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 THREE:

SOFT ACTUALISM DEFINED AND ALGEBRAICIZED

(1) **Goals of Lecture Three**

a. Review the philosophical stance called **Soft Actualism**

b. Develop some math concepts about **strict boolean preorders** that we will use for modelling Soft Actualism.

c. Apply the math to modelling Soft Actualism in an informal way. [Here 'informal' means we are working in the metalanguage, with ZFC as the ambient set theory; later we will formalize our entire semantic theory in higher-order logic.]

SOFT ACTUALISM DEFINED

(2) Some Stances toward Worlds

a. **Standard PWS:** Nonactual worlds exist (at least in the sense of being countenanced by the theory) as primitives, not constructed out of anything else.

b. **David Lewis-style PWS:** A subspecies of the preceding. There is nothing special about the actual world, as compared with (so-called) nonactual ones, except that we are in it; 'nonactual' worlds are just as real to their respective inhabitants as ours is to us.

c. **Hard Actualism:** There are no non-actual worlds, period.

d. **Soft Actualism:** Nonactual worlds exist, in the sense of being logically constructed out of things in the actual world. (This is made more specific below.)

(3) **Robert Adams' (1974) Soft Actualism**

a. Propositions exist as primitives, in the sense of not being constructed out of other things (e.g. worlds).
b. The set of propositions is equipped with a 'logical structure' that enables us to specify certain subsets as the maximal consistent ones.
c. Worlds exist *only* as maximal consistent sets of propositions.
d. So there are no 'primitive' worlds that compete with the constructed ones for which get to be the possible ways things might be.
e. Adams thought of propositions as in some sense being of the actual world, so that nonactual worlds are constructed out of things in the real world.
f. But we could just as well (and indeed will) say that propositions are out there (in 'Frege's Heaven') independently of worlds, and worlds are maximal consistent sets of them.
g. This variant form of soft actualism shares with David Lewis's view that the actual world has no distinguished status in the theory.

(4) Definitions (Preorder and Induced Equivalence)

a. A binary relation $\sqsubseteq$ on a set A is called a **preorder** on A iff it is reflexive and transitive.

b. If $\sqsubseteq$ is a preorder on A, then the **equivalence on** A **induced by** $\sqsubseteq$, written $\equiv_{\sqsubseteq}$ (or just $\equiv$ if no confusion will arise), is defined by $a \equiv b$ iff $a \sqsubseteq b$ and $b \sqsubseteq a$.

   $\equiv$ is obviously an equivalence relation.

(5) **Definitions (Least and Greatest Elements)**

Suppose $\sqsubseteq$ is a preorder on A, $B \subseteq A$, and $a \in B$. Then a is said to be:

a. a **least** element of B (relative to $\sqsubseteq$) iff, for every $b \in B$, $a \sqsubseteq b$

b. a **greatest** element of B (relative to $\sqsubseteq$) iff, for every $b \in B$, $a \sqsubseteq b$.

Note that all least elements of B are equivalent, and all greatest elements of B are equivalent.

(6) **Definitions (Top and Bottom Elements)**

Suppose $\sqsubseteq$ is a preorder on A, and $a \in A$. Then a is said to be:

a. a **bottom** element of A (relative to $\sqsubseteq$) iff it is a least element of A

b. a **top** element of A (relative to $\sqsubseteq$) iff, it is a greatest element of A.

Note that all bottom elements of A are equivalent, and all top elements of A are equivalent.

(7) **Definitions (Upper Bounds and Lower Bounds)**

Suppose $\sqsubseteq$ is a preorder on A, $B \subseteq A$, and $a \in A$. Then a is said to be:

a. an **upper bound** of B iff, for every $b \in B$, $b \sqsubseteq a$

b. a **least upper bound (lub)** of B iff it is a least member of the set of upper bounds of B

c. a **lower bound** of B iff, for every $b \in B$, $a \sqsubseteq b$

d. a **greatest lower bound (glb)** of B iff it is a greatest member of the set of lower bounds bounds of B.

