Frege's sense and denotation as algorithm and value

Yiannis N. Moschovakis
University of California, Los Angeles
and University of Athens

Göteborg, October 23, 2014

A theory of logical form, meaning and synonymy

- Brief review of Frege's views on sense and denotation
- Some aspects of the theory (to be developed tomorrow)

The key idea, as a slogan:

- *The meaning of a linguistic expression A is faithfully represented by its referential intension* $\mathrm{int}(A)$, *an (abstract) algorithm which is determined by the syntactic structure of A and computes its denotation* (Meaning $\sim$ logical meaning, *what the words say*)

- A refinement of Richard Montague's fundamental work in the 1970's and an enrichment of it with a robust notion of meaning

References, all but the last 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 (ESSLLI 2010)
★ *The logical form and meaning of attitudinal sentences* (in preparation)

Frege 1892, *On sense and denotation*

$N \equiv 1 + 1 = 2$ vs. $P \equiv$ *there are infinitely many prime numbers*

- Same truth value but different thoughts are expressed, so one may understand both and know N but not know P

$ES \equiv$ *The evening star* vs. $MS \equiv$ *The morning star*

- Both ES and MS denote the planet Venus but "in different ways", so one may understand both but not believe that $ES = MS$

- The examples are of the same kind for Frege: he viewed declarative sentences like N and P as terms (denoting phrases, proper names, signs, ... like MS and ES) which denote their truth value tt or ff

- Central Frege doctrine: *Every term has a sense* (meaning) *which determines its denotation* (reference),

$$A \mapsto \text{sense}(A) \mapsto \text{den}(A)$$

Frege on sense

"*A proper name (word, sign, sign combination, expression) expresses its sense, stands for or denotes its reference*"

"*[the sense of a sign] may be the common property of many people*" Meanings are public (abstract?) objects

"*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*"

Speakers of the language know the meanings 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

Frege did not define sense

The closest he came to it is the famous quote:

> *It is natural, now, to think of there being connected with a sign . . . besides that to which the sign refers . . . what I should like to call the sense of the sign,*

wherein the mode of presentation [of the reference] *is contained*

Compare to

- Euclid: *A point is that which has no parts*
- Cantor: *By a set we are to understand any collection into a whole of definite and separate objects of our intuition or our thought*
- This suggests an axiomatic approach to our understanding of senses, and Church developed one (with several variations) in work spanning the period 1946 – 1974
- Church's theory is very complex because of the problem of indirect senses

Compositionality principles

• Compositionality Principle for Senses (implicit but clear in Frege 1892) so that, for example, there is a function F_{and} *on senses* such that

$$\begin{aligned}\text{sense}(&1+1=2 \text{ and } 3 \text{ is prime})\\ &= F_{\text{and}}(\text{sense}(1+1=2), \text{sense}(3 \text{ is a prime}))\end{aligned}$$

• Compositionality Principle for Denotations (explicit in Frege 1892) so that, for example, there is a function G_{and} *on truth values* such that

$$\begin{aligned}\text{den}(&1+1=2 \text{ and } 3 \text{ is prime})\\ &= G_{\text{and}}(\text{den}(1+1=2), \text{den}(3 \text{ is a prime}))\end{aligned}$$

• In a language with propositional attitudes (knowledge, belief, etc.), the Compositionality Principle for Denotations implies that

$$\begin{aligned}\text{den}(&\text{George knows that } 1+1=2)\\ &= \text{den}(\text{George knows that there are infinitely many primes})\end{aligned}$$

which is false for most Georges

Indirect senses

- According to Frege, the second argument of "knows" in such attitudinal sentences occurs indirectly, and

$$\boxed{\textit{The indirect denotation of a word is its customary sense}} \qquad (*)$$

so that

$$\text{den}(\text{George knows that } 1+1=2)$$
$$= H_{\text{knows}}(\text{den}(\text{George}), \text{sense}(1+1=2))$$

$(*)$ leads to very complex "nested senses", e.g., the sense of

John believes that George knows that there are infinitely many primes

- The Compositionality Principle for Denotations makes it possible to develop a rigorous, sense-independent theory of models for classical, mathematical logic which leaves meaning at the "premathematical" level

