

# 依存型意味論とオントロジーによる述語の選択制限と論理的多義の分析

木下 恵梨子 (指導教員：戸次 大介)

## 1 はじめに：現象説明

述語には主語や目的語に対する選択制限が存在する。たとえば、自動詞 *cry* は主語に有生の存在物を要求する。本研究では、述語の選択制限の素朴な分析では捉えられない2つの現象に着目する。

1つ目は、論理的多義性を持つ名詞が copredication 構文を容認する現象である。自然言語には複数の意味を持つ名詞が存在し、それらは偶然的多義と論理的多義に分類される [1]。同一の表層形を共有する複数の語が、それぞれ互いに関係のない異なる意味を持つ場合、偶然的多義という。たとえば「銀行」と「土手」の2つの意味を持つ名詞 *bank* があげられる。一方、1つの語自体がある概念の複数の異なる相を表す場合、論理的多義という。たとえば「本」ではあるがその中で「情報」と「物理的実体」の2つの意味を持つ名詞 *book* があげられる。この2種類の多義性は以下の構文により区別できる。

- (1) a. # The bank is open and is steep.  
b. John memorized and burned the book.

このような、異なる選択制限を持つ2つ以上の述語が同じ項を共有する構文を copredication という。(1a) は偶然的多義の例である。*is open* は主語が銀行であることを、*is steep* は土手であることを要求する。一方 (1b) は論理的多義の例である。*memorized* は目的語が情報であることを、*burned* は物理的実体であることを要求する。ここで (1a) は許容されず、(1b) は許容されるという違いがある。しかし述語に選択制限があるのであれば、項にそれぞれ異なる性質の名詞を求めている2つの述語が1つの名詞を共有しているこの構文は、一様に不適切であるはずであり、(1b) のみが許容されるという現象は説明を要する。

2つ目は、コアーション現象が存在することである。コアーション現象とは、たとえば「喫茶店でオムレツを食べた客が一人いた」という文脈が与えられれば、以下の (2) を *The man who ate the omelet laughed.* の意味に取ることができる現象である。

- (2) The omelet laughed.

このとき述語に選択制限があるのであれば、動詞 *laugh* が主語に取ることができるのは有生の名詞のみのはずである。しかし実際は、(2) のように無生の名詞 *omelet* を取ることができ、これも説明が求められる。

本研究では、これらの現象を分析する枠組みとして、統語論として CCG [6] を、意味論として依存型意味論 [2] を用いる。述語の選択制限と名詞の論理的多義についての知識は、オントロジーを用いて記述することができる。また、上記2つの現象が観測されるということは、述語と項の間に何かしらの意味の変換があるということを示唆している。そこで、述語と項の間の意味変換を行う演算子を、オントロジーによって記述された知識に基づいて依存型意味論に導入することで、論理的多義性を持つ名詞の copredication 文が容認される現象やコアーション現象の説明を試みる。

## 2 依存型意味論とオントロジー

### 2.1 依存型意味論 (DTS)

依存型意味論 (Dependent Type Semantics; DTS) とは、依存型理論 [4] に基づいた証明論の意味論であり、そのため意味論について証明論的な推論が可能である。DTS による意味表示は  $\Pi$  型と  $\Sigma$  型が重要な役割を果たす。 $\Pi$ 、 $\Sigma$  はそれぞれ自然言語の全称量化、存在量化に対応しており、これらを用いることで先行する文脈を考慮した意味表示を記述することが可能となる。たとえば、(3a) の文の意味表示は DTS では (3b) のように与えられる。

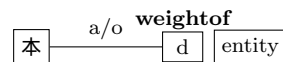
- (3) a. Every man entered.  
b.  $\left( u : \left[ \begin{array}{l} x : \text{entity} \\ \text{man}(x) \end{array} \right] \right) \rightarrow \text{enter}(\pi_1(u))$

*Every* は全称量化として扱われるため、文全体の意味表示は  $\Pi$  型となる。 $u$  は型 *entity* である項  $x$  と、 $x$  に依存した型  $\text{man}(x)$  からなる  $\Sigma$  型をもち、 $\text{enter}(\pi_1(u))$  の  $\pi_1(u)$  は  $u$  の第一要素である *entity* を指す。

DTS には聞き手にとって未知の意味表示 (underspecified representation) を表す演算子  $@$  が用意されている [8]。これにより文脈依存性を持つ現象である照応を扱うことができ、文脈によって解釈が変わるコアーション現象の分析に  $@$  を含んだ意味表示を用いることが可能である。

