Logical Grammar: Introduction to Hyperintensional Semantics

Carl Pollard

Department of Linguistics
Ohio State University

July 8, 2011

What was Wrong with Montague Semantics? (1/2)

Montague (late 1960s) was first to systematically apply the methods of mathematical logic to the analysis of NL meaning. Of course there were some shortcomings:

- Insufficiently fine-grained meaning distinctions, arising from modelling of meanings as intensions (functions whose domain is the set of worlds).

  Solution: hyperintensional semantics, e.g. Thomason 1980, Muskens 2005, Pollard 2008. We'll look at the last of these today.

- Inconvenient interface to semantics, complicated the analysis of quantification and unbounded dependencies.

  Solution: Replace Montague's primitive CG by a logic-based CG with hypothetical reasoning, such as Lambek calculus or linear grammar. We'll focus on this next week.

What was Wrong with Montague Semantics? (2/2)

- Didn't handle cross-sentential anaphora, donkey anaphora, novelty condition on indefinites, presupposition.

  Solution: dynamic semantic approaches such as DRT (Kamp 1981), FCS (Heim 1982), DMG (Groenendijk and Stokhof 1990).

  More recent dynamic approaches share with Montague semantics the advantage of being formulated entirely within HOL (Muskens 1994 and 1996, Beaver 2001, de Groote 2006).

  Combining dynamic semantics with hyperintensional semantics is the subject of ongoing research (Martin and Pollard 2010, 2011, in preparation).

Sources of Montague Semantics: Frege 1892

- A (NL) expression has a **sense** (which doesn't depend on how things are) and a **reference** (which does).
- For a declarative sentence, the sense is a **proposition** and the reference is that proposition's **truth value**.
- The sense of an expression is a function of the senses of its syntactic constituents.

Sources of Montague Semantics: Carnap 1947

- Worlds are **complete state descriptions**, i.e. sets of closed formulas in a certain logical language.
- An expression's sense is an **intension**, i.e. a function mapping each world to the expression's reference at that world.
- Thus propositions (intensions of sentences) map worlds to truth values.
- So propositions are essentially sets of worlds.

Kripke 1963

Worlds are unanalyzed primitives (contra Carnap and contra Kripke 1959),

Montague Grammars

- A Montague grammar defines a relation between word strings and intensions.
- This relation is defined by a primitive categorial grammar ('primitive' = no hypothetical proof).
- More precisely: the grammar defines a set of triples of (1) a string, (2) a syntactic type, and (3) an intension.
- Some of the triples are given in advance (the **lexicon**).
- Each grammar rule is equipped with:
  - a recipe for combining (usually by concatenation) the strings of the constituents to get a new string, and
  - a recipe for combining the intensions of the constituents (usually by function application) to get a new intension.

The Types of Montague's Semantic Theory

- The theory was written in an idiosyncratic higher-order language (no proof theory) called IL.
- But Gallin (1975) showed how to translate IL into Henkin's (1950) HOL, so we'll ignore IL and pretend that MS was written in HOL all along.
- Besides the truth value type t (Henkin's o) provided by the logic, there are two basic types:
  - e (Henkin's ι), the type of **entities**
  - w (Montague's s), the type of **worlds**. (Here inspired by Kripke 1963, not Carnap or Kripke 1959.)
- The type p of propositions is *defined* to be w $\rightarrow$ t (sets of worlds). This follows Carnap, modulo replacement of complete state descriptions by primitive worlds.

Propositions in Montague Semantics

- For p to be **true** at w is for w to be a member of p.
- The intensions for the NL 'logic words' are the expected boolean operation on propositions, e.g. (here $\rightsquigarrow$ is 'is translated as'):

$$\begin{gathered} \textit{and} \rightsquigarrow \lambda_{wpq}.(p\,w) \wedge (q\,w) : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p} \\ \textit{implies} \rightsquigarrow \lambda_{wpq}.(p\,w) \rightarrow (q\,w) : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{p} \end{gathered}$$

  i.e. intersection and relative complement of sets of worlds, respectively.

