

Chapter 2. Executing the Fregean Program

- ・ 文の「意味」、文の外延(“extension”)、真理条件(“truth-condition”)
- ・ Frege の構成性の原理(“compositionality”)—関数適用(“functional application”)—による意味計算
- ・ 計算に必要な道具立て：関数・集合・タイプ・ λ -notation

2.1. First example of a Fregean interpretation

文が指し示すもの(“denotation” = “extension”)：現実世界における真理値

真理値：真(1) or 偽(0)

固有名(“proper names”)の外延：現実世界においてその名で指し示される個体

自動詞の外延：個体から真理値への関数(“function”)

→未飽和(“unsaturated”)である関数が、項(“argument”)として NP を受

け入れることで飽和(“saturate”)し、真理値(“truth value”)を返す

解釈関数(“interpretation function”：“[[]]”)：言語表現に外延を割り当てる関数

→言語表現を項として受け入れることで、その外延を値として返す

Extensional semantics for the fragment of English

A. Inventory of denotations

- ・ D の要素 (D: 現実世界において存在する全ての個体(“individual”)の集合)
- ・ $\{0,1\}$ の要素 (False or True)
- ・ D から $\{0,1\}$ への関数 (D を項に取り真理値を返す関数)

B. Lexicon

[[**Ann**]] = Ann (etc. for other proper names)

[[**works**]] = $f: D \rightarrow \{0,1\}$

For all $x \in D$, $f(x) = 1$ iff x works. (etc. for other intransitive verbs)

C. Rules for non-terminal nodes

(S1): (関数適用)

$$\begin{array}{c} s \\ \wedge \\ \beta \quad \gamma \end{array}$$

If α has the form $\begin{array}{c} s \\ \wedge \\ \beta \quad \gamma \end{array}$, then $[[\alpha]] = [[\gamma]]([[\beta]])$.

(S2~S5): (Non-branching nodesにおける外延の継承)

An Example

“[[**Ann smokes**]] = 1 iff Ann smokes ”

$[[\text{Ann smokes}]] = [f: D \rightarrow \{0,1\} \text{ For all } x \in D, f(x) = 1 \text{ iff } x \text{ smokes. }](\text{Ann})$

$[[\text{smoke}]] = \begin{bmatrix} \text{Ann} & \rightarrow & 1 \\ \text{Jan} & \rightarrow & 1 \\ \text{Maria} & \rightarrow & 0 \end{bmatrix}$

→ “We don’t know of every existing individual whether or not (s)he smokes.”

2.2. Sets and their characteristic functions

自動詞は個体の集合としても見なすことができる

e.g. $[[\text{sleeps}]] = \{ x \in D: x \text{ sleeps} \}$

→ 主語と述語(“predicate”)を組み合わせる意味計算には別の規則が必要

特性関数(“characteristic function”)

Let A be a set. Then char_A , the *characteristic function* of A , is that function f such that, for any $x \in A$, $f(x) = 1$, and for any $x \notin A$, $f(x) = 0$.

Let f be a function with range $\{0,1\}$. Then char_f , the set characterized by f , is $\{ x \in D: f(x) = 1 \}$.

E.g. $D = \{ \text{Ann}, \text{Jan}, \text{Maria} \}$

• denoting sets

(3) $[[\text{sleeps}]] = \{ \text{Ann}, \text{Jan} \}$

(4) $[[\text{snores}]] = \{ \text{Ann} \}$

(5) $\text{Ann} \in [[\text{sleeps}]]$

(6) $[[\text{snores}]] \subseteq [[\text{sleeps}]]$

(7) $| [[\text{snores}]] \cap [[\text{sleeps}]] | = 1$

• characteristic functions of (3)(4)

(3') $[[\text{sleeps}]] = \begin{bmatrix} \text{Ann} & \rightarrow & 1 \\ \text{Jan} & \rightarrow & 1 \\ \text{Maria} & \rightarrow & 0 \end{bmatrix}$

(4') $[[\text{snores}]] = \begin{bmatrix} \text{Ann} & \rightarrow & 1 \\ \text{Jan} & \rightarrow & 0 \\ \text{Maria} & \rightarrow & 0 \end{bmatrix}$

