3O3-OS-04a-3

依存型意味論による述語の選択制限の前提としての分析

An Analysis of Selectional Restrictions as Presuppositions with Dependent Type Semantics

木下恵梨子 *1　峯島宏次 *1*3　戸次大介 *1*2*3
Eriko Kinoshita　Koji Mineshima　Daisuke Bekki

*1お茶の水女子大学 Ochanomizu University
*2国立情報学研究所 National Institute of Informatics
*3独立行政法人科学技術振興機構, CREST CREST, Japan Science and Technology Agency

Predicates in natural languages impose selectional restrictions on their arguments. In this paper, we analyze selectional restrictions of predicates as presuppositions within the framework of Dependent Type Semantics (DTS), a framework of natural language semantics based on dependent type theory. We also analyze two phenomena, coercion and copredication, which present challenges to a simple analysis of selectional restrictions, in terms of operators that shift the meanings of predicates.

1. はじめに

述語にはその項に対する選択制限 (seletional restriction) が存在する。たとえば、他動詞 *marry* は主語と目的語に人を表す表現を要求する。この選択制限により、たとえば、(1) の発話から、*Bob* と *Ann* が人であることを推論することができる。

(1) Bob married Ann.

この推論を説明する一つの方法は、述語の選択制限を文の含意 (entailment) の一部として扱うことである。この分析によれば、(1) には次のような解釈が割り当てられることになる。

(2) $\mathbf{marry}(\mathbf{bob}, \mathbf{ann}) \wedge \mathbf{human}(\mathbf{bob}) \wedge \mathbf{human}(\mathbf{ann})$

しかし、この分析では次のような推論を扱うことが困難である。

(3) Bob didn't marry Ann. ⇒ Bob and Ann are human.

(3) の否定文の発話からも、*Bob* や *Ann* が人であることを推論することが可能である。しかし、もし述語の選択制限が含意の一部であるならば、(3) は (4) の解釈をもつことになり、このような推論は不可能であることになってしまう。

(4) $\neg(\mathbf{marry}(\mathbf{bob}, \mathbf{ann}) \wedge \mathbf{human}(\mathbf{bob}) \wedge \mathbf{human}(\mathbf{ann}))$

一般に、述語の選択制限の情報は、否定やモダリティ、条件文などの演算子のスコープの外に置かれることが知られている [6, 2]。これは、前提の投射 (presupposition projection) と呼ばれる推論に共通する特徴であり、このため、選択制限を含意ではなく前提として分析する理論が最近になっていくつか提案されている [1, 5]。たとえば、(5) は、*Mary's sister* という表現が *Mary has a sister* という前提を引き起こす例であり、この推論パターンは、(3) の推論パターンと一致する。

(5) John didn't love Mary's sister. ⇒ Mary has a sister.

前研究 [11] では、依存型意味論 [4] の枠組みのもとで選択制限とそれに関連する言語現象の分析を与えた。本稿では、それをふまえて、依存型意味論による照応・前提の分析 [10] の枠組みのもとで、選択制限を前提として扱う新たな分析を提示する。

また、前研究から引き続き、述語の選択制限の単純な分析では説明することのできない 2 つの現象についても考察する。

連絡先: 木下恵梨子，お茶の水女子大学理学部情報科学科，g1220517@is.ocha.ac.jp

第 1 の現象は、コアーション現象である [8]。たとえば「喫茶店でオムレツを食べた客がいた」という文脈が与えられれば、(6a) を (6b) の意味にとることができる。

(6) a. The omelet escaped.
b. The man who ate the omelet escaped.

このように実際の項の性質が、述語の選択制限で指定される性質と異なるにもかかわらず解釈が許容される現象をコアーション現象という。

第 2 の現象は、論理的多義性を持つ名詞が copredication 構文を容認するという現象である。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]。この 2 種類の多義性は copredication と呼ばれる以下の構文により区別できる。

(7) a. # The bank is closed and is muddy.
b. Mary memorized and burned the book.