NL Entailment in Montague Semantics

- The centrally important relation of NL semantics, **entailment** between propositions, is modelled by subset inclusion in $\mathrm{w} \rightarrow \mathrm{t}$:

$$\mathsf{entails} =_{\mathrm{def}} \lambda_{pq}.\forall_{w}.(p\,w) \rightarrow (q\,w) : \mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{t}$$

- Unfortunately, it follows from the definition of entailment that mutually entailing propositions must be equal, i.e. entailment is an antisymmetric relation.
- The antisymmetry of entailment is generally seen as a grave foundational problem of MS.

Antisymmetry of Entailment in Montague Semantics

- *Example (logical omniscience)*: since there is only one logical truth, the set of all worlds, it is predicted that anyone who knows one of them (e.g. that Justin Bieber is Justin Bieber) knows them all (e.g. the Riemann Hypothesis or its denial, whichever is true).
- *Example (donkeys and asses)*: Since *Chiquita is a donkey* and *Chiquita is an ass* are mutually entailing, it is predicted that anyone who believes the first also believes the second, but this seems empirically wrong.
- Moral: it would be better to model NL entailment with a relation that is not antisymmetric

Toward Hyperintensional Semantics: Choice of Primitives

- In Montague semantics, worlds are a basic type, propositions are defined as sets of worlds, and entailment as the subset inclusion relation on propositions, which is unfortunately antisymmetric.
- In the kind of hyperintensional semantics (HS) considered here, we take propositions as a basic type and axiomatize the entailment relation in such a way that it is not forced to be antisymmetric.
- We then use the subtyping facility of our HOL to *define* the type of worlds as a certain subtype of the type of sets of propositions (namely: the **maximal consistent** sets of propositions).

HS: Starting to Get Technical (1/2)

- Besides the truth value type t provided by the logic and and the type e for entities, our only other basic type is p for propositions, *not* the type of worlds as in MS.
- Rather than defining entailment as in MS, we introduce entailment with a basic constant of type $\mathrm{p} \rightarrow \mathrm{p} \rightarrow \mathrm{t}$, and write axioms in HOL saying that entailment is a **preorder** (reflexive and transitive, **not** necessarily antisymmetric).
- We introduce more constants for the meanings of the NL "logic words" and type them as operations on propositions.
- Then we write more axioms in HOL which say that the set of propositions preordered by entailment is a **preboolean algebra** (like a boolean algebra, minus antisymmetry).
- As a result, we predict (like MS) that the logic of NL entailment is classical, but avoid the granularity problem.

HS: Starting to Get Technical (2/2)

- Using the subtyping facility of our HOL, we define worlds to be maximal consistent sets of propositions.
- We also add an axiom which says that the algebra of propositions "has enough worlds".
- This has as a result that, for any two proposiitons p and q, p entails q iff q is true in every world where p is true.
- That is, entailment 'works the way linguists expect it to'.
- We can then define, for each meaning (= hyperintension), what its **extension** is at each world.
- Worlds and extensions are of philosophical interest, but the grammar does not ever have to make reference to these.

Types of (Static) HS

- Basic types from the logic:
  t (truth values)
  n (natural numbers: needed for dynamic HS)
  1 (unit type)
- Basic static semantic types:
  e (entities)
  p (propositions)
- some nonbasic static semantic types:
  $\mathrm{p}_1 =_{\mathrm{def}} \mathrm{e} \rightarrow \mathrm{p}$ (unary static properties)
  $\mathrm{p}_{n+1} =_{\mathrm{def}} \mathrm{e} \rightarrow \mathrm{p}_n$ (n-ary static properties, $n > 1$)
  $\mathrm{p}_1 \rightarrow \mathrm{p}$ (static generalized quantifiers)
  $\mathrm{p}_1 \rightarrow \mathrm{p}_1 \rightarrow \mathrm{p}$ (static determiners)

Conventions for Variables

a. p, q, and r are of type p
b. A is a type metavariable
c. x is a variable of type A
c. P is a variable of type $A \to p$