### 2.2 オントロジー

オントロジーとは、概念と概念間の関係 (意味リンク) を用いて記述される体系的な辞書であり [7]、概念間の関係を明示することでその背景にある情報を利用することが可能となる。意味リンクの1つである attribute-of リンクは各概念が持つ性質を記述するのに用いる。たとえば、「本は重さを持つ」という知識は以下のように表現される<sup>1</sup>。



本研究では構築したオントロジーを文の意味表示とあわせて推論に用いるため、DTS の公理へ変換する必要がある。たとえば上図の attribute-of リンクは以下のように DTS の公理へ変換される [5]<sup>2</sup>。

$$(x : \text{entity}) \rightarrow (\text{book}(x) \rightarrow \left[ \begin{array}{l} d : \text{entity} \\ \text{weightof}(d, x) \end{array} \right])$$

## 3 依存型意味論への各概念の導入と argument shift operator

### 3.1 述語の選択制限の導入

本研究では、述語の選択制限の概念を、各述語のイベントオントロジー [3] における attribute-of リンクとして導入する。主語のみを求める述語は 1、主語と目的語の両方を求める述語は 2、3 のように DTS の公理へと変換される。

1.  $\left[ \begin{array}{l} \text{cry} \\ \text{entity} \end{array} \right] \xrightarrow{\text{a/o animate}} \left[ \begin{array}{l} \text{entity} \\ \text{animate} \end{array} \right] \text{ (sub)}$   
 $(x : \text{entity}) \rightarrow (\text{cry}(x) \rightarrow \text{animate}(x))$
2.  $\left[ \begin{array}{l} \text{marry} \\ \text{entity} \end{array} \right] \xrightarrow{\text{a/o human}} \left[ \begin{array}{l} \text{entity} \\ \text{human} \end{array} \right] \text{ (sub)}$   
 $(x : \text{entity}) \rightarrow (y : \text{entity}) \rightarrow (\text{marry}(x, y) \rightarrow \text{human}(x))$
3.  $\left[ \begin{array}{l} \text{marry} \\ \text{entity} \end{array} \right] \xrightarrow{\text{a/o human}} \left[ \begin{array}{l} \text{entity} \\ \text{human} \end{array} \right] \text{ (obj)}$   
 $(x : \text{entity}) \rightarrow (y : \text{entity}) \rightarrow (\text{marry}(x, y) \rightarrow \text{human}(y))$

図 1 は、イベントオントロジーより合成される述語の選択制限を表す関数の例である。

- $l : (x : \text{entity}) \rightarrow (\text{laugh}(x) \rightarrow \text{animate}(x))$
- $m1 : (x : \text{entity}) \rightarrow (y : \text{entity}) \rightarrow (\text{memorize}(x, y) \rightarrow \text{animate}(x))$
- $m2 : (x : \text{entity}) \rightarrow (y : \text{entity}) \rightarrow (\text{memorize}(x, y) \rightarrow \left[ \begin{array}{l} z : \text{entity} \\ \text{infoof}(y, z) \end{array} \right])$
- $b2 : (x : \text{entity}) \rightarrow (y : \text{entity}) \rightarrow (\text{burn}(x, y) \rightarrow \left[ \begin{array}{l} z : \text{entity} \\ \text{phyobjof}(y, z) \end{array} \right])$
- $e1 : (x : \text{entity}) \rightarrow (y : \text{entity}) \rightarrow (\text{eat}(x, y) \rightarrow \text{animate}(z))$

図 1: 述語の選択制限を表す関数

### 3.2 論理的多義の導入

名詞の論理的多義性は、オントロジーに名詞と各性質の attribute-of リンクを追加することで記述される。たとえば、名詞 *book* に対しては図 2 のような関数がオントロジーより得られる。

<sup>1</sup>オントロジーの意味リンク、および意味リンクから変換された DTS の公理の記法は本論文に合わせて変更している。

<sup>2</sup>DTS の公理への変換の際、性質を表す意味表示が一項であるか二項であるかの差異が生じる。これはオントロジーの構造の違いによるものと考えられるが、紙面の都合で割愛する。

$$\begin{aligned} \text{book1} &: (x : \text{entity}) \rightarrow (\text{book}(x) \rightarrow \left[ \begin{array}{l} y : \text{entity} \\ \text{infoof}(y, x) \end{array} \right]) \\ \text{book2} &: (x : \text{entity}) \rightarrow (\text{book}(x) \rightarrow \left[ \begin{array}{l} y : \text{entity} \\ \text{phyobjof}(y, x) \end{array} \right]) \end{aligned}$$

