Semantic Typology for Human i-Languages

Paul M. Pietroski, University of Maryland
Dept. of Linguistics, Dept. of Philosophy
Human Languages

- acquirable by normal human children given ordinary courses of experience

- pair unboundedly many “meanings” with unboundedly many pronunciations

- how many _types_ of meanings?

- one answer, via Frege on _ideal_ languages:
  
  - `<e>` entity-denoters if `<α>` and `<β>` are types,
  - `<t>` truth-evaluable sentences then so is `<α, β>`
Human Languages

• acquirable by normal human children given ordinary courses of experience

• pair unboundedly many “meanings” with unboundedly many pronunciations

• how many types of meanings?

• one answer, via Frege on ideal languages:
  <e> entity-denoters
  <t> truth-evaluable sentences

\[
\text{SADIE}_e \quad \text{IS-A-HORSE}(_e)_{e, t}
\]
Human Languages

- acquirable by normal human children given ordinary courses of experience

- pair unboundedly many “meanings” with unboundedly many pronunciations

- how many *types* of meanings?
- one answer, via Frege on *ideal* languages:
  - `<e>` entity-denoters
  - `<t>` truth-evaluable sentences
Barbara Partee (2006), “Do We Need Two Basic Types?”

“...Carstairs-McCarthy argues that the apparently universal distinction in human languages between sentences and noun phrases cannot be assumed to be inevitable....His work suggests...that there is also no conceptual necessity for the distinction between basic types e and t.... If I am asked why we take e and t as the two basic semantic types, I am ready to acknowledge that it is in part because of tradition, and in part because doing so has worked well....”

Some of Partee’s Suggested Ingredients for an Alternative

• eventish semantics for VPs: *barked* ➞ Bark(e, e’) & Past(e)  
  *chase* ➞ Chase(e, e’, e’’)
• Heim/Kamp for indefinite NPs: *a dog* ➞ Dog(e)  
• entity/event neutrality, and  
  maybe predicate/sentence neutrality (cp. Tarski)
Human Languages

• acquirable by normal human children given ordinary courses of experience

• pair unboundedly many “meanings” with unboundedly many pronunciations

• how many types of meanings (basic or not)?

• one answer, via Frege on ideal languages:
  <e> entity-denoters if <α> and <β> are types,
  <t> truth-evaluable sentences then so is <α, β>
if $<\alpha>$ and $<\beta>$, then $<\alpha, \beta>$

that’s a lot of types

0. $<e> <t>$

1. $<e, e> <e, t> <t, e> <t, t>$ (4) of $<0, 0>$

2. eight of $<0, 1>$
   sixteen of $<1, 1>$

   eight of $<1, 0>$ (32), including $<e, et>$ and $<et, t>$

3. 64 of $<0, 2>$
   128 of $<1, 2>$
   1024 of $<2, 2>$

   64 of $<2, 0>$
   128 of $<2, 1>$
   (1408), including $<e, <e, et>>$ and $<et, <et, t>>$

4. 2816 of $<0, 3>$
   5632 of $<1, 3>$
   45,056 of $<2, 3>$
   1,982,464 of $<3, 3>$

   2816 of $<3, 0>$
   5632 of $<1, 3>$
   45,056 of $<3, 2>$
   (2,089,472), including $<e, <e, <e, et>>$
possible languages

Fregean Languages with expression types: <e>, <t>, and if <α> and <β> are types, so is <α, β>

Human i-Languages

Level-n Fregean Languages with expression types: <e>, <t>, and the nonbasic types up to Level-n

Pseudo-Fregean Languages with expression types: <e>, <t>, and a few of the nonbasic types
Human Languages

• acquirable by normal human children given ordinary courses of experience

• pair unboundedly many “meanings” with unboundedly many pronunciations

• how many *types* of meanings (basic or not)?

• another answer:

  - <M> monadic predicates
  - <D> dyadic predicates

\[
\begin{align*}
\text{Horse} &\quad (_{<e, t>}) \\
\text{On} &\quad (_{<e, <e, t>})
\end{align*}
\]
We can imagine/invent a language that has...

(1) finitely many *atomic monadic* predicates: $M_1(\_)$ ... $M_k(\_)$

(2) finitely many *atomic dyadic* predicates: $D_1(\_, \_)$ ... $D_j(\_, \_)$

(3) boundlessly many *complex monadic* predicates

$$\text{Monad + Monad} \rightarrow \text{Monad} \quad \text{Dyad + Monad} \rightarrow \text{Monad}$$

$$\text{BROWN}(\_) + \text{HORSE}(\_) \rightarrow \text{BROWN}(\_)^{\text{HORSE}(\_)}$$

$$\text{FAST}(\_) + \text{BROWN}(\_)^{\text{HORSE}(\_)} \rightarrow \text{FAST}(\_)^{\text{BROWN}(\_)^{\text{HORSE}(\_)}}$$
We can imagine/invent a language that has...

