Synonymy

Yiannis N. Moschovakis
UCLA and ΜΠΛΑ

Mathematical methods in philosophy
BIRS, 2 February, 2007

Frege on synonymy

Frege in a letter to Husserl (1906):

> *"It seems to me that we must have an objective criterion for recognizing a thought as the same thought, since without such a criterion a logical analysis is not possible."*

- Does Frege want a precise definition or a decision procedure for synonymy?
- The question is in the context of Fregean semantics, which are concerned with the logical analysis of language

$$\begin{aligned} A \approx B &\iff A \text{ is synonymous with } B \\ &\iff A \text{ and } B \text{ have the same meaning (sense)} \end{aligned}$$

Outline

(1) Some comments about synonymy (briefly)

(2) Meaning and synonymy in the formal system $\mathsf{L}^{\lambda}_{r}(K)$

(3) The decision problem for synonymy

References (in www.math.ucla.edu/$\sim$ynm)

- *Sense and Denotation as Algorithm and Value*, Logic Colloquium '90, Lectures Notes in Logic v. 2 (1994), pp. 210–249. (With a correction posted on the URL above.)
- *A logical calculus of meaning and synonymy*, Linguistics and Philosophy v. 29 (2006), pp. 27–89.
- (with E. Kalyvianaki) *Two aspects of situated meaning*, to appear.

Intuitions about synonymy

$$A \,\&\, B \overset{?}{\approx} B \,\&\, A$$

Abelard loves Eloise $\overset{?}{\approx}$ Eloise $\boxed{\text{is loved by}}$ Abelard (Frege: yes)

$$2 + 3 = 6 \overset{?}{\approx} 3 + 2 = 6 \qquad (+ \text{ a primitive})$$

Busch is the president $\overset{?}{\approx}$ He is the president (context: He = Busch)

Do we have robust intuitions about synonymy?

- A theory of synonymy should primarily explain the difference between propositions with the same truth value—e.g.,

  $2 + 2 = 4$ and there are infinitely many prime numbers

- Three additional reasons to study synonymy

(a) To give an account of faithful translation

Frege:

> *"the same sense has different expressions in different languages, or even in the same language"*
>
> *"the difference between a translation and the original should properly [preserve sense]"*

- Faithful translation = synonymy in the union of two languages
- Bilingual speakers generally agree on translation accuracy
  ...to within reason
- Better source for intuitions about meaning than entailment:
  "If Hamlet were bald and $2 + 2 = 4$, then Hamlet was bald"
  ...can be easily translated, but is the inference justified?

(b) To understand the logic of propositional attitudes

Othello believed that Desdemona and Cassio were lovers

(probably in Italian—and so he did not believe the English sentence but its meaning)

- The belief carriers are meanings—but what kind of meanings?

In a situation where he = Busch:

I believe that Busch is the President

$\stackrel{?}{\approx}$ I believe that he is the President

- Synonymy theory should help clarify what sort of meanings are the belief carriers

(c) To interpret self-referential sentence

liar $\equiv$ this sentence is false

truthteller $\equiv$ this sentence is true

Notwithstanding any other provisions of this agreement, ...

Article V. The Congress, whenever two thirds of both houses shall deem it necessary, shall propose amendments to this Constitution,

...

Clearly liar $\not\approx$ truthteller but why?

Outline

(1) Some comments about synonymy (briefly)

(2) Meaning and synonymy in the formal system $\mathsf{L}_r^\lambda(K)$

(3) The decision problem for synonymy

The typed λ-calculus with recursion $\mathsf{L}_r^\lambda(K)$ is a common extension of the *Formal Language of Recursion* FLR (1994) and the *Typed λ-calculus with acyclic recursion* $\mathsf{L}_{ar}^\lambda(K)$ (2006), which extends Montague's Language of Intensional Logic

The typed λ-calculus with recursion $\mathsf{L}^{\lambda}_{r}(K)$ - types

Basic types $b \equiv e \mid t \mid s$ (entities, truth values, states)

Types: $\sigma :\equiv b \mid (\sigma_1 \to \sigma_2)$

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

Every non-basic type is uniquely of the form

$$\sigma \equiv \sigma_1 \times \cdots \times \sigma_n \to b$$

$\mathrm{level}(b) = 0$
$\mathrm{level}(\sigma_1 \times \cdots \times \sigma_n \to b) = \max\{\mathrm{level}(\sigma_1), \ldots, \mathrm{level}(\sigma_n)\} + 1$

$\mathsf{L}_r^\lambda(K)$ - syntax

Pure Variables: v_0^σ, v_1^σ, ..., for each type σ $(v : \sigma)$

Recursive variables: p_0^σ, p_1^σ, ..., for each type σ $(p : \sigma)$

Constants: A set K, and for each $\mathrm{c} \in K$, $\mathrm{c} : \sigma_1 \times \cdots \times \sigma_n \to b$