図 2: 名詞の論理的多義性を表す関数

### 3.3 argument shift operator

述語と項の間での意味変換を行うために、まず意味変換後の項を特定する条件について考察する。第一の条件は、意味変換後の項が述語の選択制限により求められている性質を持つことである。第二の条件は、意味変換前の項と変換後の項の間には何かしらの関係があり、さらに意味変換前の項とその関係にある項はただひとつしか存在しないという条件である。この関係を関係  $R$  とすると、関係  $R$  は以下のように定義される  $\text{injection}(R)$  を満たすということになる。

$$\text{injection}(R) \stackrel{\text{def}}{=} (x : \text{entity}) \rightarrow (y : \text{entity}) \rightarrow (z : \text{entity}) \rightarrow \left[ \begin{array}{l} Rxy \\ Rxz \end{array} \right] \rightarrow y = z$$

この2つの条件を形式化するために、argument shift operator (図3、図4) を導入する。この演算子を用いることで、述語の選択制限により  $Pz$  もしくは  $Pzx$  で  $z$  の持つ性質に制限がかかる。さらに  $@$  に付加された関係  $R$  を文脈から探し出す。これにより意味変換後の項  $z$  を特定することができる。

$$\frac{\epsilon}{(S \setminus NP) \setminus (S \setminus NP)} : \lambda P. \lambda x. \left[ \begin{array}{l} z : \text{entity} \\ \pi_1 @ : \left[ \begin{array}{l} R : e \rightarrow e \rightarrow \text{type} \\ \text{injection}(R) \end{array} \right] \end{array} \right] Pz \quad xz$$

図 3: argument shift operator (自動詞用)

$$\frac{\epsilon}{(S \setminus NP / NP) \setminus (S \setminus NP / NP)} : \lambda P. \lambda y. \lambda x. \left[ \begin{array}{l} z : \text{entity} \\ \pi_1 @ : \left[ \begin{array}{l} R : e \rightarrow e \rightarrow e \rightarrow \text{type} \\ (x : \text{entity}) \rightarrow \text{injection}(Rx) \end{array} \right] \end{array} \right] Pzx \quad xyz$$

図 4: argument shift operator (他動詞用)

## 4 分析

### 4.1 論理的多義の問題

*memorized the book* は他動詞用の argument shift operator (図4) を用いると図5のように合成される。

$$\frac{\frac{\text{memorized}}{S \setminus NP / NP} : \lambda y. \lambda x. \text{memorize}(x, y) \quad \frac{\epsilon}{(S \setminus NP / NP) \setminus (S \setminus NP / NP)} : \lambda P. \lambda y. \lambda x. \left[ \begin{array}{l} z : \text{entity} \\ \pi_1 @ : \left[ \begin{array}{l} R : e \rightarrow e \rightarrow e \rightarrow \text{type} \\ (x : \text{entity}) \rightarrow \text{injection}(Rx) \end{array} \right] \end{array} \right] Pzx \quad xyz}{S \setminus NP / NP} \quad \frac{\text{the book}}{NP} : b}{S \setminus NP} : \lambda y. \lambda x. \left[ \begin{array}{l} z : \text{entity} \\ \pi_1 @ : \left[ \begin{array}{l} R : e \rightarrow e \rightarrow e \rightarrow \text{type} \\ (x : \text{entity}) \rightarrow \text{injection}(Rx) \end{array} \right] \end{array} \right] \text{memorize}(x, z) \quad xyz \\ \lambda x. \left[ \begin{array}{l} z : \text{entity} \\ \pi_1 @ : \left[ \begin{array}{l} R : e \rightarrow e \rightarrow e \rightarrow \text{type} \\ (x : \text{entity}) \rightarrow \text{injection}(Rx) \end{array} \right] \end{array} \right] \text{memorize}(x, z) \quad xbz$$

図 5: *memorized the book* の意味合成

$\text{memorize}(x, z)$  と図1の関数  $m2$  により  $z$  は性質  $\text{info}$  を持つという制限がかかる。これと図2における関数  $\text{book1}$  を用いることで関係  $R$  は以下のように構成される。

$$\lambda x. \lambda y. \lambda z. \left[ \begin{array}{l} u : \text{book}(y) \\ \pi_1(\text{book1}(y, u)) = \text{entity } z \end{array} \right]$$

$@$  をこの具体的な関係  $R$  で埋めることで文 *John memorized the book.* は以下のような意味表示となり、*John memorized information of the book.* の意味に解釈できるようになる。

$$\left[ \begin{array}{l} z : \text{entity} \\ \left[ \begin{array}{l} u : \text{book}(b) \\ \pi_1(\text{book1}(b, u)) = \text{entity } z \end{array} \right] \\ \text{memorize}(j, z) \end{array} \right]$$

