

- これまでに、
 - 複雑な表現の意味がその部分の意味と意味規則に基づいて構成的に計算されること（構成性の原理）
 - 文の表す命題を論理形式で表す方法
 - 文の表す命題の真理条件を集合論により捉えることを学んだ。
- 今回は、部分の意味が構成される方法を学ぶ。

(1) 形式的に意味構成を扱うための道具立て

- a. **タイプ (type):** 結合の可能性に基づく外延の分類
- b. **ラムダ計算 (lambda calculus):** 部分の意味が結合し、全体の意味を構成されるプロセスの分析に使う

1 タイプ (§4.1)

(2) 2の基本的タイプ

- a. **タイプ e :** 存在物 (entity)
「健」、「府中」などの固有名の外延はタイプ e
- b. **タイプ t :** 真理値 (truth values)
文の外延はタイプ t

- その他のタイプは、2つの基本的タイプから作られ、関数である。

2 関数 (§4.2)

(3) モスクワはパリの東側だ。EAST(m, p) —完全 (タイプ t)

- | | |
|---|--|
| <p>(4) a. 項—完全 (タイプ e)</p> <p>「モスクワ」: m</p> <p>「パリ」: p</p> | <p>b. 述語—項なしでは不完全</p> <p>「の東側だ」: EAST(x, y)</p> |
|---|--|

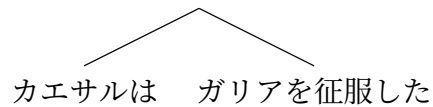
(5) Frege (1891/1980:31)

一般に言明 (statement) は [...] 2つの部分に分けられることが想定できる。一方はそれ自体で完全であり、もう一方は補う物が必要、つまり「非飽和 (unsaturated)」である。例えば、「カエサルはガリアを征服した」という文は、「カエサル (は)」と「ガリアを征服した」に分けられる。後者は「非飽和」であり、空所を1つ含む。この空所が固有名あるいは固有名を置き換える表現で満たされなければ、完全な意義は生じない。ここでも私はこの「非飽和」の部分が表すものに「関数」という名前を与える。この場合、その項は「カエサル」である。

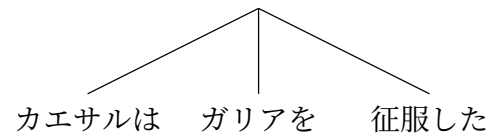
(6) フレーゲ

- a. すべての意味構成は関数適用 (function(al) application)、つまり関数とその項の結合である。
- b. 結合は二項から成る (binary)。一つの演算で結合されるのは、2つの表現（1つの関数と1つの項）のみ。

(7) a. 二項の結合



b. 三項の結合



3 関数の種類 (§4.3)

(8) 合成タイプ (composite types) (統語論)

a, b がタイプならば、 $\langle a, b \rangle$ もタイプである。

- (9) a. $\langle e, t \rangle$
- b. $\langle e, \langle e, t \rangle \rangle$
- c. $\langle \langle e, t \rangle, \langle e, \langle e, t \rangle \rangle \rangle$

(10) 簡略表記: $\langle e, t \rangle = et$

(9b) $\langle e, \langle e, t \rangle \rangle = \langle e, et \rangle$

(9c) $\langle \langle e, t \rangle, \langle e, \langle e, t \rangle \rangle \rangle = \langle et, \langle e, et \rangle \rangle$

(11) 合成タイプ (意味論)

タイプ $\langle a, b \rangle$ の表現は、 a を取って b を作る関数 (= a から b への関数) を外延とする。

(12) 例: $\langle e, t \rangle$

- a. 存在物 (タイプ e) から真理値 (タイプ t) への関数
= タイプ e の表現 (例: 固有名) と結び付き、タイプ t の表現 (文) を作る。



