

Hyperintensional Questions

Carl Pollard

INRIA-Lorraine, Universitat Rovira i Virgili, and Ohio State University
pollard@ling.ohio-state.edu

1 Introduction

It has been known for decades that Montague's (1974 [1970]) possible-worlds semantics, which follows Kripke 1963 in treating worlds as unanalyzed primitives and propositions as sets of worlds, does not provide enough meaning distinctions to make correct predictions about a wide range of natural-language entailment patterns.¹ This **granularity** problem, as it has come to be known, has many dimensions, of which the best known is that two declarative sentences which entail each other must express the same proposition. This is because entailment is modelled by the subset inclusion relation on the powerset of the set of propositions, and that relation is irretrievably antisymmetric. The most notorious consequence of this antisymmetry of entailment is the so-called **logical omniscience** problem, that (assuming knowledge is a relation between individuals and propositions) anyone who knows at least one necessary truth (e.g. that s/he is self-identical, or that two is even) must know **every** necessary truth, even an unresolved mathematical conjecture or its denial (whichever is true). Thus, e.g. if Paris Hilton knows that Paris Hilton is Paris Hilton, then she must also know that every nontrivial zero of the zeta-function has real part $1/2$, if that is indeed the case, or else she must know that this is *not* the case, if indeed it is not. In short, it seems to be a consequence of MS that a celebrity hotel heiress devoted to parties and shopping knows whether the Riemann Hypothesis is true. This is just one of the unsavory consequences of MS.

From the point of view of MS, the most conservative response to Granularity, and one which remains popular among linguistic semanticists, is to argue, along the lines of Stalnaker (1984), that it is naive to perceive it as a real problem. In effect, Paris Hilton really *does* either know that R or know that not R, she just doesn't *know* that she does. On the other hand, Granularity has been quite widely viewed as a serious foundational problem, and a wide range of ingenious

¹ For much useful discussion, some of it in the distant past, and comments on earlier versions, I wish to thank David Dowty, Jonathan Ginzburg, Howard Gregory, Martin Jansche, Brad Kolb, Tim Leffel, Scott Martin, Drew Moshier, Reinhard Muskens, Andy Plummer, Phil Scott, and Ken Shan. For their help in providing the conditions that made this research possible, I am grateful to Carlos Martin Vide, Philippe de Groote, and the Department of Linguistics and College of Humanities of Ohio State University. The research reported here was supported by grant no. 2006PIV10036 from the Agència de Gestió d'Ajuts Universitaris i de Recerca of the Generalitat de Catalunya.

and technically sophisticated replacements for MS have been proposed in response. Among these, to mention just a few of the best known proposals, have been Intentional Semantics (Thomason 1980), Situation Semantics (Barwise and Perry 1983), and Property Theory (Chierchia and Turner 1988). For summaries and critical assessment of these and other proposals, together with a more sophisticated version of Property Theory, see Fox and Lappin 2005.

In general, these alternative proposals depart radically from MS, e.g. by embracing impossible worlds in addition to possible ones; by countenancing not just possible worlds but also partial possible worlds; by rejecting possible worlds altogether; by working in a type theory that abandons one or more of the usual structural rules; by moving from typed to untyped lambda calculus, etc. By contrast, the approach adopted here, described in detail in Pollard 2008, holds that MS was close to the mark, and that a relatively minor repair job eliminates the Granularity problem along with certain other of MS's foundational problems.² The repair involves five ideas, of which only the last two are original.

First, we must treat propositions as primitives, not as sets of worlds. In connection with natural-language semantics, this was already advocated by Thomason (1980), though the philosophical roots of the idea can be traced back to Wittgenstein or perhaps even Bolzano.

Second, we must treat worlds not as primitives, but rather as ultrafilters over the boolean structure on propositions. In some form or other, this battle-tested idea is present in Adams (1974), Kripke (1959), Jónsson and Tarski (1951), Carnap (1947), and Stone (1936,1937).

Third, the entailment relation on propositions must not be antisymmetric! I am indebted to Howard Gregory³ for making me explicitly conscious of the central and nonnegotiable character of this requirement.

Fourth, the boolean structure on propositions, in terms of which worlds are defined as ultrafilters, has to be induced by the entailment relation on propositions. That is, rather than *define* entailment as subset inclusion of sets of worlds, we *axiomatize* entailment to be a boolean preorder on the set of propositions. Then, since the type theory we will work in has Choice⁴, it will follow from the (internal) Stone Representation Theorem that this preorder really deserves to be called entailment, in the sense that, for any two propositions p and q , p entails q iff every ultrafilter with p as a member also has q as a member.