(7a) の *bank* は「銀行」と「土手」の 2 つの意味をもち、偶然的多義の例である。偶然的多義の場合は、copredication 構文を許容しない。一方、(7b) の *book* は「物理的対象としての本」と「情報としての本」の 2 つの意味をもち、論理的多義と呼ばれる。論理的多義性をもつ名詞は、copredication 構文を許容する。しかし、述語に選択制限があるのであれば、(7a) 同様に、(7b) も不適切であるはずである。選択制限の理論はこの現象を説明する必要がある。

この 2 つの現象は、述語と項の間に意味の変換があるということを示唆している。本研究では述語と項の間の意味変換を行う演算子を依存型意味論に導入することにより、この 2 つの現象の統一的な説明を試みる。

2. 先行研究

この節では、型理論の枠組みのもとで述語の選択制限とコアーション現象を分析した先行研究を紹介する。

2.1 Type Composition Logic

Type Composition Logic (TCL) [1] は、標準的な内包的意味論に、圏論や証明論の解釈と型システムを統合した理論である。 述語の選択制限は、TCL では型の照合として分析される。すなわち、述語が引数に型 τ を持つことを必要としているとき、引数の型 σ が τ と一致すれば、その引数の性質は述語の選択制限で指定されている性質と一致していることになる。

また、コアーション現象の解決のために、polymorphic type functor が導入される。これをインスタンス化した functor を述語と各引数を適用した後に適用することで、元の項と求める型をもつ項が置き換えられる [3]。

(8) a. Julie enjoyed the book.

b. $\lambda\Psi\lambda\Phi\lambda\pi\ \Phi(\pi * AG)\lambda\upsilon\ \Psi(\pi)(\lambda y_1\lambda\pi_1(enjoy(\upsilon, y_1\pi_1 * ARG_2^{enjoy} : EVT - \epsilon(HD(\Phi), HD(\Psi)))))$

c. $\lambda P\lambda\pi\exists x(book(x, \pi * BOOK) \wedge P(x)(\pi * BOOK))$

d. $\lambda\Psi\lambda\Phi\lambda\pi\ \Phi(\pi * AG)(\lambda\upsilon\exists x\ (book(x, \pi * BOOK) \wedge \lambda y_1\lambda\pi_1(enjoy(\upsilon, y_1\pi_1 * [ARG_2^{enjoy} : EVT - \epsilon(HD(\Phi), BOOK)]))(x)(\pi * [ARG_2^{enjoy} : EVT - \epsilon(HD(\Phi), BOOK)])))$

e. $\lambda P\lambda u\lambda\pi''(\exists z : \epsilon(RVT, BOOK)\exists z_1 : AG(P(\pi'')(z) \wedge \phi_{\epsilon(AG,BOOK)}(z, z_1, u, \pi'')))$

f. $\exists x\exists z\lambda\pi(enjoy(j, z, \pi) \wedge book(x) \wedge \phi_{\epsilon(AG,BOOK)}(z, j, x, \pi))$

たとえば、(8a) の文は *Julie enjoyed reading the book* の意味に変換される。(8b) の *enjoy* の語彙項目に (8c) の *book* の語彙項目を関数適用したものが (8d) であるが、*enjoy* は引数に event 型を求めているのに対し、実際は book 型が渡されているため矛盾している。そこで (8e) でインスタンス化された functor を適用することで最終的に (8f) となり、*enjoy* の引数の項は book 型をもつ項 x から項 z にシフトされ意味変換が生じる。$\phi_{\epsilon(AG,BOOK)}$ は文脈等により推測される。

しかしこの方法では、意味変換毎に functor をインスタンス化しなければならないのが難点である。さらに意味表示が冗長で直感的に分かりにくい点も問題点として挙げられる。

2.2 MTTs with Coercive Subtyping

Modern Type Theory (MTT) [5, 3] は、普通名詞を型として扱う many-sorted な論理システムである。コアーション現象は、context に coercive subtyping [5] を導入することにより説明される。局所的コアーションや依存型を用いることで、同じ文上で 1 つの名詞が複数の意味へ変換される場合も扱うことができる [3]。

(9) a. Mary started the book which John finished after many years.

