A Single-Type Semantics for Natural Language

Kristina Liefke

Munich Center for Mathematical Philosophy (MCMP),
LMU Munich

NYU Semantics Group,

December 6, 2013

Partee's Conjecture

Barbara Partee, "Do We Need Two Basic Types?" (2006)

Single-Type Conjecture

- Montague's distinction between individuals and propositions is inessential for the construction of a rich semantic ontology.
- The PTQ-fragment can be modelled through ONE basic type.

Montague Semantics (EFL)

- Basic types: e (for individuals) and $(s;t)$ (for propositions);
- Derived types: $(\alpha_1 \ldots \alpha_n; e)$ and $(\alpha_1 \ldots \alpha_n; (s;t))$ for all types $\alpha_1, \ldots, \alpha_n$.

Single-Type Semantics (STS)

- Basic type: o (for individuals and propositions);
- Derived types: $(\alpha_1 \ldots \alpha_n; o)$ for all types $\alpha_1, \ldots, \alpha_n$.

Partee's Conjecture

Barbara Partee, "Do We Need Two Basic Types?" (2006)

Syntactic Category	EFL type	STS type
Proper name	e	o
Sentence	$(s; t)$	o
Complement phrase	$(s; t)$	o
Common noun	$(e; (s; t))$	$(o; o)$
Complementizers	$((s; t); (s; t))$	$(o; o)$
Sentence adverb	$((s; t); (s; t))$	$(o; o)$
Other categories	Replace e and $(s; t)$ by o	

Objective Provide formal support for Partee's conjecture:
Develop a single-type semantics for the PTQ-fragment.

Guiding Questions

- What happens if we replace e and $(s; t)$ by a single basic type?
- Under what conditions is this possible?
- What does a suitable interpretive domain for o look like?
- What are its properties?

- What effects does this change of type system have on our semantics' ability to model natural language?
- How does it influence our understanding of the relations between different objects?
- Does it make Montague's type system dispensable?

The Plan

1. Partee's Conjecture
2. Support for Partee's Conjecture
3. Challenges in Modelling the Conjecture
4. Meeting the Challenges
5. A Single-Type Semantics for the PTQ-Fragment
6. Conclusion

Support for Partee's Conjecture

Three kinds of considerations:

1. Empirical considerations (greater?) modeling power of single-type semantics w.r.t. Montague semantics
2. Formal considerations the possibility of constructing single-type models for the PTQ-fragment
3. Methodological considerations the methodological desirability of a single basic type (cf. unification ➥ simplicity, etc.)

A 'minimality test' for Montague's type system

By formulating a STS **without** reference to the types e or $(s; t)$, we provide evidence against the need for Montague's basic-type distinction.

Partee's Motivation

Andrew Carstairs-McCarthy, *The Origins of Complex Language* (1999)

Single-Category Conjecture

- The distinction between sentences and noun phrases is inessential for the generation of complex modern languages.
- All synt. categories can be obtained from ONE basic category.

Categorial Grammar (CG)

- Basic categories: NP (for noun phrases) and S (for sentences);
- Derived categories: A/B for all categories A, B.

Single-Category Syntax (SCG)

- Basic category: X (for noun phrases and sentences);
- Derived categories: A/B for all categories A, B.

Empirical Support for Partee's Conjecture (1)

1. Language development The NP/S-distinction is a contingent property of grammar, cf. (Carstairs-McCarthy, 1999)

Only STS (but not Montague semantics) explains the ff. facts:

2. Lexical syntax Many verbs select a complement that can be realized as an NP or a CP, cf. (Kim and Sag, 2005)

(2.1) a. Pat remembered [$_{\mathrm{NP}}$Bill].
b. Pat remembered [$_{\mathrm{CP}}$that Bill was waiting for her].

(2.2) a. Chris noticed [$_{\mathrm{NP}}$the problem].
b. Chris noticed [$_{\mathrm{CP}}$that the types didn't match].

- In MS, all occur's of an expr. are interpreted in the same type.
- In MS, names and CPs are interpreted in diff. types (e, $(s;t)$).

➡ MS fails to model (2.1/2a) or (2.1/2b).

Empirical Support for Partee's Conjecture (1)

2. Lexical syntax Many verbs select a complement that can be realized as an NP or a CP, cf. (Kim and Sag, 2005)

(2.1) a. Pat remembered [$_{NP}$Bill].
b. Pat remembered [$_{CP}$that Bill was waiting for her].

(2.2) a. Chris noticed [$_{NP}$the problem].
b. Chris noticed [$_{CP}$that the types didn't match].

- In MS, all occur's of an expr. are interpreted in the same type.
- In MS, names and CPs are interpreted in diff. types (e, $(s; t)$).

➥ MS fails to model (2.1/2a) or (2.1/2b).

- In STS, names and CPs are interpreted in the same type, o.

➥ STS models both (2.1/2a) and (2.1/2b).

Empirical Support for Partee's Conjecture (2)

3. Coordination English has coordinate structures with a proper name- and a CP-conjunct, cf. (Bayer, 1996).

(2.3) Pat rememb'd [$_{NP}$Bill] and [$_{CP}$that he was waiting for her]

(2.4) C. noticed [the problem] viz. [that the types didn't match]

Coordinability requirement To allow coordination, expressions must receive an interpretation in the same type.

4. CP equatives Some copular sentences equate the referents of an NP and a CP, cf. (Potts, 2002).

(2.5) [$_{NP}$The problem] is [$_{CP}$that the types don't match].

Equatability requirement To allow equation, the referents of expressions must have the same type, cf. (Heycock and Kroch, 1999).