And fifth, the type theory must countenance a Separation-like notion of subtyping, along the lines of Lambek and Scott 1986. This will ensure that the worlds, in spite of having been defined as ultrafilters of propositions, constitute a type of the underlying type theory. That is because the property of being an ultrafilter is an internally definable property of propositions. More specifically, we can write a formula (boolean term) $u[S]$ (where S is a variable of the

² Another approach that is similar in spirit, though not in technical detail, is that of Muskens (2005). Unfortunately Muskens and I separately adopted the term “hyperintensional” for our approaches.

³ Personal communication to Shalom Lappin and me, 1999.

⁴ Actually, the Boolean Prime Ideal Theorem would be enough.

type of sets of propositions) which says of S that it is an ultrafilter, and then use that formula to form a subtype—call it World—of the type $\text{Prop} \supset \text{Bool}$. This type will play a role in the semantic theory analogous in crucial respects to that played by the world-type s in MS. However, unlike MS, meanings (including propositions) will *not* be modelled as intensions (functions from worlds to extensions), but rather as the considerably more fine-grained *hyperintensions*.

The purpose of this paper is to show how a semantic theory along these lines sheds light on the understanding of *questions*, the meanings expressed by interrogative sentences. Our questions will turn out to be closely related to the two standard modellings of questions in the linguistic semantics literature, due respectively to Karttunen (1977) (hereafter, K) and Groenendijk and Stokhof (1984) (hereafter, G). The gist of the relationship is as follows. First, the hyperintensional theory of meanings has the property that it contains an ‘isomorphic copy of MS’, together with a type-parametrized family of functions ext_A that maps each hyperintension to the corresponding Montagovian intension. In general this function is many-to-one, but it becomes a bijection if we (perversely!) add to our meaning theory an axiom that makes entailment antisymmetric. In the special case of the type Prop of propositions, this function is just the Stone function that maps each member of the preboolean algebra of propositions to a clopen subset of the corresponding Stone space (the one whose members are the ultrafilters of which that proposition is a member.) Under this mapping, each hyperintensional question is mapped to a K-question intension. And the corresponding G-question is in turn obtained from that, by taking the induced equivalence relation on worlds.

Thus, our theory of meanings is a natural generalization of MS (dropping the antisymmetry of entailment), and our theory of questions within it is a natural generalization of the standard MS modellings of questions. The greater generality will enable us to make finer-grained distinctions among questions than the standard theories can make. For example, on K’s account (and therefore, on G’s as well) these two complementized denote the same questions:

- (1) a. whether Paris Hilton is Paris Hilton
- b. whether Britney Spears is Britney Spears

But on our account (hereafter, H), they do not. And on G’s account, these two questions are the same:

- (2) a. Which students are vegetarians?
- b. Which vegetarians are students?

but on our account they are distinct. The remainder of the paper is organized as follows. In section 2, we describe the version of higher order logic (our counterpart of Montague’s IL or Gallin’s Ty2) in which the theory of meanings is written. Section 3 sketches the meaning theory itself, while section 4 deals with worlds and extensions at them. Section 5 extends the meaning theory to include questions. And in section 6, we conclude by showing how our theory of questions relates to those of Karttunen and Groenendijk-Stokhof.

2 Higher Order Logic (HOL) with Subtypes

We work in a classical HOL broadly similar to that of Henkin 1950 or Gallin 1975, but augmented with machinery for handling subtyping along the lines of Lambek and Scott 1986. This logic is built on top of positive typed lambda calculus (TLC) (positive in the sense that the underlying type theory is positive intuitionistic propositional logic, so that we have the nullary type constructor \top (unit type) and product \wedge as well as the usual exponential \supset). For the term constructors, we write $*$ for the constant of type \top , $(-, -)$ for pairing, π and π' for the projections, and $f(a)$ for $\text{eval}(f, a)$. As usual, we have the basic types Ent (entities, corresponding to Montague's e) and Bool (truth values, corresponding to Montague's t^5). In addition we will have two basic meaning types Prop (propositions, which will serve as the meanings of declarative sentences) and Ind (individual concepts, which will serve as the meanings of names and other expressions that refer to entities). We also have plenty of constants (see (20) below).

To get from positive TLC to HOL, we follow Lambek and Scott and add an equality symbol $=_A : (A \wedge A) \supset \text{Bool}$ for each type A , and then define the usual logical connectives and quantifiers in terms of λ and equality, as shown in the Appendix (13). There we also list some of the logical axioms (or theorems, depending on the choice of axiomatization) of the HOL (14-18).

