The logical form and meaning of attitudinal sentences

Yiannis N. Moschovakis
University of California, Los Angeles

February 22, 2014, Workshop at UCI

Two Frege quotes on sense

★ "*The sense of a proper name is grasped by everyone who is sufficiently familiar with the language . . . Comprehensive knowledge of the thing denoted . . . we never attain*" Language speakers know the meanings but not always the denotations of terms

"*The same sense has different expressions in different languages or even in the same language*" "*The difference between a translation and the original text should properly not overstep the [level of the idea]*" Faithful translation should preserve meaning

★ *Faithful translation* $\sim$ *synonymy in the join of two languages*

My topic is the logic of meaning and synonymy in a (mostly) Fregean tradition.

It can be viewed as modifying and extending Montague semantics, most significantly by adding to it a robust notion of meaning

A logic of meaning and synonymy (simplified, all lies are white)

Language. The *typed λ-calculus with acyclic recursion* $\mathsf{L}^{\lambda}_{\mathsf{ar}}$,
an extension of the typed λ-calculus Ty_2 into which Richard Montague's *language of intensional logic* can be translated (Gallin)

Interpretation. In every suitable *type structure* $\mathfrak{M}$, each *closed term* A of $\mathsf{L}^{\lambda}_{\mathsf{ar}}$ is assigned

$$\boxed{\text{a value } \mathrm{den}^{\mathfrak{M}}(A) \quad \textbf{and} \quad \text{a referential intension } \mathrm{int}^{\mathfrak{M}}(A)}$$

$\mathrm{int}^{\mathfrak{M}}(A)$ models the meaning of A in $\mathfrak{M}$ and determines $\mathrm{den}^{\mathfrak{M}}(A)$

Will assume a "standard structure", our universe, and skip the $\mathfrak{M}$

Denotational equality: $\models A = B \iff \mathrm{den}(A) = \mathrm{den}(B)$

Synonymy: $A \approx B \iff \mathrm{int}(A) = \mathrm{int}(B)$

★ $\mathrm{int}(A)$ captures the logical meaning of A, *what the words say*

The key idea (simplified, all lies are white)

- *The sense of a term A is (faithfully represented by) an abstract procedure which computes its denotation*

Dummett, Tichy, ...and perhaps (implicity) Davidson.
Evans: "ideal verificationism" ... "scant evidence for it [in Frege]"

★ *The meaning* int(A) *of A is the algorithm which computes* den(A)

★ The relevant (abstract) algorithms are precisely defined and

- for every term A of $\mathsf{L}^{\lambda}_{\mathrm{ar}}$, int($A$) is an object of our universe, and
- the operation $A \mapsto \mathrm{int}(A)$ can be defined in $\mathsf{L}^{\lambda}_{\mathrm{ar}}$

★ This makes it possible to derive equivalences of the form

$$\textit{George believes that } A \iff \textit{George believes}^{*}\ \mathrm{int}(A)$$

where believes* is denotational. (No "higher order" senses)

★ The theory imports many more ideas from programming languages, (assignments, a robust state, ...)

Rendering (of fragments of natural language into $\mathsf{L}^{\lambda}_{\mathrm{ar}}$)

The logical analysis of a phrase from natural language starts by rendering (translating) it into the formal language $\mathsf{L}^{\lambda}_{\mathrm{ar}}$

every man loves some woman (λ-calculus)

$$\xrightarrow{\text{render}} \text{every}(\text{man})\Big[\lambda(u)\Big(\text{some}(\text{woman})(\lambda(v)\text{loves}(u,v))\Big)\Big]$$

coordination: (λ-calculus)

Abelard loved and honored Eloise

$$\xrightarrow{\text{render}} \lambda(u,v)\Big(\text{loved}(u,v) \text{ and } \text{honored}(u,v)\Big)(\text{Abelard}, \text{Eloise})$$

coindexing (anaphora): (new)

Abelard loved Eloise and (he) honored her

$$\xrightarrow{\text{render}} \text{loved}(\dot{a},\dot{e}) \text{ and } \text{honored}(\dot{a},\dot{e}) \ \underline{\text{where}}\ \{\dot{a} := \text{Abelard}, \dot{e} := \text{Eloise}\}$$

propositional attitudes: (different from Montague)