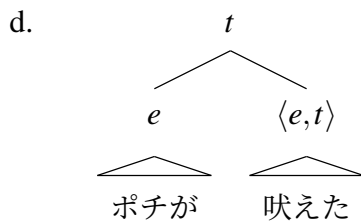
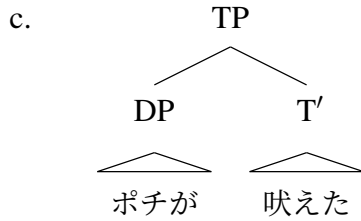
Q. (9b) と (9c) について、同様の樹形図を描きなさい。

一項述語

(13) a. 「犬」(普通名詞)、「赤い」(形容詞)、「歩く、吠える」(自動詞)

b. $\langle e, t \rangle$

存在物と結び付き、言明を作る関数



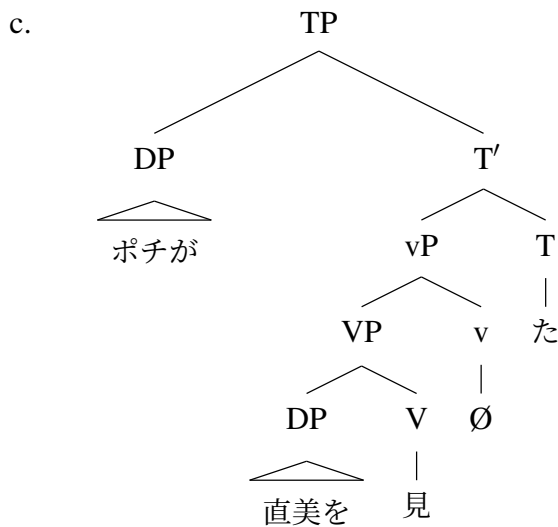
二項述語

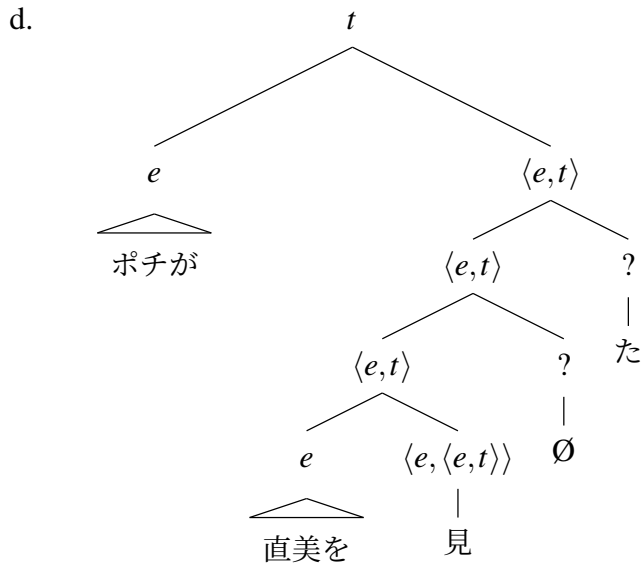
(14) a. 「殴る」「見る」「好む」(他動詞)、「友達」「兄弟」「同僚」「好き」(普通名詞、形容動詞)、「憎い」、higher (形容詞)、beside、under、in、on (前置詞)