The crucial feature of our HOL, which will make it possible to internally define a type for worlds (and later, as we will see, also for questions), is the machinery for handling subtypes, adapted from Lambek and Scott 1986. The motivation for this machinery is that the familiar HOLs employed in linguistic semantics provide no way to say that A is a *subtype* of B . In a set-theoretic interpretation I of the logic, this should mean $I(A) \subseteq I(B)$.

(3) Subtypes (after Lambek and Scott 1986)

If A is a type and a an A -predicate (i.e. a closed term of type $A \supset \text{Bool}$), then

- a. A_a is a type
- b. embed_a is a term of type $A_a \supset A$; and
- c. Axioms:
 - i. $\vdash \forall_{y,z \in A_a} [(\text{embed}_a(y) = \text{embed}_a(z)) \supset y = z]$
 - ii. $\vdash \forall_{x \in A} [a(x) \leftrightarrow \exists_{y \in A_a} x = \text{embed}_a(y)]$

In a set-theoretic interpretation I of the logic, $I(\text{embed}_a)$ is the function that embeds into $I(A)$ the subset whose characteristic function is $I(a)$.⁶

⁵ Or to Lambek and Scott's Ω (subobject classifier). The name Bool will be justified because the HOL of truth-value-typed terms will be classical, and correspondingly the categorical models will be boolean toposes.

⁶ More generally, in a topos model, $I(\text{embed}_a)$ is the char of $I(a)$.

3 The Theory of Meanings

We now use our HOL to express a theory about the things that will serve as meanings of (utterances of) natural language expressions. We call this a theory of meanings, not a semantic theory, because we reserve the latter term for a theory that connects linguistic expressions with their meanings. (Some aspects of such a theory are developed in Pollard (2007, submitted, in preparation 1, in preparation 2).

We start by recalling that our basic types are the two extensional types Ent (entities) and Bool (truth values), and the two meaning types Ind (individual concepts) and Prop (propositions). As we'll see, entities are the kinds of things that can be the extensions (at worlds) of individual concepts, and truth values are the kinds of things that can be the extensions of propositions. Note that World is not a basic type! Instead, World will be *defined* as the subtype of $\text{Prop} \supset \text{Bool}$ consisting precisely of the ultrafilters. This works because ultrafilterhood is an internally definable property (see Appendix, (19).

We now define a certain set of types, the **hyperintensional** types, that will serve as meaning types. This is the set obtained from the basic meaning types Ind and Prop by closing under the TLC type constructors (T , \wedge , \supset) and subtype formation. Intuitively, the things of these types are the things that have the potential to be meanings. They play a role in our semantic theory analogous to the role played by intensions in MS: they are mathematical models of Fregean senses. (Unavoidably, we will have intensions as well; but we won't use them to model meanings.) Some illustrative examples of constants of various hyperintensional types (corresponding to word meanings) are given in the Appendix (20).

Next, roughly following Montague, we define a function Ext that assigns to each meaning type the type for the corresponding extensions, i.e. any meaning of type A will have, at each world, an extension of type $\text{Ext}(A)$.

(4) Extensional Types

- a. $\text{Ext}(\text{Prop}) =_{\text{def}} \text{Bool}$
- b. $\text{Ext}(\text{Ind}) =_{\text{def}} \text{Ent}$
- c. $\text{Ext}(\text{T}) =_{\text{def}} \text{T}$
- d. $\text{Ext}(A \wedge B) =_{\text{def}} \text{Ext}(A) \wedge \text{Ext}(B)$
- e. $\text{Ext}(A \supset B) =_{\text{def}} A \supset \text{Ext}(B)$
- f. $\text{Ext}(A_a) =_{\text{def}} \text{Ext}(A)$

(5) Linguistic Consequences

At any world:

- a. Declarative sentences denote truth values.
- b. Names denote entities.
- c. Dummy pronouns have vacuous reference.

- d. The list of complements of a verb denotes the ordered tuple of the denotations the complements.
- e. A verb that expresses a function from A 's to propositions denotes (the characteristic function of) a set of A 's.

