A Single-Type Semantics for Natural Language

Kristina Liefke

Tilburg Center for Logic and Philosophy of Science (Netherlands)

Formal Semantics and Pragmatics: Discourse, Context, and Models
University of Latvia, Riga, November 19, 2010
Partee’s Conjecture

Barbara H. Partee, “Do We Need Two Basic Types?” (2006)

Single-Type Hypothesis

- The distinction between entities and propositions is inessential for the construction of a rich linguistic ontology.
- All object-types can be bootstrapped from a single basic type.

Montague Semantics

- Basic types: $e$ (for entities) and $\langle s, t \rangle$ (for propositions);
- Derived types: $\langle e, \langle s, t \rangle \rangle$ (for properties), $\langle e, \langle e, \langle s, t \rangle \rangle \rangle$ (relations).

Single-Type Semantics

- Basic type: $q$ (for entities and propositions);
- Derived types: $\langle q, q \rangle$ (for properties), $\langle q, \langle q, q \rangle \rangle$ (for relations).
Partee’s Motivation


**Single-Category Hypothesis**
- The distinction between NPs and sentences is inessential for the generation of complex modern languages.
- All categories can be constructed from a **single basic category**.

**Categorial Syntax**
- Basic categories: **NP** (for noun phrases) and **S** (for sentences);
- Derived categories: **NP\S**, (**NP\S**)/**NP** (for (in-)transitive verbs).

**Monocategoric Syntax**
- Basic category: **X** (for noun phrases and sentences);
- Derived categories: **X\X**, (**X\X**)/**X** (for (in-)transitive verbs).
Arguments for the Single-Category Hypothesis

1. Evolutionary linguistics  The NP/S-distinction is a contingent property of modern grammar.

2. Nominalization  Many sentences can be converted into NPs (cf. Carstairs-McCarthy’s ‘Nominalized English’).

3. Language acquisition  The function of NPs is often ambiguous between reference and assertion (Snedeker et al., 2007).

- Arguments 2, 3 have direct counterparts in semantics.
- Especially, the construction of a single-base syntax for NL can be paralleled on the level of semantics.

Objective  Define such a semantics!
The Plan

1. Single-Type Semantics
   - Survey the objects in our domains.
   - Describe their interrelations.

2. Linguistic Application
   - Show that single-type semantics models the PTQ-fragment.

3. Wrap-Up
Objects

Single-Type Objects

Basic and Derived Objects

Let \( A := D_e \) be the domain of entities.

- **Individuals**: Filters (or ideals) in \( \mathcal{P}(A) \)
- **Propositions**: Filters (or ideals) in \( \mathcal{P}(A) \)
- **Worlds**: Filters (or ideals) in \( \mathcal{P}(A) \)

\[ \begin{align*}
\{ & \text{basic} \\
\{ & \text{derived} \\
\end{align*} \]

- **Individual Concepts**: Fct’s from worlds to individuals
- **Proposit’l Concepts**: Fct’s from worlds to proposit’s
- **Properties**: All fct’s in the domain hierarchy
Single-Type Objects: Individuals

- Entities and propositions are unsuitable single-type domains:
  - Entities (type $e$) lack an algebraic structure;
  - Propositions (type $\langle s, t \rangle$) cannot represent entities.
- We adopt basic sets of sets of entities in $D_{\langle e, t \rangle, t} := \mathcal{P}^2(A)$:
  
  \[
  \text{John} := \{ \text{is self-identical, is a man} \} \\
  \text{Mary} := \{ \text{is self-identical, is a woman} \}
  \]
- We interpret connectives via set-theoretic operations:
  
  \[
  [\text{John and Mary}] := \text{John} \cap \text{Mary} = \{ \text{is self-identical} \} \\
  [\text{John or Mary}] := \text{John} \cup \text{Mary} = \{ \text{is self-identical, ...} \} \\
  [\text{not John}] := \text{John}' = \{ \text{is a woman} \}
  \]
Single-Type Objects: Propositions

- We identify individuals with filters in $\mathcal{P}(A)$:
  - Individuals are closed under finite intersection:
    
    $$\text{John} := \{\text{is self-identical, is a man}\}$$
    $$\implies \text{John} := \{\text{is self-identical} \cap \text{is a man}\}$$
  