Note that all lubs of B are equivalent, and all glbs of B are equivalent.

(8) **Observations about Least and Greatest Elements**

Suppose $\sqsubseteq$ is a preorder on A, $B \subseteq A$, and $a \in B$. Then:

a. The following three conditions on a are equivalent:
   i. a is a greatest element of B;
   ii. a is an upper bound of B; and
   iii. a is a lub of B.
b. The following three conditions on a are equivalent:
   i. a is a least element of B;
   ii. a is a lower bound of B; and
   iii. a is a glb of B.

Caveat: These equivalences no longer hold if $a \notin B$!

(9) **Definition (Strict Bicartesian Preorder)**

A **strict bicartesian preorder** is a set A together with a preorder $\sqsubseteq$, two distinguished elements $\top$ and $\bot$, and two binary operations $\sqcap$ and $\sqcup$, such that:

a. $\top$ is a top;
b. $\bot$ is a bottom;
c. for any $a, b \in A$, $a \sqcap b$ is a glb of $\{a, b\}$; and
d. for any $a, b \in A$, $a \sqcup b$ is a lub of $\{a, b\}$.

(10) **Basics of Strict Bicartesian Preorders**

a. Another name for strict bicartesian preorders is **bounded prelattices**.
b. A bounded lattice is the same thing as a strict bicartesian *order*.
c. Strict bicartesian preorders satisfy all the equations for bounded lattices, with equality replaced by equivalence.

(11) **Definition (Strict Bicartesian Closed Preorder)**

A **strict bicartesian closed preorder** is a strict bicartesian preorder (with same notation as above) together with an additional binary operation $\Rightarrow$ such that, for any $a, b \in A$, $a \Rightarrow b$ is a **relative pseudocomplement (rpc)** of a and b (relative to the given glb operation $\sqcap$), i.e.

$a \Rightarrow b$ is a greatest member of the set $\{c \in A \mid a \sqcap c \sqsubseteq b\}$.

(12) **Basics of Strict Bicartesian Closed Preorders**

a. Another name for strict bicartesian closed preorders is **pre-heyting algebras**.
b. A heyting algebra is the same thing as a strict bicartesian closed *order*.
c. Strict bicartesian closed preorders satisfy all the equations for heyting algebras, with equality replaced by equivalence.

(13) **Definition (Strict Boolean Preorder)**

A **strict boolean preorder** is a strict bicartesian closed preorder (with same notation as above) together with an additional unary operation $\neg$ such that, for every $a \in A$,

a. $\neg a \equiv a \Rightarrow \bot$; and
b. $\neg\neg a \equiv a$.

(14) **Summary of Strict Boolean Preorders**

Putting all the pieces together: a strict boolean preorder is a set A together with a preorder $\sqsubseteq$, two distinguished elements $\top$ and $\bot$, a unary operation $\neg$, and three binary operations $\sqcap$, $\sqcup$, and $\Rightarrow$, such that

a. $\top$ is a top;
b. $\bot$ is a bottom;
c. $\sqcap$ is a glb operation;
d. $\sqcup$ is a lub operation;
e. $\Rightarrow$ is an rpc operation relative to $\sqcap$;
f. for every $a \in A$,
   i. $\neg a \equiv a \Rightarrow \bot$; and
   ii. $\neg\neg a \equiv a$.

(15) **Basic Fact about Strict Boolean Preorders**

a. Another name for strict boolean preorders is **pre-boolean algebras**.

b. A boolean algebra is the same thing as a strict boolean *order*.

c. Strict boolean preorders satisfy all the equations for boolean algebras, with equality replaced by equivalence.

d. The condition (13b) in the definition of strict boolean preorder can be replaced by the condition that, for all $a \in A$,
$a \sqcup \neg a \equiv \top$.

(16) **Tonicity (aka Functoriality) of Boolean Operations**

In a strict boolean preorder:

a. $\sqcap$ and $\sqcup$ are monotonic (aka covariant) in both arguments;

b. $\Rightarrow$ is antitonic (aka contravariant) in the first argument and monotonic in the second; and

c. $\neg$ is antitonic.

