HYPERINTENSIONAL SEMANTICS

Carl Pollard

ESSLLI 2007

Dublin, August 6—10, 2007

THESE SLIDES ARE AVAILABLE AT

`http://www.ling.ohio-state.edu/~hana/hog/`

TENTATIVE COURSE OVERVIEW (DOUBTLESS OVERLY AMBITIOUS)

Day One

Lecture 1: Introduction and Motivation

Lecture 2: Problems with Standard Possible-Worlds Semantics

Day Two

Lecture 3: Soft Actualism Defined and Algebraicized

Lecture 4: The Positive Typed Lambda Calculus

Day Three

Lecture 5: Higher Order Logic with Subtypes

Lecture 6: Hyperintensions and Entailment

Day Four

Lecture 7: Worlds, Extensions, and Equivalence

Lecture 8: Quantifiers and Modality

Day Five

Lecture 9: Questions

Lecture 10: Wrap-Up

LECTURE FIVE:

HIGHER-ORDER LOGIC WITH SUBTYPES

(1) **Goals of Lecture Five**

- Review how to extend positive TLC to a HOL
- Show (following roughly Lambek and Scott 1986) how to add a subtyping schema that functions in the HOL analogously to the Axiom Schema of Separation in axiomatic set theory.

FROM TYPED LAMBDA CALCULUS TO HIGHER-ORDER LOGIC

(2) **Generalities on Extending a TLC to an HOL**

The overall approach is indebted to Church (1940), Henkin (1950), and Lambek and Scott (1986).

a. Start with a positive TLC with a basic type Bool; terms of this type are called **formulas**.
b. In an interpretation, Bool is interpreted as a set with two elements called **truth values**.
c. Add an equality symbol $=_A$: $(A \wedge A) \supset \mathrm{Bool}$ for each type A.
d. Define the usual logical connectives and quantifiers (for formulas) in terms of λ and equality.
e. Add suitable axioms and rules for proving formulas.
f. We end up with *two logics*:
   i. the *type logic* (the intuitionistic logic of the type system)
   ii. the *term logic* (the classical logic of formulas).

(3) **In a A Logic Defined in this Way:**

- all the usual TLC lambda equivalences (‘conversion’) are provable equalities of the term logic
- anything you would expect to be able to prove in classical first-order predicate logic is also provable
- quantification is permitted over variables of all types

(4) **Early Development of HOL**

a. Church's (1940) Simple Theory of Types (STT) introduced constants for boolean negation, disjunction, and universal quantification, and then defined equality via Leibniz's Law:

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

b. Henkin (1950):

i. added to STT a key axiom (*Boolean Extensionality*) identifying boolean equality with bi-implication:

$\vdash \forall_{s,t \in \mathrm{Bool}}[(s \leftrightarrow t) \supset (s = t)]$

ii. proved completeness relative to the class of set-theoretic models that bear his name.

(5) **Continued Development of HOL**

a. Gallin (1975) showed that Henkin's HOL with two basic types (besides Bool) was equivalent (in a precise sense) to Montague's IL.

b. Groenendijk and Stokhof (1980s) started using Ty2 instead of IL for NL semantics.

c. Lambek and Scott (1986):

   i. generalized to allow for an intuitionistic term logic

   ii. added *subtyping* (analogous to the Axiom of Separation in set theory)

   iii. allowed a wider class of (not necessarily set-theoretic) models (*toposes*).

(6) **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. $\mathsf{true} =_{\mathrm{def}} * = *$;

b. $\forall_x \phi =_{\mathrm{def}} \lambda_x \phi = \lambda_x \mathsf{true}$;

c. $\mathsf{false} =_{\mathrm{def}} \forall_t t$

d. $\phi \wedge \psi =_{\mathrm{def}} (\phi, \psi) = (\mathsf{true}, \mathsf{true})$;

e. $\phi \supset \psi =_{\mathrm{def}} \phi = (\phi \wedge \psi)$;

f. $\phi \leftrightarrow \ \psi =_{\mathrm{def}} [(\phi \supset \psi) \wedge (\psi \supset \phi)]$;

g. $\sim \phi =_{\mathrm{def}} \phi \supset \mathsf{false}$;

h. $\phi \vee \psi =_{\mathrm{def}} \sim [(\sim \phi) \wedge (\sim \psi)]$; and

i. $\exists_x \phi =_{\mathrm{def}} \sim \forall_x \sim \phi$.

