Plural Theory in HOL

Carl Pollard

August 30, 2018

Background (1/2)

- Linguistic semantics of plurals starts with Montague semanticists like G. Link and F. Landman in the 1980s.
- They focused on set-theoretic modelling, and the models were talked about informally in English or semiformally in a kind of logical language that mixes first-order logic and lambda calculus (tricky because plurality is fundamentally non-first order).
- The key idea was to replace the traditional universe of entities (or individuals) with a more general one that includes **plural** entities as well as the familiar **singular** ones (also called **atoms**).
- The enlarged set of entities forms a kind of algebra called a **complete atomic join semilattice**, and the join operation is used to combine entities into larger ones.
- Predicates are classified according to whether they predicate only of atoms, only of plurals, or of both.

Background (2/2)

Some central issues in plural theory:

- Do we need to distinguish between **sums**), such as *Kim and Sandy*, which seem to be uniquely determined by their atoms, and things (sometimes called **groups**), such as committees and rock bands, which are not?
- Can there be plural plurals (some of whose atoms are themselves plurals)?

  The swarms converged. (ambiguous)
  The philosophers and the linguists baffle each other. (three ways ambiguous)
- How can we distinguish plural **senses** ((hyper-)intensions) from their extensions?

  If the Beatles hadn't fired Pete Best, there might have been five of them.
  The ancients thought that The Morning Star and the Evening Star were different heavenly bodies.
  Are Jennifer Lopez and J. Lo the same person?

Today's Presentation

- We revisit plural semantics, but instead of focusing on set-theoretic models, we axiomatize everything in HOL, starting with a Link-like extensional theory.
- This leads to a simpler theory, and also one which can be more easily ported into modern type theories that may not have set-theoretic models.
- Unlike the 1980s theories, we have plurals not just of entities, but of all semantic types (and this is linguistically motivated—give examples).
- Since this is LLIC, *of course* there is a monad lurking in the background (the nonempty powerset monad).
- Other issues, such as plural plurals and (hyper-)intensionality, we hope to address next week.

Link 1983 Basics

N.B.: Link's theory includes portions of matter (interpretations of mass NPs), but we ignore them here.

- The domain of individuals E is a complete atomic boolean algebra with join operation $\bigsqcup$ called **sum**, order $\sqsubseteq$ called **part-of**, and set of atoms A called **singulars**.
- The members of $\mathsf{E} \setminus \mathsf{A}$, which are sums of more than one atom, are called **plurals**.
- The bottom element of E, called 0, isn't used.
- $a \sqcup b$ abbreviates $\bigsqcup\{a, b\}$.
- Names (e.g. *Kim*) denote singulars.
- Plural NPs (e.g. *Kim and Sandy, the children*) denote plurals.

Link 1983 Predicates

- Most predicates (e.g. *built the raft, carried the piano*) denote subsets of $\mathsf{E} \setminus \{0\}$.
- Singulars of distributive predicates (e.g. *is a pop star, is dying, sees Kim* denote subsets of A.
- Plurals of distributive predicates (e.g. *are pop stars, are dying, see Kim*) work like this: if the corresponding singular predicate denotes $\mathsf{S} \subseteq \mathsf{A}$, then the plural predicate denotes the set of all the plurals that are the sum of a subset of S.

  So if P is a singular predicate (e.g. *is a pop star, is dying*) with denotation S, then the corresponding plural predicate $\mathsf{plu}(P)$ denotes $[\mathsf{S}] \setminus \mathsf{S}$, where $[\mathsf{S}]$, the **complete join semilattice generated by** $[\mathsf{S}]$, is defined by

$$[\mathsf{S}] := \{x \in \mathsf{E} | \exists T \subseteq S.(T \neq \emptyset) \wedge (x = \bigsqcup T)\}$$

Link 1983 Quantification

- Singular existential and universal QPs (e.g. *a book, every book*) get the standard Montague treatment.
- A singular definite NP like *the book* is translated as $\lambda P.P(\iota x\ \mathsf{book}'(x))$, where it's presupposed that book' denotes a singleton.
- A plural definite NP like *the books* is translated as $\lambda P.P(\sigma^* x\ \mathsf{book}'(x))$, where it's presupposed that book' denotes a set with at least two elements.

  Here, if a singular predicate Q denotes $\mathsf{S} \subseteq \mathsf{A}$, then $\sigma^* x\ Q(x)$ is taken to denote $\bigsqcup \mathsf{S}$, the plural which is the sum of the entities in the denotation of Q.