Empirical Support for Partee's Conjecture (3)

3. Nonsentential speech Names are often used to assert a proposition about their type-*e* referent, cf. (Merchant, 2008):

Context: A woman is entering the room

Interpret $[_{\text{NP}}$Barbara Partee] as $[_{\text{S}}$Barbara Partee is arriving], or as $[_{\text{S}}$Barbara Partee is (the woman) entering the room]

➥ 'Barbara Partee' is true/false in this context.

➥ 'B. Partee' is equivalent to 'B. Partee is entering the room'.

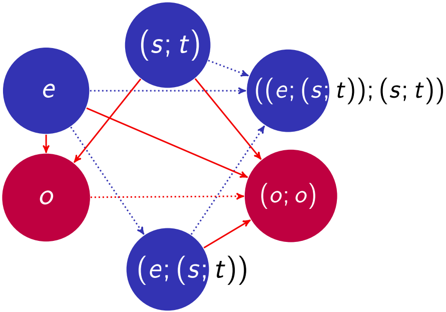
Cf. C. noticed [the problem] viz. [that the types don't match]

But 'Type-shifting' enables an easier accommodation of these facts: Some occur's of an expr. are interpreted in different types.

➥ Empirical support for Partee's conjecture will only exert little confirmatory force.

Methodological Support for Partee's Conjecture

1. Unification of types The type o bootstraps all PTQ-referents (w. the expected consequ's: simplicity, confirmation).
2. Relations between types STS extends the representational relations from Flexible Montague Grammar (Hendriks, '90):



➥ yields insight into the apparatus of types in formal semantics.

Formal Support for Partee's Conjecture

Show Single-type models exist:

1. Identify the type o (properties of Kratzer-style situations)
2. Give an o-based semantics for a fragment of English:

$[\![$you$]\!]$	the property of (being) a minimal situation containing you
$[\![$a snake$]\!]$	the property of (being) a snake-containing situation
$[\![$see$]\!]$	a fct. from two situation-properties p_1 and p_2 to a property p_3 which holds of a situation s_3 if s_3 contains 2 situations, s_1 and s_2, with the p'ties p_1, resp. p_2, where (sth. in) s_1 sees (sth. in) s_2
$[\![$You see a snake$]\!]$	the p'ty of (being) a situation in which you see a/the snake

Problem Partee's fragment is *very* small (4 words).
The presentation of its semantics is informal.

➥ It does not give compelling support for Partee's conjecture

Objective Formalize and extend Partee's model.

Challenges

1. Existing single-type semantics (e.g. models of the untyped lambda calculus, or of Henkin's theory of propositional types) are unsuitable for our purpose.
2. Simple adaptations of these semantics disable an easy definition of core semantic notions (e.g. truth, equivalence).
3. Successful single-type semantics require the introduction of new constants, and employ layered structures.

1. The Usual Suspects . . . Don't Work

Untyped λ-logic (Church, 1985; Beeson, 2005)

- Single basic type: the universal type.
- ✗ We cannot use semantics to explain the well-formedness of NL expressions.
- ✗ Untyped λ-models are quite different from models of IL (TY_2).

Theory of Propositional Types (Henkin, 1963)

- Single basic type: the Boolean type t.
- ✗ Boolean values do NOT represent individuals/propositions. (There is NO injective function from D_e (or $D_{(s;t)}$) to $\{\mathbf{T}, \mathbf{F}\}$.)

2. *Variants of* the Usual Suspects also Don't Work

Solution Replace t by another type in the theory of propos. types:

- Single basic type: the primitive type o.
- ✗ The constants $\bot_t$, $\Rightarrow_{(\alpha\,\alpha;t)}$, etc. are no longer available.
- ✗ We cannot give easy truth-conditions for sentences.
- ✗ We cannot identify equivalence relations between proper names and sentences.
- ➥ We cannot model empirical support for Partee's conjecture: esp. support from non-sentential speech (Merchant, 2008).

Our Strategy (1)

Replace the type t by $(s;t)$ in the theory of propositional types.

➥ Our STS is a model of an $(s;t)$-based subsystem of TY_2.

The type $(s;t)$ has desirable properties, cf. (Liefke, 2013):

1. Familiarity The type $(s;t)$ is closest to Partee's single-choice.
2. Algebraicity $(s;t)$ enables the truth-evaluation of sentences.
3. Representability The type $(s;t)$ enables an injective function f from individuals/propositions to type-$(s;t)$ objects:
   - f sends prop's φ to themselves, $\{w_s \mid w \in \varphi\}$.
   - f sends individuals a to sets $\{w_s \mid a \text{ exists in } w\}$.

!!! To ensure injectivity, we postulate that no two individuals exist in exactly the same indices.

Our Strategy (2)

3. Representability The type $(s; t)$ enables an injective function f from individuals/propositions to type-$(s; t)$ objects:
   - f sends prop's φ to themselves, $\{w_s \mid w \in \varphi\}$.
   - f sends individuals a to sets $\{w_s \mid a \text{ exists in } w\}$.

➥ Sentences X are true at @ iff $@ \in [\![X_{(s;t)}]\!]$.

➥ Names Y are true at @ iff $[\![Y_e]\!]$ exists in @.

➥ Names Y are equivalent to sentences X iff they are true/false at the same indices.

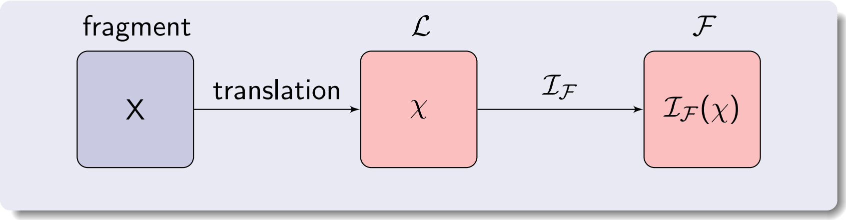
NB1 To use this strategy, we still need a multi-typed metatheory with types e, s, t (e.g. TY_2).