b. $\langle e, \langle e, t \rangle \rangle$

存在物と結び付き、一項述語を作る関数

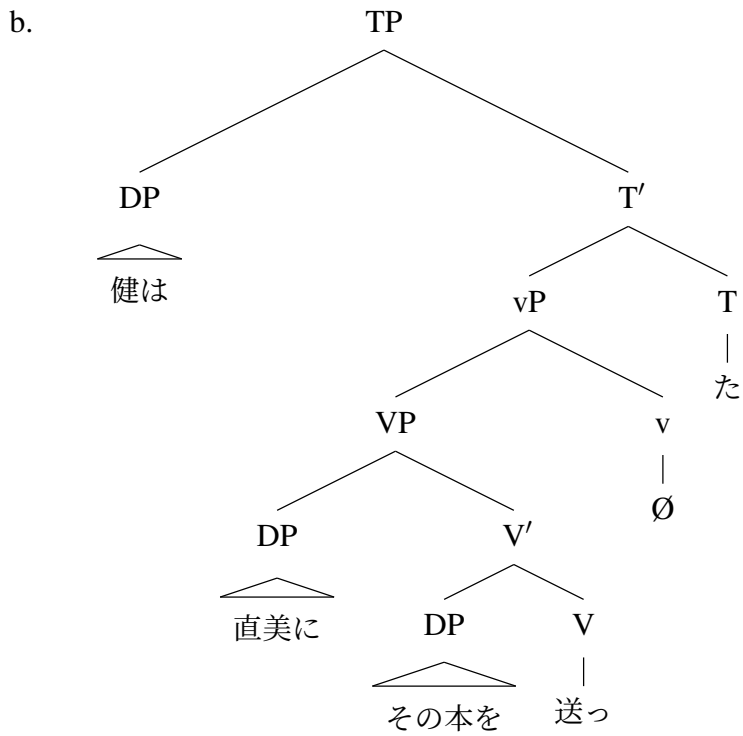
=存在物と結び付き、存在物と結び付き、言明を作る関数を作る関数

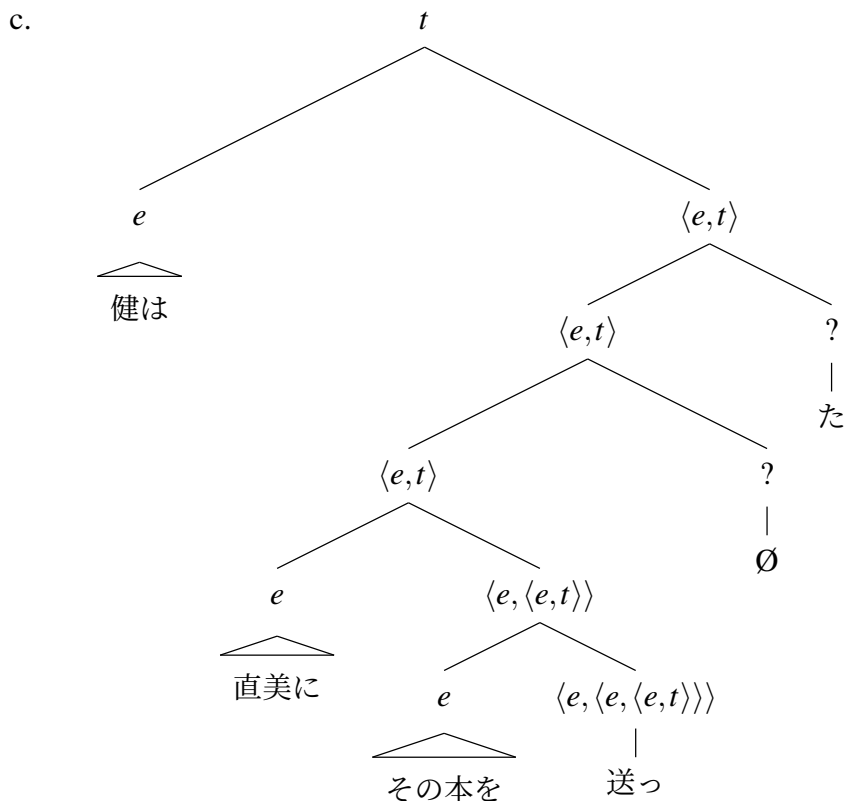




三項述語

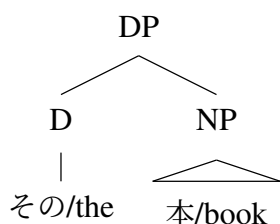
- (15) a. $\langle e, \langle e, \langle e, t \rangle \rangle \rangle$
 存在物と結び付き、二項述語を作る関数