We now turn to the axiomatization of *entailment*, the binary relation between propositions that semantics is centrally concerned with. We represent this relation by the constant \models of type $(\text{Prop} \wedge \text{Prop}) \supset \text{Bool}$. Loosely speaking, we also say that one (utterance of a) declarative sentence entails another if the the proposition it expresses entails the one expressed by the other. Of course every sentence entails itself, and entailment is transitive, so we start with these (nonlogical!) axioms:

(6) Preorder Axioms for Entailment

- a. $\vdash \forall_p(p \models p)$
- b. $\vdash \forall_{p,q,r}((p \models q) \wedge (q \models r)) \supset (p \models r)$

We use the constant \equiv for mutual entailment:

(7) Mutual Entailment

- a. $\vdash \forall_{p,q}[(p \equiv q) = (p \models q \wedge q \models p)]$
- b. Nothing in our theory will let us prove
 $\vdash \forall_{p,q}[(p \equiv q) \supset (p = q)]$

That is, entailment is not antisymmetric.

We now introduce the constants **Truth** : Prop, **Falsity** : Prop, **not'** : Prop \supset Prop, **and'** : (Prop \wedge Prop) \supset Prop, **or'** : (Prop \wedge Prop) \supset Prop, and **implies'** : (Prop \wedge Prop) \supset Prop. These will be interpreted as the operations in the pre-boolean algebraic structure induced by entailment on the set of propositions. As the spelling suggests, some of these operations will also serve as the meanings of the English “logic words” (see Appendix (21) for details). These are axiomatized (Appendix (22)) so that, in an interpretation, the propositions preordered by entailment form a preboolean algebra (roughly, a boolean algebra without antisymmetry, i.e. the usual boolean facts obtain, but with equality replaced by mutual entailment). These axioms essentially say that the usual natural deduction rules of classical propositional logic are valid for natural language argumentation.

4 Worlds and Extensions at Them

So far we have not brought worlds into our theory of meanings. In fact, our view is that worlds are not relevant to meaning; meanings are “out there” (in Frege’s Heaven, if you will) and are independent of contingent fact. On the other hand, we must get involved with worlds to deal with reference; since the reference of a linguistic expression is the extension of its meaning, and what that extension is *does* depend on how things are. The most obvious example of this is

that the reference of a declarative sentence is the extension—a truth value—of the proposition it expresses. Fortunately, for us, a world will just be a set of propositions (more specifically, an ultrafilter of the preboolean algebra induced by entailment), and so for that proposition to be true at a given world is simply to be a set-theoretic member of it. This turns MS on its head, since there a proposition is true at a world if the world is a member of *it*.

Now pretheoretically, for p to entail q is supposed to mean that, no matter how things are, if p is true when things are that way, then so is q . So for this to be the case in our framework, it must be the case that p entails q iff every ultrafilter with p as a member also has q as a member. Fortunately for us, this is a theorem of ZFC, not about entailment specifically of course, but about any preboolean algebra (in our setting, the preorder is entailment). In fact this is just a slight generalization (dropping the antisymmetry requirement) of the Boolean Prime Ideal (BPI) Theorem, the key ingredient of Stone’s Representation Theorem for boolean algebras. Of course, we are not working in ZFC; but there are topos-theoretic versions of Choice appropriate to the kind of HOL we are working in, and once we add one, we are home free. To summarize, once we add a suitable form of Choice to our axioms, we can prove an internal version of BPI, and this will guarantee that entailment, by virtue of being a boolean preorder, really does behave the way that, pretheoretically, we expect entailment to behave.

We express the connection between hyperintensions, worlds, and extensions using a family of constants ext_A of type $A \supset (\text{World} \supset \text{Ext}(A))$, where A ranges over the hyperintensional types. A full account is given in Pollard 2008, but the three key cases are these:

(8) Extensions at Worlds

$$\begin{aligned} &\vdash \forall_{p,w} [\text{ext}_{\text{Prop}}(p)(w) = p@w] \\ &\vdash \forall_{w,z} [\text{ext}_{A \wedge B}(z)(w) = (\text{ext}_A(\pi(z)))(w), \text{ext}_B(\pi'(z))(w)] \\ &\vdash \forall_{w,f} [\text{ext}_{A \supset B}(f)(w) = \lambda_{x \in A} \text{ext}_B(f(x))(w)] \end{aligned}$$

Here $p@w$ abbreviates $\text{emb}(w)(p)$, where emb is the term constructor (from the subtyping schema) that denotes the subtype embedding of World into $\text{Prop} \supset \text{Bool}$.⁷ Note that ext_{Prop} denotes the Stone embedding (the function that maps each element of a preboolean algebra to the set of ultrafilters of which it is a member); that is, ext extends Stone duality from just propositions to all meaning types.