  - Individuals are closed under entailment:
    
    $$\{\text{is self-identical} \cap \text{is a man}\} \subseteq \{\text{is self-identical}\}$$
    $$\implies \text{John} := \{X | \text{is self-identical} \cap \text{is a man} \subseteq X\}$$

- Trivially true propositions assert a property’s membership in a filter:
  $$\{X | \text{is self-identical} \cap \text{is a man} \subseteq X\} \cap \{\text{is a man}\}$$

- Informative propositions constitute proper filter extensions:
  $$\text{John}_{\text{new}} := \{X | \text{is self-identical} \cap \text{is a man} \cap \text{runs} \subseteq X\}$$
Objects

Single-Type Objects: Properties

Property attribution := Filter extension

- We interpret predicates as functions from filters to filters.
- A filter in the function’s domain may not be more informative than the filter from its range:
  \[
  \text{[runs]} : \text{John} \rightarrow \text{John}_{\text{new}}
  \]
  s.t. \( \text{John} \subseteq \text{John}_{\text{new}} \).
- Relations have a similar representation:
  \[
  \text{[loves]} : \langle \text{John, Mary} \rangle \rightarrow \langle \text{John}_{\text{new}}, \text{Mary}_{\text{new}} \rangle
  \]
  where \( \text{John}_{\text{new}} := \text{John} \cap \{ \text{loves Mary} \} \),
  \( \text{Mary}_{\text{new}} := \text{Mary} \cap \{ \text{is loved by John} \} \).
Single-Type Objects: Individual Concepts

To enable proper filter extensions, we associate individual constants with families of filters in \( \mathcal{P}(A) \):

1. Interpret individual constants as individual concepts, i.e. functions \( f : \mathcal{P}^2(A) \rightarrow \mathcal{P}^2(A) \);
2. Apply individual concepts to different worlds in \( \mathcal{P}^2(A) \);
3. Obtain different world-specific individuals.

We have two sorts of basic-type objects: individuals, worlds. Worlds will always be at least as informative as their individuals:
Single-Type Objects: Example

- Let \( w := \{ X \mid \text{is self-identical } \cap \text{is a man} \subseteq X \} \), where
  
  \[
  \begin{align*}
  \text{is self-identical} & := \{ \text{John, Mary, the Moon} \}, \\
  \text{is a man} & := \{ \text{John} \}, \\
  \text{is not a man} & := \{ \text{Mary} \}.
  \end{align*}
  \]

- Then, since
  
  \[
  \begin{align*}
  \{ \text{is self-identical} \} & \subseteq \text{John}(w), \\
  \{ \text{is a man} \} & \subseteq \text{John}(w), \\
  \implies \text{John}(w) = w & = \{ X \mid \text{is self-identical } \cap \text{is a man} \subseteq X \}
  \end{align*}
  \]

- But, since
  
  \[
  \begin{align*}
  \{ \text{is a man} \} & \not
  \subseteq \text{the Moon}(w), \\
  \{ \text{is not a man} \} & \not
  \subseteq \text{the Moon}(w), \\
  \implies \text{the Moon}(w) & \neq w
  \end{align*}
  \]
Single-Type Objects: Partiality

We need to separate a constant’s denotation and complement:

1. Only thus can we account for truth-value gaps.
2. Only thus can we enable proper filter extensions.

Implementation Remove LEM from the axioms of our algebra:

- Weaken the structure on single-type domains;
- Split the interpretation and assignment functions;
- Partialize the algebraic operations.
Partial Truth

- Single-type objects are defined by their properties & privations:
  
  \[ \text{John} := \langle \text{John}^+, \text{John}^- \rangle, \text{ with} \]
  
  \[ \text{John}^+ \text{ The set of properties John is known to have;} \]
  
  \[ \text{John}^- \text{ The set of properties John is known to lack.} \]

- We split worlds into a denotation- and a complement-world.

- Truth at a world is defined as an object’s inclusion in a world.

- Compatibility with a world is defined as an object’s inclusion of a world.