NB2 To use this strategy, we interpret s, t in the partial logic TY_2^3.

From PTQ to an $(s;t)$-Based Single-Type Semantics

Indirect interpretation We interpret the PTQ-fragment via its translation into the language of a single-type logic:

1. Develop the lang. $\mathcal{L}$ and models $\langle \mathcal{F}, \mathcal{I}_{\mathcal{F}} \rangle$ of the logic STY^3_1.
2. Provide a set of translation rules from expressions X of the PTQ-fragment to terms χ of the logic.



The Single-Type Logic STY_1^3

STY_1^3 is a subsystem of TY_2^3 that only has one basic type, $(s;t)$.

On being 'basic'

The type $(s;t)$ is a basic STY_1^3 type, because the TY_2^3 types s and t disqualify as STY_1^3 types.

➥ The type $(s;t)$ cannot be obtained from lower-rank types through the usual type-forming rules.

Definition (STY_1^3 types)

The smallest set of strings 1Type s.t., if $\alpha_1, \ldots, \alpha_n \in$ 1Type, then $(\alpha_1 \ldots \alpha_n\, s;t) \in$ 1Type.

1Type $\ni \{\, (s;t)\, ,\, ((s;t)\, s;t)\, ,\, ((s;t)\,(s;t)\, s;t)\, ,\, (((s;t)\, s;t)\, s;t)\, \}$

Terms

Basic STY_1^3 terms

- A set, $L := \cup_{\alpha \in 1\mathrm{Type}} \cup \{ ⓐ, \circledast, \dot{\Rightarrow} \}$, of non-log. constants

  $\textcircled{\perp}, \circledast, \dot{\Rightarrow} :=$ STY_1^3 stand-ins for $\perp, *, \Rightarrow$

- A set, $\mathcal{V}$, of variables.

Definition (STY_1^3 terms)

i. $L_\alpha, \mathcal{V}_\alpha \subseteq T_\alpha$, $\quad \textcircled{\perp}, \circledast \in T_{(s;t)}$;

ii. If $\mathbf{A} \in T_{(\beta\alpha_1 \dots \alpha_n\, s;t)}$ and $\mathbf{B} \in T_\beta$, then $(\mathbf{A}(\mathbf{B})) \in T_{(\alpha_1 \dots \alpha_n\, s;t)}$;

iii. If $\mathbf{A} \in T_{(\alpha_1 \dots \alpha_n\, s;t)}$, $\mathbf{x} \in \mathcal{V}_\beta$, then $(\lambda \mathbf{x}.\, \mathbf{A}) \in T_{(\beta\alpha_1 \dots \alpha_n\, s;t)}$;

iv. If $\mathbf{A}, \mathbf{B} \in T_\alpha$, then $(\mathbf{A} \dot{\Rightarrow} \mathbf{B}) \in T_{(s;t)}$.

- We will enforce on $\textcircled{\perp}, \circledast, \dot{\Rightarrow}$ the behavior of $\perp, *, \Rightarrow$.
- Stand-ins for other constants are defined as in (Henkin, 1950):

Models and Truth

- Frames $F = \{D_{(\alpha_1 \dots \alpha_n\, s;t)}\}$ have the usual definitions, where
$$D_{(\alpha_1 \dots \alpha_n\, s;t)} \subseteq \left\{f \mid f : (D_{\alpha_1} \times \cdots \times D_{\alpha_n} \times D_s) \to \mathbf{3}\right\}$$

➡ We can evaluate the truth or falsity of basic STY_1^3 terms.

- Since $s, t \notin$ 1Type, this evaluation proceeds in models of TY_2^3:

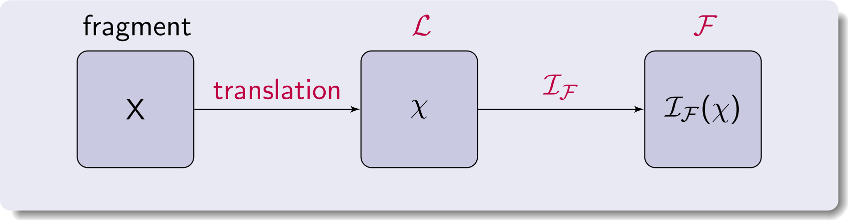
Definition (STY_1^3 truth)

- Let $M^2 = \langle F^2, I_{F^2}, V_{F^2}\rangle$ be an 'embedding' TY_2^3 model for
$M \;\; = \langle F^{2 \restriction 1\mathrm{Type}}, I_{F^{2 \restriction 1\mathrm{Type}}}, V_{F^{2 \restriction 1\mathrm{Type}}}\rangle$.

Then,
$$w_s \models_M \mathbf{A}_{(s;t)} \quad \text{iff} \quad w \models_{M^2} \mathbf{A} \quad \text{iff} \quad [\![\mathbf{A}]\!]^{M^2}(w) = \mathbf{T},$$
$$w_s \Dashv_M \mathbf{A}_{(s;t)} \quad \text{iff} \quad w \Dashv_{M^2} \mathbf{A} \quad \text{iff} \quad [\![\mathbf{A}]\!]^{M^2}(w) = \mathbf{F}.$$

- Since entailment is a relation of the type $(\alpha\, \alpha; t)$, it is also defined in the metatheory, TY_2^3.