5 Questions in a Hyperintensional Setting

Our modelling of questions is inspired by Karttunen’s (1977) idea that the extension of a question at a world is the set of its true answers there. That is, whatever the type of questions is—for now let’s just call it Que —the corresponding extensional type should be sets of propositions, i.e. $\text{Ext}(\text{Que})$ should

⁷ This is how to say in this kind of HOL that p is a member of w .

be $\text{Prop} \supset \text{Bool}$. Correspondingly, Que should be $\text{Prop} \supset \text{Prop}$ or some subtype thereof. In short, questions are (certain) propositional operators.

To start with the easy case, how should we analyze yes/no questions, or embedded *whether*-questions? (We will analyze root and embedded interrogatives the same way, for both polar and constituent questions.) On Karttunen's account, the extension at w of *whether*'(p) is the singleton set whose only member is either p or its denial, whichever is true at w . Thus, e.g. *Is Bush crazy*, or equivalently *whether Bush is crazy*, denotes the singleton of the proposition that Bush is crazy in worlds where he is, and the singleton of the proposition that he isn't in worlds where he isn't. We accept this analysis, except that for us the question itself has to be a hyperintension rather than an intension. The key ingredient in our analysis is the meaning *whether*' of the interrogative complementizer *whether*, for which we give the following meaning postulate:

(9) **Meaning Postulate for Whether**

$$\vdash \text{whether}' = \lambda_{p'} \lambda_p [p \text{ and}' ((p \text{ equals}' p') \text{ or}' (p \text{ equals}' \text{not}'(p')))]$$

Here *equals*' is the constant whose interpretation is the meaning of the verb *equals*, which refers to true equality (Appendix (23)).

(10) **Example (Polar Question)**

a. Is Bush crazy?/whether Bush is crazy

b. Meaning:

$$\lambda_p [p \text{ and}' ((p \text{ equals}' \text{crazy}'(\text{Bush}')) \text{ or}' (p \text{ equals}' \text{not}'(\text{crazy}'(\text{Bush}')))]$$

c. Extension at w :

$$\lambda_p [p@w \wedge ((p = \text{crazy}'(\text{Bush}')) \vee (p = \text{not}'(\text{crazy}'(\text{Bush}')))]$$

What about constituent questions, such as *which dog barked*? For Karttunen, the extension at w is the set of all propositions true at w of the form *barked*'(x) for x an individual such that *dog*'(x) is true at w . That is, Karttunen-extensions for constituent questions contain only *positive* true answers. As discussed in Pollard ms. 2, we propose to include also the *negative* true answers; that is, in a world where Fido barks and Spot doesn't, the extension of the 'plus-or-minus' K - (hereafter, K_{\pm}) question will have as members, possibly *inter alia*, both the proposition that Fido barks *and* the proposition that Spot *doesn't* bark. The essence of this analysis is captured in our semantics for the interrogative determiner *which*:

(11) **The Meaning of which**

$$\vdash \text{which}' = \lambda_P \lambda_Q \lambda_p \text{exists}'(P)(\lambda_x ((\text{whether}'(Q(x))(p))))$$

Here $x : A$ for A a hyperintensional type; $P, Q : A \supset \text{Prop}$; and $p : \text{Prop}$.

Then our analysis of a simple constituent question looks like this;

(12) **Example (Constituent Question)**

- a. Which dogs bark?
- b. Meaning: $\text{which}'(\text{dog}')(\text{bark}')$
- c. Expanding the abbreviation: $\lambda_p[\text{exists}'(\text{dog}')(\lambda_x(p \text{ and}' ((p \text{ equals}' \text{bark}'(x))\text{or}' (p \text{ equals}' \text{not}'(\text{bark}'(x)))))))]$
- d. Extension at w :
 $\lambda_p[\exists_x[\text{dog}'(x)@w \wedge (p@w \wedge ((p = \text{bark}'(x)) \vee (p = \text{not}'(\text{bark}'(x)))))]]$

Here the constant exists' denotes the hyperintensional counterpart of the usual existential generalized determiner. (Meaning postulates for some representative hyperintensional generalized determiners are given in the Appendix (24).) The upshot is that, at each world w , the reference of the interrogative sentence *which dogs bark* is the range of the function that maps each w -dog to the proposition which correctly answers, for w , the question of whether or not s/he barks.

6 What Have We Gained? What Have We Lost?

We have developed a new modelling of questions within a fine-grained theory of linguistic meanings, a theory which was independently motivated by a number of long-standing problems not specifically connected with questions. And this modelling in turn permits us to distinguish questions from each other in an appropriately more fine grained way.

For example, suppose S_1 and S_2 are English sentences that express mutually entailing but nonidentical propositions. It seems clear that one might wonder whether S_1 without wondering whether S_2 . This is problematic for MS, but not for us, because it is easy to show that the function denoted by $\text{whether}'$ is injective.