I.e., if $a \sqsubseteq b$ and $c \sqsubseteq d$, then:

a. $a \sqcap c \sqsubseteq b \sqcap d$ and $a \sqcup c \sqsubseteq b \sqcup d$;

b. $b \Rightarrow c \sqsubseteq a \Rightarrow d$; and

c. $\neg b \sqsubseteq \neg a$.

(17) **Substitutivity of Boolean Operations**

An immediate corollary of Tonicity is that, in a strict boolean preorder, the boolean operations respect equivalence, i.e. if $a \equiv b$ and $c \equiv d$, then

a. $a \sqcap c \equiv b \sqcap d$;
b. $a \sqcup c \equiv b \sqcup d$;
c. $b \Rightarrow c \equiv a \Rightarrow d$; and
d. $\neg b \equiv \neg a$.

(18) **Caveat about Tonicity and Substitutivity**

a. It's important to remember that tonicity and substitutivity are *special* properties possessed by the boolean operations.
b. That is, not just any old operation on a strict boolean preorder can be expected to have these properties.
c. That would be like expecting every function from the reals to the reals to be either nondecreasing or nonincreasing.
d. This will be directly relevant to our analysis of Logical Omniscience.

(19) **Generalizing Ultrafilters to Strict Boolean Preorders**

In a strict boolean preorder, a set $U \subseteq A$ is called an **ultrafilter** of A iff the following three conditions hold for all $a, b \in A$:

a. if $a, b \in U$ then $a \sqcap b \in U$;

b. if $a \in U$ and $a \sqsubseteq b$, then $b \in U$; and

c. For every $a \in A$, either $a \in U$ or $\neg a \in U$, but not both.

Note 1: It follows from the definitions that $\top \in U$, but $\bot \notin U$.

Note 2: Compare this definition with the one for power set algebras, to see in what sense it is a generalization.

(20) **(Non-)Principal Ultrafilters Generalized**

In a strict boolean preorder:

a. An ultrafilter is called **principal** iff it has a least element (called the **generator**), and **nonprincipal** otherwise.

b. The principal ultrafilters are the sets of the form $\{x \in A \mid a \sqsubseteq x\}$ where a is an **atom** (i.e. a member of A such that, for any $b \in A$, if $b \sqsubseteq a$ but $b \not\equiv a$, then $b \equiv \bot$).

c. If A has only a finite number of equivalence classes (with respect to the equivalence induced by the preorder), then every ultrafilter is principal.

d. But otherwise, it can be proved in ZFC that not every ultrafilter is principal.

(21) **Stone's Lemma and its Consequences**

a. It can be proved in ZFC that for any a, b in a strict boolean preorder, $a \sqsubseteq b$ iff every ultrafilter with a as a member also has b as a member. [Note: the ' only if' direction follows from the definition of ultrafilter.]

b. This is a very slight generalization of the principal lemma that Marshall Stone used to prove:

   Stone Representation Theorem: Any boolean algebra can be isomorphically embedded into the power set of the set of its ultrafilters, by mapping each element to the set of ultrafilters containing it. (This mapping is called the **Stone embedding**.)

c. As we'll see, Stone's Lemma is the foundation on which our modelling of Soft Actualism (and once we logicize everything, hyperintensional PWS) will be built.

d. An immediate corollary of Stone's Lemma is the following:

   Boolean Equivalence Theorem: Two elements of a strict boolean preorder are equivalent (relative to the equivalence relation induced by the preorder) iff they belong to the same ultrafilters.

(22) **A Historical Footnote**

a. The Stone Representation Theorem is the foundation for the the theory of n-ary boolean operators—and the accompanying algebraicization of modal logic—worked out by Tarski, McKinsey, and Jónsson in the 1940's and early 1950's.

b. In this line of work, worlds are ultrafilters, not primitives.

c. Kripke's (1959) modal semantics is essentially the same (though Kripke was not then aware of the earlier development); Kripke's *complete assignments of truth values to formulas* are just characteristic functions of ultrafilters of formulas.

d. Kripke acknowledged as much (in a footnote to his 1963).

e. Carnap's (1947) *state descriptions* are likewise just ultrafilters of formulas.

f. This whole line from Stone 1936 up to Kripke 1959 is compatible with Soft Actualism.