決定詞「その/the」

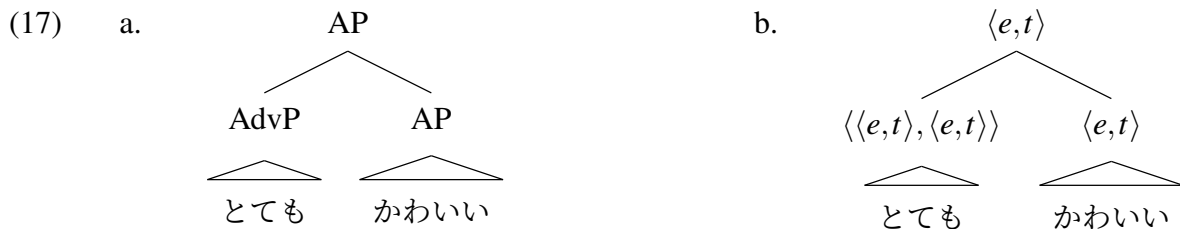
Q. 上の樹形図に示したように、定名詞句 (DP) 「その本を」全体はタイプ e の表現である。普通名詞のタイプは $\langle e, t \rangle$ である。定名詞句の構造が次のようになっているとすると、決定詞「その/the」は、どのようなタイプか？



修飾語

- (16) a. タマはかわいい。
 b. タマはとてもかわいい。

- 「かわいい」も「とてもかわいい」も一項述語 (タイプ $\langle e, t \rangle$)。
- よって、「とても」のような修飾語はタイプを変化させない。修飾語はタイプ X の表現と結び付き、タイプ X の表現を作る。
- 修飾語のタイプは、 $\langle X, X \rangle$ の形を取る。



4 ラムダ抽象 (§4.4)

述語の表示法

- (18) a. $GIVE(x, y, z)$
 —3つの項が非飽和の開いた命題 (open proposition).
 b. $GIVE(k, \text{その本}, n)$
 —3つの項が飽和された閉じた命題・文「健は直美にその本をあげた」。タイプ t 。

- $GIVE$ が閉じた命題ではなく関数であること、そして3つの項をどのような順で取るかをきちんと表示する方法として、**ラムダ計算** (lambda calculus) を用いる。

λ 演算子

- (19) a. 健は直美にその本をあげた。 $GIVE(k, \text{その本}, n)$
 ↓ 「その本を」を変数で置き換える
 b. 健は直美に x あげた。 $GIVE(k, x, n)$
 ↓ 変数 x を λ 演算子で束縛する
 c. $\lambda x[\text{健は直美に } x \text{ あげた}]$ $\lambda x[GIVE(k, x, n)]$
 【その他の表記法】 $\lambda x.$ 健は直美に x あげた $\lambda x.GIVE(k, x, n)$

- (20) $\lambda x[\text{健は直美に } x \text{ あげた}]$ $\lambda x[GIVE(k, x, n)]$
 a. x が埋まれば、命題になる
 b. 「健が直美にあげた存在物であるという特性を持つ」を表す一項述語
 c. 個体 x から命題への関数
 d. タイプ的には $\langle e, t \rangle$ 、つまり e から t への関数

- このように命題から関数を作る方法（例：(19a) → (19c)）は、**ラムダ抽象** (lambda abstraction) と呼ばれる。

Q. 以下の式の外延は何か？

- (21) a. $\lambda x[GIVE(x, \text{その本}, n)]$
 b. $\lambda x[GIVE(k, \text{その本}, x)]$

(22) $\lambda x[\dots x \dots](x$ を埋める項)

• ラムダ簡約 (lambda reduction) (あるいはラムダ変換 (lambda conversion)) は

1. λ により束縛された位置に項を入れ、
2. λ を削除する

操作である。

- (23) a. $\lambda x[\text{GIVE}(k, x, n)](\text{その本}) \Rightarrow \text{GIVE}(k, \text{その本}, n)$
 b. $\lambda x[\text{GIVE}(x, \text{その本}, n)](k) \Rightarrow \text{GIVE}(k, \text{その本}, n)$
 c. $\lambda x[\text{GIVE}(k, \text{その本}, x)](n) \Rightarrow \text{GIVE}(k, \text{その本}, n)$