Terms – with assumed type restrictions and assigned types $(A : \sigma)$

$$A :\equiv v \mid p \mid c(A_1, \dots, A_n) \mid B(C) \mid \lambda(v)(B)$$
$$\mid A_0 \text{ where } \{p_1 := A_1, \dots, p_n := A_n\}$$

$$C : \sigma, B : (\sigma \to \tau) \implies B(C) : \tau$$
$$v : \sigma, B : \tau \implies \lambda(v)(B) : (\sigma \to \tau)$$
$$A_0 : \sigma \implies A_0 \text{ where } \{p_1 := A_1, \dots, p_n := A_n\} : \sigma$$

Abbreviation: $\quad A(B, C, D) \equiv A(B)(C)(D)$

Rendering natural language in $L_r^\lambda(K)$

$$\text{Abelard loves Eloise} \xrightarrow{\text{render}} \text{loves(Abelard,Eloise)} : \tilde{t}$$

$$\text{Busch is the president} \xrightarrow{\text{render}} \text{eq(Busch,the(president))} : \tilde{t}$$

$$\text{liar} \xrightarrow{\text{render}} p \text{ where } \{p := \neg p\} : t$$

$$\text{truthteller} \xrightarrow{\text{render}} p \text{ where } \{p := p\} : t$$

$$\tilde{t} \equiv (s \to t) \qquad \text{(type of Carnap intensions)}$$

$$\tilde{e} \equiv (s \to e) \qquad \text{(type of individual concepts)}$$

$$\text{Abelard, Eloise, Busch} : \tilde{e}$$

$$\text{president} : \tilde{e} \to \tilde{t}, \text{eq} : \tilde{e} \times \tilde{e} \to \tilde{t}$$

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

Rendering natural language in $\mathsf{L}^{\lambda}_{r}(K)$

John stumbled and fell $\xrightarrow{\text{render}} \lambda(x)\Big(\text{stumbled}(x)\ \&\ \text{fell}(x)\Big)(\text{John})$

(predication after coordination)

This is in Montague's LIL, the language of Intensional logic
(as it is interpreted in $\mathsf{L}^{\lambda}_{r}(K)$)

John stumbled and he fell $\xrightarrow{\text{render}} \text{stumbled}(j)\ \&\ \text{fell}(j)$ where $\{j := \text{John}\}$

(conjunction after co-indexing)

The logical form of this sentence cannot be captured faithfully in LIL — recursion models co-indexing preserving logical form

$\mathsf{L}^{\lambda}_{r}(K)$ - denotational semantics

- We are given basic sets $\mathbb{T}_e, \mathbb{T}_t, \mathbb{T}_s$ for the basic types

$$\mathbb{T}_{\sigma\to\tau} = \text{the set of all functions } f : \mathbb{T}_\sigma \to \mathbb{T}_\tau$$
$$\mathbb{P}_b = \mathbb{T}_b \cup \{\bot\} = \text{the ``flat poset'' of } \mathbb{T}_b$$
$$\mathbb{P}_{\sigma\to\tau} = \text{the set of all functions } f : \mathbb{T}_\sigma \to \mathbb{P}_\tau$$

Each $\mathbb{P}_\sigma$ is a complete poset (with the pointwise ordering)

- We are given a monotone $\bar{c} : \mathbb{P}_{\sigma_1} \times \cdots \times \mathbb{P}_{\sigma_n} \to \mathbb{P}_b$

  for each constant $c : \sigma_1 \times \cdots \times \sigma_n \to b$

  - Pure variables of type σ vary over $\mathbb{T}_\sigma$; recursive ones over $\mathbb{P}_\sigma$
  - If $A : \sigma$ and π is a type-respecting assignment to the variables, then $\mathrm{den}(A)(\pi) \in \mathbb{P}_\sigma$.
  - Recursive terms are interpreted by the taking of least-fixed-points $\quad \mathrm{den}(\mathrm{liar}) = \mathrm{den}(\mathrm{truthteller}) = \bot$

Meaning in $\mathsf{L}^{\lambda}_{r}(K)$

- In slogan form: *The meaning of a term A is faithfully modeled by an algorithm* int(A) *which computes den$(A)(\pi)$ for every assignment π*
- The referential intension int(A) is (compositionally) determined from A
- int(A) is an abstract (not necessarily implementable) recursive algorithm which can be defined in $\mathsf{L}^{\lambda}_{r}(K)$
- Referential synonymy: $A \approx B \iff \text{int}(A) \sim \text{int}(A)$ (where $\sim$ is a natural isomorphism relation between abstract, recursive algorithms)
- Claim: Meanings are faithfully modeled
- Claim: Synonymy is captured (defined)

Is this notion of meaning Fregean?

Evans (in a discussion of Dummett's similar, computational interpretations of Frege's sense):

> *"This leads [Dummett] to think generally that the sense of an expression is (not a way of thinking about its [denotation], but) a method or procedure for determining its denotation. So someone who grasps the sense of a sentence will be possessed of some method for determining the sentence's truth value*
> *. . . ideal verificationism*
> *. . . there is scant evidence for attributing it to Frege"*

Converse question: If you posses a method for determining the truth value of a sentence A, do you then "grasp" the sense of A?