b. $Evt : Human \rightarrow Type$
$start, finish\ :\ \Pi h : Human.(Evt(h) \rightarrow Prop)$
$read, write\ :\ \Pi h : Human.(Book \rightarrow Evh(h))$
$Book <_{c(h)} Evt(h)$
$$c(h,b) = \begin{cases} write(h,b) & if\ he\ wrote\ b \\ read(h,b) & otherwise \end{cases}$$

たとえば、(9a) では、動詞 *started* と *finished* はそれぞれ、*reading the book* 及び *writing the book* を項として要求している。そこで、(9b) を context に加えることで、coercive subtyping によりそれぞれ求めている意味への変換が可能となる。

しかし、この方法にもいくつか問題点がある。(9) の例では、context のコアーション関数 c の定義により、「著者が自分自身が書いた本を読む」という内容の意味変換を行うことが不可能である。また、(6a) のような例では、context に subtyping をどのように定義すればよいのかが不明確である。

3. 今回提案する手法：DTS

本稿では、統語論として CCG [9] を、意味論として依存型意味論 (DTS) [4] を用い、述語の選択制限の分析を試みる。システムの全体像を図 1 に示す。述語の選択制限を前提として扱うには、図 1 における型チェックの段階で選択制限が満たされているかどうかを計算する必要がある。そこで本稿では、選択制限の情報を DTS の語彙項目に記述することで、型チェックの段階での選択制限の分析を試みる。

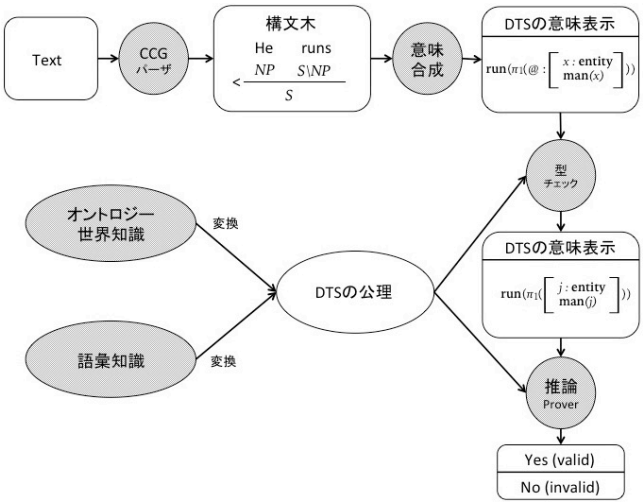

図 1: システムの全体像

3.1 依存型意味論 (DTS)

依存型意味論 (Dependent Type Semantics; DTS) とは、依存型理論 [7] に基づいた証明論的意味論であり、推論という観点から意味を分析する点に特徴がある。DTS による意味表示では Π 型と Σ 型が重要な役割を果たす。Π、Σ 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、(10a) の文の意味表示は DTS では (10b) のように与えられる。

(10) a. Every man entered.

b. $\left(u : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix} \right) \rightarrow \mathbf{enter}(\pi_1(u))$

Every は全称量化として扱われるため、文全体の意味表示は Π 型となる。u は型 **entity** である項 x と、x に依存した型 $\mathbf{man}(x)$ からなる Σ 型をもち、$\mathbf{enter}(\pi_1(u))$ 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指示対象が未知である表示 (underspecified representation) を表す演算子 @ が用意されている [10]。これにより文脈依存性を持つ現象である前提・照応を扱うことが可能となる。たとえば、代名詞 *he* の意味表示は $@ : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix}$ となり、@ が含まれている。これを用いることで (11a) の意味表示は以下のように与えられる。

(11) a. Every man entered.

b. $\mathbf{whistle}\left(\pi_1(@ : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix}) \right)$