- Both notions are double-barrelled.
Worlds

In single-type semantics, possible worlds serve a triple-duty:

1. Worlds obtain families of individuals (propositions);
2. Worlds enable the evaluation of their truth value;
3. Worlds define the modal operators.

Worlds are well-defined:

- Worlds are partial and consistent;
- Worlds are partially ordered;
- Worlds are extensional.
Worlds and Accessibility

The accessibility relation $R_{\langle q,q \rangle, q}$ inherits its properties from the partial order on $D_q$:

**Definition (Accessibility)**

Let $\lambda j \lambda i. R_{ij}$ formalize $j$ includes the information of $i$. Then,

i. $\forall i. R_{ii}$ (Reflexivity);
ii. $\forall i \forall j \forall k. (R_{ij} \land R_{jk}) \rightarrow R_{ik}$ (Transitivity).

- Other properties (e.g. symmetry, euclideanness) can be stipulated via non-logical axioms.
- From $R$, the modal operators are standardly defined.
**An Acid Test**

**Objective** Define a single-type semantics for NL!

- To compare our semantics’ modeling power with that of multi-type systems, we model the PTQ-fragment.
- **Measure for success** Our semantics’ ability to interpret all expressions of the fragment.
- We associate complex expressions with sets of syntactic structures.
- We render syntactic structures into single-type terms via type-driven translation.
Show that single-type semantics models the PTQ-fragment.

Conventions

We employ the following typographical conventions:

<table>
<thead>
<tr>
<th>Variable</th>
<th>$TY^3_0$ type</th>
<th>$TY_2$ type</th>
<th>Objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>$i, j$</td>
<td>$q$</td>
<td>$\langle\langle e, t\rangle, t\rangle := s$</td>
<td>worlds</td>
</tr>
<tr>
<td>$z$</td>
<td>$q$</td>
<td>$\langle\langle e, t\rangle, t\rangle := e$</td>
<td>individuals</td>
</tr>
<tr>
<td>$x, y$</td>
<td>$\langle q, q\rangle$</td>
<td>$\langle s, e\rangle$</td>
<td>ind. concepts</td>
</tr>
<tr>
<td>$P_1, P_2$</td>
<td>$\langle\langle q, q\rangle, \langle q, q\rangle\rangle$</td>
<td>$\langle s, \langle e, \langle s, t\rangle\rangle\rangle$</td>
<td>FO properties</td>
</tr>
<tr>
<td>$Q$</td>
<td>$\langle\langle q, q\rangle, \langle q, q\rangle\rangle, \langle q, q\rangle\rangle$</td>
<td>$\langle s, \langle\langle s, \langle e, \langle s, t\rangle\rangle, t\rangle\rangle$</td>
<td>SO properties</td>
</tr>
</tbody>
</table>
Show that single-type semantics models the PTQ-fragment.

### Basic PTQ-Translations

<table>
<thead>
<tr>
<th>Words</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>John, Mary, Bill, ...</td>
<td>(\lambda i.\text{john }i,\ldots)</td>
</tr>
<tr>
<td>runs, walks, talks, ...</td>
<td>(\lambda i.\text{run }i,\ldots)</td>
</tr>
<tr>
<td>man, woman, unicorn</td>
<td>(\lambda i.\text{man }i,\ldots)</td>
</tr>
<tr>
<td>finds, loves, ...</td>
<td>(\lambda Q\lambda y\lambda i.Q(\lambda x.\text{find }yx,\ldots))</td>
</tr>
<tr>
<td>seeks</td>
<td>(\lambda i.\forall P\forall x(\text{seek }xP \leftrightarrow \text{try }x \text{ find }P))</td>
</tr>
<tr>
<td>rapidly, allegedly, ...</td>
<td>(\lambda i.\text{rapidly }i,\ldots)</td>
</tr>
<tr>
<td>necessarily</td>
<td>(\lambda z.\forall i((\Omega i \land Ri) \rightarrow (z \rightarrow i)))</td>
</tr>
<tr>
<td>in</td>
<td>(\lambda Q\lambda P\lambda y\lambda i.Q(\lambda x.\text{in }xPy))</td>
</tr>
<tr>
<td>believes that</td>
<td>(\lambda y\lambda x\lambda i.\text{believe }xy,\ldots)</td>
</tr>
<tr>
<td>tries to, wishes to</td>
<td>(\lambda P\lambda x\lambda i.\text{try }xP,\ldots)</td>
</tr>
<tr>
<td>is</td>
<td>(\lambda Q\lambda y.Q(\lambda x\lambda i.x = y))</td>
</tr>
<tr>
<td>some, a</td>
<td>(\lambda P_2\lambda P_1\lambda x.\exists x(P_2x \land P_1x))</td>
</tr>
<tr>
<td>every</td>
<td>(\lambda P_2\lambda P_1\lambda x.\forall x(P_2x \rightarrow P_1x))</td>
</tr>
<tr>
<td>the</td>
<td>(\lambda P_2\lambda P_1\lambda x.\exists x((\forall y(P_2y) \leftrightarrow y = x) \land P_1x))</td>
</tr>
<tr>
<td>(t_n)</td>
<td>(v_{\alpha_n} \text{ with } \alpha \in \text{Monotype})</td>
</tr>
</tbody>
</table>
Show that single-type semantics models the PTQ-fragment.