Abelard believed that Eloise loved him

$$\xrightarrow{\text{render}} \text{Believed}(\dot{a},\ \underline{\text{that}}\ \text{loved}(\text{Eloise}, \dot{a})) \text{ where } \{\dot{a} := \text{Abelard}\}$$

Some methodological points

- Compositionality Principle
- Logical form
- State

• Logic cannot solve philosophical or linguistic problems, and I will say nothing specific to belief, knowledge, etc., or the "rendering" process. What logic can do is to relate some philosophical and linguistic views to precise, technical, problems and help eliminate inconsistent or incoherent proposals

• Models or faithfully represents is the mathematical version of is

- $\{\{x\}, \{x, y\}\}$ models the ordered pair (x, y)
- $\lambda x\, a, \lambda x \lambda y\, a, \ldots$ can be used to model the object a

A model of a notion should "code" all its important properties and characterize it up to a natural relation of "isomorphism".

It can be viewed as a "weak explication" of the notion

Outline

Introduction (already done)

1. The syntax and denotational semantics of $\mathsf{L}^{\lambda}_{\mathsf{ar}}$, 8 pages
2. Referential intensions and synonymy, 3 pages
3. Attitudinal application, 6 pages
4. Uses of states, last page

References, posted on http://www.math.ucla.edu/∼ynm

Sense and denotation as algorithm and value (1994)
A logical calculus of meaning and synonymy (2006)
Two aspects of situated meaning, with E. Kalyvianaki (2008)
A logic of meaning and synonymy, with Fritz Hamm
(Lecture notes for an advanced course in ESSLLI 2010)

The λ-calculus with acyclic recursion $\mathsf{L}^{\lambda}_{\mathrm{ar}}$: types

Basic types Entities : e Truth values : t States : s

$$\boxed{\sigma :\equiv \mathsf{e} \mid \mathsf{t} \mid \mathsf{s} \mid (\sigma_1 \to \sigma_2)}$$

Interpretation (standard)

$$\mathbb{T}_{\mathsf{e}} = \text{a given set (or class) of people, objects, etc.}$$

$$\mathbb{T}_{\mathsf{s}} = \text{a given set of states}$$

$$\{0, 1, er\} \subseteq \mathbb{T}_{\mathsf{t}} = \text{a given set of truth values } \subseteq \mathbb{T}_{\mathsf{e}}$$

$$\boxed{\mathbb{T}_{(\sigma\to\tau)} = (\mathbb{T}_\sigma \to \mathbb{T}_\tau) = \text{the set of all functions } p : \mathbb{T}_\sigma \to \mathbb{T}_\tau}$$

State $a = (\mathrm{world}(a), \mathrm{time}(a), \mathrm{location}(a), \mathrm{agent\ (speaker)}(a), \delta)$

$$\delta(\mathrm{He}_1) = \ldots, \quad \delta(\mathrm{this}) = \ldots, \text{ etc.}$$

$$er = \text{ error} \quad \Big(\mathrm{den}(\text{the King of France is bald})(a) = er\Big)$$

$$\boxed{x : \sigma \iff x \in \mathbb{T}_\sigma \quad (x \text{ is an object of type } \sigma)}$$

Pure and state-dependended types and objects

Pure types $\boxed{\sigma :\equiv \mathsf{e} \mid \mathsf{t} \mid (\sigma_1 \to \sigma_2)}$ (for mathematical objects)

$$\tilde{\mathsf{t}} :\equiv (\mathsf{s} \to \mathsf{t}) \quad \text{(Carnap intensions)}$$
$$\tilde{\mathsf{e}} :\equiv (\mathsf{s} \to \mathsf{e}) \quad \text{(individual concepts)}$$

Natural language types $\boxed{\sigma :\equiv \tilde{\mathsf{e}} \mid \tilde{\mathsf{t}} \mid (\sigma_1 \to \sigma_2)}$

- ★ *The terms which render natural language phrases are (hereditarily) of natural language type*, but
- ★ *the Gallin translation of Intensional Logic is not into the natural language fragment of* $\mathsf{L}^{\lambda}_{\mathsf{ar}}$