Toward a Higher Order Link-ish Theory of Plurals

- We'll work in standard HOL with basic types e and t, where for any type A, a 'set of A's' means something of type $A \to \mathsf{t}$.
- We abbreviate some useful functional terms:

$$\{x\}_A := \lambda xy : A.x = y$$
$$\mathsf{nonempty}_A := \lambda S : A \to \mathsf{t}.\exists x : A.S\ x$$
$$\mathsf{singleton}_A := \lambda S : A \to \mathsf{t}.\exists x : A.S = \{x\}$$
$$\mathsf{injective}_{A,B} := \lambda f : A \to B.\forall xy : A.[(f\ x) = (f\ y)] \to x = y$$
$$\subseteq_A := \lambda ST : A \to \mathsf{t}.\forall x : A.(S\ x) \to (T\ x)$$
$$\bigcup_A := \lambda S : (A \to \mathsf{t}) \to \mathsf{t}.\lambda x : A.\exists T : A \to \mathsf{t}.(S\ T) \wedge (T\ x)$$
$$\cup_A := \lambda ST : A \to \mathsf{t}.\lambda x : A.(S\ x) \vee (T\ x)$$

The Basic Ingredients of the Theory

- We introduce a unary type constructor written as a superscript prime, e.g. A'. For example, e' is our counterpart of Link's complete atomic join semilattice $\mathsf{E} \setminus \{0\}$ of singular and plural entities (but it is a type, not a model-theoretic object).
- We introduce the family of constants
$$\bigsqcup_A : (A' \to t) \to A'$$
For example, $\bigsqcup_e$ is our counterpart of Link's sum.
- We introduce the family of constants
$$\mathsf{atoms}_A : A' \to A \to A$$
The intuition is that for an A-singular or plural m, $(\mathsf{atoms}\ m)$ is the set of A's that 'belong' to it. So, for example, if a is an entity, there will be an e-singular m such that $(\mathsf{atoms}\ m) = \{a\}_A$. (There is nothing in Link that corresponds to this.)

Some Defined Notions

- $\sqsubseteq_A := \lambda mn : A'(\mathsf{atoms}\ m) \subseteq (\mathsf{atoms}\ n)$

  This corresponds to Link's part-of order (but at all types).
- $\mathsf{sng}_A := \lambda m : A'.\mathsf{singleton}\ (\mathsf{atoms}\ m)$
- $\mathsf{plu}_A := \lambda m : A'.\neg(\mathsf{sng}\ m)$

The Four Axioms

- The first axiom says that applying $\bigsqcup$ to a set of singulars and/or plurals corresponds to unioning their sets of atoms:

$$\vdash \forall S : A' \to \mathsf{t}.\mathsf{atoms}(\bigsqcup S) =$$
$$\bigcup(\lambda T : A \to \mathsf{t}.\exists m : A'.(S\ m) \wedge (T = (\mathsf{atoms}\ m)))$$

- And the rest say that atoms maps the A-singulars and plurals bijectively to the nonempty sets of A's:

$$\vdash \mathsf{injective}\ \mathsf{atoms}_A$$
$$\vdash \forall m : A'.\mathsf{nonempty}\ (\mathsf{atoms}\ m)$$
$$\vdash \forall S : A \to \mathsf{t}.(\mathsf{nonempty}\ S) \to \exists m : A'.S = (\mathsf{atoms}\ m)$$

Next Up

- It follows from these axioms that the types A' are all complete atomic join semilattices (without bottom).
- Moreover, we can *define* the families of functions needed to make $(-)'$ into the (type-level part of) the functor of a monad.
- That monad is (up to isomorphism) the nonempty powerset monad.

Monads in HOL

In HOL, a monad consists of

a. A unary type constructor M

b. a type-parametrized family of constants
$\vdash \eta_A : A \to \mathsf{M}A$ (*(monadic) unit*)

c. A doubly type-parametrized family of constants
$\vdash^{*A,B} : \mathsf{M}A \to (A \to \mathsf{M}B) \to \mathsf{M}B$ (*Klesili star*)

subject to three axioms given below.

Equivalently, these constants can be introduced by natural-deduction rules:

$$\frac{\Gamma \vdash a : A}{\Gamma \vdash \eta_A a : \mathsf{M}A}$$

$$\frac{\Gamma \vdash m : \mathsf{M}A \qquad \Delta, x : A \vdash f : \mathsf{M}B}{\Gamma, \Delta \vdash m^{*A,B}(\lambda x.f) : \mathsf{M}B}$$

Monad Axioms in HOL

1. left identity:
   $\vdash \forall x : A.\forall f : A \to \mathsf{M}B.(\eta x)^* f = f\ x$
2. right identity:
   $\vdash \forall m : \mathsf{M}A.m^* \eta = m$
3. associativity:
   $\vdash \forall m : \mathsf{M}A.\forall f : A \to \mathsf{M}B.\forall g : B \to \mathsf{M}C.$
   $(m^* f)^* g = m^*(\lambda x.(f\ x)^* g)$

The μ transformation

$$\mu_A := \lambda k : \mathsf{M}^2 A.k^*(\lambda m : \mathsf{M} A.m)$$

Monads Can be Defined at the Term Level

$\mathsf{M}_{A,B} := \lambda f : A \to B.\lambda m : \mathsf{M}A.m^*(\lambda x : A.\eta(f\ x))$

This shows that, categorically, a monad is a *functor*.

So-Called Monadic Application

$$\mathsf{app}_{A,B} := \lambda m : \mathsf{M}(A \to B).\lambda k : \mathsf{M}A.m^{*_{A\to B,B}}(\lambda f : A \to B.\ k^{*_{A,B}}(\lambda x : A.\eta_B(f\ x)))$$