(24) $\lambda x[\text{GIVE}(k, x, n)](\text{その本}) = \text{GIVE}(k, \text{その本}, n)$

一項述語

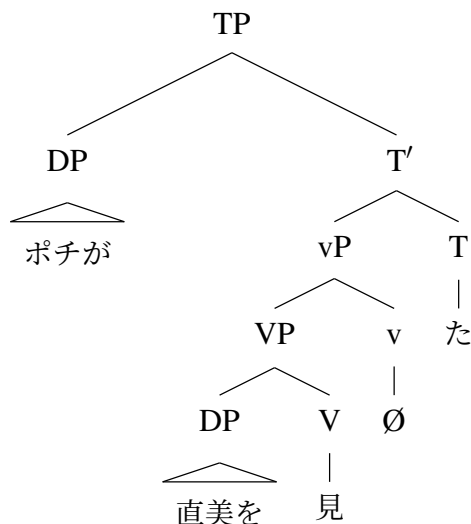
(25) $\lambda x[P(x)]$ (タイプ $\langle e, t \rangle$)

- a. $[[\text{犬 (だ)}]] = \lambda x[\text{DOG}(x)]$
- b. $[[\text{吠える}]] = \lambda x[\text{BARK}(x)]$
- c. $[[\text{赤い}]] = \lambda x[\text{RED}(x)]$

二項述語

(26) a. ポチが直美を見た。

b.



(27) a. 主語の λ 抽象

$\lambda x[x$ 直美を見た]

$\lambda x[\text{SAW}(x, n)]$

b. 目的語の λ 抽象

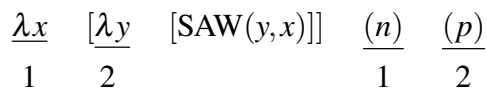
$\lambda y[\lambda x[x$ y 見た]]

$\lambda y[\lambda x[\text{SAW}(x, y)]]$

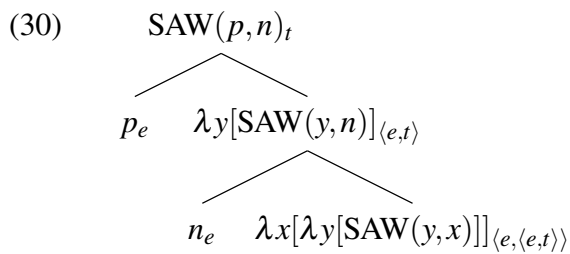
- c. その他の表記法
 $\lambda y[\lambda x[\text{SAW}(x,y)]]$
 $\lambda y\lambda x[\text{SAW}(x,y)]$
 $\lambda y.\lambda x.\text{SAW}(x,y)$
 $\lambda y\lambda x.\text{SAW}(x,y)$

• これに項を加えると、...