John という男性が存在することが文脈により証明されている場合、この意味表示の型チェックの証明木は以下のようになる (各推論規則の詳細は [10] を参照)。

$$\dfrac{\mathbf{whistle} : \mathbf{entity} \rightarrow \mathbf{type}\ Con \qquad \dfrac{\dfrac{\dfrac{}{john : \mathbf{entity}}\ Con \quad \dfrac{}{t : \mathbf{man}(john)}\ Con}{(john, t) : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix}}\ \Sigma I \qquad \dfrac{}{\begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix} : \mathbf{type}}}{(@ : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix}) : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix}}\ @}{\pi_1(@ : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix}) : \mathbf{entity}}\ \Sigma E}{\mathbf{whistle}\left(\pi_1(@ : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{man}(x) \end{bmatrix}) \right) : \mathbf{type}}\ \Pi E$$

@を具体的な項 $(john, t)$ と置き換えることにより、(11b) の意味表示は最終的に $\mathbf{whistle}(john)$ となる。このように、前提・照応解決は、型チェックにより文の意味表示から @ の型を特定し、その型をもつ具体的な項を証明探索により構成することで行われる。以下では、この @ を述語の意味表示に組み込むことで、述語の選択制限を前提として導入することを試みる。

4. 各概念と演算子の導入

4.1 述語の選択制限

自動詞と他動詞の語彙項目を次のように定義する。

例	統語範疇／意味表示
cry	$S\backslash NP$
	$\lambda x.\mathbf{cry}(x, @ : \mathbf{animate}(x))$
marry	$S\backslash NP/NP$
	$\lambda y.\lambda x.\mathbf{marry}(x, @_i : \mathbf{human}(x))(y, @_j : \mathbf{human}(y))$

この意味表示に現れる述語 **cry** と **marry** の型は依存型を用いてそれぞれ以下のように定義される。

$$\mathbf{cry} : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{animate}(x) \end{bmatrix} \to \mathsf{type}$$

$$\mathbf{marry} : \begin{bmatrix} x : \mathbf{entity} \\ \mathbf{human}(x) \end{bmatrix} \to \begin{bmatrix} y : \mathbf{entity} \\ \mathbf{human}(y) \end{bmatrix} \to \mathsf{type}$$

たとえば、自動詞 *cry* では、語彙項目の意味表示内の @ : $\mathbf{animate}(x)$ より、型チェックの段階で $\mathbf{animate}(x)$ を証明する必要が生じる。これにより、動詞 *cry* の主語は **animate** の性質をもつことが保証される。

4.2 論理的多義

名詞の論理的多義性は、前研究 [11] で、たとえば名詞 *book* に対しては図 2 のような関数として導入した。このとき関数 $book_{\mathbf{InfoOf}}$, $book_{\mathbf{PhyObjOf}}$ はそれぞれ、名詞 *book* が情報としての側面、物理的実体としての側面を持つことを表している。この関数を本稿でも採用する。

$book_{\mathbf{InfoOf}}$:

$$(x : \mathbf{entity}) \to (\mathbf{book}(x) \to \begin{bmatrix} y : \mathbf{entity} \\ \mathbf{InfoOf}(y, x) \end{bmatrix})$$

$book_{\mathbf{PhyObjOf}}$:

$$(x : \mathbf{entity}) \to (\mathbf{book}(x) \to \begin{bmatrix} y : \mathbf{entity} \\ \mathbf{PhyObjOf}(y, x) \end{bmatrix})$$

図 2: 名詞の論理的多義性を表す関数

4.3 Argument operator

述語と項の間の意味変換を行う演算子は、argument operator（図 3、図 4）として定義する。これを用いることで、意味変換前の項と何らかの関係があり、かつその中でも一番関係の強い項が照応解決により文脈上から探し出され、その項と意味変換前の項が置き換えられる *1。TCL の手法では問題であった functor の多様性は、この 2 つの operator だけですべての意味変換に対応することができる点により解消される。

$$\frac{\epsilon}{(S\backslash NP)\backslash(S\backslash NP)} : \lambda P.\lambda x.P\left(\pi_1(@_1 : \begin{bmatrix} z : \mathbf{entity} \\ \begin{bmatrix} R : \mathbf{entity} \to \mathbf{entity} \to \mathsf{type} \end{bmatrix} \\ Rxz \end{bmatrix})\right)$$