Next, consider a constituent interrogative embedded under a “question-resolving” verb, e.g. *Mary knows (in precise detail) which students in Linguistics 680 are vegetarians*. For this to be true, Mary needs to know, at minimum, for each Linguistics 680 student, whether or not s/he is a vegetarian. But she does not need to know, for each vegetarian, whether or not s/he is taking Linguistics 680. And it would seem that a minimum requirement for predicting this fact must be that the two sentences in (2) express different questions. The G-semantics does not meet this requirement, but the K_{\pm} -semantics (both in its original form and in the hyperintensional generalization) does.

To summarize, the hyperintensional semantics of questions generalizes the K_{\pm} -semantics, in the sense that for any hyperintensional question, the corresponding K_{\pm} -question (should it be needed) is its Stone dual, directly obtainable by application of ext . Moreover, recalling that for any function f , the induced equivalence relation on the domain is the relation of being mapped to the same value by f , it is not hard to see (Pollard ms. 2) that the corresponding G-question is exactly the equivalence relation on worlds induced by the K_{\pm} -question. So it too is easily recovered if needed. Thus, nothing is lost in comparison with the traditional accounts.

References

- Adams, R.: Theories of Actuality. *Noûs* 8, 211–231 (1974)
- Barwise, J., Perry, J.: *Situations and Attitudes*. Bradford Books, Cambridge (1983)
- Carnap, R.: *Meaning and Necessity*. University of Chicago Press, Chicago (1947)
- Chierchia, G., Turner, R.: Semantics and property theory. *Linguistics and Philosophy* 11, 261–302 (1988)
- Fox, C., Lappin, S.: *Foundations of Intensional Semantics*. Blackwell, Oxford (2005)
- Gallin, D.: *Intensional and Higher Order Modal Logic*. North-Holland, Amsterdam (1975)
- Groenendijk, J., Stokhof, M.: *Studies on the Semantics of Questions and the Pragmatics of Answers*. Ph. D. dissertation, University of Amsterdam (1984)
- Hamblin, C.: Questions in Montague English. *Foundations of Language* 10, 41–53 (1973)
- Henkin, L.: Completeness in the Theory of Types. *Journal of Symbolic Logic* 15, 81–91 (1950)
- Johnstone, P.: *Stone Spaces*. Cambridge University Press, Cambridge (1982)
- Jónsson, B., Tarski, A.: Boolean algebras with operators, part 1. *American Journal of Mathematics* 73(4), 891–939 (1951)
- Karttunen, L.: Syntax and semantics of questions. *Linguistics and Philosophy* 1, 3–44 (1977)
- Kripke, S.: A completeness theorem in modal logic. *Journal of Symbolic Logic* 24, 1–14 (1959)
- Kripke, S.: Semantic analysis of modal logic I: normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 9, 67–96 (1963)
- Lambek, J., Scott, P.: *Introduction to Higher-Order Categorical Logic*. Cambridge University Press, Cambridge (1986)
- Montague, R.: The proper treatment of quantification in ordinary English. In: Thomason, R. (ed.) *Formal Philosophy: Selected Papers of Richard Montague*, pp. 247–270. Yale University Press, New Haven (1974)
- Muskens, R.: Sense and the computation of reference. *Linguistics and Philosophy* 28(4), 473–504 (2005)
- Pollard, C.: Nonlocal dependencies via variable contexts. In: Muskens, R. (ed.) *Workshop on New Directions in Type-Theoretic Grammar. ESSLLI 2007, Dublin (2007)*; Revised and extended version under review for a special issue of *Journal of Logic, Language, and Information*
- Pollard, C.: Hyperintensions. *Journal of Logic and Computation* 18(2), 257–282 (2008)
- Pollard, C.: Covert movement in logical grammar. In: *ESSLLI 2008 Workshop on Ludics and Symmetric Calculi, Hamburg, August 2008* (submitted)
- Pollard, C.: Ms. 1. Stone dual semantics for natural language. Ohio State University and Universitat Rovira i Virgili. In: *Semantics in Paris 2: Semantics beyond Set Theory, CNRS/ENS, Paris, October 2007* (unpublished paper)
- Pollard, C.: Ms. 2. What do interrogative sentences refer to? Ohio State University, Universitat Rovira i Virgili, and INRIA-Lorraine. In: *Workshop on Reference to Abstract Objects, Universitat Pompeu Fabra, Barcelona, March 2008* (unpublished paper)
- Pollard, C.: 1. The *whether* underground. In: *Workshop on Ludics, Dialogue, Game Theory, and Questions, Autrans, France, May 2008* (in preparation)