STY$^3_1$-Based Single-Type Semantics



The Language $\mathcal{L}$

CONSTANT		STY$^3_1$ TYPE	
$\bigwedge, \bigvee$	$(\alpha\,(s;t)\,s;t)$,	$\wedge, \vee$	$((\vec{\alpha}\,s;t)\,(\vec{\alpha}\,s;t)\,\vec{\alpha}\,s;t)$
$\dot{\Rightarrow}, \dot{=}, \dot{\neq}, \dot{\rightarrow}, \dot{\leftrightarrow}$	$(\alpha\,\alpha\,s;t)$,	$\neg$	$((\vec{\alpha}\,s;t)\,\vec{\alpha}\,s;t)$
$\top$, $\bot$, **bill**, **pat**, **partee**, ...			$(s;t)$
$\boxdot$, $\diamond$, **woman**, **unic.**, **walk**, **wait**, **arrive**, **E**			$((s;t)\,s;t)$
find, **remember**, **believe**, ...			$((s;t)\,(s;t)\,s;t)$
seek, ...			$((((s;t)\,s;t)\,s;t)\,(s;t)\,s;t)$
for			$((s;t)\,((s;t)\,s;t)\,(s;t)\,s;t)$

VARIABLE		STY$^3_1$ TYPE	
$\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{y}, \mathbf{z}, \mathbf{u}, \mathbf{p}, \mathbf{q}, \mathbf{r}$			$(s;t)$
$\mathbf{P}, \mathbf{P}_1, \ldots, \mathbf{P}_n$	$((s;t)\,s;t)$,	$\mathbf{Q}$	$(((s;t)\,s;t)\,s;t)$
$\mathbf{R}, \mathbf{R}_1$	$(\alpha_1 \ldots \alpha_n\,s;t)$,	$\vec{\mathbf{X}}$	seq. $\alpha_1, \ldots, \alpha_n$

$\mathcal{I}_{\mathcal{F}} : \mathcal{L} \to \mathcal{F}$ respects the conventional relations bw. content words.

PTQ-to-STY$^3_1$ Translation

LFs are translated via type-driven translation (Klein and Sag, '85):

Definition (Basic STY$^3_1$ translations)

Bill	$\rightsquigarrow$	**bill**;	Pat	$\rightsquigarrow$	**pat**;
Partee	$\rightsquigarrow$	**partee**;	woman	$\rightsquigarrow$	**woman**;
unicorn	$\rightsquigarrow$	**unicorn**;	walks	$\rightsquigarrow$	**walk**;
waits	$\rightsquigarrow$	**wait**;	arrives	$\rightsquigarrow$	**arrive**;
exists	$\rightsquigarrow$	**E**;	finds	$\rightsquigarrow$	$\lambda \mathbf{Q}\, \lambda \mathbf{x}. \mathbf{Q}\, (\lambda \mathbf{y}.\, \mathbf{find}\, (\mathbf{y}, \mathbf{x}))$;
seeks	$\rightsquigarrow$	**seek**;	remembers	$\rightsquigarrow$	$\lambda \mathbf{Q}\, \lambda \mathbf{x}. \mathbf{Q}\, (\lambda \mathbf{y}.\, \mathbf{remember}\, (\mathbf{y}, \mathbf{x}))$;
is	$\rightsquigarrow$	$\lambda \mathbf{Q}\, \lambda \mathbf{x}. \mathbf{Q}\, (\lambda \mathbf{y}.\, \mathbf{x} \doteq \mathbf{y})$;	for	$\rightsquigarrow$	$\lambda \mathbf{Q}\, \lambda \mathbf{P} \lambda \mathbf{x}. \mathbf{Q}\, (\lambda \mathbf{y}.\, \mathbf{for}\, (\mathbf{y}, \mathbf{P}, \mathbf{x}))$;
that	$\rightsquigarrow$	$\lambda \mathbf{p}.\, \mathbf{p}$;	believes	$\rightsquigarrow$	$\lambda \mathbf{p}\, \lambda \mathbf{x}.\, \mathbf{believe}\, (\mathbf{p}, \mathbf{x})$;
necessarily	$\rightsquigarrow$	$\lambda \mathbf{p}.\, \boxdot\, \mathbf{p}$;	a/some	$\rightsquigarrow$	$\lambda \mathbf{P}_1 \lambda \mathbf{P} \bigvee \mathbf{x}.\, \mathbf{P}_1(\mathbf{x}) \wedge \mathbf{P}\, (\mathbf{x})$;
t_n	$\rightsquigarrow$	$\mathbf{x}_n$, for each n;	every	$\rightsquigarrow$	$\lambda \mathbf{P}_1 \lambda \mathbf{P} \bigwedge \mathbf{x}.\, \mathbf{P}_1(\mathbf{x}) \dot{\rightarrow} \mathbf{P}\, (\mathbf{x})$;
the	$\rightsquigarrow$	$\lambda \mathbf{P}_1 \lambda \mathbf{P} \bigvee \mathbf{x} \bigwedge \mathbf{y}.\, (\mathbf{P}_1(\mathbf{y}) \dot{\leftrightarrow} \mathbf{x} \doteq \mathbf{y}) \wedge \mathbf{P}\, (\mathbf{x})$;			
and	$\rightsquigarrow$	$\lambda \mathbf{R}_1 \lambda \mathbf{R}\, \lambda \vec{\mathbf{X}}. \mathbf{R}\, (\vec{\mathbf{X}}) \wedge \mathbf{R}_1(\vec{\mathbf{X}})$			

Derived Translations

STY_1^3 translates the example sentences from (Montague, 1973), (2):