図 3: argument operator（自動詞用）

$$\frac{\epsilon}{(S\backslash NP/NP)\backslash(S\backslash NP/NP)} : \lambda P.\lambda y.\lambda x.P\left(\pi_1(@_1 : \begin{bmatrix} z : \mathbf{entity} \\ \begin{bmatrix} R : \mathbf{entity} \to \mathbf{entity} \to \mathbf{entity} \to \mathsf{type} \end{bmatrix} \\ Rxyz \end{bmatrix})\right)(x)$$

図 4: argument operator（他動詞用）

5. 分析

以上で導入した手法を使用して、述語の選択制限とその際の問題であった 2 つの現象を分析し、各構文の意味合成を示す。以降簡略化のため、型 **entity** を **e** と記す。

5.1 述語の選択制限

まず選択制限に違反した例 *The chair cried.* を取り上げる。この文の意味表示を $\mathbf{cry}(chair, @ : \mathbf{animate}(chair))$ とする。この意味表示の型チェックの証明木は以下のようになる。

$$\dfrac{\mathbf{cry} : \begin{bmatrix} x : \mathbf{e} \\ \mathbf{animate}(x) \end{bmatrix} \to \mathsf{type} \quad Con \quad \dfrac{\overline{chair : \mathbf{e}}\ Con \quad \vdots \quad (@ : \mathbf{animate}(chair)) : \mathbf{animate}(chair)}{(chair, @ : \mathbf{animate}(chair)) : \begin{bmatrix} x : \mathbf{e} \\ \mathbf{animate}(x) \end{bmatrix}}\ \Sigma I}{\mathbf{cry}(chair, @ : \mathbf{animate}(chair)) : \mathsf{type}}\ \Pi E$$

@ を含む開いた枝により、$\mathbf{animate}(chair)$ の証明を文脈から探すことが要求される。つまり、「自動詞 *cry* の主語である項 *chair* は **animate** の性質を持つ」という選択制限は、意味表示が well-typed であるための前提である。しかし、*the chair* は無生物であるため、そのような証明は存在しない。よってこの文は（後述のコアーションなどの操作を行わないかぎり）well-typed ではない。

また、否定文 *The chair didn't cry.* の意味表示は $\neg\mathbf{cry}(chair, @ : \mathbf{animate}(chair))$ である。型チェックは、否定 ($\neg$) の形成規則を使用して以下のように進行する。

$$\dfrac{\dfrac{\mathbf{cry} : \begin{bmatrix} x : \mathbf{e} \\ \mathbf{animate}(x) \end{bmatrix} \to \mathsf{type} \quad Con \quad \dfrac{\overline{chair : \mathbf{e}}\ Con \quad \vdots \quad (@ : \mathbf{animate}(chair)) : \mathbf{animate}(chair)}{(chair, @ : \mathbf{animate}(chair)) : \begin{bmatrix} x : \mathbf{e} \\ \mathbf{animate}(x) \end{bmatrix}}\ \Sigma I}{\mathbf{cry}(chair, @ : \mathbf{animate}(chair)) : \mathsf{type}}\ \Pi E}{\neg\mathbf{cry}(chair, @ : \mathbf{animate}(chair)) : \mathsf{type}}\ \neg F$$

このときも同様に @ を含む枝により、$\mathbf{animate}(chair)$ の証明が要求される。よって、選択制限が否定のスコープから投射するという前提の推論パターンを導くことが可能となる。

5.2 コアーション現象

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

$$\dfrac{\dfrac{\dfrac{\text{escaped}}{S\backslash NP} : \lambda x.\mathbf{escape}(x, @_1 : \mathbf{animate}(x)) \quad \dfrac{\epsilon}{(S\backslash NP)\backslash(S\backslash NP)} : \lambda P.\lambda x.P\left(\pi_1(@_2 : \begin{bmatrix} z : \mathbf{e} \\ \begin{bmatrix} R : \mathbf{e} \to \mathbf{e} \to \mathsf{type} \end{bmatrix} \\ Rxz \end{bmatrix})\right)}{S\backslash NP}\ <}{S}\ <$$