Note that once the definitions are unpacked, all these formulas are just equations between two lambda terms.

(7) **Numerous Options for Axiomatizing HOL**

- Gallin (Ty2, 1975) essentially follows Henkin 1950.
- Carpenter (1997) essentially follows Andrews 1986.
- Lambek and Scott (1986) have $\wedge$ in the underlying type logic, subtyping, and the option of having the term logic be intuitionistic.
- We remain agnostic about how to best axiomatize HOL, and just mention some useful rules and theorems (or axioms, depending on the axiomatization).

(8) **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)

(9) **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. $\vdash (\alpha = \beta) \supset (\lambda_x\alpha = \lambda_x\beta)$

(10) **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. $\vdash (\pi(\gamma), \pi'(\gamma)) = \gamma$

(11) **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)

(12) **Axioms for Boolean Equality**

a. $\vdash \phi = (\phi = \mathsf{true})$

b. If $\vdash \phi$ and $\vdash \phi = \psi$, then $\vdash \psi$

c. $\vdash \phi$ iff $\vdash \phi = \mathsf{true}$

d. $\vdash \forall_{s,t}[(s \leftrightarrow t) \supset (s = t)]$ (Boolean Extensionality)

ADDING SUBTYPING
TO HIGHER-ORDER LOGIC

(13) **Motivation for Subtypes**

- Standard HOL has no way to say A is a *subtype* of B.
- In an intepretation I, this should mean $I(A) \subseteq I(B)$.
- Syntactic example: we might want to say that the type $\mathrm{NP}_{\mathsf{acc}}$ of NPs that can be objects of verbs is a subtype of NP.
- Semantic example: we might want to say that the type World is a subtype of the type Prop $\supset$ Bool of sets of propositions (namely the ones which are maximal consistent sets).

(14) **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 is a type
- embed_a is a term of type $A_a \supset A$; and
- Axioms:

$$\vdash \forall_{y,z \in A_a}[(\mathsf{embed}_a(y) = \mathsf{embed}_a(z)) \supset y = z)]$$
$$\vdash \forall_{x \in A}[a(x) \leftrightarrow \ \exists_{y \in A_a} x = \mathsf{embed}_a(y)]$$

(15) **What Subtypes Mean in an Interpretation I**

- $I(a)$ is a function from $I(A)$ to truth values
- $I(\mathsf{embed}_a)$ is a one-to-one function from $I(A_a)$ to $I(A)$
- the members of $I(A)$ that $I(a)$ maps to $I(\mathsf{true})$ are the ones that are embedded images of members of $I(A_a)$.

In short: $I(\mathsf{embed}_a)$ is the function that embeds into $I(A)$ the subset whose characteristic function is $I(a)$.

LECTURE SIX:

A HIGHER-ORDER THEORY OF HYPERINTENSIONS AND ENTAILMENT

(16) **Goals of Lecture Six**

- To introduce the various types of hyperintensions and their uses, and their correspponding extensional types
- To lay out the high-order theory of natural language entailment and the meanings of the English 'logic words' ("natural-language natural deduction")

INTRODUCING HYPERINTENSIONS

(17) **Review of Basic Types**

a. Recall that out only basic types are:

   i. Prop, for **propositions**, the kind of things that can be meanings of utterances of declarative sentences

   ii. Ind, for **individual concepts** (**individuals** for short), the kind of things that can be meanings of utterances of names

   iii. Bool, for **truth values**, the kind of things that can be extensions of propositions (at worlds)

   iv. Ent, for **entities**, the kind of things that can be extensions of individuals (at worlds)

b. In particular World is not a basic type.

c. But World will be *defined* as a certain subtype of the type Prop $\supset$ Bool.

d. So in an intepretation, worlds will be certain sets of propositions.

(18) **The Notion of a Kind**

a. By a **kind**, we mean a recursively defined set of types.

b. ‘Recursively defined’ here means defined at the level of the metalanguage using the informal ambient set theory (ZFC).

c. In other words, we are incorporating a form of **schematic** or **abbreviatory** polymorphism.

d. To define kinds internally, we would need a richer type theory (i.e. at the level of types we would need both lambda abstraction and a fixed-point operator).