- For languages with propositional attitudes, $(*)$ does not eliminate senses—which (some say) is what Frege wanted to do

Denotational identity and synonymy

- Without precise definitions, let us temporarily write for any A, B,

$$A = B \iff \mathrm{den}(A) = \mathrm{den}(B) \qquad \text{(identity)},$$
$$A \approx B \iff \mathrm{sense}(A) = \mathrm{sense}(B) \qquad \text{(synonymy)}$$

With one of Frege's basic examples,

$$ES = MS \text{ because } \mathrm{den}(ES) = \mathrm{den}(MS) = \text{Venus}, \text{ but } ES \not\approx MS$$

- In a precisely formulated language, the problem of defining "sense" is equivalent to defining "synonymy", on the (natural) assumption that synonymy is an equivalence relation; because then

$$\mathrm{sense}(A) = \Big\{B : A \approx B\Big\}$$

Structural approaches

- In later work (1980's), Church, Cresswell and van Heijenoort use variants of the following idea:
  You get the sense of a term A by replacing in the syntactic construction of A all the primitive words by their denotations, e.g.,

$$\text{sense}(\textit{Phaliron lies between Piraeus and Sounion}) \\ = \langle \text{lies}, \text{Phaliron}, \langle \text{between}, \text{Piraeus}, \text{Sounion} \rangle \rangle$$

- This does not allow for semantic import into meaning, as it gives

$$\textit{Phaliron lies between Piraeus and Sounion} \\ \not\approx \textit{Phaliron lies between Sounion and Piraeus}$$

- A language speaker would know that for all x, y, z,

$$\text{between}(x, y, z) \iff \text{between}(x, z, y)$$

- Frege: *Abelard loves Eloise* $\approx$ *Eloise is_loved_by Abelard*

. . . and we would really like to know what meanings are!

Paraphrasing Donald Davidson:

> *Theaetetus and the property of flying do not (by themselves) amount to the meaning of "Theaetetus flies": we would like to know just what kind of objects meanings are, and how the meaning of "Theaetetus flies" is determined by the meanings of "Theaetetus" and "flying"*

. . . and also what is the meaning of the more interesting, true sentence

"*Theaetetus proved that* $\sqrt{17}$ *is irrational*"

and how it is determined by the meanings of "Theaetetus", "prove", "$\sqrt{17}$", and "irrational number"?

- Well, as Bill Clinton said, $\boxed{\textit{it depends of what the meaning of is is}}$
- In mathematics, we understand is to mean faithfully represented by and we often identify objects which are naturally isomorphic

sense(A) *contains the mode of presentation of* den(A)

- This (rough) quote from Frege has led many philosophers (including Dummett, Tichy and implicitly Davidson) to propose that

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

and if we replace in this proposal the vague "abstract procedure" by a rigorous notion of (abstract) algorithm, then it turns into a mathematical explication of the notion of sense, which is our aim

★ Thesis: for a precisely formalized fragment of natural language:

$\boxed{\textit{The meaning of a term } A \textit{ is an algorithm which computes } \mathrm{den}(A)}$

- I say "meaning" rather than "sense" because it is not entirely clear whether this is a "Fregean" notion of meaning. In critiquing Dummett, Gareth Evans called this approach

"ideal verificationism ... [for which there is] scant evidence [in Frege]"

So what is an algorithm?

...well, that's another, long (and complicated) story

- There is no generally accepted, precise and widely applicable definition of *algorithms*

- For rigorous work, especially in complexity theory, algorithms are typically specified by various computation models (Turing or Random access machines, etc.)