(Sounds more like Davidson rather than Frege)

Reduction, Canonical Forms and the Synonymy Theorem

- A reduction relation $A \Rightarrow B$ is defined on terms of $\mathrm{L}_r^\lambda(K)$
- Each term A is reducible to a unique (up to congruence) irreducible recursive term, its canonical form

$$A \Rightarrow \mathrm{cf}(A) \equiv A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\}$$

- $\boxed{\mathrm{int}(A) = (\mathrm{den}(A_0), \mathrm{den}(A_1), \ldots, \mathrm{den}(A_n))}$
- The parts $A_0, \ldots, A_n$ of A are irreducible, explicit terms
- $\mathrm{cf}(A)$ models the logical form of A
- *Synonymy Theorem. $A \approx B$ if and only if*

$$B \Rightarrow \mathrm{cf}(B) \equiv B_0 \text{ where } \{p_1 := B_1, \ldots, p_m := B_m\}$$

so that $n = m$ and for $i \leq n$, $den(A_i) = den(B_i)$

Some of the examples

$$A \mathbin{\&} B \approx B \mathbin{\&} A$$

Abelard loves Eloise $\approx$ Eloise $\boxed{\text{is loved by}}$ Abelard

$$\text{loves}(a, e) \text{ where } \{a := \text{Abelard}, e := \text{Eloise}\}$$

$$\approx \boxed{\text{is loved by}}(e, a) \text{ where } \{e := \text{Eloise}, a := \text{Abelard}\}$$

$$2 + 3 = 6 \approx 3 + 2 = 6$$

$$\text{eq}(s, r) \text{ where } \{\boxed{s := t + t'}, t := 2, t' := 3, r := 6\}$$

$$\text{eq}(s, r) \text{ where } \{\boxed{s := t' + t}, t' := 3, t := 2, r := 6\}$$

Busch is the president $\not\approx$ He is the president

(at state a, where He $=$ Busch)

$$\text{eq}(b, p)(\bar{a}) \text{ where } \{\boxed{b := \text{Busch}}, p := \text{the}(\text{President})\}$$

$$\not\approx \text{eq}(b, p)(\bar{a}) \text{ where } \{\boxed{b := \text{He}}, p := \text{the}(\text{President})\}$$

Outline

(1) Some comments about synonymy (briefly)

(2) Meaning and synonymy in the formal system $\mathsf{L}^{\lambda}_{r}(K)$

(3) The decision problem for synonymy

Is referential synonymy decidable?

Synonymy Theorem. $A \approx B$ if and only if

$$A \Rightarrow \mathrm{cf}(A) \equiv A_0 \text{ where } \{p_1 := A_1, \ldots, p_n := A_n\}$$
$$B \Rightarrow \mathrm{cf}(B) \equiv B_0 \text{ where } \{p_1 := B_1, \ldots, p_n := B_n\}$$

so that for $i = 0, \ldots, n$ and all π, $den(A_i)(\pi) = den(B_i)(\pi)$.

- Synonymy is reduced to denotational equality for explicit, irreducible terms
- Denotational equality for arbitrary terms is undecidable (there are constants, with fixed but arbitrary interpretations)
- The explicit, irreducible terms are very special

— but by no means trivial!

The synonymy problem for $\mathsf{L}_r^\lambda(K)$ (with finite K)

- The decision problem for $\mathsf{L}_r^\lambda(K)$-synonymy is open

Theorem *If the set of constants K is finite, then synonymy is decidable for terms of* **adjusted level** ≤ 2

These terms include the renditions in $\mathsf{L}_r^\lambda(K)$ of most (perhaps all) sentences of natural language—those involving nouns, adjectives, verbs, adverbs, logical and modal operators, descriptions, etc. They also include all sentences about mathematical structures which are included among the basic types

Proof is by reducing this claim to the Main Theorem in the 1994 paper

(There was a gap in the 1994 proof; a correct version is posted in www.math.ucla.edu/$\sim$ynm)

Explicit, irreducible identities that must be known

- Los Angeles = LA   (Athens = Αθήνα)
- $x \,\&\, y = y \,\&\, x$
- $\text{between}(x, y, z) = \text{between}(x, z, y)$
- $\text{love}(x, y) = \text{be\_loved}(y, x)$

A dictionary is needed—but what kind and how large?

$$\text{ev}_2(\lambda(u_1, u_2) r(u_1, u_2, \vec{a}), b, z) = \text{ev}_1(\lambda(v) r(v, z, \vec{a}), b)$$

Evaluation functions: both sides are equal to $r(b, z, \vec{a})$

The dictionary line which determines this is (essentially)

$$\lambda(s) x(s, z) = \lambda(s) y(s) \implies \text{ev}_2(x, b, z) = \text{ev}_1(y, b)$$

The form of the decision algorithm

- A finite list of true dictionary lines is constructed, which codifies the relationships between the constants
- Given two explicit, irreducible terms A, B of adjusted level ≤ 2, we construct (effectively) a finite set $L(A, B)$ of lines
- $A = B$ if and only if every line in $L(A, B)$ is congruent to one in the dictionary

- It is a lookup algorithm, justified by a finite basis theorem