(19) **The Kind** HYPER **of Hyperintensions**

a. Intuitively, hyperintensions are the kind of thing that can be meanings.

b. Hyperintensions can be thought of as mathematical models of Fregean senses (competing with the intensional modelling of standard PWS).

c. In fact, we will still have intensions; but we won't use them as meanings (but rather to pick out equivalences classes of meanings).

d. There are lots of different types of hyperintensions.

e. We collect these types together into a kind called HYPER.

f. Informally, the kind HYPER is obtained by closing the set of basic hyperintensional types $\{\mathrm{Prop}, \mathrm{Ind}\}$ under the positive TLC type constructors and subtyping.

(20) **The Kind** HYPER **Defined**

a. Prop $\in$ HYPER and Ind $\in$ HYPER

b. T $\in$ HYPER

c. If $A, B \in$ HYPER, then $A \wedge B \in$ HYPER.

d. If $A, B \in$ HYPER, then $A \supset B \in$ HYPER.

e. If $A \in$ HYPER and a is an A-predicate (i.e. a closed term of type $A \supset$ Bool), then $A_a \in$ HYPER.

Note that this last clause just says that any lambda-definable subtype of a hyperintensional type is also a hyperintensional type.

f. Nothing else is a hyperintensional type.

(21) **Hyperintensional Types and Syntactic Categories**

a. Since we are trying to stay neutral about NL syntax, we have no precise inventory of syntactic categories.

b. But informally, we can give some rough correspondences between hyperintensional types and categories of linguistic expressions.

c. Examples follow.

(22) **Some Constants for Word Meaning**

a. Ind corresponds to names, e.g. Fido'

b. T corresponds to dummy pronouns. Recall that up to provable equality, the only closed term of this type is $*$.

c. T $\supset$ Prop corresponds to weather verbs and other intransitive verbs with dummy subjects, e.g. rain'

d. Ind $\supset$ Prop corresponds to ordinary intransitive verbs, and to common nouns, e.g. dog', bark'

e. (Ind $\wedge$ Ind) $\supset$ Prop corresponds to ordinary transitive verbs, e.g. bite'

f. (Ind $\wedge$ Ind $\wedge$ Ind) $\supset$ Prop corresponds to ordinary ditransitive verbs, e.g. give'

g. (Ind $\wedge$ Prop) $\supset$ Prop corresponds to verbs with sentential complements (including propositional attitude verbs), e.g. know'

h. ((Ind $\supset$ Prop) $\wedge$ (Ind $\supset$ Prop)) $\supset$ Prop corresponds to determiners, e.g. every'.

(23) **Extensional Types**

a. We define a function Ext from hyperintensional types to types.

b. For each $A \in \mathrm{HYPER}$, $\mathrm{Ext}(A)$ is called the **extensional** type **corresponding** to A.

c. At any world (to be defined later), the extension at that world of any hyperintension of type A will be of type $\mathrm{Ext}(A)$.

(24) **Correspondence between Hyperintensional and Extensional Types**

The correspondence is recursively defined as follows:

a. $\text{Ext}(\text{Prop}) =_{\text{def}} \text{Bool}$

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

c. $\text{Ext}(1) =_{\text{def}} 1$

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

(25) **Linguistic Consequences**

At any world:

a. Declarative sentences denote truth values

b. Names denote to 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.

(26) **An Obvious Example**

a. For example, a VP that takes a sentential subject expresses a function from propositions to propositions, but denotes a set of propositions.

b. More specifically, at any world, *is obvious* denotes the set of propositions that are obvious in that world.

c. To make this more precise, we need to extend our theory to include worlds and extensions of hyperintensions at worlds (Lecture 7).

A HIGHER-ORDER THEORY OF ENTAILMENT

(27) **Entailment Recalled**

a. Recall that entailment is a certain preorder on propositions.

b. Pretheoretically: p entails q iff q is true in every world where p is true.

c. In standard PWS (say, in Ty2 to be specific), there is a complex closed term of type Prop $\supset$ (Prop $\supset$ Bool) (where Prop is defined as World $\supset$ Bool) that in any interpetation is interpreted as the entailment relation, namely $\lambda_p \lambda_q \lambda_w [q(w) \supset p(w)]$.

d. This is curried so that the sentential complement is the first argument and the sentential subject is the second argument (of the sentence p *entails* q.

e. Careful: the symbol $\supset$ here is for boolean implication (we never use this symbol for reverse set-inclusion).

f. So propositions are (characteristic functions of) sets of worlds, and entailment is (the curried form of the characteristic function of) the subset inclusion relation on sets of worlds.