(23) **More History**

a. But Kripke 1963 (for reasons not clear to me) switched to taking worlds as primitives.

b. Montague (for reasons not clear to me) followed Kripke 1959, not Kripke 1963.

c. The pre-1963 (worlds-as-ultrafilters) approach to modal logic remains underappreciated (notwithstanding Goldblatt 1991), sometimes characterized as 'uninsightful' or 'a mess' (for reasons not clear to me).

d. The development of natural language semantics might have been quite different if Montague had followed Kripke 1959 (or Tarski and Jónsson 1951) instead of Kripke 1963.

(24) **Review of Strict Boolean Preorders**

Recall from (14) above that a strict boolean preorder is a set A together with a preorder $\sqsubseteq$, two distinguished elements $\top$ and $\bot$, a unary operation $\neg$, and three binary operations $\sqcap$, $\sqcup$, and $\Rightarrow$, such that

a. $\top$ is a top;
b. $\bot$ is a bottom;
c. $\sqcap$ is a glb operation;
d. $\sqcup$ is a lub operation;
e. $\Rightarrow$ is an rpc operation relative to $\sqcap$;
f. for every $a \in A$,
   i. $\neg a \equiv a \Rightarrow \bot$; and
   ii. $\neg\neg a \equiv a$.

(25) **Modelling Soft Actualism**

a. We use a strict boolean preorder A, **but not a power set algebra**.
b. The propositions are the elements of A.
c. Entailment is the preorder. **Nothing makes it antisymmetric.**
d. Truth-conditional equivalence is mutual entailment.
e. The meanings of analytically true sentences are tops (i.e. equivalent to $\top$). Nothing makes them **equal** to $\top$.
f. The meanings of analytically false sentences are bottoms (i.e. equivalent to $\bot$). Nothing makes them **equal** to $\bot$.
g. The meaning of *and* is $\sqcap$.
h. The meaning of *or* is $\sqcup$.
i. The meaning of *if . . . then* is $\Rightarrow$.
j. The meaning of *it is not the case that* is $\neg$.
k. The worlds are **all** the ultrafilters (**not just the principal ones**).
l. For a proposition to be true at a world is to a set-theoretic member of it, **not the other way around**.

(26) **Entailment and Worlds**

a. As stated above, we model entailment as the preorder $\sqsubseteq$.

b. But entailment is supposed to be the relation that holds between two propositions a and b when, no matter how things are, if a is true when things are that way, so is b.

c. This is just as it should be, because in this setting, Stone's Lemma just says that a entails b iff b is true in every world where a is true.

d. This is why we said (in (21c) above) that Stone's Lemma is the foundation on which our modelling of Soft Actualism is built.

(27) **Some Soft-Actualist Solutions**

We can now see how Soft Actualism solves three problems:

a. one part of the Granularity Problem, viz. Logical Omnisicence
b. the Nonprincipal Ultrafilters Problem
c. the Omniscience (*Simpliciter*) of Paris Hilton

(28) **The Logical Ominscience Problem Solved**

a. Suppose, e.g. a is the meaning expressed by *Paris Hilton is Paris Hilton*, and b is the meaning expressed by whichever sentence is true, the Riemann Hypothesis or its denial.

b. Even though a and b are both true at every world, nothing in Soft Actualism forces them to be the same proposition (as they would have to be in standard PWS).

c. Now let f be the propositional operator expressed by *Paris Hilton knows that* .... Then, even though a and b are equivalent, nothing forces $f(a)$ and $f(b)$ to be equivalent, since there is no reason to think f is tonic (functorial); there is no *general* principal of substitutivity for operators on a strict boolean preorder.

d. So there is no reason why there could not be an ultrafilter that has $f(a)$ as a member but not $f(b)$.

e. So it might well be that Paris Hilton knows that Paris Hilton is Paris Hilton, without knowing whether the Riemann Hypothesis is true.