1. $[_{\mathrm{NP}}\text{B. Partee}] \rightsquigarrow \textbf{partee}$
2. $[_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]] \rightsquigarrow \textbf{arrive}$
3. $[_{\mathrm{S}}[_{\mathrm{NP}}\text{B. Partee}][_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \textbf{arrive}\,(\textbf{partee})$

$$[_{\mathrm{S}}[_{\mathrm{NP}}[\text{a}][_{\mathrm{N}}\text{woman}]][[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \bigvee \mathbf{x}.\ \textbf{woman}\,(\mathbf{x}) \wedge \textbf{arrive}\,(\mathbf{x})$$

$$[_{\mathrm{S}}[[\text{every}][\text{woman}]][[\text{arrives}]]] \rightsquigarrow \bigwedge \mathbf{x}.\ \textbf{woman}\,(\mathbf{x}) \dot{\rightarrow} \textbf{arrive}\,(\mathbf{x})$$

$$[_{\mathrm{S}}[[\text{the}][\text{woman}]][[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \bigvee \mathbf{x} \bigwedge \mathbf{y}.(\textbf{woman}(\mathbf{y}) \dot{\leftrightarrow} \mathbf{x} \doteq \mathbf{y}) \wedge \textbf{arrive}(\mathbf{x})$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}\text{Bill}][[\text{finds}][[\text{a}][\text{unic.}]]]] \rightsquigarrow \bigvee \mathbf{x}.\ \textbf{unicorn}\,(\mathbf{x}) \wedge \textbf{find}\,(\mathbf{x}, \textbf{bill})$$

$$[_{\mathrm{S}}[_{\mathrm{NP}}\text{Bill}][[\text{seeks}][[\text{a}][\text{unic.}]]]] \rightsquigarrow \textbf{seek}\,([\lambda \mathbf{P} \bigvee \mathbf{x}.\textbf{unicorn}\,(\mathbf{x}) \wedge \mathbf{P}\,(\mathbf{x})], \textbf{bill})$$

$$[_{\mathrm{S}}[[\text{a}][\text{unic.}]]^0\ [_{\mathrm{S}}[\text{B.}][[\text{seeks}]\ t_0]]] \rightsquigarrow \bigvee \mathbf{x}.\textbf{unicorn}\,(\mathbf{x}) \wedge \textbf{seek}\,([\lambda \mathbf{P}.\mathbf{P}\,(\mathbf{x})], \textbf{bill})$$

Derived PTQ-Translations

STY_1^3 translates the STS-supporting sentences from (2.1)–(2.5):

NP/CP-neutrality of complements:

$[_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Pat}][_{\mathrm{VP}}[_{\mathrm{TV}}\mathrm{remembers}][_{\mathrm{NP}}\mathrm{Bill}]]] \rightsquigarrow \mathbf{remember}(\mathbf{bill}, \mathbf{pat})$

$[_{\mathrm{S}}[\mathrm{Pat}]^1\ [t_1\ [_{\mathrm{VP}}[\mathrm{remembers}][_{\mathrm{CP}}[\mathrm{that}][_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Bill}][_{\mathrm{VP}}[\mathrm{waits}][_{\mathrm{PP}}[\mathrm{for}][\mathrm{she}_1]]]]]]]]$

$\rightsquigarrow \mathbf{remember}(\mathbf{for}(\mathbf{pat}, \mathbf{wait}, \mathbf{bill}), \mathbf{pat})$

NP/CP-coordinability:

$[_{\mathrm{S}}[\mathrm{Pat}]^1\ [_{\mathrm{S}}\, t_1\ [_{\mathrm{VP}}[_{\mathrm{TV}}\mathrm{remembers}][_{\mathrm{NP}}\mathrm{Bill}]]]]$

$[[_{\mathrm{CONJ}}\mathrm{and}][_{\mathrm{CP}}[\mathrm{that}][_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Bill}][_{\mathrm{VP}}[\mathrm{waits}][_{\mathrm{PP}}[\mathrm{for}][\mathrm{she}_1]]]]]]]$

$\rightsquigarrow \mathbf{remember}((\mathbf{bill} \wedge \mathbf{for}(\mathbf{pat}, \mathbf{wait}, \mathbf{bill})), \mathbf{pat})$

CP-equatives:

$[_{\mathrm{S}}[_{\mathrm{NP}}[\mathrm{the}][_{\mathrm{N}}\mathrm{problem}]][_{\mathrm{VP}}[\mathrm{is}][_{\mathrm{CP}}[\mathrm{that}][_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Mary}][_{\mathrm{VP}}[\mathrm{hates}][_{\mathrm{NP}}\mathrm{Bill}]]]]]]$

$\rightsquigarrow \bigvee \mathbf{x} \bigwedge \mathbf{y}.(\mathbf{problem}(\mathbf{y}) \dot{\leftrightarrow} \mathbf{x} \doteq \mathbf{y}) \wedge \mathbf{x} \doteq \mathbf{hate}(\mathbf{bill}, \mathbf{mary})$

Derived PTQ-Translations

STY_1^3 translates the STS-supporting sentences from (2.1)–(2.5):

NP/CP-coordinability:

$[_{\mathrm{S}}[\mathrm{Pat}]^1\, [_{\mathrm{S}}\, t_1\, [_{\mathrm{VP}}[_{\mathrm{TV}}\mathrm{remembers}][_{\mathrm{NP}}\mathrm{Bill}]]]]$

$[[_{\mathrm{CONJ}}\mathrm{and}][_{\mathrm{CP}}[\mathrm{that}][_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Bill}][_{\mathrm{VP}}[\mathrm{waits}][_{\mathrm{PP}}[\mathrm{for}][\mathrm{she}_1]]]]]]]]$

$$\rightsquigarrow \mathbf{remember}\,((\mathbf{bill} \wedge \mathbf{for}\,(\mathbf{pat}, \mathbf{wait}, \mathbf{bill})), \mathbf{pat})$$

CP-equatives:

$[_{\mathrm{S}}[_{\mathrm{NP}}[\mathrm{the}][_{\mathrm{N}}\mathrm{problem}]][_{\mathrm{VP}}[\mathrm{is}][_{\mathrm{CP}}[\mathrm{that}][_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Mary}][_{\mathrm{VP}}[\mathrm{hates}][_{\mathrm{NP}}\mathrm{Bill}]]]]]]]$

$$\rightsquigarrow \bigvee \mathbf{x} \bigwedge \mathbf{y}.(\mathbf{problem}\,(\mathbf{y}) \dot{\leftrightarrow} \mathbf{x} \doteq \mathbf{y}) \wedge \mathbf{x} \doteq \mathbf{hate}\,(\mathbf{bill}, \mathbf{mary})$$

NP/S-equivalence: Is $\mathcal{I}_{\mathcal{F}}(\mathbf{partee}) = \mathcal{I}_{\mathcal{F}}(\mathbf{arrive}\,(\mathbf{partee}))$?

