of A_a.)[21]

The preceding says of a set-theoretic model that 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 $A \Rightarrow \text{Bool}$) is also in the model.[22] To summarize: the set-theoretic interpretations are Henkin models which are closed under (1) finite cartesian products, and (2) taking of subsets whose characteristic functions are lambda-definable.

[21]In categorical terms, this means that a model is not only cartesian closed, but also a topos, with $I(\mathsf{true}) : I(1) \rightarrow I(\text{Bool})$ as its subobject classifier.

[22]Together with (42), it says of a categorical model that it is a *boolean topos*. It is a fact about such models that coproducts (the categorical generalizations of disjoint unions, and the disjunction in the underlying type logic) are definable, and moreover that Bool is isomorphic to $1 + 1$, with $I(\mathsf{true})$ and $I(\mathsf{false})$ being the canonical injections of the cofactors. So even though Bool was introduced as a basic nonlogical type, with the addition of (42), (43), and (44c), Bool is actually a logical type, in the sense of being definable in the underlying type logic. In the special case of the set-theoretic models, this has as a bivalence as a consequence (i.e. that true and false are the only truth values). For the categorical models, more work is needed to enforce bivalence. First, we require the following axiom that forces true and false to be distinct:

(i) **Nondegeneracy**

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

And second, we must impose the following condition on provability:

(ii) **Disjunctivity**
We require that for all boolean terms ϕ and ψ, if $\vdash \phi \vee \psi$, then $\vdash \phi$ or $\vdash \psi$.

In the presence of the axioms already imposed, this condition can be shown to be equivalent to bivalence. The categorical models are *local boolean toposes*, or equivalently, *bivalent boolean toposes*. Up to isomorphism, the set-theoretic interpretations are the local boolean toposes which are *well-pointed* (i.e. have no uninhabited types other than the counit (null coproduct) type. It is an open question which class of models is more suitable for NL semantics. But for familiarity, I will speak of the models *as if* they were well pointed, e.g. 'set' for 'object', 'subset' for 'subobject', 'function' for 'arrow', 'preorder' for 'internal preorder object', etc.

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.[23] Crucially, there is no basic type World.

(45) **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)

d. Bool, the type of truth values, is supplied by the HOL.

e. There is *not* a basic type World!

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:

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

a. 1 is a hyperintensional type;

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.

[23]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).

For simplicity, let us assume that the syntactic (i.e. tectogrammatical) part of our linguistic theory provides the basic syntactic types NP, $\mathrm{NP}_{\mathsf{it}}$, $\mathrm{NP}_{\mathsf{there}}$, N, and S, and that $\times$ and $\Rightarrow$ are the syntactic type constructors.[24] Then at the level of types, the mapping from linguistic expressions (i,e. signs, or syntactic derivations) to their senses (hyperintensions) is defined recursively as follows:

(47) **NL Semantic Interpretation is Structure-Preserving**[25]

a. $\mathsf{Sem}(\mathrm{NP}_{\mathsf{it}}) = \mathsf{Sem}(\mathrm{NP}_{\mathsf{there}}) = 1$;
b. $\mathsf{Sem}(\mathrm{NP}_{\mathsf{name}}) = \mathrm{Ind}$;
c. $\mathsf{Sem}(\mathrm{S}) = \mathrm{Prop}$;
d. $\mathsf{Sem}(\mathrm{N}) = \mathrm{Ind} \Rightarrow \mathrm{Prop}$
e. $\mathsf{Sem}(X \times Y) = \mathsf{Sem}(X) \times \mathsf{Sem}(Y)$; and
f. $\mathsf{Sem}(X \Rightarrow Y) = \mathsf{Sem}(X) \Rightarrow \mathsf{Sem}(Y)$.

Again at the level of types, the mapping from hyperintensions to extensions is defined recursively as follows:[26]

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

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.

[24]That is, we follow Curry, de Groote, Muskens, Pollard, Ranta, and others in assuming that tectogrammatical combinatorics is nondirectional, with word order determined by the interface between tectogrammar and phenogrammar.

[25]Categorically: Sem is a cartesian closed functor.

[26]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. We also defer to Pollard (in preparation) the question of what the extensional type corresponding to a subtype is.

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 object-language constant

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

and equivalence of propositions is defined as mutual entailment:

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

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

(49) **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 $=_{\text{Prop}}$.[27]

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

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

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

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

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

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

e. $\mathsf{if'} \ldots \mathsf{then'}$ 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. 22):

[27] Cf. (43), which says $\leftrightarrow$ **is** equal to $=_{\text{Bool}}$.

