More Plural Theory in HOL

Carl Pollard

September 13, 2018

Today's Presentation

- At first, we work in HOL with basic types $\mathtt{e}$ and $\mathtt{t}$.
- For A a type, an 'A-set' means something of type $A \to \mathtt{t}$.
- We review and elaborate the Link-isch theory, with A' renamed to Agg.
- We show that Agg is a monad.
- We then start to develop a more refined theory, for classifying predicates and handling fancier pluralities.
- The more refined theory has to be framed in a richer type theory (at least with dependent sums indexed by the natural numbers).
- (Hyper-)intensionality will have to wait.

Review of Useful Defined Terms for Set-ish Business

$$\{-\}_A := \lambda xy : A.x = y$$

$$\mathsf{nonempty}_A := \lambda S : A \to \mathsf{t}.\exists x : A.S\ x$$

$$\mathsf{singleton}_A := \lambda S : A \to \mathsf{t}.\exists x : A.S = \{x\}$$

$$\mathsf{pluralton}_A := \lambda S : A \to \mathsf{t}.\exists xy : A.(x \neq y) \wedge (S\ x) \wedge (S\ y)$$

$$\mathsf{injective}_{A,B} := \lambda f : A \to B.\forall xy : A.[(f\ x) = (f\ y)] \to x = y$$

$$\subseteq_A := \lambda ST : A \to \mathsf{t}.\forall x : A.(S\ x) \to (T\ x)$$

$$\bigcup_A := \lambda S : (A \to \mathsf{t}) \to \mathsf{t}.\lambda x : A.\exists T : A \to \mathsf{t}.(S\ T) \wedge (T\ x)$$

$$\cup_A := \lambda ST : A \to \mathsf{t}.\lambda x : A.(S\ x) \vee (T\ x)$$

Link-isch Theory Basics (1/2)

- We introduce a unary type constructor Agg of *aggregates* (previously written as a superscript prime).
- We introduce the type-indexed family of constants

$$\mathsf{atoms}_A : (\mathsf{Agg}\ A) \to A \to \mathsf{t}$$

axiomatized as bijections from the A-aggregates to the nonempty A-sets:

$$\vdash \mathsf{injective\ atoms}_A$$

$$\vdash \forall m : \mathsf{Agg}\ A.\mathsf{nonempty}\ (\mathsf{atoms}\ m)$$

$$\vdash \forall S : A \to \mathsf{t}.(\mathsf{nonempty}\ S) \to \exists m : \mathsf{Agg}\ A.S = (\mathsf{atoms}\ m)$$

- We define singular and plural aggregates straightforwardly:

$\mathsf{singular}_A := \lambda m : \mathsf{Agg}\ A.\mathsf{singleton}\ (\mathsf{atoms}\ m)$

$\mathsf{plural}_A := \lambda m : \mathsf{Agg}\ A.\mathsf{pluralton}\ (\mathsf{atoms}\ m)$

Link-isch Theory Basics (2/2)

- We define our counterpart of Link's part-of order as follows:

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

which makes the bijection from A-aggregates to nonempty A-sets into an order-isomorphism.

- We introduce a family of constants corresponding to Link's sum:

$$\bigsqcup_A : ((\mathsf{Agg}\ A) \to t) \to (\mathsf{Agg}\ A)$$

axiomatized so that applying $\bigsqcup$ to a set of aggregates corresponds to unioning their sets of atoms:

$$\vdash \forall S : (\mathsf{Agg}\ A) \to \mathsf{t}.\mathsf{atoms}(\bigsqcup S) = \\ \bigcup(\lambda T : A \to \mathsf{t}.\exists m : \mathsf{Agg}\ A.(S\ m) \wedge (T = (\mathsf{atoms}\ m)))$$

This makes the order isomorphism into a complete join-semilattice isomorphism.