- (28) $\lambda x[\lambda y[\text{SAW}(y,x)]](n)(p)$
 —項は結合の順番に、左から右へと並ぶ。

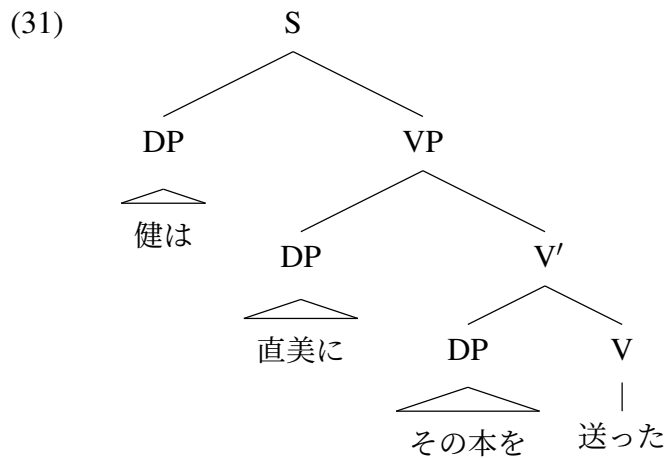


- (29) λ 簡約
 $\lambda x[\lambda y[\text{SAW}(y,x)]](n)(p)$
 =
 =



三項述語

Q. 次の（略式）統語構造に対応する意味樹形図をタイプとともに示しなさい。



等位構造

- (32) Fred [_{VP} danced and sang].
 $\lambda x[\text{DANCE}(x) \wedge \text{SING}(x)](f) = \text{DANCE}(f) \wedge \text{SING}(f)$

5 意味構成の一般規則 (§4.5)

終端節点 (terminal node)

- (33) 終端節点規則
 終端節点の解釈はレキシコンから得られる。

- | | | | | |
|------|----|-----|----|------------------------------|
| (34) | a. | 統語論 | b. | 意味論 |
| | | N | | [[N]] |
| | | | | = [[犬]] |
| | | 犬 | | = $\lambda x[\text{DOG}(x)]$ |

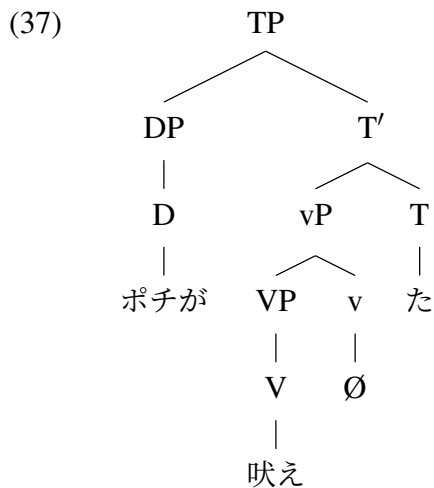
非枝分かれ節点 (non-branching node)

- (35) 非枝分かれ節点規則
 非枝分かれ節点の外延はその単一の姉妹の外延と等しい。

- | | | | | |
|------|----|-----|----|------------------------------|
| (36) | a. | 統語論 | b. | 意味論 |
| | | NP | | [[NP]] |
| | | | | = [[N]] |
| | | N | | = [[犬]] |
| | | | | = $\lambda x[\text{DOG}(x)]$ |
| | | 犬 | | |

枝分かれ節点 (branching node) (1) : 関数+項の場合

- 関数適用 (functional application)。その後、 λ 簡約。



(38) 終端節点規則と非枝分かれ節点規則により、

- a. $[[DP]] = p$
- b. $[[VP]] = \lambda x[\text{BARK}(x)]$

(39) 簡潔化のため、 \emptyset と「た」は意味的に何もしないとしておくと、
 $[[T']] = [[vP]] = [[VP]] = \lambda x[\text{BARK}(x)]$

(40) $[[TP]] = [[T']]([DP])$
 $= \lambda x[\text{BARK}(x)](p)$
 $= \text{BARK}(p)$

関数適用
 λ 簡約

(41) $\text{BARK}(p)_t$
 $p_e \quad \lambda x[\text{BARK}(x)]_{(e,t)}$

枝分かれ節点 (2)：一方が他方を修飾する 2 つの関数の場合

(42) 交差的修飾語 (intersective modifier)
 「青い傘」 $\text{BLUE}(x) \wedge \text{UMBRELLA}(x)$

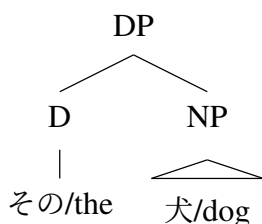
(43) 交差的修飾語 (intersective modifier) 規則
 $[[A]_{(e,t)} + B]_{(e,t)} = \lambda x[A(x) \wedge B(x)]$