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

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:

(52) **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 ⇒ 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:

(53) **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 $\forall_{(p,q)}[(s(p) \wedge s(q)) \rightarrow s(p \;\mathsf{and}'\; q)]$; 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.

(54) **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 :: World, not w :: Prop $\Rightarrow$ Bool.

c. But World = $[\text{Prop} \Rightarrow \text{Bool}]_u$ where u is defined as in (53), so ker_u :: World $\Rightarrow$ (Prop $\Rightarrow$ 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 $\mathsf{ker}_u(w)(p)$.

e. For this reason, I will usually abbreviate $\mathsf{ker}_u(w)(p)$ 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 functions from worlds to extensions of the appropriate type. These functions are constrained by the following axioms:

(55) **Axioms for Extensions**

a. $\vdash \forall_{w,p}(\mathsf{ext}_{\text{Prop}}(p)(w) = p@w)$
b. $\vdash \mathsf{ext}_1(w)(*) = *$
c. $\vdash \forall_{w,c}(\mathsf{ext}_{A\times B}(c)(w) = (\mathsf{ext}_A(\pi(c))(w), \mathsf{ext}_B(\pi'(c))(w)))$
d. $\vdash \forall_{w,f}(\mathsf{ext}_{A\Rightarrow B}(f)(w) = \lambda_{x\in A}\mathsf{ext}_B(f(x))(w))$

The last of these is the most interesting, because it makes explicit why it is that, even though we are good Fregeans as far as the compositionality of meanings (hyperintensions) is concerned[28], we part company with Frege with respect to compositionality of reference. For example, we translate the sentential adverb obviously :: S $\Rightarrow$ S by a constant obviously' : Prop $\Rightarrow$ Prop, which is interpreted set-theoretically as a function from propositions to propositions. So the reference of obviously at a world w is the extension at w of that function, which is (the characteristic function of) a set of propositions, viz. the ones which are obvious at that world.[29]

For A a hyperintensional type, we call a closed term of type $A \Rightarrow$ Prop an A-**predicate** and its interpretation an A-**property**. For many properties that serve as NL meanings, for any world w, whether something a of type A has the property at w depends only on the extension of a at w. Such properties (and by extension, predicates whose interpretations are such properties) are called **extensional**.

(56) **Extensionality for Predicates**

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

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

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

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

For example, NL determiners are (A-parametrized) families of extensional $(A \Rightarrow \text{Prop}) \times (A \Rightarrow \text{Prop})$-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 would be a consequence of the following nonlogical axiom scheme:

[28] Composiitonality of meaning is embodied by the cartesian-closed functoriality of the mapping Sem from linguistic expressions to meanings.

[29] Thus we avoid having to say that in certain contexts the reference of a sentence is its customary sense.

(57) **Meaning Postulate for every**

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

8.4 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 $a :: A$, $\mathsf{ext}(a)$ is of type World $\Rightarrow$ 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 of the same type **equivalent** iff "ext maps them to the same intension", i.e.:

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

Correspondingly, we call two hyperintensions **equivalent** if, at every world w, they have the same extension at w. This generalizes the notion of equivalent propositions as ones that are true at the same worlds. 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 Ext(a)?

(58) **Intensional Hyperintensions**
Call a closed hyperintensional term $f :: A \Rightarrow B$ **intensional** iff

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

Consider, for example, an S5-style neccessity operator as follows[30]:

(59) **S5 Necessity**

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

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

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.[31]

9 Conclusion

For over 60 years, it's 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 straightforward 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 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, categorical model theory is also available should one care to explore the consequences of setting aside familar assumptions (such as wellpointedness or double negation). Worlds and intensions are still available, for whatever semantic uses one might choose to put them to; only the worlds are constructed rather than

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

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

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 less idiosynacratic. As far as I can tell, we can still do everything we wanted to do in mainstream semantics, without accepting (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] Church, A. On Carnap's analysis of statements of assertion and belief. *Analysis*, 10:97–99, 1950.

[6] 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.

[7] 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.

[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, 190

[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] C.I. Lewis. The modes of meaning. Reprinted in L. Linsky, ed., *Semantics and the Philosophy of Language.* University of Illinois Press, Urbana, 1952.

[20] R. Montague. The proper treatment of quantification in ordinary English. In R. Thomason, editor, *Formal Philosophy: Selected Paperds of Richard Montague*, pages 247–270. Yale University Press, New Haven, 1974.

[21] 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.

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

[23] 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*.

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

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

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

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

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