The Aggregate Monad (1/2)

Since the nonempty powerset functor has a well-known monad structure (namely the nondetermism monad), we can use the complete join-semilattice isomorphism described above to transfer that structure to Agg. This results in the following axioms (and in the case of μ, a definition):

Unit:

$\eta_A : A \rightarrow (\mathsf{Agg}\ A)$

$\vdash \forall x : A.(\mathsf{atoms}\ (\eta_A\ x)) = \{x\}$

Multiplication:

$\mu_A : (\mathsf{Agg}^2 A) \rightarrow (\mathsf{Agg}\ A)$

$\mu_A := \lambda m : \mathsf{Agg}^2\ A. \bigsqcup_A (\mathsf{atoms}\ m)$

The Aggregate Monad (2/2)

Functor at level of arrows:

$\mathsf{Agg}_{A,B} : (A \to B) \to (\mathsf{Agg}\ A) \to (\mathsf{Agg}\ B)$

$\vdash \forall f : A \to B. \forall m : \mathsf{Agg} A. \mathsf{atoms}\ (\mathsf{Agg}_{A,B}\ f\ m) =$
$\lambda y : B. \exists x : A. (\mathsf{atoms}\ m\ x) \wedge y = (f\ x)$

Monadic application:

$\mathsf{mapp}_{A,B} : (\mathsf{Agg}(A \to B)) \to (\mathsf{Agg}\ A) \to (Agg\ B)$

$\vdash \forall h : \mathsf{Agg}\ (A \to B). \forall m : \mathsf{Agg}\ A. (\mathsf{atoms}\ (\mathsf{mapp}\ h\ m)) =$
$\lambda y : B. \exists f : A \to B. \exists x : A. (\mathsf{atoms}\ h\ f) \wedge (\mathsf{atoms}\ m\ x) \wedge$
$(y = (f\ x))$

Kleisli star:

$\star_{A,B} : (\mathsf{Agg}\ A) \to (\mathsf{A} \to (\mathsf{Agg}\ B)) \to (\mathsf{Agg}\ B)$

$\vdash \forall m : \mathsf{AggA}. \forall f : A \to (\mathsf{Agg}\ B). (\mathsf{atoms}\ m \star f) =$
$\lambda y : B. \exists x : A. (\mathsf{atoms}\ m\ x) \wedge (\mathsf{atoms}\ (f\ x)\ y)$

Nonquantificational NPs

- As in traditional accounts, we translate names of entities with constants of type e, e.g. $\mathsf{j} : \mathsf{e}$ (*John*), $\mathsf{m} : \mathsf{e}$ (*Mary*).
- *And* is treated as ambiguous between its familiar boolean meaning (for conjoining truth values or functions with final result type t) and the new meaning $\sqcup$ (finite sum of aggregates).
- Entities can't be summed, but the corresponding singular aggregates can, e.g. $(\eta\ \mathsf{j}) \sqcup (\eta\ \mathsf{m}) : \mathsf{Agg}\ e$ (*John and Mary*).

Indifferent Predicates

- Predicates which can predicate of both singlars and plurals, such as *performed*, are treated as sets of aggregates, i.e. (for entities) $(\mathsf{Agg}\ \mathsf{e}) \to t$:

  $\mathsf{perform}\ ((\eta\ \mathsf{m}) \sqcup (\eta\ \mathsf{j}))$ (*Mary and John performed.* [as a unit])
  $\mathsf{perform}\ (\eta\ \mathsf{m})$ (*Mary performed.*)

- But *Mary and John performed* also has a distributive) reading, usually expressed using boolean conjunction.
- And *Mary performed* is usually analyzed as having an entity predicate (type $\mathsf{e} \to \mathsf{t}$).

Distributivity (1/2)

- We define an aggregate predicate to be *distributive* provided it holds of an aggregate iff it holds to the aggregate's singular subparts:

$$\mathsf{distrib}_A := \lambda T : (\mathsf{Agg}\ A) \rightarrow \mathsf{t}.\forall m : \mathsf{Agg}\ A.(T\ m) \leftrightarrow$$
$$(\forall n : \mathsf{Agg} A.((\mathsf{singular}\ n) \wedge (n \sqsubseteq m)) \rightarrow (T\ n))$$

- We analyze distributive predicates (e.g. *die*) as aggregate predicates which are *axiomatically* distributive:

$$\vdash \mathsf{die} : (\mathsf{Agg}\ \mathsf{e}) \rightarrow \mathsf{t}$$
$$\vdash (\mathsf{distrib}\ \mathsf{die})$$

Distributivity (2/2)

- For each type A we can define two functions that set up a bijection between the A-predicates and the distributive $(\mathsf{Agg}\ A)$-predicates, called **individualization** and **distributivization**:

$$\mathsf{indiv}_A := \lambda T : (\mathsf{Agg}\ A) \to \mathsf{t}.\lambda x : A.T(\eta\ x)$$

$$\mathsf{dist}_A := \lambda S : A \to \mathsf{t}.\lambda m : \mathsf{Agg}\ A.\forall x : A.(\mathsf{atoms}\ m\ x) \to (S\ x)$$