- An alternative approach is to specify algorithms by systems of mutually recursive equations which can be implemented by many, distinct computation models. These are the recursive programs introduced by John MacCarthy in the first order case, but they can also be used to specify infinitary (not implementable) algorithms as well as algorithms which interact with their environment

- *To represent meanings, we will need only very simple* acyclic algorithms, and so we can avoid a discussion of the general notion of *algorithm*

Acyclic recursors

- An acyclic recursor $\alpha : X \rightsquigarrow W$ is a tuple $\alpha = (\alpha_0, \alpha_1, \dots, \alpha_n)$ of functions such that the following hold with suitable sets $D_1, \dots, D_n$:

(R1) For $i = 1, \dots, n$, $\alpha_i : X \times D_1 \times \cdots \times D_n \to D_i$ so that with $d = (d_1, \dots, d_n)$, the system of equations

$$\left\{ d_1 = \alpha_1(x, d),\ d_2 = \alpha_2(x, d),\ \dots,\ d_n = \alpha_n(x, d) \right\} \qquad (*)$$

makes sense. It is the body of α

(R2) $\alpha_0 : X \times D_1 \times \cdots \times D_n \to W$. This is the head of α

(R3) There are numbers $\mathrm{rank}(1), \dots, \mathrm{rank}(n)$ such that

$$\mathrm{rank}(j) \leq \mathrm{rank}(i) \implies \alpha_j(x, d_1, \dots, d_n) \text{ is independent of } d_i$$

So, if $\mathrm{rank}(j)$ is least, then $\alpha_j(x, d_1, \dots, d_n) = f(x)$ for some $f : X \to D_j$ and the system $(*)$ unique solutions $\overline{d}_1(x), \dots, \overline{d}_n(x)$ for each $x \in X$

- The function $\overline{\alpha} : X \to W$ computed by α is

$$\overline{\alpha}(x) = \alpha_0(x, \overline{d}_1(x), \dots, \overline{d}_n(x))$$

Acyclic algorithm identity

- We can specify succinctly a recursor $\alpha = (\alpha_0, \ldots, \alpha_n) : X \rightsquigarrow W$ by

$$\boxed{\alpha(x) = \alpha_0(x, d) \underline{\textit{where}} \{d_1 = \alpha_1(x, d), \ldots, d_n = \alpha_n(x, d)\}}$$

with $d = (d_1, \ldots, d_n)$

- For the intended interpretation, the order of the equations in the body $\{d_1 = \alpha_1(x, d), \ldots, d_n = \alpha_n(x, d)\}$ of α is irrelevant
- The recursor $\alpha : X \rightsquigarrow W$ above is naturally isomorphic (equal) with

$$\boxed{\beta(x) = \beta_0(x, e) \underline{\textit{where}} \{e_1 = \beta_1(x, e), \ldots, e_m = \beta_m(x, e)\} : X \rightsquigarrow W}$$

if they have the same dimension ($m = n$), and there is a permutation

$$\pi : \{0, 1, \ldots, n\} \rightarrowtail\!\!\!\!\twoheadrightarrow \{0, 1, \ldots, n\} \text{ with } \pi(0) = 0,$$

such that $D_{\pi(i)} = E_i$ for $i = 1, \ldots, n$, and for $x \in X, d_i \in D_i, i \leq n$,

$$\alpha_{\pi(i)}(x, d_1, \ldots, d_n) = \beta_i(x, d_{\pi(1)}, \ldots, d_{\pi(n)})$$

- *Equal recursors model the same acyclic algorithm*

About modelling (real) algorithms by recursors

- A recursor $\alpha : X \rightsquigarrow W$ (in general) is specified by a tuple

$$\alpha = (\alpha_0, \alpha_1, \dots, \alpha_n),$$

such that for suitable $D_0 = W, D_1, \dots, D_n$,

$$\alpha_i : X \times D_1 \times \cdots \times D_n \to D_i$$

and we describe it succinctly by

$$\boxed{\alpha(x) = \alpha_0(x, d)\ \underline{\textit{where}}\ \{d_1 = \alpha_1(x, d), \dots, d_n = \alpha_n(x, d)\}}$$

- To insure that the system in the body of α has canonical solutions:
  - We may assume that it is acyclic, as we will do here, or
  - we may assume that each D_i is a directed complete poset and each α_i is Scott continuous, to develop algorithmic semantics for programming languages, or
  - we may assume that each D_i is a directed complete poset and each α_i is monotone, for higher type recursion theory, or
  - ...other things, e.g., to model concurrent algorithms