Some Static Hyperintensional Constants

a. $\vdash \mathsf{truth} : \mathrm{p}$

b. $\vdash \mathsf{falsity} : \mathrm{p}$

c. $\vdash \mathsf{not} : \mathrm{p} \to \mathrm{p}$ (translates *it is not the case that*)

d. $\vdash \mathsf{and} : \mathrm{p} \to \mathrm{p} \to \mathrm{p}$ (translates *and*)

e. $\vdash \mathsf{or} : \mathrm{p} \to \mathrm{p} \to \mathrm{p}$ (translates *or*)

f. $\vdash \mathsf{implies} : \mathrm{p} \to \mathrm{p} \to \mathrm{p}$ (translates episodic *if ... then*)

g. $\vdash \mathsf{exists}_A : (A \to \mathrm{p}) \to \mathrm{p}$

h. $\vdash \mathsf{forall}_A : (A \to \mathrm{p}) \to \mathrm{p}$

i. $\vdash \mathsf{entails} : \mathrm{p} \to \mathrm{p} \to \mathrm{t}$

j. $p \equiv q =_{\mathrm{def}} (p \mathrel{\mathsf{entails}} q) \wedge (q \mathrel{\mathsf{entails}} p)$

Axioms that Say Entailment is a Preorder

a. $\vdash \forall_p . p$ entails p

b. $\vdash \forall_{p,q,r}.((p \text{ entails } q) \wedge (q \text{ entails } r)) \rightarrow (p \text{ entails } r)$

Axioms that Say NL Entailment is Classical

a. $\vdash \forall_p . p$ entails truth

b. $\vdash \forall_p .$falsity entails p

c. $\vdash \forall_{p,q} .(p$ and $q)$ entails p

d. $\vdash \forall_{p,q} .(p$ and $q)$ entails q

e. $\vdash \forall_{p,q,r} .((p$ entails $q) \wedge (p$ entails $r)) \rightarrow (p$ entails $(q$ and $r))$

f. $\vdash \forall_{p,q} . p$ entails $(p$ or $q)$

g. $\vdash \forall_{p,q} . q$ entails $(p$ or $q)$

h. $\vdash \forall_{p,q,r} .((p$ entails $r) \wedge (q$ entails $r)) \rightarrow ((p$ or $q)$ entails $r)$

i. $\vdash \forall_{p,q} .(p$ implies $q)$ and $p)$ entails q

j. $\vdash \forall_{p,q,r} .((r$ and $p)$ entails $q) \rightarrow (r$ entails $(p$ implies $q))$

k. $\vdash \forall_p .($not $p) \equiv (p$ implies falsity$)$

l. $\vdash \forall_p .($not $($not $p))$ entails p

Axioms for exists and forall

Note: these will be used to define the meanings of the NL determiners *some* and *every*.

a. $\vdash \forall_{xP}.(P\ x)\ \ \textsf{entails}\ \ (\textsf{exists}\ P)$

b. $\vdash \forall_{pP}.(\forall_{x}.(P\ x)\ \ \textsf{entails}\ \ p) \rightarrow ((\textsf{exists}\ P)\ \ \textsf{entails}\ \ p)$

c. $\vdash \forall_{xP}.(\textsf{forall}\ P)\ \ \textsf{entails}\ \ (P\ x)$

d. $\vdash \forall_{pP}.(\forall_{x}.p\ \ \textsf{entails}\ \ (P\ x)) \rightarrow (p\ \ \textsf{entails}\ \ (\textsf{forall}\ P))$

Worlds in HS (1/2)

Recall: if A is a type and a an A-predicate (i.e. a closed term of type $A \rightarrow \mathrm{t}$), then

- A_a is a type
- embed_a is a term of type $A_a \rightarrow A$; and
- Axioms:

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

- Then we call A_a a **subtype** of A.
- In an interpretation I, $I(\mathsf{embed}_a)$ denotes a one-to-one function from $I(A_a)$ to $I(A)$.