(29) **The Nonprincipal-Ultrafilters Problem Solved**

a. The problem in standard PWS is that the only ultrafilters that correspond to (primitive) worlds are the principal ones.

b. But in Soft Actualism, there are no primitive worlds, just ultrafilters, and so the principal ultrafilters (if indeed there are any, which is far from certain) have no special status.

c. In particular, the nonprincipal ultrafilters are 'first-class citizens', from the point of view of 'counting' as worlds.

(30) **The Paris Hilton Ominscience Problem Solved**

a. The problem in standard PWS is that Paris Hilton seems to know the proposition which is the conjunction of all the propositions that are true in the actual world.

b. But in Soft Actualism, there is no reason to think that such a proposition even exists.

c. For such a proposition to exist is equivalent to the actual world being a principal ultrafilter, with that proposition being the atom that generates it.

d. But there is no reason to think that the actual world is a principal ultrafilter.

e. Indeed, there is not even any reason to think that the preorder of propositions has *any* principal ultrafilters!

f. To put it another way, the preorder of propositions might well be atomless.

(31) **Where do we Go from Here?**

a. We appear to be off to a promising start.

b. But it will be easier to extend our semantic theory in a precise and consistent way if we work inside a formal theory instead in the metalanguage.

c. For this, we will do what lingustic semanticists usually do, and work inside a higher-order logic (HOL).

d. To get started, we first lay out the typed lambda calculus (TLC) that our HOL is built upon.

LECTURE FOUR:

THE POSITIVE TYPED LAMBDA CALCULUS

(32) **Typed Lambda Calculi (TLCs)**

- Originated by Church and Curry in 1930's
- Can be viewed in a proof-theoretic (Curry-Howard) way (not so relevant for us here) or in a model-theoretic (Henkin-Montague) way (crucial for us)
- Underlies higher-order logic (HOL), the tool of choice for formalizing theories of natural language meaning
- Different kinds of TLCs, depending on what kind of logic the type system is based on
- The TLC we will use, **positive** TLC, is based on **positive intutitionistic propositional logic (PIPL)**.

(33) **TLC Overview: Syntax**

Syntactically, a TLC consists of:

a. a set of **types**;

b. for each type, a set of **terms** of that type; and

c. an **equivalence relation** on terms ('lambda equivalence')

Caveat: Lambda equivalence relation must not be confused with the equivalence relation on meanings of having the same extension at every world (of which mutual entailment of propositions is a special case)! In hyperintensional semantics, terms denoting meanings which are equivalent in this sense are generally *not* lambda-equivalent.

(34) **TLC Overview: Semantics**

In a set-theoretic interpretation of a TLC:

a. types are interpreted as sets;

b. a term of a given type is interpreted as a member of the set that interprets that type; and

c. lambda-equivalent terms have the same interpretation.

(35) **Types of Positive TLC**

a. Some **basic** types are given. The ones we will use are:

   Prop, for **propositions**, the kind of meanings expressed by utterances of declarative sentences

   Ind, for **individuals**, the kind of meanings expressed by utterances of names

   Bool, for **booleans** aka **truth values**, the kind of things that can be extensions of propositions

   Ent, for **entities**, the kind of things that can be extensions of individual concepts

b. T is a type, called the **unit** type

c. there are two binary **type constructors** $\wedge$ (**conjunction**) and $\supset$ (**implication**), so that if A and B are types, then so are $A \wedge B$ and $A \supset B$.

   Caveat: Linguists usually write $\langle A, B\rangle$ instead of $A \supset B$. This obscures the implicative character of the constructor.