State-dependent unary quantifier type

$$\mathsf{some(girl), every(boy)} : \tilde{\mathsf{q}} :\equiv ((\tilde{\mathsf{e}} \to \tilde{\mathsf{t}}) \to \tilde{\mathsf{t}})$$

Abbreviations $\sigma_1 \times \sigma_2 \to \tau :\equiv (\sigma_1 \to (\sigma_2 \to \tau))$

Constants; the lexicon

Denotational empirical constants:

Entities	0, 1, 2, . . . , *er* :	e
Names, indexicals	John, I, he, him:	$\tilde{e}$
Common nouns	man, unicorn, temperature:	$\tilde{e} \to \tilde{t}$
Adjectives, adverbs	tall, young, rapidly:	$(\tilde{e} \to \tilde{t}) \to (\tilde{e} \to \tilde{t})$
Propositions	it rains:	$\tilde{t}$
Intransitive verbs	stand, run, rise:	$\tilde{e} \to \tilde{t}$
Transitive verbs	find, love, be:	$(\tilde{e} \times \tilde{e}) \to \tilde{t}$
Description operator	the	$(\tilde{e} \to \tilde{t}) \to \tilde{e}$

★ **Attitudinal constants:** Believes, Knows, Claims : $(\tilde{e} \times \cdots \times \tilde{e} \times \tilde{t}) \to \tilde{t}$

Pure type logical constants:

$$=_\sigma \ : \ \sigma \times \sigma \to t$$
$$\neg \ : \ t \to t$$
$$\&, \vee, \Rightarrow \ : \ t \times t \to t$$
$$\forall_\sigma, \exists_\sigma \ : \ (\sigma \to t) \to t$$

Natural type logical constants

not, $\Box$, in the future : $\tilde{t} \to \tilde{t}$

and, or, if .. then .. : $\tilde{t} \times \tilde{t} \to \tilde{t}$

every, some : $(\tilde{e} \to \tilde{t}) \to \tilde{q}$

the : $(\tilde{e} \to \tilde{t}) \to \tilde{e}$

Typed variables

★ **Two kinds of (typed) variables**

- Pure variables of type σ: $v_0^\sigma, v_1^\sigma \dots$
- Recursion variables or locations of type σ: $\dot{v}_0^\sigma, \dot{v}_1^\sigma \dots$

- Both v_i^σ and $\dot{v}_i^\sigma$ are interpreted by arbitrary objects $x : \sigma$

  ... but they are treated differently in the syntax

★ Pure variables are bound by λ (as in the typed λ-calculus)

★ Locations are used to make (formal) assignments

$$\dot{p} := A$$

and are bound by the recursion construct $\boxed{\text{where}}$

★ *There are no variables over entities in the natural language fragment, variables over individual concepts are used instead*

Terms (defined recursively, with suitable restrictions)

$$A :\equiv x \mid \mathsf{c} \mid A_1(A_2) \mid \lambda(u)(A_1)$$
$$\mid A_0 \text{ where } \{\dot{p}_1 := A_1, \ldots, \dot{p}_n := A_n\}$$
$$\mid \mathsf{C}(A_1, \ldots, A_n, \text{that } A_0) \quad \text{(attitudinal application)}$$

- Four formations rules: application, λ-abstraction
  acyclic recursion (where) and attitudinal application (that)
- c is a denotational constant and C an attitudinal constant
- Each term is assigned a type, $A : \mathrm{type}(A)$
  - If $A_1, \ldots, A_n : \tilde{\mathrm{e}}$ and $A_0 : \tilde{\mathrm{t}}$, then $C(A_1, \ldots, A_n, \text{that } A_0) : \tilde{\mathrm{t}}$
- Free and bound occurrences of variables are specified
  (C is treated like c, for recursive terms on the next slide)
- $\mathrm{den}(A)(\mathrm{g})$ = the denotation of A for the valuation g
  (which assigns correctly typed values to the variables)

John loves Mary and dislikes her husband

$$\boxed{\begin{aligned} A \equiv \dot{p} \text{ and } \dot{q} \text{ where } \{&\dot{p} := \text{loves}(\dot{j}, \dot{m}), \dot{q} := \text{dislikes}(\dot{j}, \dot{h}), \\ &\dot{h} := \text{husband}(\dot{m}), \dot{j} := \text{John}, \dot{m} := \text{Mary}\} : \tilde{\text{t}} \end{aligned}}$$