(1) finitely many *atomic monadic* predicates: $M_1(_) \ldots M_k(_)$

(2) finitely many *atomic dyadic* predicates: $D_1(_, _) \ldots D_j(_, _)$

(3) boundlessly many *complex monadic* predicates

Monad + Monad $\rightarrow$ Monad  
Dyad + Monad $\rightarrow$ Monad

$\Phi(_) \wedge \Psi(_) \text{ applies to } e \text{ iff}$

$\Phi(_) \text{ applies to } e \text{ and}$

$\Psi(_) \text{ applies to } e$

$ON(_, _) + HORSE(_) \rightarrow \exists[ON(_, _) \wedge HORSE(_)]$
We can imagine/invent a language that has...

(1) finitely many *atomic monadic* predicates: $M_1(\_)$ ... $M_k(\_)$

(2) finitely many *atomic dyadic* predicates: $D_1(\_, \_)$ ... $D_j(\_, \_)$

(3) boundlessly many *complex monadic* predicates

Monad + Monad $\rightarrow$ Monad  
Dyad + Monad $\rightarrow$ Monad

$\Phi(\_)^\Psi(\_)$ applies to e iff $\Delta(\_, \_)^\Phi(\_)$ applies to e iff e bears $\Delta(\_, \_)$ to *something* that $\Phi(\_)$ applies to

$\Phi(\_)$ applies to e *and*  
$\Psi(\_)$ applies to e

$\exists [\text{ON}(\_, \_)^\text{HORSE}(\_)]$
Monad + Monad \(\rightarrow\) Monad

\[\Phi(_\_)^{\psi(_\_)}\] applies to e iff

\[\Phi(_\_)\] applies to e and

\[\psi(_\_)\] applies to e

Dyad + Monad \(\rightarrow\) Monad

\[\Delta(_\_, _\_)^{\Phi(_\_)}\] applies to e iff

e bears \[\Delta(_\_, _\_)\] to something

that \[\Phi(_\_)\] applies to

\[\text{FAST(\_\_)}^{\text{BROWN(\_\_)}}^{\text{HORSE(\_\_)}}\]

\[\text{FAST(e)} \& \text{BROWN(e)} \& \text{HORSE(e)}\]

\[\exists [\text{ON(\_\_, _\_)}^{\text{HORSE(\_\_)}}]\]

\[\exists e'[\text{ON(e, e')} \& \text{HORSE(e')}]\]

but ‘&’ and ‘\(\exists v[...]\)’ permit \textit{a lot} more than ‘^’ and ‘\(\exists\)’

\[\exists v[\text{BETWEEN(x, y, z)} \& \text{SOLD(z, w, v, x)}]\]

Triad & Tetrad \(\rightarrow\) Pentad

\[\exists v[\text{Pentad(...v...)}] \rightarrow\) Tetrada
Monad + Monad $\rightarrow$ Monad

$\Phi(\_)^{\Psi(\_)}$ applies to $e$ iff

$\Phi(\_)$ applies to $e$ \textit{and}

$\Psi(\_)$ applies to $e$

Dyad + Monad $\rightarrow$ Monad

$\Delta(\_,\_)^{\Phi(\_)}$ applies to $e$ iff

e bears $\Delta(\_,\_)$ to \textit{something}

that $\Phi(\_)$ applies to

$\exists[\text{ON}(\_,\_)^{\text{HORSE}(\_)}]$

$\exists[e'[\text{ON}(e, e') \& \text{HORSE}(e')]]$

$\exists[\text{AGENT}(\_,\_)^{\text{HORSE}(\_)^{\text{FAST}(\_)^{\text{RUN}(\_)}}}$

$\exists[e'[\text{AGENT}(e, e') \& \text{HORSE}(e')] \& \text{FAST}(e) \& \text{RUN}(e)]$

$\exists[\text{AGENT}(\_,\_)^{\text{HORSE}(\_)^{\text{FAST}(\_)^{\text{RUN}(\_)}}$

$\exists[e'[\text{AGENT}(e, e') \& \text{HORSE}(e') \& \text{FAST}(e')] \& \text{RUN}(e)]$

$\exists[\text{AGENT}(\_,\_)^{\text{HORSE}(\_)^{\text{FAST}(\_)^{\text{RUN}(\_)}}$

$\exists[e'[\text{AGENT}(e, e') \& \text{HORSE}(e') \& \text{FAST}(e')] \& \text{RUN}(e)]$
Monad + Monad $\Rightarrow$ Monad

$\Phi(_{\_})^{\Psi(_{\_})}$ applies to e iff
$\Phi(_{\_})$ applies to e and
$\Psi(_{\_})$ applies to e

Dyad + Monad $\Rightarrow$ Monad

$\Delta(_{\_, \_})^{\Phi(_{\_})}$ applies to e iff
e bears $\Delta(_{\_, \_})$ to something
that $\Phi(_{\_})$ applies to