The methodology of formal, Fregean semantics

- An interpreted formalized language of terms L is selected
- A fragment of natural language is rendered in L:

  natural language expression + context

  $\xrightarrow{\text{render}}$ formal expression of L [+state]

★ *Rendering* is a matter of linguistics; I will only illustrate it by example

- Semantic values (denotations, meanings, etc.) are defined rigorously for the terms of L and pulled back to natural language expressions using the rendering

- Montague (1970): *English as a formal language*

  His *Language of Intensional Logic* was interpreted by Daniel Gallin into the *typed λ-calculus* (which he called Ty_2)

- The slogan for this work: *English as a typed programming language*

$\boxed{\text{L : } \textit{the typed } \lambda\textit{-calculus with acyclic recursion and propositional attitudes}}$

The precise definition of terms, denotations, referential intensions (meanings) and the corresponding notion of referential synonymy in L will be given in the research lecture

In the remaining three slides I will consider intuitively some simple examples which illustrate the ideas and point to possible applications

Some referential synonymies and non-synonymies

- There are infinitely many primes $\not\approx 1 + 1 = 2$
- $A \,\&\, B \approx B \,\&\, A$,
  A is between B and $C \approx A$ is between C and B
- Abelard loves Eloise $\approx$ Eloise $\boxed{\text{is loved by}}$ Abelard (Frege)
- $2 + 3 = 6 \approx 6 = 3 + 2$ (with $+$ and the numbers primitive)

- The morning star is the evening star
  $\approx$The evening star is the morning star
  (This fails with Montague's renderings of these terms)
- John stumbled and he fell $\xrightarrow{\text{render}}$
  $$A \equiv \text{stumbled}(\mathsf{J}) \,\&\, \text{fell}(\mathsf{J}) \text{ where } \{\mathsf{J} := \text{John}\}$$

• L extends the typed λ-calculus and its denotational fragment can be interpreted into the typed λ-calculus

. . . but L is intensionally richer: the term A in the last example is not referentially synonymous with any λ-calculus term

Examples of rendering into L

every man loves some woman (λ-calculus)

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

John loves himself (coindexing, anaphora with λ)

$$\xrightarrow{\text{render}} \text{loves(John, himself)} \xrightarrow{\text{coindex}} \lambda(u)\text{loves}(u,u)(\text{John})$$

(∗) Abelard loved and honored Eloise (coordination with λ)

$$\xrightarrow{\text{render}} \text{(loved and honored)(Abelard, Eloise)}$$

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

(∗∗) Abelard loved Eloise and he honored her (coindexing in L)

$$\xrightarrow{\text{render}} \text{loved(Abelard, Eloise) and honored(He, her)}$$

$$\xrightarrow{\text{coindex}} \text{loved}(\mathtt{A},\mathtt{E}) \text{ and honored}(\mathtt{A},\mathtt{E}) \text{ where } \{\mathtt{A} = \text{Abelard}, \mathtt{E} = \text{Eloise}\}$$

- (∗) is a predication while (∗∗) is a conjunction
- The rendering in L preserves the logical form

Propositional attitudes

In a certain state u,

- $A \equiv$ John believes that the blond likes him (belief$_1$, de dicto)

  Actually, the blond is Elaine who hates John, and so

- $B \equiv$ John does not believe that Elaine likes him

  No contradiction, because letting $\approx_u$ stand for local synonymy in u

  $$\boxed{\text{the blond likes him} \not\approx_u \text{Elaine likes him}}$$

- $C \equiv$ John believes of the blond that she likes him (belief$_2$, de re)
- $D \equiv$ John does not believe of Elaine that she likes him
- In the de re rending, $\boxed{\text{the blond likes him} \approx_u \text{Elaine likes him}}$

  so that $D =_u \neg C$, which causes a more serious paradox

- The terms belief$_1$ and belief$_2$ are Quine's whose treatment of "quantifying in" is close to ours. Not so with this other Quine quote:

  Intensions are creatures of darkness, and I shall rejoice with the reader when they are exorcized