- (44) a. $[[\text{青い傘}]] = \lambda x[\text{BLUE}(x) \wedge \text{UMBRELLA}(x)]$
- b. $[[\text{真理子が殺した男}]] = \lambda x[\text{MAN}(x) \wedge \text{KILL}(m,x)]$

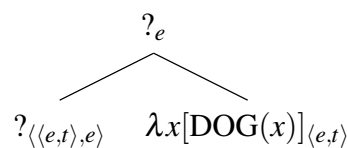
(45) 単純な関係節規則
 関係化される項を λ 抽象せよ。
 例： $[[\text{真理子が殺した}]] (\text{男}) = \lambda x[\text{KILL}(m,x)]$

決定詞「その/the」

(46) a. 統語論



b. 意味論



- 決定詞「その/the」は、全称量化子の一種（文脈により一義的に特定される集合の要素すべて）であると分析した。
- この意味の側面を捉えるのに、イオタ (*i*) 演算子が用いられることがある。

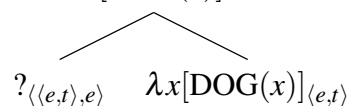
(47) *e* を形成する決定詞のための単純な規則

$$\llbracket \text{その/the } P \rrbracket = i x [P(x)]$$

$i [P(x)]$ はタイプ *e*

(48) a. $\llbracket \text{その犬} \rrbracket = i x [\text{DOG}(x)]$

b. $i x [\text{DOG}(x)]_e$



- $i x [\text{DOG}(x)]$ は、文脈により一義的に特定される犬を外延とする。
- この犬を *a* とすると、(49) が成り立つ。

(49) $i x [\text{DOG}(x)] = a$ iff $\exists y [\text{DOG}(y) \wedge \forall z [\text{DOG}(z) \rightarrow z = y] \wedge y = a]$

「ある犬 *y* がいたとして、どの犬 *z* も *y* と同じ存在物（つまり *y* が唯一の犬）であり、*y* が *a* と同じ存在物である場合、そしてその場合にのみ、『その犬』は存在物 *a* である。」

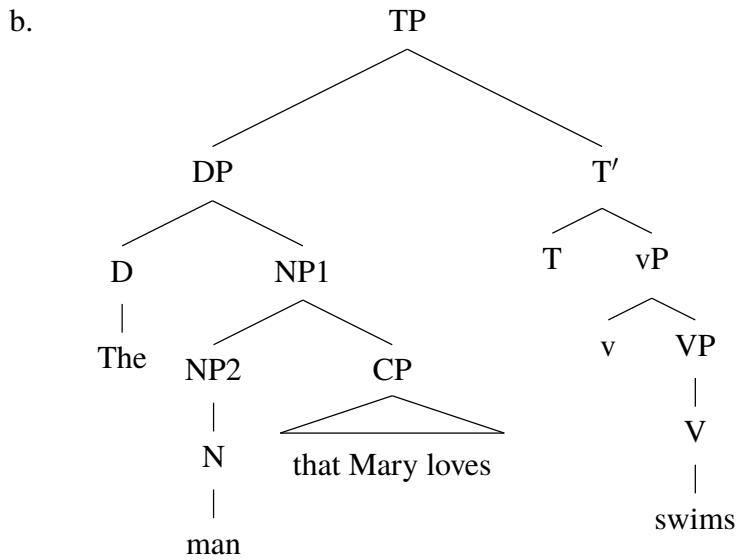
- $i x [P(x)]$ は、定数を付与することにより簡略化できる。

(50) タイプ *e* の名詞句への定数付与

$$i x [P(x)] = a$$

例

(51) a. The man that Mary loves swims.