PAST(_{\_})^{\SEE(_{\_})^{\exists[THEME(_{\_, \_})^{HORSE(_{\_})}]}}$

PAST(e) & SEE(e) & $\exists e'[THEME(e, e') & HORSE(e')]$

\[
\begin{align*}
\text{PAST(_{\_})^{\SEE(_{\_})^{\exists[THEME(_{\_, \_})^{HORSE(_{\_})}]}} & \\
\text{PAST(e) & SEE(e) & } & \exists e'[THEME(e, e') & HORSE(e')] \\
/ \backslash & \\
\text{saw} & \text{a horse}
\end{align*}
\]

\[
\begin{align*}
\text{PAST(_{\_})^{\SEE(_{\_})^{\exists[THEME(_{\_, \_})^{HORSE(_{\_})}]}} & \\
\text{PAST(e) & SEE(e) & } & \exists e'[THEME(e, e') & HORSE(e')] \\
/ \backslash & \\
\text{saw} & \text{a horse}
\end{align*}
\]

\[
\begin{align*}
/ \backslash & \\
\text{run} & \\
/ \backslash & \\
\text{a horse}
\end{align*}
\]

\[
\begin{align*}
\text{PAST(e) & SEE(e) & } & \exists e'[THEME(e, e') & HORSE(e')] \\
\text{RUN(e') & } & \exists e''[AGENT(e', e'')^{HORSE(e'')}] \\
\end{align*}
\]
How many *types* of meanings in human languages?

How do the meanings *combine*?

\[
\begin{align*}
<t> & / \backslash <\beta> \\
/e & e, t \quad \alpha, \beta \quad / \backslash \\
<e> & e, t, e \quad <\alpha> \\
/e, e, t > & e \quad <\alpha> \\
/e, e, t, t > & e, t \\
& \ldots \quad \ldots
\end{align*}
\]

- function application
- *and*
- (type-shifting or)
- a rule for
- a rule for
- adjunction
- adjunction

\[
\begin{align*}
<M> & / \backslash <M> \\
/M & <M> \quad D \quad <M> \\
<e, t > & BROWN(\_)^\text{HORSE}(\_) \\
\exists[\text{ON}(\_, \_)^\text{HORSE}(\_)]
\end{align*}
\]

brown horse
Human Languages

• *acquirable* by normal human children given ordinary courses of experience

• pair unboundedly many “meanings” with unboundedly many pronunciations in accord with language-specific *constraints*

Bingley was ready _ to please _.  
Georgiana was eager _ to please _.  
Darcy was easy _ to please _.  

ambiguous  
unambiguous  
unambiguous  
(the other way)
Human Languages

- *acquirable* by normal human children given ordinary courses of experience

- pair unboundedly many “meanings” with unboundedly many pronunciations in accord with language-specific *constraints*

hiker lost kept walking circles
Human Languages

- **acquirable** by normal human children given ordinary courses of experience

- pair unboundedly many “meanings” with unboundedly many pronunciations in accord with language-specific *constraints*

The hiker who was lost kept walking in circles.

The hiker who lost was kept walking in circles.

Was the hiker who lost kept walking in circles? (meaning 2)
Human Languages

- acquirable by children
- unbounded but constrained
  
  and so presumably

- i-Languages in Chomsky’s (Church-inspired) sense

  function-in-intension vs. function-in-extension

  -- a *procedure* that pairs inputs with outputs in a certain way

  -- a *set* of ordered pairs (with no \(<x,y>\) and \(<x,z>\) where \(y \neq z\)
Human Languages

- acquirable by children
- unbounded but constrained

and so presumably

- i-Languages in Chomsky’s (Church-inspired) sense

function-in-intension vs. function-in-extension

\[ |x - 1| \quad +V(x^2 - 2x + 1) \]

\{..., (-2, 3), (-1, 2), (0, 1), (1, 0), (2, 1), ...\}

\[ \lambda x . |x - 1| \neq \lambda x . +V(x^2 - 2x + 1) \]

\[ \lambda x . |x - 1| = \lambda x . +V(x^2 - 2x + 1) \]

\[ \text{Extension}[\lambda x . |x - 1|] = \text{Extension}[\lambda x . +V(x^2 - 2x + 1)] \]
Human Languages

• acquirable by children
• unbounded but constrained
• biologically implemented procedures that pair “meanings” with sounds/gestures in a human way

• how many types of meanings?

• one hypothesis, via Frege on ideal languages:
  
  <e> entity-denoters
  <t> truth-evaluable sentences

  if <α> and <β> are types, then so is <α, β>
Human Languages

• natural generative *procedures*

• how many *types* of meanings?