(36) **Terms of Positive TLC**

Note: we write '$\vdash a : A$' to mean that term a is of type A.

a. There can be **nonlogical constants** of any type. In our setting, many of these will be interpreted as word meanings.

b. There is a **logical constant** $\vdash * : \mathrm{T}$.

c. For each type A there are variables $\vdash x_i^A : A$ $(i \in \omega)$.

d. There are **term constructors** $(\cdot, \cdot)$ (**pairing**), π (**left projection**), π' (**right projection**), and app (**application**), such that:

   i. if $\vdash a : A$ and $\vdash b : B$, then $\vdash (a, b) : A \wedge B$

   ii. if $\vdash a : A \wedge B$, then $\vdash \pi(a) : A$ and $\vdash \pi'(a) : B$

   iii. If $\vdash f : A \supset B$ and $\vdash a : A$, then $\vdash \mathsf{app}(f, a) : B$

e. There is a **variable binder** λ (**lambda**) such that if $\vdash x : A$ is a variable and $\vdash b : B$, then $\vdash \lambda_x b : A \supset B$

(37) **Notes on Positive TLC Terms**

a. Usually $\mathsf{app}(f, a)$ is abbreviated to $f(a)$. But conceptually it is important to remember that application is a term constructor (*modus ponens* in the Curry-Howard perspective).

b. The logical constant $*$ will be intepreted as the 'vacuous meaning' (e.g. of dummy pronouns).

c. The presence of the pairing constructor obviates the need to curry functions (while not disallowing it). E.g. we can have nonlogical constants that are interpreted as word meanings such as:

$\vdash \mathsf{Mary'} : \text{Ind}$

$\vdash \mathsf{bark'} : \text{Ind} \supset \text{Prop}$

$\vdash \mathsf{bite'} : (\text{Ind} \wedge \text{Ind}) \supset \text{Prop}$

$\vdash \mathsf{give'} : (\text{Ind} \wedge \text{Ind} \wedge \text{Ind}) \supset \text{Prop}$

$\vdash \mathsf{believe'} : (\text{Ind} \wedge \text{Prop}) \supset \text{Prop}$

$\vdash \mathsf{bother'} : (\text{Prop} \wedge \text{Ind}) \supset \text{Prop}$

(38) **Positive TLC Term Equivalences (Postponed)**

a. Usually a TLC comes equipped with an equivalence relation on terms (lambda equivalence) such that equivalent terms have the same model-theoretic interpretation.

b. Usually this relation is described in the metalanguage with the help of an equational calculus ('lambda conversion').

c. But since we will later extend our TLC to a logic with equality, we will wait and express the term equivalences *in the logic* as axioms about equality.

(39) **Set-Theoretic Interpretation of Positive TLC**

An **interpretation** I of a positive TLC assigns to to each type A a set $I(A)$, and to each constant $\vdash a : A$ a member $I(a) \in I(A)$, such that:

a. $I(\mathrm{T})$ is a singleton;

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

c. $I(A \supset B) \subseteq I(A) \Rightarrow I(B)$.

(40) **Variable Assignments (Relative to an Interpretation)**

A **variable assignment** is a function that maps each variable x to a member the set that interprets its type.

(41) **Extending an Interpretation**

Given a variable assignment α relative to an interpretation I, there is a unique extension of I to *all* terms, written I_α, such that:

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

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

c. if $\vdash a : A$ and $\vdash b : B$, then $I_\alpha((a, b)) = \langle I_\alpha(a), I_\alpha(b)\rangle$;

d. if $\vdash p : A \wedge B$, then $I_\alpha(\pi(p))$ is the first component (= projection onto $I(A)$) of $I_\alpha(p)$; and $I_\alpha(\pi'(p))$ is the second component (= projection onto $I(B)$) of $I_\alpha(p)$;

e. if $\vdash f : A \supset B$ and $\vdash a : A$, then $I_\alpha(f(a)) = (I_\alpha(f))(I_\alpha(a))$; and

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

(42) **Observations about Interpretations**

a. For any term a, $I_\alpha(a)$ depends only on the restriction of α to the free variables of a.

b. In particular, if a is a closed (i.e. has no free variables), then $I_\alpha(a)$ is independent of α so we can simply write $I(a)$.

c. Thus, an interpretation for the basic types and constants extends uniquely to all types and all closed terms.