(52) レキシコンより

a. $\llbracket \text{man} \rrbracket =$

b. $\llbracket \text{loves} \rrbracket =$

c. $\llbracket \text{swims} \rrbracket =$

(53) 終端節点規則

a. $\llbracket \text{N} \rrbracket = \llbracket \text{man} \rrbracket =$

b. $\llbracket \text{V} \rrbracket = \llbracket \text{loves} \rrbracket =$

c. $\llbracket \text{V} \rrbracket = \llbracket \text{swims} \rrbracket =$

(54) 非枝分かれ節点規則 + v、T を無視

a. $\llbracket \text{NP2} \rrbracket = \llbracket \text{N} \rrbracket = \llbracket \text{man} \rrbracket =$

b. $\llbracket \text{T}' \rrbracket = \llbracket \text{vP} \rrbracket = \llbracket \text{VP} \rrbracket = \llbracket \text{V} \rrbracket = \llbracket \text{swims} \rrbracket =$

(55) 単純な関係節規則

$\llbracket \text{CP} \rrbracket = \llbracket \text{that Mary loves} \rrbracket =$

(56) 交差的修飾語規則

[[NP1]] =

(57) e を形成する決定詞のための単純な規則

[[DP]] =

(58) (58) に定数 j を付与すると、

[[DP]] =

(59) 関数適用により、

[[TP]] =

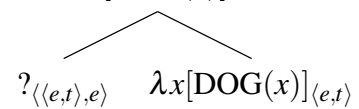
Q. タイプ付き意味樹形図を描きなさい。

6 述語変数 (§4.6)

- (48b) の「その/the」には、まだきちんとした外延がない。

(60) a. その犬/the dog

b. $\iota x[\text{DOG}(x)]_e$



- 個体ではなく、述語を表す変数が必要。
- 述語を表す変数には通常、 P, Q, R が用いられる。

$$(61) \quad \llbracket \text{その/the} \rrbracket = \lambda P[\iota x[P(x)]]$$

(62) 「その犬」の意味構成

a. レキシコンより

$$\llbracket \text{その} \rrbracket = \lambda P[\iota x[P(x)]]$$

$$\llbracket \text{犬} \rrbracket = \lambda y[\text{DOG}(y)]$$

b. 関数適用

$$\llbracket \text{その犬} \rrbracket = \llbracket \text{その} \rrbracket(\llbracket \text{犬} \rrbracket)$$

$$= \lambda P[\iota x[P(x)]](\lambda y[\text{DOG}(y)])$$

c. λ 簡約 (λP)

$$= \iota x[\lambda y[\text{DOG}(y)](x)]$$

d. λ 簡約 (λy)

$$= \iota x[\text{DOG}(x)]$$

- 交差的修飾語を含む名詞句の意味は、特別な意味規則 (43) を用いて、計算した。
- しかし、述語を表す変数を用いて、交差的修飾語の外延を表現することで、関数適用だけで意味計算をすることも可能。^{*1}

(63) 青い傘

$$\llbracket \text{青い} \rrbracket = \lambda P[\lambda x[\text{BLUE}(x) \wedge P(x)]]$$

(64) $\llbracket \text{青い傘} \rrbracket = \llbracket \text{青い} \rrbracket(\llbracket \text{傘} \rrbracket)$

=

(65) 述語名詞中の *a/an*

$$\llbracket a/an \rrbracket = \lambda P[P] \quad (\text{恒等関数 (identity function)})$$

(66) $\llbracket a \text{ dentist} \rrbracket = \llbracket a \rrbracket(\llbracket \text{dentist} \rrbracket)$

=

参考文献

Frege, Gottlob. 1891/1980. Function and concept. In *Translations from the Philosophical Writings of Gottlob Frege*, ed. P. T. Geach and Max Black, 22–41. Oxford: Blackwell, third edition.

^{*1} ただし、この場合、同じ語が述語の場合と修飾語の場合で異なる意味を持つことになる： $\langle e, t \rangle$ (述語として) vs. $\langle \langle e, t \rangle, \langle e, t \rangle \rangle$ (修飾語として)。交差的修飾語規則を設定する場合、このような曖昧性は生じない。