• one hypothesis, via Frege on *ideal* languages:
  
  \(<e> \quad <t> \quad \text{if } <\alpha> \text{ and } <\beta>, \text{ then } <\alpha, \beta>\)

• **THREE CONCERNS:** available evidence suggests that...
  
  the proposed generative principle overgenerates (massively)

  Human Languages don’t generate expressions of type *<e>*

  Human Languages don’t generate expressions of type *<t>*
<e>  <t>

if <α> and <β>, then <α, β>

• one worry...overgeneration

0. <e>  <t>  \( (2) \)

1. <e, e> <e, t> <t, e> <t, t>  \( (4) \)

2. eight of <0, 1> eight of <1, 0>
   sixteen of <1, 1>  \( (32) \)

3. 64 of <0, 2> 64 of <2, 0>
   128 of <1, 2> 128 of <2, 1>
   1024 of <2, 2>  \( (1408) \)

4. 2816 of <0, 3> 2816 of <3, 0>
   5632 of <1, 3> 5632 of <1, 3>
   45,056 of <2, 3> 45,056 of <3, 2>  \( (2,089,471) \)
   1,982,464 of <3, 3>

possible languages
Human Languages

• **natural generative procedures**

• how many *types* of meanings?

• one hypothesis, via Frege on *ideal* languages:

  \[
  \langle e \rangle \quad \langle t \rangle \quad \text{if } \langle \alpha \rangle \text{ and } \langle \beta \rangle, \text{ then } \langle \alpha, \beta \rangle
  \]

• **one worry...overgeneration**

\[
\langle e, e, e, e, e, t \rangle \quad \lambda v. \lambda w. \lambda z. \lambda y. \lambda x. \text{GRONK}(x, y, z, w, v)
\]

\[
\langle \langle \langle e, t \rangle, \langle \langle e, t \rangle, \langle e, t \rangle \rangle \rangle \quad \lambda z. \lambda y. \lambda x. \text{GLONK}(X, Y, Z)
\]

\[
\langle \langle e, t \rangle, \langle t, e \rangle \rangle \quad \forall x [X(x) \lor Y(x) \lor Z(x)]
\]

\[
\langle \langle e, t \rangle, e \rangle \quad \exists x [X(x)] \land \exists x [Y(x) \land Z(x)]
\]

\[
\langle e, e \rangle \quad ???
\]

\[
\langle e, e \rangle \quad ???
\]
Human Languages

• natural generative \textit{procedures}

• how many \textit{types} of meanings?

• one hypothesis, via Frege on \textit{ideal} languages:
  
  \[
  \langle e \rangle \quad \langle t \rangle \quad \text{if } \langle \alpha \rangle \text{ and } \langle \beta \rangle, \text{ then } \langle \alpha, \beta \rangle
  \]

• one worry...overgeneration
  
  \[
  \langle e, e, e, e, e, t \rangle \quad \lambda w. \lambda z. \lambda y. \lambda x. \lambda e. \text{SELL}(e, x, y, z, w)
  \]
  
  \[
  \langle e, e, e, e, t \rangle \quad \lambda z. \lambda y. \lambda x. \lambda e. \text{GIVE}(e, x, y, z)
  \]
  
  \[
  \langle e, e, e, t \rangle \quad \lambda y. \lambda x. \lambda e. \text{KICK}(e, x, y)
  \]

\textit{claim}: verbs don’t provide evidence for “supradyadical” types
a linguist sold a car to a friend for a dollar
a linguist sold a friend a car for a dollar
Why not just...

A triple-object construction

'a linguist sold a friend a car a dollar'

'\lambda z. \lambda y. \lambda w. \lambda x. x sold y to z for w'

\lambda z. \lambda y. \lambda w. \lambda x. \lambda e. e was a selling by x of y to z for w
a thief jimmied a lock with a knife
(x) (y) (z)
Why not instead…

\[
\text{‘jimmied’} \Rightarrow \lambda z. \lambda y. \lambda x. x \text{ jimmied } y \text{ with } z
\]
Why not…

'a rock betweens a lock a knife'

'betweens' → \( \lambda z. \lambda y. \lambda x. x \text{ is between } y \text{ and } z \)
a linguist (x) tossed kicked gave a coin (y)
a linguist kicked gave a coin to a friend
a linguist tossed a friend a coin

‘tossed’ \( \rightarrow \lambda z . \lambda y . \lambda x . x \text{ tossed } y \text{ to } z \)

‘kicked’ \( \rightarrow \lambda z . \lambda y . \lambda x . x \text{ kicked } y \text{ to } z \)

\( \lambda z . \lambda y . \lambda x . \lambda e . \)

e was a kicking by x of y to z
if Kratzer and others are on the right track...