(5') $[[\text{sleeps}]](\text{Ann}) = 1$

(6') $\{ x: [[\text{snores}]](x) = 1 \}$

$\subseteq \{ x: [[\text{sleeps}]](x) = 1 \}$

(7') $| \{ x: [[\text{snores}]](x) = 1 \}$

$\cap \{ x: [[\text{sleeps}]](x) = 1 \} | = 1$

特性関数によって述語を表すと、(5)(6)(7)のような直観を表すのに難儀になる。

2.3. Adding transitive verbs: semantic types and denotation domains

他動詞の導入

“Ann likes John.” $\rightarrow$ [s [NP Ann] [VP likes John]]

このような構造を認めるならば、他動詞のdenotationは何なのか？

- ・意味の合成は関数適用により行われる (Frege's conjecture)
- ・NPが指示するのは個体
- ・VPが指示するのは、個体から真理値への関数

$\rightarrow$ likeの指示対象は『「個体」から「個体から真理値への関数」への関数』

(直接目的語と項として取って、自動詞相当の関数を返す。1項述語)

$\rightarrow$ $[[like]] = f: D \in \{g: g \text{ is a function from } D \text{ to } \{0, 1\}\}$

For all $x, y \in D$, $[f(x)](y) = 1$ iff y likes x

ここでpp.26のfig.のような句構造の意味計算をするために、次の解釈規則を加える：

(S6) If α has the form $\begin{array}{c} \text{VP} \\ \wedge \\ \beta \quad \gamma \end{array}$, then $[[\alpha]] = [[\beta]]([[\gamma]])$. (cf. (S3) pp.14)

Types and Denotation Domains

Saturated denotation (eとtはBasic type)

- (1) e は個体を表すタイプ $D_e := D$
- (2) t は真理値を表すタイプ $D_t := \{0, 1\}$

Unsaturated denotation ($\langle \sigma, \tau \rangle : \sigma$ を項に取って τ を返す関数)

- (3) VP (自動詞) $D_{\langle e, t \rangle} := \{f: f \text{ is a function from } D_e \text{ to } D_t\}$
- (4) 他動詞 $D_{\langle e, \langle e, t \rangle \rangle} := \{f: f \text{ is a function from } D_e \text{ to } D_{\langle e, t \rangle}\}$

2.4. Schönfinkelization

Syntax-Semantics間に想定する事項

- ・ Binary Branch

他動詞は直接目的語と結合しVPを形成する。VPは主語と結合し文を形成する

- ・ Locality

意味解釈の規則は局所的に作用する。どの非終端ノードの指示対象も、その娘ノードの指示対象から計算される

- ・ Frege's conjecture

意味の合成は関数適用により行われる

論理学では、他動詞は2項述語、つまり順序付きのペアとして表される。しかしここでは他動詞は1項述語として扱っているが。。。

$D = \{\text{Sebastian, Dimitri, Leopord}\}$, D における“is-bigger-than”関係： R_{bigger} について：

集合による表記

$$R_{\text{bigger}} = \{ \langle \text{Sebastian, Dimitri} \rangle, \langle \text{Sebastian, Leopord} \rangle, \langle \text{Dimitri, Leopord} \rangle \}$$

関数による表記

R_{bigger} は $D \times D^1$ から真理値への関数であり、次の2項関数 f のように表される：

$$f_{\text{bigger}} = \begin{bmatrix} \langle L, S \rangle \rightarrow 0 \\ \langle L, D \rangle \rightarrow 0 \\ \langle L, L \rangle \rightarrow 0 \\ \langle S, L \rangle \rightarrow 1 \\ \langle S, D \rangle \rightarrow 1 \\ \langle S, S \rangle \rightarrow 0 \\ \langle D, L \rangle \rightarrow 1 \\ \langle D, S \rangle \rightarrow 0 \\ \langle D, D \rangle \rightarrow 0 \end{bmatrix} \xrightarrow{\text{右向き Schönfinkelization}} f'_{\text{bigger}} = \begin{bmatrix} L \rightarrow \begin{bmatrix} L \rightarrow 0 \\ S \rightarrow 0 \\ D \rightarrow 0 \end{bmatrix} \\ S \rightarrow \begin{bmatrix} L \rightarrow 1 \\ S \rightarrow 0 \\ D \rightarrow 1 \end{bmatrix} \\ D \rightarrow \begin{bmatrix} L \rightarrow 1 \\ S \rightarrow 0 \\ D \rightarrow 0 \end{bmatrix} \end{bmatrix}$$