- Pollard, C.: 2. Continuations for comparatives. *Semantics and Linguistic Theory* 19, Columbus, April 2009 (submitted)
- Stalnaker, R.: *Inquiry*. Bradford Books/MIT Press, Cambridge (1984)
- Stone, M.: The theory of representation for boolean algebras. *Transactions of the American Mathematical Society* 40, 37–111 (1936)
- Stone, M.: Topological representation of distributive lattices and Brouwerian logics. *Časopis pešt. mat. fys.* 67, 1–25 (1937)
- Thomason, R.: A model theory for propositional attitudes. *Linguistics and Philosophy* 4, 47–70 (1980)

Appendix

(13) Classical Connectives and Quantifiers are Definable

Here ϕ and ψ are metavariables over formulas, x is a variable of type A , and t is a variable of type Bool :

- a. $\text{true} =_{\text{def}} * = *$;
- b. $\forall_x \phi =_{\text{def}} \lambda_x \phi = \lambda_x \text{true}$;
- c. $\text{false} =_{\text{def}} \forall_t t$
- d. $\phi \wedge \psi =_{\text{def}} (\phi, \psi) = (\text{true}, \text{true})$;
- e. $\phi \supset \psi =_{\text{def}} \phi = (\phi \wedge \psi)$;
- f. $\phi \leftrightarrow \psi =_{\text{def}} [(\phi \supset \psi) \wedge (\psi \supset \phi)]$;
- g. $\sim \phi =_{\text{def}} \phi \supset \text{false}$;
- h. $\phi \vee \psi =_{\text{def}} \sim [(\sim \phi) \wedge (\sim \psi)]$; and
- i. $\exists_x \phi =_{\text{def}} \sim \forall_x \sim \phi$.

(14) Equality is an Equivalence Relation

In the following, $\alpha, \beta, \gamma, \delta$ are metavariables over terms, and ϕ, ψ are metavariables over formulas,

- a. $\vdash \alpha = \alpha$ (reflexivity)
- b. $\vdash (\alpha = \beta) \leftrightarrow (\beta = \alpha)$ (symmetry)
- c. $\vdash [(\alpha = \beta) \wedge (\beta = \gamma)] \supset (\alpha = \gamma)$ (transitivity)

(15) Substitution of Equals

- a. $\vdash [(\alpha = \gamma) \wedge (\beta = \delta)] \supset ((\alpha, \beta) = (\gamma, \delta))$
- b. $\vdash [(\alpha = \gamma) \wedge (\beta = \delta)] \supset (\alpha(\beta) = \gamma(\delta))$
- c. $\vdash (\alpha = \beta) \supset (\pi(\alpha) = \pi(\beta))$
- d. $\vdash (\alpha = \beta) \supset (\pi'(\alpha) = \pi'(\beta))$
- e. $\vdash (\alpha = \beta) \supset (\lambda_x \alpha = \lambda_x \beta)$

(16) Axioms for Cartesian Products

- a. $\vdash \alpha = *$ (α a term of type T)
- b. $\vdash \pi(\alpha, \beta) = \alpha$
- c. $\vdash \pi'(\alpha, \beta) = \beta$
- d. $\vdash (\pi(\gamma), \pi'(\gamma)) = \gamma$

(17) Axioms for Lambda Conversion

- a. $\vdash \lambda_{x \in A} \gamma[x] = \lambda_{y \in A} \gamma[y]$ if y is substitutable for x in γ (Rule α)
- b. $\vdash [\lambda_{x \in A} \gamma[x]](a) = \gamma[a]$ if a is substitutable for x in γ (Rule β)
- c. $\vdash \lambda_x(\alpha(x)) = \alpha$ if x is not free in α (Rule η)

(18) Axioms for Boolean Equality

- a. $\vdash \phi = (\phi = \text{true})$
- b. If $\vdash \phi$ and $\vdash \phi = \psi$, then $\vdash \psi$
- c. $\vdash \phi$ iff $\vdash \phi = \text{true}$
- d. $\vdash \forall_{s,t} [(s \leftrightarrow t) \supset (s = t)]$ (Boolean Extensionality)

(19) Ultrafilterhood is Definable

- 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 **(not' p)** is in S .
- b. To be explicit:
 - i. $a(S)$ is $\forall_{(p,q)} [(S(p) \wedge p \models q) \supset S(q)]$;
 - ii. $b(S)$ is $\forall_{(p,q)} [(S(p) \wedge S(q)) \supset S(\text{and}' q)]$; and
 - iii. $c(S)$ is $\neg S(\text{falsity}') \wedge \forall_p (S(p) \vee S(\text{not}' p))$.