'a coin was tossed (by a linguist)'

'tossed' $\Rightarrow \lambda y. \lambda e . e$ was a tossing of $y$

'kicked' $\Rightarrow \lambda y. \lambda e . e$ was a kicking of $y$

representing an Agent is as optional as representing a Recipient
if Kratzer and others are on the right track…

∃[AGENT(_, _)^LINGUIST(_)^PAST(_)^∃[KICK(_, _)^COIN(_)]

or ... ∃[PAST(_)^KICK(_)^∃[PATIENT(_, _)^COIN(_)]
Chris hailed from Boston.

Why ...

and not ...

Chris frommed Boston

\( \lambda y . \lambda x . \lambda e . \)

e was a hailing by x from y
Chris hailed from Boston.

Chris was taller than Sam.

Why … and not…

Chris frommed Boston

Chris tallled frommed than outheighted Boston Sam

and not…
Human Languages

• natural generative *procedures*

• how many *types* of meanings?

• one hypothesis, via Frege on *ideal* languages:

  \[ <e> \quad <t> \quad \text{if } <\alpha> \text{ and } <\beta>, \text{ then } <\alpha, \beta> \]

• THREE CONCERNS: it seems that...

✔ the proposed generative principle overgenerates (massively)

  Human Languages don’t generate expressions of type \(<e>\)
  Human Languages don’t generate expressions of type \(<t>\)
Which expressions are of type <e>?

\[ \text{NAMES} \]

\[
\begin{align*}
\| \text{Robin}_{<e>} \| &= \quad \lambda P . P(\text{®}) \\
\| \text{Cruso}_{<e>} \| &= \quad \lambda P . P(\text{©}) \\
\| \text{Robin}_{<e, t>} \| &= \quad \lambda x . x = \text{®} \\
&\quad \lambda x . \text{Robin}(x) \\
\| \text{Cruso}_{<e, t>} \| &= \quad \lambda x . x = \text{©} \\
&\quad \lambda x . \text{Cruso}(x) \\
\| [D1_{<e, t} \text{Robin}_{<e, t}>]_{<e, t>} \| &= \quad \lambda x . \text{Indexes}(1, x) & \& \text{Called}(x, \text{‘Robin’})
\end{align*}
\]
Proper Nouns

• even *English* tells against the idea that *lexical proper nouns* are i-language expressions of type <e>

• Every Tyler I saw was a philosopher
  Every philosopher I saw was a Tyler
  That Tyler stayed late, and so did this one
  There were three Tylers at the party, and Tylers are clever
  The Tylers are coming to dinner
  (That nice) Professor Tyler Burge
  was sitting next to John Jacob Jingleheimer Schmidt

• proper *nouns* seem to be of type <M>, even if they are *related to* singular *concepts* of type <e>
every tiger
the Tyler
this

most tigers
some Tylers
those

that <e, t>
nice ?
Professor Burge <e>

<e, t>

that I saw Tyler(s)

<e, t>

Prof. <e, t>
<e, t> / \
Tyler Burge <e, t> <e, t>
Which expressions are of type <e>?

\[ V(P) \rightarrow \exists [AGENT(\_ , \_)^\text{THAT}(\_)^\text{1}(\_)^\text{WOMAN}(\_)]^\text{\ldots} \]

\[
\begin{array}{c}
D(P) \\
/ \ \\
\text{D} \ \\
/ \\
D(P)
\end{array} \quad \begin{array}{c}
V(P) \\
/ \\
\text{N} \\
/ \\
D(P)
\end{array}
\]

that \ \text{woman} \ \text{tossed} \quad / \ \text{1} \\
this \ \text{coin} \ \text{2}

if the nouns are of type <e, t>
then presumably, the indexed determiners are \textit{not} of type <e>;

if ‘that’ and ‘this’ are of type <e, t>
then presumably, the indices are \textit{not} of type <e>;
Which expressions are of type <e>?

\[ V(P) \implies \exists [AGENT(_, _)^{\text{\textregistered}}FEMALE(_)^{1(_)}]^\ldots \]

\[
\begin{array}{c}
D(P) \\
\downarrow \\
P \downarrow \\
\downarrow \downarrow \\
\text{she}_1 \quad \text{tossed} \\
\downarrow \\
\text{it}_2
\end{array}
\]

If ‘the pronouns ‘she’ and ‘it’ are of type <e, t>
then presumably, the indices are \textit{not} of type <e>

\[
\| \text{she}_{e, t} \| = \lambda x . \text{\textregistered} x \text{ is female} \\
\| 1_{e, t} \| = \lambda x . \text{Indexes}(1, x)
\]
Human Languages

• natural generative **procedures**

• how many **types** of meanings?

• one hypothesis, via Frege on **ideal** languages:
  
  $<e> <t>$  
  if $<\alpha>$ and $<\beta>$, then $<\alpha, \beta>$

• **THREE CONCERNS:** it seems that...