Challenge Ensure that STY_1^3 interpretations respect our strategy.
Ensure that Partee arrives denotes $\{w_s \mid [\![arrive]\!]([\![partee]\!], w)\}$.

The Language $\mathcal{L}^2$

To constrain the interpretations of STY_1^3 translations, we specify

CONSTANT	TY_2^3 TYPE
bill, *pat*, *partee*, ...	e
woman, *unicorn*, *walk*, *wait*, *arrive*, *E*, ...	$(e\,s;t)$
find, *remember*, ...	$(e\,e\,s;t)$
believe, ...	$((s;t)\,e\,s;t)$
seek, ...	$(((e\,s;t)\,s;t)\,e\,s;t)$
for	$(e\,(e\,s;t)\,e\,s;t)$

$\mathcal{L} \subseteq \mathcal{L}^2$

VARIABLE	TY_2^3 TYPE				
i, j, k	s	x, y, z	e		
p, q, r	$(s;t)$	P, P_1	$(e\,s;t)$	Q	$((e\,s;t)\,s;t)$

$\mathcal{V} \subseteq \mathcal{V}^2$

We define $\mathcal{F} = \mathcal{F}^2{\restriction}_{1\mathrm{Type}}$ and $\mathcal{I}_\mathcal{F} = \mathcal{I}_{\mathcal{F}^2}{\restriction}_{1\mathrm{Type}}$

Constraints on Basic PTQ-Interpretations

Semantic constraints specify, for every STY_1^3 term, which element in the 'embedding' TY_2^3 model it designates.

Definition (Constraints for $\mathcal{L}$-constants)

$$
\begin{array}{llcl}
\text{C1.} & \circledperp & = & \lambda i.\, \perp, \qquad \circledast \;=\; \lambda i.\, *; \\
\text{C2.} & (\mathbf{A} \dot{\Rightarrow} \mathbf{B}) & = & \lambda i.\, \mathbf{A}(i) \Rightarrow \mathbf{B}(i); \\
\text{C3.} & \mathbf{partee} & = & \lambda i.\, E(\mathit{partee}, i); \\
\text{C4.} & \mathbf{arrive} & = & \lambda \mathbf{x}\, \lambda i.\, \mathit{arrive}\big([\iota x.\, \mathbf{x} = (\lambda j.\, E(x, j))], i\big), \\
 & \mathbf{E} & = & \lambda \mathbf{x}\, \lambda i.\, E\big([\iota x.\, \mathbf{x} = (\lambda j.\, E(x, j))], i\big); \\
\text{C5.} & \mathbf{woman} & = & \lambda \mathbf{x}\, \lambda i.\, \mathit{woman}\big([\iota x.\, \mathbf{x} = (\lambda j.\, E(x, j))], i\big); \\
\text{C6.} & \mathbf{believe} & = & \lambda \mathbf{p} \lambda \mathbf{x}\, \lambda i.\, \mathit{believe}\big(\mathbf{p}, [\iota x.\, \mathbf{x} = (\lambda j.\, E(x, j))], i\big); \\
\text{C7.} & \mathbf{remember} & = & \lambda \mathbf{y}\, \lambda \mathbf{x}\, \lambda i.\, \mathit{remember}\big([\iota y.\, \mathbf{y} = (\lambda j.\, E(y, j))], \\
\text{C8.} & \ldots & = & \ldots \qquad [\iota x.\, \mathbf{x} = (\lambda k.\, E(x, k))], i\big);
\end{array}
$$

Defining the Translation of 'Partee arrives'

1. $[_{\mathrm{NP}}\text{B. Partee}] \rightsquigarrow \textbf{partee} = \lambda i.\, E(\mathit{partee}, i)$
2. $[_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]] \rightsquigarrow \textbf{arrive} = \lambda \mathbf{x}\, \lambda i.\, \mathit{arrive}\big([\iota x.\, \mathbf{x} = (\lambda j.\, E(x, j))], i\big)$
3. $[_{\mathrm{S}}[_{\mathrm{NP}}\text{B. Partee}][_{\mathrm{VP}}[_{\mathrm{IV}}\text{arrives}]]] \rightsquigarrow \textbf{arrive}\,(\textbf{partee})$

$$= \lambda \mathbf{x}\, \lambda i.\, \mathit{arrive}\big([\iota x.\, \mathbf{x} = (\lambda j.\, E(x, j))], i\big)\,\big[\lambda k.\, E(\mathit{partee}, k)\big]$$
$$= \lambda i.\, \mathit{arrive}\Big(\big[\iota x.\, [\lambda k.\, E(\mathit{partee}, k)] = (\lambda j.\, E(x, j))\big], i\Big)$$
$$= \lambda i.\, \mathit{arrive}\big([\iota x.\, \mathit{partee} = x], i\big)$$
$$= \lambda i.\, \mathit{arrive}\,(\mathit{partee}, i)$$

➥ The STY_1^3 interpretation of Partee arrives is the interpretation from (Montague, 1973), cf. (Gallin, 1975).

➥ The STY_1^3 interpret'n of Partee arrives respects our strategy.