## More PTQ-Translations

<table>
<thead>
<tr>
<th></th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>[Bill walks]</td>
</tr>
<tr>
<td></td>
<td>( \text{walk (bill)} )</td>
</tr>
<tr>
<td>(2)</td>
<td>[[a man] walks]</td>
</tr>
<tr>
<td></td>
<td>( \lambda i \exists x. \text{man } x i \land \text{walk } x i )</td>
</tr>
<tr>
<td>(3)</td>
<td>[John finds a unicorn]</td>
</tr>
<tr>
<td></td>
<td>( \lambda i \exists x. \text{unicorn } x i \land \text{find john } x i )</td>
</tr>
<tr>
<td>(4)</td>
<td>[John [seeks [a unicorn]]]</td>
</tr>
<tr>
<td></td>
<td>( \lambda i. \text{try john } \exists x (\text{unicorn } x i \land \text{find john } x i) )</td>
</tr>
<tr>
<td></td>
<td>[[a unicorn(^1)[John [seeks (t_1])]]]</td>
</tr>
<tr>
<td></td>
<td>( \lambda i. \exists x (\text{unicorn } x i \land \text{try john find john } x i) )</td>
</tr>
<tr>
<td>(5)</td>
<td>[Bill [is Mary]]</td>
</tr>
<tr>
<td></td>
<td>( \lambda i. \text{bill } = \text{mary } i )</td>
</tr>
<tr>
<td>(6)</td>
<td>[Bill [is a man]]</td>
</tr>
<tr>
<td></td>
<td>( \lambda i. \exists x (\text{man } x i \land x = \text{bill } i) )</td>
</tr>
<tr>
<td>(7)</td>
<td>[Necessarily [Bill [is Bill]]]</td>
</tr>
<tr>
<td></td>
<td>( \forall i ((\Omega i \land R i) \rightarrow (\text{bill } = \text{bill } i \rightarrow i)) )</td>
</tr>
<tr>
<td>(8)</td>
<td>[Possibly [Bill [is Mary]]]</td>
</tr>
<tr>
<td></td>
<td>( \neg \forall i ((\Omega i \land R i) \rightarrow (\neg \text{bill } = \text{mary } i \rightarrow i)) )</td>
</tr>
</tbody>
</table>
We have developed a single-type semantics for NL:

- The domain $D_q$ unifies individuals, propositions, and worlds.
- From $D_q$, we can construct individual/propositional concepts, properties, and relations.
- Single-type models assign every PTQ-rendering a denotation and complement in the partial algebra.

We have developed a single-type semantics for ‘Montague’-English:

- To model larger fragments, we must define more operations (nominalization, collectivization, grinding, ...).
Thank you!
References


Partee, Barbara H. *Do We Need Two Basic Types?*, Snippets: Special Issue in Honor of Manfred Krifka (S. Beck and H. Gärtner, eds.), Vol. 20, Berlin, 2009.