Stage 1: $\bar{\jmath} :=$ John : $\tilde{\text{e}}$, $\overline{m} :=$ Mary : $\tilde{\text{e}}$
Stage 2: $\overline{h} :=$ husband$(\overline{m})$ = Mary's husband : $\tilde{\text{e}}$
$\overline{p} :=$ loves$(\bar{\jmath}, \overline{m})$: $\tilde{\text{t}}$
Stage 3: $\overline{q} :=$ dislikes$(\bar{\jmath}, \overline{h})$: $\tilde{\text{t}}$
Stage 4: den$(A) = \overline{p}$ and $\overline{q}$: $\tilde{\text{t}}$

For any state α,
den$(A)(\alpha) = (\overline{p}$ and $\overline{q})(\alpha) = \overline{p}(\alpha)$ and $\overline{q}(\alpha)$
= the truth value ofJohn loves Mary and dislikes her husband
in state α

(= *er* if Mary does not have exactly one husband in state α)

Acyclic recursion

$$\boxed{A \equiv A_0 \text{ where } \{\dot{p}_1 := A_1, \ldots, \dot{p}_n := A_n\}}$$

- The sequence of (correctly-typed) term assignments

$$\{\dot{p}_1 := A_1, \ldots, \dot{p}_n := A_n\} \quad (\text{type}(\dot{p}_i) = \text{type}(A_i))$$

to the distinct locations $\dot{p}_1, \ldots, \dot{p}_n$ is acyclic, i.e., there are numbers $\text{rank}(\dot{p}_1), \ldots, \text{rank}(\dot{p}_n)$, such that

$$\text{if } \dot{p}_j \text{ occurs free in } A_i, \text{ then } \text{rank}(\dot{p}_j) < \text{rank}(\dot{p}_i)$$

- $A : \text{type}(A_0)$
- All occurrences of $\dot{p}_1, \ldots, \dot{p}_n$ are bound in A
- $\text{den}(A)(\mathrm{g}) = \text{den}(A_0)(\mathrm{g}\{\dot{p}_1 := \overline{p}_1, \ldots, \dot{p}_n := \overline{p}_n\})$ where $\overline{p}_1, \ldots, \overline{p}_n$ are the unique solutions of the system

$$\overline{p}_i = \text{den}(A_i)(\mathrm{g}\{\dot{p}_1 := \overline{p}_1, \ldots, \dot{p}_n := \overline{p}_n\}) \quad (i = 1, \ldots, n)$$

Abbreviations, congruence, term replacement

Abbreviations and misspellings:

$$A(B)(C) \equiv A(B,C),$$

$$A[B(C,D)] \equiv A(B(C)(D)),$$

$$A \text{ where } \{\ \} \equiv A, etc.$$

Term Congruence: $A \equiv_c B$ is an equivalence relation on terms such that

- $A \equiv_c B$ if B is constructed from A by alphabetic changes of bound variables and
- $A \text{ where } \{\dot{p} := B, \dot{q} := C\} \equiv_c A \text{ where } \{\dot{q} := C, \dot{p} := B\}$

Free term replacement:

$$A\{x :\equiv B\} \equiv \text{the result of replacing every free occurrence of the variable } x \text{ in } A \text{ by the term } B$$

(used only if no free variable of B is bound in $A\{x :\equiv B\}$)

⋆ The Reduction Calculus

We define a reduction relation between terms so that intuitively

$$\boxed{\begin{aligned} A \Rightarrow B \iff\ & A \equiv_c B \quad (A \text{ is congruent with } B) \\ & \text{or } A \text{ and } B \text{ have the same meaning} \\ & \quad \text{and } B \text{ expresses that meaning "more directly"} \end{aligned}}$$

- $A \Rightarrow B$ is defined by ten simple rules, like a proof system
- Compositionality: $C_1 \Rightarrow C_2 \implies A\{x :\equiv C_1\} \Rightarrow A\{x :\equiv C_2\}$
- A term A is irreducible if $\boxed{A \Rightarrow B \implies A \equiv_c B}$