Worlds in HS (2/2)

- We now *define* the type w of worlds to be the subtype $(\mathrm{p} \to \mathrm{t})_u$ of the type of sets of propositions, where $u : (\mathrm{p} \to \mathrm{t}) \to \mathrm{t}$ is the predicate on sets of propositions such that $(u\ s)$ says "s is a maximal consistent set of propositions". (See Pollard 2008 for details.)
- Then the way to say "p is true at w" is **not** $(p\ w)$ as in MS, but rather $(\mathsf{embed}_u\ w\ p)$.
- This is abbreviated $p@w$, read "p is true at w", or simply "p at w".

The Axiom that Says there are Enough Worlds

- We need to make sure there are enough worlds to guarantee that our NL entailment relation really behaves the way a linguist expects entailment to behave.
- If we were working in a set theory with the Axiom of Choice (such as ZFC) this would come for free. But the type theory we are working in is much weaker than ZFC.
- The axiom in question is:

$$\vdash \forall_{pq}.\neg(p \mathsf{\ entails\ } q) \rightarrow \exists_w.p@w \wedge \neg q@w$$

- In English: for any two propositions p and q, if p does not entail q, then there is some world where p is true but q is false.

Extensional and Intensional Types in HS

- We define the **hyperintensional** types to be p, e, and types obtained from these using the type constructors (including 1).
- For each hyperintensional type A, the corresponding **extensional** type $\mathrm{Ext}(A)$ is defined as follows:

$$\begin{aligned}
&\mathrm{Ext}(\mathrm{p}) = \mathrm{t}\\
&\mathrm{Ext}(\mathrm{e}) = \mathrm{e}\\
&\mathrm{Ext}(1) = 1\\
&\mathrm{Ext}(A \times B) = \mathrm{Ext}(A) \times \mathrm{Ext}(B)\\
&\mathrm{Ext}(A \rightarrow B) = A \rightarrow \mathrm{Ext}(B)
\end{aligned}$$

- and the corresponding **intensional** type $\mathrm{Int}(A)$ is defined as $\mathrm{w} \rightarrow \mathrm{Ext}(A)$.
- So there are intensions.
- But they aren't the meanings; hyperintensions are.

Extensions at Worlds in HS

- The **extension** of a hyperintension $a : A$ at a world w, written $a@w : \mathrm{Ext}(A)$, is defined as follows:
  - This was already defined for $A = \mathrm{p}$.
  - $a@w = a$ for $A = \mathrm{e}$.
  - $*@w = *$
  - $\langle a, b\rangle\, @w = \langle a@w, b@w\rangle$
  - $a@w = \lambda_x.(a\ x)@w$ for $A = B \to C$.
- For each hyperintensional type A, the **intensionalizer** function is

$$\mathbf{int}_A =_{\mathrm{def}} \lambda_{xw}.x@w : A \to \mathrm{Int}(A).$$

- **int** a is called the **intension corresponding to** a.

Propositions and their Intensions

- $\mathbf{int}_{\mathrm{p}} : \mathrm{p} \to \mathrm{w} \to \mathrm{t}$ is the function that maps each proposition to the set of worlds which contain it
- Hence the family of morphisms $\mathbf{int}_A$ amounts to a generalized Stone dual at all hyperintensional types (Pollard 2011).
- For each $p : \mathrm{p}$, $\mathbf{int}\, p$ is a function from worlds to truth values.
- Hence $\mathbf{int}\, p$ is much like a Carnapian proposition, modulo the replacement of 'complete state descriptions' of (syntactic!) formulas by (semantic!) maximal consistent sets of propositions.

The Big Differences between HS and MS

- MS is written in Henkin-style HOL; HS in Lambek-Scott-style HOL.
- In MS propositions are sets of worlds; in HS it is the other way around.
- More generally: in MS meanings are intensions; in HS meanings are hyperintensions and there is a function that maps each hyperintension to its corresponding intension.
- The reason intensional semantics is not fine-grained enough is because this function isn't one-to-one.