(28) **Hyperintensional Entailment**

a. But in hyperintensional semantics, propositions and entailment are not defined in terms of worlds.

b. Instead, the term that is interepreted as the entailment relation is just a constant of type (Prop $\wedge$ Prop) $\supset$ Bool, written $\models$. (We don't bother to curry the type.)

c. At every world, the entailment relation will be the denotation of the verb *entails*.

d. So the meaning of this verb is rigid: whether one sentence utteranance follows from another is not contingent on how things are!

e. This is confusing at first for many people used to thinking of $\models$ only as a metalanguage symbol (for the semantical consequence relation between boolean terms).

f. Get used to it! This is a logical theory *about* entailment.

(29) **(Equivalence Revisited**

a. We use the constant $\equiv$ of type (Prop $\wedge$ Prop) $\supset$ Bool for mutual entailment, by adding this axiom to our theory (the first one in our semantic theory):

$\vdash \forall_{p,q}[(p \equiv q) = (p \models q \wedge q \models p)]$

b. There will never be anything in our theory that will let us prove

$\vdash \forall_{p,q}[(p \equiv q) \supset (p = q)$

c. That is, unlike the situation in intensional semantics, entailment in hyperintensional semantics is not antisymmetric.

d. Of course boolean implication is still antisymmetric:

$\vdash \forall_{s,t}[(s \leftrightarrow \ t) \supset (s = t)]$

(Remember, that is Henkin's Axiom of Boolean Extensionality).

(30) **Preorder Axioms for Entailment**

a. In accordance with (algebraicized) Soft Actualism, we will axiomatize entailment so that it is a strict boolean preorder, with the meanings of the English logic words as the boolean connectives.

b. So we start with the preorder axioms:

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

   ii. $\vdash \forall_{p,q,r} (p \models q) \supset ((q \models r) \supset (p \models r)))$

c. The natural-language significance of these is, respectively:

   i. Every declarative sentence follows from itself.

   ii. Hypothetical Syllogism is a valid rule of natural-language argumentation.

(31) **Boolean Operations on Propositions**

Next we introduce the constants that will be interpreted as the desgnated boolean operations with respect to the entailment preorder on propositions (once appropriate axioms have been provided).

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

(32) **NL Significance of the Boolean Operations**

a. Meanings of analytically true sentences will be equivalent to $\top$.

b. Meanings of analytically false sentences will be equivalent to $\bot$.

c. $\neg$ will be the meaning of *it is not the case that*.

d. $\sqcap$ will be the meaning of the sentential coordinate conjunction *and*.

e. $\sqcup$ will be the meaning of the sentential coordinate conjunction *or*.

f. $\Rightarrow$ will be the meaning of the discontinuous sentential conjunction *if . . . then*.

(33) **Axioms for Truth and $\mathsf{Falsity}$**

a. The interpretation of Truth is a top (analytic truth) with respect to entailment:

$\vdash \forall_p (p \models \mathsf{Truth})$

b. The interpretation of $\mathsf{Falsity}$ is a bottom (analytic falsehood) with respect to entailment:

$\vdash \forall_p (\mathsf{Falsity} \models p)$

(34) **NL Significance of theAxioms for Truth and $\mathsf{Falsity}$**

a. An analytically true sentence follows from anything.

b. Anything follows from an analytically false sentence

(35) **Axioms for and'**

The meaning of *and* is a glb operation with respect to entailment:

a. $\vdash \forall_{p,q}((p \textsf{ and' } q) \models p)$

b. $\vdash \forall_{p,q}((p \textsf{ and' } q) \models q)$

c. $\vdash \forall_{p,q,r}[((p \models q) \wedge (p \models r)) \supset (p \models (q \textsf{ and' } r))]$

(36) **NL Significance of the Axioms for and'**

The familiar natural deduction rules of Conjunction Elimination and Introduction are valid rules of natural language argumentation.