Defining the Translations of (2.1)–(2.5)

TY_2^3 defines the translations of STS-supporting sentences:

$$[_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Pat}][_{\mathrm{VP}}[_{\mathrm{TV}}\mathrm{remembers}][_{\mathrm{NP}}\mathrm{Bill}]]] \rightsquigarrow \mathbf{remember}(\mathbf{bill}, \mathbf{pat}) = \lambda i.\, remember(bill, pat, i)$$

$$[_{\mathrm{S}}[\mathrm{Pat}]^1\, [t_1\, [_{\mathrm{VP}}[\mathrm{remembers}][_{\mathrm{CP}}[\mathrm{that}][_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Bill}][_{\mathrm{VP}}[\mathrm{waits}][_{\mathrm{PP}}[\mathrm{for}][\mathrm{she}_1]]]]]]]]$$
$$\rightsquigarrow \mathbf{remember}(\mathbf{for}(\mathbf{pat}, \mathbf{wait}, \mathbf{bill}), \mathbf{pat})$$
$$= \lambda i.\, remember([\iota y.[\lambda k.\, for(pat, wait, bill, k)] = (\lambda j.\, E(y, j))], pat, i)$$

To enable this definition, we require that TY_2^3 models contain type-e correlates of propositions:

$$[_{\mathrm{S}}[_{\mathrm{NP}}[\mathrm{the}][_{\mathrm{N}}\mathrm{problem}]][_{\mathrm{VP}}[\mathrm{is}][_{\mathrm{CP}}[\mathrm{that}][_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Mary}][_{\mathrm{VP}}[\mathrm{hates}][_{\mathrm{NP}}\mathrm{Bill}]]]]]]$$
$$\rightsquigarrow \bigvee \mathbf{x} \bigwedge \mathbf{y}.(\mathbf{problem}(\mathbf{y}) \dot{\leftrightarrow} \mathbf{x} \doteq \mathbf{y}) \wedge \mathbf{x} \doteq \mathbf{hate}(\mathbf{bill}, \mathbf{mary})$$
$$= \lambda i \exists x \forall y.(problem(y, i) \leftrightarrow x = y) \wedge x = [\iota z.(\lambda k.\, hate(bill, mary, k)) = (\lambda j.\, E(z, j))]$$

Single-Type Truth

- M^2 abbreviates $M_{\mathcal{F}^2}$, M abbreviates $M_{\mathcal{F}}$.
- g^2 and $g = g^{2\restriction 1\text{Type}}$ are the assignments of M^2, resp. M.

Definition (STY_1^3-based truth)

Let $X \rightsquigarrow \mathbf{A}_{(s;t)}$. Then, X is true (or false) at w in M^2 under g^2, i.e. $\text{TRUE}_{M,w}(X)$ (resp. $\text{FALSE}_{M,w}(X)$), iff $w \models_{M^2} \mathbf{A}$ ($w \Dashv_{M^2} \mathbf{A}$).

- STY_1^3-STS gives standard truth-conditions for sentences:

$$\text{TRUE}_{M,@}\left([_{\text{S}}[_{\text{NP}}\text{Partee}][_{\text{VP}}[_{\text{IV}}\text{arrives}]]]\right)$$
$$\text{iff} \quad @ \models_M \mathbf{arrive}\,(\mathbf{partee}) \quad \text{iff} \quad @ \models_{M^2} \lambda i.\, arrive\,(partee, i)$$

$$\text{FALSE}_{M,@}\left([_{\text{S}}[_{\text{NP}}\text{Partee}][_{\text{VP}}[_{\text{IV}}\text{arrives}]]]\right)$$
$$\text{iff} \quad @ \Dashv_M \mathbf{arrive}\,(\mathbf{partee}) \quad \text{iff} \quad @ \Dashv_{M^2} \lambda i.\, arrive\,(partee, i)$$
$$\text{iff} \quad @ \models_M \neg\,\mathbf{arrive}\,(\mathbf{partee}) \quad \text{iff} \quad @ \models_{M^2} \lambda i.\, \neg arrive\,(partee, i)$$

Single-Type Truth for Names

- STY_1^3-STS gives truth-conditions for proper names:

$$\mathrm{TRUE}_{M,@}\big([_{\mathrm{NP}}\mathrm{Partee}]\big)$$
$$\text{iff} \quad @ \models_M \mathbf{partee} \quad \text{iff} \quad @ \models_{M^2} \lambda i.\, E(partee, i)$$

$$\mathrm{FALSE}_{M,@}\big([_{\mathrm{NP}}\mathrm{Partee}]\big)$$
$$\text{iff} \quad @ \not\models_M \mathbf{partee} \quad \text{iff} \quad @ \not\models_{M^2} \lambda i.\, E(partee, i)$$

- Since exists $\rightsquigarrow \mathbf{E} = \lambda \mathbf{x}\, \lambda i.\, E\big([\iota x.\, \mathbf{x} = (\lambda j.\, E(x, j))], i\big)$, names have the truth-conditions of existential sentences:

$$\mathrm{TRUE}_{M,@}\big([_{\mathrm{NP}}\mathrm{Partee}]\big) \quad \text{iff} \quad \mathrm{TRUE}_{M,@}\big([_{\mathrm{S}}[_{\mathrm{NP}}\mathrm{Partee}][_{\mathrm{VP}}[_{\mathrm{IV}}\mathrm{exists}]]]\big)$$
$$\text{iff} \quad @ \models_M \mathbf{partee} \quad \text{iff} \quad @ \models_M \mathbf{E}(\mathbf{partee}) \quad \text{iff} \quad @ \models_{M^2} \lambda i.\, E(partee, i)$$

➥ STY_1^3-STS identifies equivalence relations between names and simple existential sentences.