✔ the proposed generative principle overgenerates (massively)

✔ Human Languages don’t generate expressions of type $<e>$

  Human Languages don’t generate expressions of type $<t>$
Which expressions are of type <t>?

\[
\begin{align*}
\text{SENTENCES} & \quad S \rightarrow \text{NP aux VP} \\
D + N & \rightarrow D(P) \\
\text{every} + \text{tree} & \rightarrow \text{every} \text{ } D \text{ } \text{N} \\
? + ?? & \rightarrow S \\
\text{NP aux VP} & \\
V + D(P) & \rightarrow V(P) \\
\text{saw} + \text{every} + \text{tree} & \rightarrow \text{saw} \text{ } \text{D} \text{ } \text{N} \\
\end{align*}
\]
Which expressions are of type <t>?

\[ S = TP \]

\[ \text{T(P)} \]
\[ \text{past} \]
\[ \text{T} \]
\[ \text{V(P)} \]
\[ \rightarrow \]
\[ \lambda e . \text{T} \equiv e \text{ is (tenselessly) an event of John seeing Mary} \]

\[ \lambda y . \lambda x . \lambda e . \text{T} \equiv e \text{ is (tenselessly) an event of x seeing y} \]
Which expressions are of type <t>?

\[ S = TP \]

\[ \begin{array}{l}
T(P) \rightarrow \text{Why think TPs are of type <t> instead of <e, t> ?}
\end{array} \]

\[ \begin{array}{l}
T
\end{array} \]

\[ \begin{array}{l}
\text{past}
\end{array} \]

\[ \begin{array}{l}
V(P)
\end{array} \]

\[ \begin{array}{l}
\lambda e . T \equiv e \text{ is (tenselessly) an event of John seeing Mary}
\end{array} \]

\[ \begin{array}{l}
D(P)
\end{array} \]

\[ \begin{array}{l}
John
\end{array} \]

\[ \begin{array}{l}
\text{see}
\end{array} \]

\[ \begin{array}{l}
V
\end{array} \]

\[ \begin{array}{l}
D(P)
\end{array} \]

Is this expression of type <<e, t>, t> or type <e, t> ?
Which expressions are of type \(<t>\)?

\[
\begin{align*}
\text{SENTENCES} & \quad S = TP \\
T(P) & \quad \Rightarrow \text{Why think TPs are of type } \langle t \rangle \text{ instead of } \langle e, t \rangle \text{ ?} \\
\begin{array}{c}
\text{TP} \\
\downarrow \text{past} \\
\text{John} \\
\downarrow \text{see} \\
\langle e, t \rangle \\
\end{array}
\Rightarrow \quad \lambda e \cdot T \equiv e \text{ is (tenselessly) an event of John seeing Mary} \\
\begin{array}{c}
\text{D(P)} \\
\downarrow \\
\text{Mary} \\
\end{array}
\Rightarrow \quad \lambda E \cdot T \equiv \exists e [e \text{ is in the past } \& E(e) = T]
\end{align*}
\]
Which expressions are of type <t>?

SENTENCES \[ S = TP^+ \]

? \quad / \quad \setminus
\quad / \quad \setminus
\quad \exists \quad T(P) \quad \Rightarrow \quad \lambda e . T \equiv e \text{ is in the past} \quad \& \quad ...
\quad / \quad \setminus
\quad past \quad / \quad \setminus
\quad T \quad V(P) \quad \Rightarrow \quad \lambda e . T \equiv e \text{ is (tenselessly) an event of John seeing Mary}
\quad / \quad \setminus
\quad D(P) \quad V(P)
\quad John \quad / \quad \setminus
\quad V \quad D(P)
\quad see \quad Mary

<e, t> \quad \lambda e . T \equiv e \text{ is in the past}
Which expressions are of type <t>?

\[ \neg \]
\[ \exists \text{T(P)} \rightarrow \lambda e. \text{T} \equiv e \text{ is in the past} \]
\[ \text{past} \]
\[ \text{D(P)} \]
\[ \text{John} \]
\[ \lambda e. \text{T} \equiv e \text{ is (tenselessly) an event of John seeing Mary} \]

\[ \text{V(P)} \]
\[ \text{V} \]
\[ \text{see} \]
\[ \text{D(P)} \]
\[ \text{Mary} \]

\[ \langle e, t \rangle \]
\[ \lambda e. \text{T} \equiv e \text{ is in the past} \]
Which expressions are of type <t>?

\[ T(P) \rightarrow T \equiv \exists e \{ \text{e is in the past &} \ldots \} \]

\[ T \rightarrow \exists e \{ \text{e is in the past &} \ldots \} \]

\[ \text{past} \]

\[ V(P) \rightarrow \lambda e \cdot T \equiv e \text{ is (tenselessly) an event of John seeing Mary} \]

\[ D(P) \quad V(P) \]

\[ D(P) \quad V(P) \]

\[ V \quad D(P) \]

\[ \text{see} \quad \text{Mary} \]

\[ \langle \langle e, t \rangle, t \rangle \quad \lambda E \cdot T \equiv \exists e \{ \text{e is in the past &} E(e) = T \} \]

is \( T \) a quantificational argument of \( V \) and a conjunctive adjunct?
Which expressions are of type <t>?