(20) Some Constants for Word Meaning

- a. Ind: names, e.g. **Fido'**
- b. T: dummy pronouns. Up to provable equality, the only closed term of this type is $*$.
- c. T \supset Prop: intransitive verbs with dummy subjects, e.g. **rain'**
- d. Ind \supset Prop: ordinary intransitive verbs and common nouns, e.g. **dog'**, **bark'**
- e. (Ind \wedge Ind) \supset Prop: transitive verbs, e.g. **bite'**
- f. (Ind \wedge Ind \wedge Ind) \supset Prop: ditransitive verbs, e.g. **give'**
- g. (Ind \wedge Prop) \supset Prop: verbs with declarative sentential complements, e.g. **believe'**
- h. ((Ind \supset Prop) \wedge (Ind \supset Prop)) \supset Prop : (individual) determiners, e.g. **every'**.

(21) **Boolean Operations on Propositions**

- a. Truth : Prop will be interpreted as \top , the designated top.
- b. Falsity : Prop will be interpreted as \perp , the designated bottom.
- c. not' : Prop \supset Prop will be interpreted as \neg , the designated complement operation.
- d. and' : (Prop \wedge Prop) \supset Prop will be interpreted as \sqcap , the designated glb operation.
- e. or' : (Prop \wedge Prop) \supset Prop will be interpreted as \sqcup , the designated lub operation.
- f. implies' : (Prop \wedge Prop) \supset Prop will be interpreted as \Rightarrow , the designated relative complement operation.

(22) **Preboolean Axioms for Entailment**

- a. $\vdash \forall_p(p \models \text{Truth})$
- b. $\vdash \forall_p(\text{Falsity} \models p)$
- c. $\vdash \forall_{p,q}((p \text{ and}' q) \models p)$
- d. $\vdash \forall_{p,q}((p \text{ and}' q) \models q)$
- e. $\vdash \forall_{p,q,r}(((p \models q) \wedge (p \models r)) \supset (p \models (q \text{ and}' r)))$
- f. $\vdash \forall_{p,q}(p \models (p \text{ or}' q))$
- g. $\vdash \forall_{p,q}(q \models (p \text{ or}' q))$
- h. $\vdash \forall_{p,q,r}(((p \models r) \wedge (q \models r)) \supset ((p \text{ or}' q) \models r))$
- i. $\vdash \forall_{p,q}((p \text{ implies}' q) \text{ and}' p \models q)$
- j. $\vdash \forall_{p,q,r}(((r \text{ and}' p) \models q) \supset (r \models (p \text{ implies}' q)))$
- k. $\vdash \forall_p((\text{not}' p) \equiv (p \text{ implies}' \text{Falsity}))$
- l. $\vdash \forall_p((\text{not}' (\text{not}' p)) \models p)$

(23) **Three Grades of Equality**

Three families of constants of type $(A \wedge A) \supset \text{Prop}$ (for $A \in \text{HYPER}$):

- a. equals_A is interpreted as the meaning of the verb *equals*. This has 'true equality' as its extension, as expressed in this meaning postulate:

$$\vdash \forall_{w,x,y}[(x \text{ equals } y)@w = (x = y)]$$

- b. equiv_A is interpreted as the meaning of the term-of-art *is hyperintensionally equivalent to*, subject to the meaning postulate.

$$\vdash \forall_{w,x,y}[(x \text{ equiv } y)@w = \forall_{w'}(\text{ext}(x)(w') = \text{ext}(y)(w'))]$$

- c. coext_A is interpreted as the meaning of the term-of-art *is coextensive with*, subject to the meaning postulate:

$$\vdash \forall_{w,x,y}[(x \text{ coext } y)@w = (\text{ext}(x)(w) = \text{ext}(y)(w))]$$

(24) **Hyperintensional Generalized Determiners**

a. Constants:

$$\vdash \text{every}'_A, \text{exists}'_A, \text{no}'_A : ((A \supset \text{Prop}) \wedge (A \supset \text{Prop})) \supset \text{Prop}$$

b. Meaning postulates:

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

$$\vdash \forall_{w,P,Q} [\text{exists}'(P, Q)@w = \exists_x (P(x)@w \wedge Q(x)@w)]$$

$$\vdash \forall_{w,P,Q} [\text{no}'(P, Q)@w = \sim \exists_x (P(x)@w \wedge Q(x)@w)]$$