⋆ Variables and some simple immediate terms $x(v), \lambda v \dot{p}(u, v) \ldots$ refer immediately and are not assigned meaning

⋆ Non-immediate, explicit irreducible terms runs(x), λvloves(u, v), ... refer directly; they have meanings, albeit trivial ones which are exhausted by their denotations

⋆ Canonical forms and referential intensions of closed terms

Canonical Form Theorem

For each term A, there is a unique (up to congruence) recursive, irreducible, denotational term

$$\boxed{\mathrm{cf}(A) \equiv A_0 \text{ where } \{\dot{p}_1 := A_1, \ldots, \dot{p}_n := A_n\}}$$

such that $A \Rightarrow \mathrm{cf}(A)$. Each A_i is explicit and irreducible

⋆ $\mathrm{cf}(A)$ models the logical form of A

⋆ The parts $A_0, A_1, \ldots, A_n$ of A act like truth conditions for A

⋆ If A is closed, then its formal referential intension is

$$\mathrm{fint}(A) \equiv \Big(\lambda\vec{p}A_0, \lambda\vec{p}A_1, \ldots, \lambda\vec{p}A_n\Big);$$

and its referential intension is

$$\mathrm{int}(A) = \Big(\mathrm{den}(\lambda\vec{p}A_0), \mathrm{den}(\lambda\vec{p}A_1), \ldots, \mathrm{den}(\lambda\vec{p}A_n)\Big)$$

⋆ If A is irreducible, then $\mathrm{cf}(A) \equiv A$ and $\mathrm{int}(A) = (\mathrm{den}(A))$

The meaning of "John loves Mary and dislikes her husband"

$$\begin{aligned}\mathrm{cf}(A) \equiv \dot{p} \text{ and } \dot{q} \text{ where } \{&\dot{p} := \mathrm{loves}(\dot{j}, \dot{m}), \dot{q} := \mathrm{dislikes}(\dot{j}, \dot{h}),\\ &\dot{h} := \mathrm{husband}(\dot{m}), \dot{j} := \mathrm{John}, \dot{m} := \mathrm{Mary}\} : \tilde{\mathrm{t}}\end{aligned}$$

With $\vec{u} \equiv (p, q, h, j, m)$:

$$\begin{aligned}\mathrm{fint}(A) = \Big(&\lambda\vec{u}\,(p \text{ and } q), \lambda\vec{u}\,\mathrm{loves}(j, m), \lambda\vec{u}\,\mathrm{dislikes}(j, h),\\ &\lambda\vec{u}\,\mathrm{husband}(m), \lambda\vec{u}\,\mathrm{John}, \lambda\vec{u}\,\mathrm{Mary}\Big)\end{aligned}$$

$$\begin{aligned}\mathrm{int}(A) = \Big(&\mathrm{den}(\lambda\vec{u}\,(p \text{ and } q)), \mathrm{den}(\lambda\vec{u}\,\mathrm{loves}(j, m)),\\ &\mathrm{den}(\lambda\vec{u}\,\mathrm{dislikes}(j, h)), \mathrm{den}(\lambda\vec{u}\,\mathrm{husband}(m)),\\ &\mathrm{den}(\lambda\vec{u}\,\mathrm{John}), \mathrm{den}(\lambda\vec{u}\,\mathrm{Mary})\Big)\end{aligned}$$

★ Attitudinal application on closed terms

Notation: $A \Rightarrow_{\mathsf{cf}} B \iff \mathsf{cf}(A) \equiv B$

Jim is smart $\xrightarrow{\text{render}}$ smart(Jim) $\Rightarrow_{\mathsf{cf}}$ smart(j) where $\{j := \text{Jim}\}$

$$\text{fint}(\text{smart}(\text{Jim})) \equiv (\lambda j\, \text{smart}(j), \lambda j\, \text{Jim})$$

George believes that Jim is smart

$$\begin{aligned} &\xrightarrow{\text{render}} \text{Believes}(\text{G}, \text{that smart}(\text{Jim})) \\ &\Rightarrow \text{Believes}^t(\text{G}, \text{fint}(\text{smart}(\text{Jim}))) \qquad \star \\ &\equiv \text{Believes}^t(\text{G}, \lambda j\, \text{smart}(j), \lambda j\, \text{Jim}) \end{aligned}$$

where Believes^t is a denotational constant

★ If $\vec{A}$ are terms, B is a closed term and C is an attitudinal constant:

$$C(\vec{A}, \text{that } B) \Rightarrow C^t(\vec{A}, \text{fint}(B))$$

where C^t is a denotational constant whose type t depends on the sequence of types of the terms of $\vec{A}, \text{fint}(B)$. How is $C^t(x, \vec{y})$ defined?