SENTENCES  \[ S = \text{TP (or TP+)} \]

if \( T \) is (semantically) the verb’s 3\(^{rd}\) argument, then why not...

\[ V(P) \rightarrow T \equiv \text{That is (tenselessly) an event of J seeing M} \]

\[ D(P) \quad \backslash \]
\[ V(P) \quad \rightarrow \lambda e. T \equiv e \text{ is (tenselessly) an event of John seeing Mary} \]

Tense may be needed (in matrix clauses). But does it do two semantic jobs: adding time information via the ‘e’-variable, like the adjunct ‘yesterday’; and closing the ‘e’ variable, as if tense is the 3\(^{rd}\) argument of a verb that can’t take a 3\(^{rd}\) argument?
Which expressions are of type \(<t>\)?

SENTENCES $\quad S = TP$

\[
T(P) \rightarrow T \equiv \exists e [PAST(e) \& \ldots]
\]

\[
\begin{array}{c}
T \\
past
\end{array}
\]

\[
\begin{array}{c}
V(P) \\
D(P)
\end{array}
\]

John

\[
\begin{array}{c}
V \\
D(P)
\end{array}
\]

see Mary

$\llangle e, t, t \rangle \quad \lambda E . T \equiv \exists e [PAST(e) \& E(e) = T]$

$(e < RefTime) \& (RefTime = SpeechTime)$

“God likes Fregean Semantics” theory of tense
Which expressions are of type <t>?

Maybe None:

```
+Polarized ➔ _ is such that ∃[PastSeeingOfMaryByJohn(_)]
  
  T(P) ➔ PastSeeingOfMaryByJohn(_)
  /
  
  T     V(P)
  past  /
  
  D(P)   V(P)
  John  /
  
  V     D(P)
  see   Mary
```

a monadic predicate \( M \) can be “polarized” into
a predicate \( +M \) that applies to \( e \) iff \( M \) applies to some \( \textit{something} \)
or a predicate \( -M \) that applies to \( e \) iff \( M \) applies to \( \textit{nothing} \)
But what about Quantification?

Not at all clear that
the “external argument” of ‘every cow’ is—
or even can be—an expression of type <et>
Fido chased every cow today.

Bessie ran every cow today.
every cow ran today
Why not…

\[
\exists (i)t_1 \text{ ran}
\]

every cow which ran

day

\[
\text{very syncategorematic}
\]
Fido chased every cow today
We can discuss the difficulties for this kind of view in Q&A. But my point is not that an <et, t> analysis of quantifiers cannot be preserved. My point is that there is no argument here for the standard typology, especially given doubts about <e>. 

\[
\begin{array}{c}
\langle et \rangle \\
/ \ \\
\langle et \rangle \ \\
/ \ \\
\langle et, t \rangle \\
/ \ \\
\langle e, et \rangle \\
/ \ \\
\langle e, et \rangle \\
/ \ \\
\langle 1, 0 \rangle, \langle 0, 1 \rangle \\
/ \ \\
\langle 2, 2 \rangle \\
/ \ \\
\langle 3 \rangle \\
\end{array}
\]
Human Quantification: Still Puzzling

But maybe...

• ‘every’ is a **plural** Dyad:
  \[\text{EVERY}(_, _)\] applies to \(<\text{the Xs, the Ys}>\) iff the Xs *include* the Ys

• ‘every cow’ is a **plural** Monad:
  \[\exists [\text{EVERY}(_, _)^{\text{THE-COWS}(\_)}]\]
  applies to the Xs iff the Xs include *the* cows

• ‘every cow ran’ is a **plural** Monad:
  \[\exists [\text{EVERY}(_, _)^{\text{THE-COWS}(\_)}]^{\text{RAN}(\_)}\]
  applies to the Xs iff the Xs include the cows & the Xs ran
Human Quantification: Still Puzzling

But maybe...

• ‘most’ is a plural Dyad:
  MOST(_, _) applies to <the Xs, the Ys> iff
  the Ys that are Xs outnumber the Ys that are not Xs

• ‘most cows’ is a plural Monad:
  \( \exists [\text{MOST}(_, _) \text{^THE-COWS(_)}] \) applies to the Xs iff
  the cows that are Xs outnumber the cows that are not Xs

• ‘most cows ran’ is a plural Monad:
  \( \exists [\text{MOST}(_, _) \text{^THE-COWS(_)}] \text{^RAN(_)} \) applies to the Xs iff
  the cows that are Xs outnumber the other cows
Human Quantification: Still Puzzling

But maybe...