(37) **Axioms for or'**

The meaning of *or* is a lub operation with respect to entailment:

a. $\vdash \forall_{p,q}(p \models (p \ \mathsf{or'} \ q))$

b. $\vdash \forall_{p,q}(q \models (p \ \mathsf{or'} \ q))$

c. $\vdash \forall_{p,q,r}[((p \models r) \wedge (q \models r)) \supset ((p \ \mathsf{or'} \ q) \models r)]$

(38) **NL Significance of the Axioms for or'**

The familiar natural deduction rules of Disjunction Introduction and Elimination are valid rules of natural language argumentation.

(39) **Axioms for implies'**

The meaning of *if ... then* is a relative pseudocomplement operation with respect to entailment:

a. $\vdash \forall_{p,q}[(p \text{ implies' } q) \text{ and' } p) \models q]$

b. $\vdash \forall_{p,q,r}[((r \text{ and' } p) \models q) \supset (r \models (p \text{ implies' } q))]$

(40) **NL Significance of the Axioms for implies'**

The familiar natural deduction rules of Implication Elimination (Modus Ponens) and Introduction (Curry's Rule) are valid rules of natural language argumetation.

(41) **The Axiom of Contradiction**

The meaning of *it is not the case that* is a pseudocomplement operation with respect to entailment:

a. $\vdash \forall_p((\mathsf{not'}\ p) \equiv (p\ \mathsf{implies'}\ \mathsf{Falsity}))$

b. NL significance: Proof by Contradiction is valid in natural language reasoning.

(42) **Summary so Far**

It is a prediction of our semantic theory thus far that standard intuitionistic propositional reasoning is valid for natural language, i.e. sentence meanings preordered by entailment form a strict heyting preorder.

(43) **The Axiom of Double Negation**

The meaning of *it is not the case that* satisfies Double Negation, and so from now on we can speak of simply (relative) complements instead of (relative) pseudocomplements:

a. $\vdash \forall_p[(\mathsf{not'}\ (\mathsf{not'}\ p)) \models p]$

b. NL significance: standard classical propositional reasoning is valid for natural language, i.e. sentence meanings preordered by entailment form a strict boolean preorder.

c. Note that so far worlds have played no role in the theory; we have not even introduced them yet!

(44) **The Semantic Theory So Far**

a. $\vdash \forall_{p,q}[(p \equiv q) = (p \models q \wedge q \models p)]$

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

c. $\vdash \forall_{p,q,r}(p \models q) \supset ((q \models r) \supset (p \models r)))$

d. $\vdash \forall_p(p \models \mathsf{Truth})$

e. $\vdash \forall_p(\mathsf{Falsity} \models p)$

f. $\vdash \forall_{p,q}((p\ \mathsf{and'}\ q) \models p)$

g. $\vdash \forall_{p,q}((p\ \mathsf{and'}\ q) \models q)$

h. $\vdash \forall_{p,q,r}[((p \models q) \wedge (p \models r)) \supset (p \models (q\ \mathsf{and'}\ r))]$

i. $\vdash \forall_{p,q}(p \models (p\ \mathsf{or'}\ q))$

j. $\vdash \forall_{p,q}(q \models (p\ \mathsf{or'}\ q))$

k. $\vdash \forall_{p,q,r}[((p \models r) \wedge (q \models r)) \supset ((p\ \mathsf{or'}\ q) \models r)]$

l. $\vdash \forall_{p,q}[(p\ \mathsf{implies'}\ q)\ \mathsf{and'}\ p) \models q]$

m. $\vdash \forall_{p,q,r}[((r\ \mathsf{and'}\ p) \models q) \supset (r \models (p\ \mathsf{implies'}\ q))]$

n. $\vdash \forall_p((\mathsf{not'}\ p) \equiv (p\ \mathsf{implies'}\ \mathsf{Falsity}))$

o. $\vdash \forall_p[(\mathsf{not'}\ (\mathsf{not'}\ p)) \models p]$

(45) **In Other Words . . .**

a. These axioms say no more, and no less, than that entailment together with the boolean connectives (meanings of the English ‘logic words’) make the set of propositions into a strict boolean preorder.

b. From these we can prove:

   i. the usual equivalences for the boolean connectives

   ii. the tonocity and substitutivity theorems for these connectives (of which metalanguage versions were discussed earlier).