Is he Scott? (after Scott-Soames, after Russell, Quine, Church, ...)

In a book-signing ceremony with a disguised Sir Walter Scott

George IV does not believe that *He is Scott*

George IV believes that *Scott is Scott*

The paradox: If α is the state of the ceremony,

$$\models \mathsf{He}(\alpha) = \mathsf{Scott}(\alpha); \tag{1}$$

but $\mathsf{He}(\alpha), \mathsf{Scott}(\alpha)$ are irreducible, and so

$$\mathsf{He}(\alpha) \approx \mathsf{Scott}(\alpha) \tag{2}$$

and so by the replacement property,

$$\Big(\mathsf{He}(\alpha) = \mathsf{Scott}(\alpha)\Big) \approx \Big(\mathsf{Scott}(\alpha) = \mathsf{Scott}(\alpha)\Big) \tag{3}$$

and so

GIV believes and does not believe the same thing in state α (4)

★ With referential synonymy, (1) – (3) are true, but (4) is false

Utterances and local meanings

Technical move: We add to $\mathsf{L}^{\lambda}_{\mathsf{ar}}$ parameters $\alpha, \beta, \ldots$ to name states

★ If $A : \tilde{\mathrm{t}}$, then $A(\alpha)$ is the utterance of A in state α, $A(\alpha) : \mathrm{t}$

★ The local meaning of A in state α is $\mathrm{int}(A(\alpha))$

★ *The objects of belief are local meanings* (standard)

Believes$(x, \text{that } A)$ is true in state α

$$\iff x(\alpha) \text{ believes in state } \alpha \text{ that } A(\alpha)$$

$$\boxed{\begin{aligned}&\Big(A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\}\Big)(B) \qquad \text{(recap rule)}\\ &\qquad\qquad \Rightarrow A_0(B) \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\}\end{aligned}}$$

$$\Big(\text{He} = \text{Scott}\Big)(\alpha) \Rightarrow_{\mathsf{cf}} (\dot{h}(\alpha) = \dot{s}(\alpha)) \text{ where } \{\dot{h} := \text{He}, \dot{s} := \text{Scott}\}$$

$$\Big(\text{Scott} = \text{Scott}\Big)(\alpha) \Rightarrow_{\mathsf{cf}} (\dot{h}(\alpha) = \dot{s}(\alpha)) \text{ where } \{\dot{h} := \text{Scott}, \dot{s} := \text{Scott}\}$$

$$\Big(\text{He} = \text{Scott}\Big)(\alpha) \not\approx \Big(\text{Scott} = \text{Scott}\Big)(\alpha)$$

- Local and global meanings $\sim$ Kaplan's content and character

$\star$ Referential intensions of terms with free variables

George believes that he is handsome

$$\xrightarrow{\text{render}} \text{Believes}(\dot{g}, \underline{\text{that handsome}(\dot{g})}) \text{ where } \{\dot{g} := \text{George}\}$$

$$\Rightarrow \text{Believes}^t(\dot{g}, \underline{\dot{g}, \lambda(g)\text{handsome}(g)}) \text{ where } \{\dot{g} := \text{George}\}$$

$\star$ If $A \Rightarrow_{\text{cf}} A_0$ where $\{p_1 := A_1, \ldots, p_n := A_n\}$
and $\vec{v} \equiv v_1, \ldots, v_k$ are all the free variables in A:

$$\text{fint}(A) \equiv (\vec{v}, \underline{\text{fint}_0(A)}) = \Big(\vec{v}, \underline{\lambda\vec{v}\lambda\vec{p}A_0, \lambda\vec{v}\vec{p}A_1, \ldots, \lambda\vec{v}\vec{p}A_n}\Big),$$

$$\text{int}(A, \vec{x}) = \Big(\vec{x}, \text{den}(\lambda\vec{v}\lambda\vec{p}A_0), \text{den}(\lambda\vec{v}\lambda\vec{p}A_1), \ldots, \text{den}(\lambda\vec{v}\lambda\vec{p}A_n)\Big)$$