$$\dfrac{\dfrac{\text{The omelet}}{NP} : o \quad : \lambda x.\mathbf{escape}\left(\pi_1(@_2 : \begin{bmatrix} z : \mathbf{e} \\ \begin{bmatrix} R : \mathbf{e} \to \mathbf{e} \to \mathsf{type} \end{bmatrix} \\ Rxz \end{bmatrix}), @_1 : \mathbf{animate}\left(\pi_1(@_2 : \begin{bmatrix} z : \mathbf{e} \\ \begin{bmatrix} R : \mathbf{e} \to \mathbf{e} \to \mathsf{type} \end{bmatrix} \\ Rxz \end{bmatrix})\right)\right)}{S}\ <$$

$$: \mathbf{escape}\left(\pi_1(@_2 : \begin{bmatrix} z : \mathbf{e} \\ \begin{bmatrix} R : \mathbf{e} \to \mathbf{e} \to \mathsf{type} \end{bmatrix} \\ Roz \end{bmatrix}), @_1 : \mathbf{animate}\left(\pi_1(@_2 : \begin{bmatrix} z : \mathbf{e} \\ \begin{bmatrix} R : \mathbf{e} \to \mathbf{e} \to \mathsf{type} \end{bmatrix} \\ Roz \end{bmatrix})\right)\right)$$

図 6: *The omelet escaped.* の意味合成

*1 前研究 [11] では、意味変換が起こる条件として、意味変換前の項と後の項には何らかの関係があり、かつ意味変換前の項とその関係にある項はただ 1 つだけであると仮定していた。しかし、実際は意味変換が生じるにもかかわらず意味合成を行うことができない文脈が存在することが問題であった。ここでは、意味変換前の項とその関係にある項が複数存在した場合は照応解決に基づいて意味変換が行われると考え、演算子の定義を修正した。

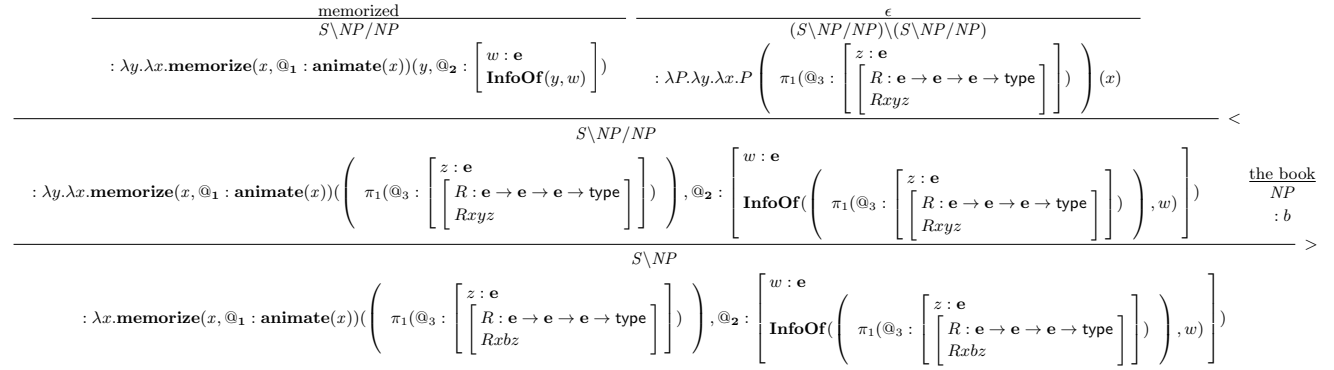

図 5: *memorized the book* の意味合成

動詞 *escape* の主語に対する選択制限は *the omelet* の項 o にかかるはずであったが、argument 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、ここでの動詞の選択制限による矛盾は解消される。ここで関係 R として動詞 *eat* の語彙項目を使用し、@ を具体的な項と置き換えると、最終的な意味表示は以下のようになる。

$$\mathbf{escape}(Z_2, @_1 : \mathbf{animate}(Z_2))$$

このとき、$Z_2 \equiv \pi_1 \left[\begin{array}{l} z : \mathbf{e} \\ \mathbf{eat}(z, @_2 : \mathbf{animate}(z))(o, @_3 : \mathbf{food}(o)) \end{array} \right]$