*burned the book* においても同様にして関係  $R$  が以下のように構成される。

$$\lambda x. \lambda y. \lambda z. \left[ \begin{array}{l} u : \text{book}(y) \\ \pi_1(\text{book2}(y, u)) = \text{entity } z \end{array} \right]$$

よって copredication を含む文 *John memorized and burned the book.* は、述語と項の間での意味変換が *memorized* と *burned* のそれぞれで行われるために意味合成が可能になる。

### 4.2 コアーション現象

文 *The omelet laughed.* は自動詞用の argument shift operator (図3) を用いることで図6のように意味合成することが可能である。

$$\frac{\frac{\text{laughed}}{S \setminus NP} : \lambda x. \text{laugh}(x) \quad \frac{\epsilon}{(S \setminus NP) \setminus (S \setminus NP)} : \lambda P. \lambda x. \left[ \begin{array}{l} z : \text{entity} \\ \pi_1 @ : \left[ \begin{array}{l} R : e \rightarrow e \rightarrow \text{type} \\ (x : \text{entity}) \rightarrow \text{injection}(Rx) \end{array} \right] \end{array} \right] Pz \quad xz}{S \setminus NP} \quad \frac{\text{The omelet}}{NP} : o}{S} : \lambda x. \left[ \begin{array}{l} z : \text{entity} \\ \pi_1 @ : \left[ \begin{array}{l} R : e \rightarrow e \rightarrow \text{type} \\ (x : \text{entity}) \rightarrow \text{injection}(Rx) \end{array} \right] \end{array} \right] \text{laugh}(z) \quad oz$$

図 6: *The omelet laughed.* の意味合成

動詞 *laugh* の主語に対する選択制限は *the omelet* にかかるとはなかったが、argument shift operator を関数適用することにより項  $z$  へと制限対象が移行されるため、ここでの矛盾は解消される。続いて意味変換後の項  $z$  を特定する。  $\text{laugh}(z)$  と図1の関数  $l$  により、 $z$  は性質  $\text{animate}$  を持つという制限がかかる。ここで関係  $R$  として  $\lambda y. \lambda x. \text{eat}(x, y)$  を構成すれば、最終的な意味表示は以下ようになる。

$$\left[ \begin{array}{l} z : \text{entity} \\ \text{eat}(z, o) \\ \text{laugh}(z) \end{array} \right]$$

これは図1の関数  $e1$  と矛盾しない。よって文 *The omelet laughed.* は *The man who ate omelet laughed.* の意味に解釈できるようになる。

## 5 まとめと今後の課題

本研究では、述語の選択制限と名詞の論理的多義の2つの概念と、述語と項の間での意味変換を行う argument shift operator を依存型意味論に導入し、論理的多義やコアーション現象の問題点であった文の意味合成を行った。

現状では、実際はコアーション現象が起こるにもかかわらず意味合成をすることができない文脈が存在すること、また、適切でない意味に合成する過剰生成が発生しうることなど問題点が残っている。くわえて、argument shift operator を適用するための判断基準が存在しないという問題もあり、これらをどのように導入していくかが今後の課題である。

## 参考文献

- [1] Nicholas Asher. *Lexical meaning in context: A web of words*. Cambridge University Press, 2011.
- [2] Daisuke Bekki. Representing anaphora with dependent types. In *Logical Aspects of Computational Linguistics*, pp. 14–29. Springer, 2014.
- [3] Ai Kawazoe, Yusuke Miyao, Takuya Matsuzaki, Hikaru Yokono, and Noriko Arai. World history ontology for reasoning truth/falsehood of sentences: Event classification to fill in the gaps between knowledge resources and natural language texts. In Yukiko Nakano, Ken Satoh, and Daisuke Bekki, editors, *New Frontiers in Artificial Intelligence*, Vol. 8417 of *Lecture Notes in Computer Science*, pp. 42–50. Springer, 2014.
- [4] Per Martin-Löf. Intuitionistic type theory. *Naples: Bibliopolis*, 1984.
- [5] Ayako Nakamura, Koji Mineshima, and Daisuke Bekki. Towards modeling natural language inferences with part-whole relations using formal ontology and lexical semantics. In *proceedings of CEUR workshop*, Vol. 1517, 2015.
- [6] Mark J. Steedman. *Surface Structure and Interpretation*. The MIT Press, Cambridge, 1996.
- [7] 溝口理一郎. *オントロジー工学*. オーム社, 2005.
- [8] 佐藤未歩, 戸次大介. 依存型意味論における型推論の定式化と実装. 言語処理学会第21回年次大会発表論文集, pp. 461–464, 2015.