Indifferent Predicates Revisited

- Any aggregate predicate T can be mapped to a distributive predicate, namely $\mathsf{dist}\ (\mathsf{indiv}\ T)$.
- For example, the distributive reading of *Mary and John performed* can be expressed (*without* boolean conjunction):

$$\mathsf{dist}\ (\mathsf{indiv\ perform})\ ((\eta\ \mathsf{m}) \sqcup (\eta\ \mathsf{j}))$$

- Also, *Mary performed* can be expressed with an entity predicate:

$$\mathsf{indiv\ perform\ m}$$

Singular and Plural Nouns (1/3)

N.B.: Here, by 'noun', we really mean 'count noun'.

- On a first pass, we'll treat plural noun denotations as distributive aggregate predicates ($(\mathsf{Agg}\ \mathsf{e}) \rightarrow \mathsf{t}$) and singular nouns as their individualizations (*a fortiori*, entity predicates ($\mathsf{e} \rightarrow \mathsf{t}$):

$$\vdash \mathsf{donkeys} : (\mathsf{Agg}\ e) \rightarrow \mathsf{t}$$
$$\vdash \mathsf{distrib\ donkeys}$$
$$\mathsf{donkey} := \mathsf{indiv\ donkeys}$$

- $\mathsf{donkeys}\ ((\eta\ \mathsf{c}) \sqcup (\eta\ \mathsf{b}))$ (*Chiquita and Burrita are donkeys.*)
  $\mathsf{donkey}\ \mathsf{c}$ (*Chiquita is a donkey.*)

Singular and Plural Nouns (2/3)

- This treatment of plural nouns isn't quite right, because plural noun denotations can't hold of entities, or even of singular aggregates:

  a. Mary is/are/wants to be pop stars.

- Rather, they (and other plural predicates, such as *be alike* and *hate each other*) can only hold of *plural* aggregates (*John and Mary, the children*).
- In fact, it seems we really should say something stronger: that they can only be *predicated* of plural aggregates

Singular and Plural Nouns (3/3)

- That is, the following examples aren't merely false:

  a. The linguist/Mary is/are wants to be pop stars.
  b. The farmer/ Pedro was/were alike.
  c. The mathematician/Fermat hated each other.

  but rather are semantically ill-typed: negating them does not improve them.

- We'll ignore this issue for now; we'll eventually resolve it by adding a separate type constructor for plurals.

Definites

- We assume *the* is ambiguous between $\mathsf{the}_{\mathsf{sng}} : (\mathsf{e} \ \rightarrow \mathsf{t}) \rightarrow \mathsf{t}$, which presupposes the argument predicate is a singleton, and $\mathsf{the}_{\mathsf{plu}} : ((\mathsf{Agg}\ \mathsf{e}) \rightarrow \mathsf{t})) \rightarrow \mathsf{t}$, which presupposes the argument predicate holds of a plural aggregate.
- $\mathsf{the}_{\mathsf{sng}} := \iota$
  $\mathsf{the}_{\mathsf{plu}} := \bigsqcup$
- $\mathsf{linguists}\ (\bigsqcup \mathsf{men})$ (*The men are linguists.*)
  $\mathsf{philosopher}\ (\iota\ \mathsf{woman})$ (*The woman is a philosopher.*)

Strictly Plural Predicates (1/2)

- *Strictly plural* predicates can hold only of plurals. (In fact, they can only be *predicated* of plurals, but our current theory doesn't account for that.)

  a. The linguists/Cynthia and Mike are alike.
  b. *The linguist/Cynthia is alike.

- Other examples: *converge, disperse, go to the same gym, see different movies, hate each other*
- Some strictly plural predicates aren't fussy about what their arguments are plurals *of*:

  c. Kim and Sandy/juggling and miming/donkeys and burros/the Riemann Hypothesis and the Goldbach Conjecture/17 and 37/conjunction and sum are alike.

Strictly Plural Predicates (1/2)

- We can analyze such predicates as families of type-indexed (ordinary) predicates, e.g.

$$\mathsf{alike}_A : (\mathsf{Agg}\ A) \rightarrow \mathsf{t}$$

- How can we analyze the ambiguity of sentences like *the linguists and the philosophers are alike*?