The referential intension of A for $\vec{v} := \vec{x}$ is the pair of the input $\vec{x}$ and the algorithm for computing the function $\vec{x} \mapsto \text{den}(A\{\vec{v} := \vec{x}\})$

$\star$ (extending Quine): If $\vec{A}, B$ are terms, $\vec{v}$ are all the variables which occur free in B, and C is an attitudinal constant,

$$C(\vec{A}, \text{that } B) \Rightarrow C^t(\vec{A}, \text{fint}(B)) \equiv C^t(A, \vec{v}, \text{fint}_0(B))$$

C^t is denotational (and depends on the sequence of types in $\vec{A}, \text{fint}(B)$)

de re attitudinal application

Repeated application of the reduction rule

$$C(\vec{A}, \text{that } B) \Rightarrow C^t(\vec{A}, \vec{v}, \text{fint}_0(B))$$

transforms every term to a synonymous denotational term;
and so there is no technical obstruction to coherent "quantifying in"

The de re attribution of belief is rendered by

A believes of B that he satisfies C $\xrightarrow{\text{render}}$ $\text{Believes\_of}(A, B, \text{that } C)$

★ For each attitudinal C we introduce its de re version C_of so that its associated denotational primitive C_of^t satisfies

$$\models C\_\text{of}^t(A, v, \text{fint}_0(B))(\alpha) = C^t(A, \underline{\lambda\beta\, v(\alpha)}, \text{fint}_0(B))(\alpha) \quad (*)$$

This construct has the crucial de re property, that for every state α,

$$\models \Big(y(\alpha) = z(\alpha)\Big) \implies C\_\text{of}(A, y, B)(\alpha) = C\_\text{of}(A, z, B)(\alpha)$$

★ We cannot view $(*)$ as an abbreviation because
the term on the right is not in the natural language fragment

The muddle of King George

GIV believes of v that he is Scott

$$\xrightarrow{\text{render}} \text{Believes\_of}(\text{GIV}, v, \text{that } v \text{ is Scott})$$

If we set

$$A \equiv \text{Believes\_of}(\text{GIV}, \text{Scott}, \text{that } v = \text{Scott})$$
$$B \equiv \text{Believes\_of}(\text{GIV}, \text{He}, \text{that } v = \text{Scott})$$

then at the state α of the book-signing

$$\models A(\alpha) = B(\alpha)$$

and so if GIV believes of Scott that he is Scott at α then GIV also believes of the person he is pointing to that he is Scott

- When GIV claims both A and $\neg B$, he is contradicting himself
- $A(\alpha) \not\approx B(\alpha)$
  so GIV can coherently claim one of them and not the other at α

Uses of state

- G: Some delegate arrived and she registered
- J: She was probably that woman from New York
- G: No, I don't mean Eleanor, it was someone else

Some delegate arrived and she registered

$$(*) \qquad \not\xrightarrow{\text{render}} (\exists x)[\text{delegate}(x) \,\&\, \text{woman}(x) \,\&\, \text{registered}(x)]$$

Some delegate arrived and she registered

$$\xrightarrow{\text{render}} S \equiv \text{arrived}(\dot{w}) \,\&\, \text{registered}(\dot{w})$$
$$\text{where } \{\dot{w} := \nu(\lambda x[\text{delegate}(x) \,\&\, \text{woman}(x)])\}$$

★ $\nu : (\tilde{\mathrm{e}} \to \tilde{\mathrm{t}}) \to \tilde{\mathrm{e}}$ is a constant (like 'the') such that for every $p : (\tilde{\mathrm{e}} \to \tilde{\mathrm{t}})$ and every state α,

$$\text{either } \nu(p, \alpha) \text{ has property } p(\alpha) \text{ or } \nu(p, \alpha) = er$$

This is similar to the use of state in Discourse Representation Theory, but (if I am right),

- The state in DRT is completely determined by what is said
- At the end, the truth conditions are set (essentially) by $(*)$