Single-Type Equivalents of Names

- STY_1^3-STS identifies equivalence relations between names and simple existential sentences:

$$\begin{array}{ll} & \textsc{means}_M\big([_{\text{NP}}\text{Partee}],\ [_{\text{S}}[_{\text{NP}}\text{Partee}][_{\text{VP}}[_{\text{IV}}\text{exists}]]]\big) \\ \text{iff} & \models_g \mathbf{partee} = \mathbf{E}\,(\mathbf{partee}) \\ \text{iff} & \models_g (\lambda i.\, E\,(partee, i)) = (\lambda i.\, E\,(partee, i)) \end{array}$$

- STY_1^3-STS does NOT identify equivalence relations between names and contextually salient sentences:

$$\begin{array}{ll} & \text{NOT } \textsc{means}_M\big([_{\text{NP}}\text{Partee}],\ [_{\text{S}}[_{\text{NP}}\text{Partee}][_{\text{VP}}[_{\text{IV}}\text{arrives}]]]\big) \\ \text{b/c} & \models_g \mathbf{partee} \neq \mathbf{arrive}\,(\mathbf{partee}) \\ \text{b/c} & \models_g (\lambda i.\, E\,(partee, i)) \neq (\lambda i.\, arrive\,(partee, i)) \\ \text{b/c} & \exists i.\, E\,(partee, i) = \top \wedge arrive\,(partee, i) = * \end{array}$$

- STY_1^3-based semantics is a WEAK single-type semantics.

Strong Single-Type Semantics

A single-type semantics which identifies equivalence relations between names and contextually salient sentences.

➥ a model of an $(\mathbf{s}; (s; t))$ (or $(\mathbf{s}\, s; t)$)-based subsystem of TY_2.

The type $(s\, s; t)$ shares the semantic properties of $(s; t)$:

Representability The type $(\mathbf{s}\, s; t)$ enables an injective fct. f from individuals/propositions to propositional concepts:

- f sends individuals a to sets
  $\{\langle w_1, w\rangle \mid \text{for all } p_{\langle s,t\rangle}, {}^{\star}\text{if } w_1 \in p \text{ and } p \text{ is about } a, {}^{\star}\text{then } w \in p\}$
- f sends propositions φ to sets
  $\{\langle w_1, w\rangle \mid w \in \varphi \text{ and, for all } p_{\langle s,t\rangle}, {}^{\star}\text{if } w_1 \in p \text{ and,}$
  $\text{for some } x, \varphi \text{ is about } x \text{ and } p \text{ is about } x, {}^{\star}\text{then } w \in p\}$

Merits of Single-Type Semantics

Empirical merits: STS extends the modeling scope of Montague S:

- STS accommodates NP/CP complement-neutral verbs.
- STS accommodates NP/CP-coordinations.
- STS accommodates CP-equatives.
- STS accommodates the truth-evaluability of names.
- Strong STS accommodates the attested equivalence relations bw. names and CPs.

Methodological merits: STS unifies Montague's semantic ontology:

- STS identifies new representational relations between different types of objects:

Weak STS	bw. individuals and propositions;
Strong STS	bw. individuals and propositional concepts,
	bw. propositions and propositional concepts

Limitations of Single-Type Semantics

Empirical limitations: STS **reduces** the explanatory scope of Montague semantics:

- STS dissolves the semantic basis for many synt. categories.
- STS cannot explain the ill-formedness of
  - Possibly $[_{\mathrm{NP}}$Partee$]$.
  - $[_{\mathrm{S}}$Partee arrives$]$ exists.
  - Bill eats $[_{\mathrm{CP}}$that a unicorn exists$]$.

Method. 'limitations': Support for Montague's original type system

- STS requires a multi-typed metatheory with types e, s, t.

➥ STS still relies on Montague types.

Future Work

- Generalize STS to explain the type-$(s;t)$ interpretation of other non-quantificational NPs.

Context: Someone is pointing towards an empty chair

Interpret $[_{\mathrm{NP}}$an editor of *Natural Language Semantics*$]$ as $[_{\mathrm{S}}$This seat is reserved for an editor of *NL Sem.*$]$

➥ Strategy: Use ST-proxies of the iota and (Skolemized) choice operators.

- Extend STS to larger fragments (with events, states, degrees).
- Investigate untyped single-'type' semantics.
- Identify the (meta-)properties of minimal formal semantics for certain fragments.

Thank you!

Bayer, Samuel. 1996. *The coordination of unlike categories*, Language **72/3**, 579–616.

Beeson, Michael. 2005. *Lambda Logic*, LPAR-05 Workshop: Empirically Successful Automated Reasoning in Higher-Order Logic (ESHOL) (Christoph Benzmüller, John Harrison, and Carsten Schürmann, eds.), 2005.

Carstairs-McCarthy, A. 1999. *The Origins of Complex Language*: *An inquiry into the evolutionary beginnings of sentences, syllables, and truth*, OUP.

Church, A. 1985. *The Calculi of Lambda Conversion*, Princeton UP.

Gallin, D. 1975. *Intensional and Higher-Order Modal Logic with Applications to Montague Semantics*, North Holland, Amsterdam.

Henkin, L. 1950. *Completeness in the theory of types*, JSL **15**, 81–91.

______. 1963. *A theory of propositional types*, Fund. Math. **52**, 323–344.

Heycock, Caroline and Anthony Kroch. 1999. *Pseudocleft connectedness*: *Implications for the LF interface level*, Linguistic Inquiry **30**, 365–397.

Kim, J.-B. and I. Sag. 2005. *It-extraposition in English*: *A constraint-based approach*, HPSG'2005, Universidade de Lisboa, 2005.

Merchant, J. 2008. *Three types of ellipsis*, Context-Dependence, Perspective, and Relativity (F. Recanati at al., ed.), de Gruyter, Berlin, 2010.

Montague, R. 1970. *English as a formal language*, Formal Philosophy.

———. 1973. *The proper treatment of quantification in ordinary English*, Formal Philosophy, Yale UP, 1976.

Partee, B. 2006. *Do we need two basic types?*, Snippets, Vol. 20, Berlin, 2009.

Potts, Christopher. 2002. *The lexical semantics of parenthetical -as and appositive -which*, Syntax **5/1**, 55–88.