• ‘every’ is a plural Dyad:
  EVERY(_ , _) applies to <the Xs, the Ys> iff the Xs include the Ys

• ‘every cow’ is a plural Monad:
  \( \exists [\text{EVERY}(_ , _) \wedge \text{THE-COWS}(\_)] \)
  applies to the Xs iff the Xs include the cows

• ‘every cow ran’ is a plural Monad:
  \( \exists [\text{EVERY}(_ , _) \wedge \text{THE-COWS}(\_)] \wedge \text{RAN}(_) \)
  applies to the Xs iff the Xs include the cows & the Xs ran
for any assignment $A$ of values to variables...

applies to $e$ iff there was an event of $A_1$ running

applies to $e$ iff $e$ was an event of $A_1$ running
for any assignment $A$ of values to variables...

so if $e$ is the value of 1 (but $A$ is otherwise the same), then the “polarized” predicate applies to $e$ iff $e$ ran
for any assignment $A$ of values to variables...

and we can define a “Tarski Relation” such that

$\text{TARSKI}(e, \text{Polarity}[t_1 \text{ ran}], 1) \iff e \text{ ran}$
for any assignment $A$ of values to variables...

$\exists A^*: A^* \approx_1 A \{ \text{Satisfies}(A^*, \text{Polarity}[t_1 \text{ ran}]) \land (e = A^*[1]) \}$
we can define a “Tarski Relation” such that
\[ \text{TARSKI}(e, \text{Polarity}[t_1 \text{ ran}], 1) \text{ iff } e \text{ ran} \]
Human Quantification: Still Puzzling

But maybe...

• ‘every’ is a **plural** Dyad:
  \[
  \text{EVERY}(\_ , \_ ) \text{ applies to } \langle \text{the Xs, the Ys} \rangle \text{ iff}\n  \text{the Xs } \text{include} \text{ the Ys}
  \]

• ‘every cow’ is a **plural** Monad:
  \[
  \exists \left[ \text{EVERY}(\_ , \_ )^\text{THE-COWS(\_ )} \right]
  \text{ applies to the Xs iff the Xs include the cows}
  \]

• ‘every cow ran’ is a **plural** Monad:
  \[
  \exists \left[ \text{EVERY}(\_ , \_ )^\text{THE-COWS(\_ )} \right]^\text{RAN(\_ )}
  \text{ applies to the Xs iff the Xs include the cows & the Xs ran}
  \]
Lots of Further Issues

- Quantification Homework (including conservativity)
- Mary saw John, *and* John didn’t see Mary
- Lexicalization of singular and relational concepts
- Lexical *inflexibilities*
  *Chris sneezed the baby*
  *Chris put the letter*
- Gleitman-esque acquisition of verbs
- Your Objection Here
Human Languages

• *acquirable* by normal human children given ordinary courses of experience

• generatively pair meanings with gestures in accord with human *constraints*
Fregean Languages with expression types: $<e>, <t>$, and if $<\alpha>$ and $<\beta>$ are types, so is $<\alpha, \beta>$

Human i-Languages

Level-$n$ Fregean Languages with expression types: $<e>, <t>$, and the nonbasic types up to Level-$n$

Pseudo-Fregean Languages with expression types: $<e>, <t>$, and a few of the nonbasic types

possible languages
Semantic Typology for Human I-Languages

THANKS!
Human Quantification: Still Puzzling

But maybe...

• **DET( _, _ )** applies to <the Xs, the Ys> only if the Xs *are among* the Ys

• **EVERY( _, _ )** applies to <the Xs, the Ys> only if the Xs *are* the Ys

• **MOST( _, _ )** applies to <the Xs, the Ys> only if #(X) > #(Y) − #(X)
Monad + Monad $\Rightarrow$ Monad

$\Phi(\_)^\Psi(\_)$ applies to e iff

$\Phi(\_)$ applies to e and $\Psi(\_)$ applies to e

Dyad + Monad $\Rightarrow$ Monad

$\Delta(\_, \_)^\Phi(\_)$ applies to e iff e bears $\Delta(\_, \_)$ to something that $\Phi(\_)$ applies to

$\text{FAST}(\_)^\text{BROWN}(\_)^\text{HORSE}(\_)$

Fast(e) & Brown(e) & Horse(e)

$\text{HORSE}(e)$

$\exists s[\text{EXEMPLIFIES}(e, s) \& \text{HORSEY}(s)] \& \exists i[\text{AT}(s, i) \& \text{NOW}(i)]$
Which expressions are of type <t>?

Maybe None

```
         <t>
         /    \
     <t>       <t, t>
Mary saw John /    \
  <t, <t, t>> <t>
and       John saw Mary
```
Mary saw John

before

John saw Mary

\[\exists [\text{Before}(\_ , \_ )^{\text{JohnSeeMary} (\_ )}] \]
Mary saw John

\[\exists f [\text{Before}(e, f) \land f \text{ is an event of John seeing Mary}]\]

\[\Rightarrow e \text{ was an event of Mary seeing John} \land \ldots\]