$$\xrightarrow{\text{左向き Schönfinkelization}} f''_{\text{bigger}} = \begin{bmatrix} L \rightarrow \begin{bmatrix} L \rightarrow 0 \\ S \rightarrow 1 \\ D \rightarrow 1 \end{bmatrix} \\ S \rightarrow \begin{bmatrix} L \rightarrow 0 \\ S \rightarrow 0 \\ D \rightarrow 0 \end{bmatrix} \\ D \rightarrow \begin{bmatrix} L \rightarrow 0 \\ S \rightarrow 1 \\ D \rightarrow 0 \end{bmatrix} \end{bmatrix}$$

- ・ Schönfinkelizationにより、どの n 項関数も1項関数へ縮約される
- ・ Fregeの意味論における動詞のdenotationは、左向きSchönfinkelizationで表される
(まず目的語を項として取り、次いで主語を項として取る)

2.5. Defining function in the λ -notation

(1) $F_{+1} = f: \mathbf{N} \rightarrow \mathbf{N}$ For every $x \in \mathbf{N}$, $f(x) = x + 1$ (自然数から自然数への関数)

(2) $F_{+1} = [\lambda x: x \in \mathbf{N}. x + 1]$ ((1))

「自然数に含まれる全ての x を $x+1$ へ写像する関数」

(3) $[\lambda \alpha: \phi. \gamma]$

↓ 関数の定義域(“domain condition”)
↓ 式(“value description”)

項を表す変数(“argument valuable”、

(4) $[\lambda x: x \in \mathbf{N}. x + 1]$ (5) $= 5 + 1 = 6$

↓ 関数 ↓ 項に代入される数

¹ “ $D \times D$ ”は、 D と D の直積集合(“Cartesian product”)であり、 D の要素からなる全ての順序対の集合である。

言語表現のλ-notation (自動詞の場合)

(6) $[[\text{smoke}]] = f: D \rightarrow \{0,1\}$ For every $x \in D$, $f(x) = 1$ iff x smokes.

(3)において式 γ が文である場合は次のように読み替える：

「 γ である場合には ϕ であるようなあらゆる α を1に写像し、そうでない場合には0へ写像する関数」

(7) $[[\text{smoke}]] := [\lambda x: x \in D. x \text{ smokes}]$

→「 x がたばこを吸っている場合には、個体であるあらゆる x を1に写像し、そうでない場合は0へ写像する関数」

(8) $[[\text{smoke}]](\text{Ann}) = [\lambda x: x \in D. x \text{ smokes}](\text{Ann}) = 1$ iff Ann smokes.
(= 0 otherwise.)

(9) $[\lambda \alpha: \phi. \gamma]$

(i) ϕ であるようなあらゆる α を γ に写像する関数 (γ が文以外の場合)

(ii) γ である場合には ϕ であるようなあらゆる α を1に写像し、そうでない場合には0へ写像する関数 (γ が文の場合)

他動詞の場合

$[[\text{love}]] := [\lambda x: x \in D. [\lambda y: y \in D. y \text{ loves } x]]$

主語 y から真理値へ写像する関数(9ii)

目的語 x から「主語 y から真理値へ写像する関数」へ写像する関数(9i)

表記上の注意

(14) 最も外側のカッコの省略、ドメインの省略

a. $[[\text{love}]] := [\lambda x: x \in D. [\lambda y: y \in D. y \text{ loves } x]]$

b. $[[\text{love}]] := \lambda x. [\lambda y. y \text{ loves } x]$

(15) カッコを外すことで生じる曖昧性

a. $\lambda x. [\lambda y. y \text{ loves } x](\text{Sue})$

b. $[\lambda x. [\lambda y. y \text{ loves } x](\text{Sue})] = \lambda x. \text{Sue loves } x$

c. $[\lambda x. [\lambda y. y \text{ loves } x]](\text{Sue}) = \lambda y. y \text{ loves Sue}$