ここで、動詞 *escaped* の主語に対する選択制限 $@_1 : \mathbf{animate}(Z_2)$ がかかるが、Z_2 は **animate** の性質をもつため矛盾しない。さらに、動詞 *eat* の目的語の選択制限 $@_3 : \mathbf{food}(o)$ についても、*The omelet* が **food** の性質をもつため矛盾は生じない。よって文 *The omelet escaped.* は *The man who ate omelet escaped.* の意味に解釈できるようになる。

5.3 論理的多義の copredication 構文

memorized the book は他動詞用の argument operator（図 4）を用いると図 5 のように合成される。argument operator を適用したことにより、動詞 *memorized* の引数が *the book* を表す項 b から項 z へシフトされ、意味変換が生じる。このとき図 2 における関数 *book*1 を用いることで関係 R は以下のように構成されうる。

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

この具体的な関係 R を用いて @ を埋めることで文 *Mary memorized the book.* は最終的に以下の意味表示となる。

$$\mathbf{memorize}(m, @_1 : \mathbf{animate}(m))(Z_1, @_2 : \left[\begin{array}{l} w : \mathbf{e} \\ \mathbf{InfoOf}(Z_1, w) \end{array} \right])$$

$$\text{このとき、} Z_1 \equiv \pi_1 \left[\begin{array}{l} z : \mathbf{entity} \\ \left[\begin{array}{l} u : \mathbf{book}(b) \\ \pi_1(book1(b, u)) =_{\mathbf{e}} z \end{array} \right] \end{array} \right]$$

ここで、動詞 *memorized* の目的語に対する選択制限 $@_2 : \left[\begin{array}{l} w : \mathbf{e} \\ \mathbf{InfoOf}(Z_1, w) \end{array} \right]$ がかかるが、Z_1 は性質 **InfoOf** の第一引数であるため矛盾しない。したがって *Mary memorized the infomation of the book.* の意味に合成することが可能である。

Mary burned the book. についても同様にして、動詞 *burned* の目的語に対する選択制限に違反しない関係 R を以下のように構成することができる。

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

よって、copredication を含む文 *Mary memorized and burned the book.* は、述語と項の間の意味変換が *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。

6. まとめと今後の課題

本稿では、DTS の枠組みのもとで、述語の選択制限を前提として扱う分析を提案した。また、述語の選択制限の単純な分析で問題となった 2 つの現象に対して、前研究 [11] で提案した意味変換を行う演算子に改良を加えることで統一的な分析を与えると同時に、意味合成の不足が起こりうるという問題点を解決した。

今後は、現在の argument operator の定義で意味合成の過剰生成・不足が生じることがないか様々な例文で検討し、それに基づき改良を続けていく必要がある。

参考文献

[1] Nicholas Asher. *Lexical Meaning in Context: A Web of Words.* Cambridge University Press, 2011.

[2] Nicholas Asher. Selectional restrictions, types and categories. *Journal of Applied Logic*, Vol. 12, No. 1, pp. 75–87, 2014.

[3] Nicholas Asher and Zhaohui Luo. Formalisation of coercions in lexical semantics. In *Sinn und Bedeutung*, Vol. 17, pp. 63–80, 2012.

[4] Daisuke Bekki. Representing anaphora with dependent types. In *Logical Aspects of Computational Linguistics*, pp. 14–29. Springer, 2014.

[5] Zhaohui Luo. Formal semantics in modern type theories with coercive subtyping. *Linguistics and Philosophy*, Vol. 35, No. 6, pp. 491–513, 2012.

[6] Ofra Magidor. *Categiry Mistakes.* Oxford University Press, 2013.

[7] Per Martin-Löf. *Intuitionistic type theory.* Bibliopolis, 1984.

[8] Geoffrey Nunberg. Transfers of meaning. *Journal of Semantics*, Vol. 12, No. 2, pp. 109–132, 1995.

[9] Mark Steedman. *Surface Structure and Interpretation.* The MIT Press, 1996.

[10] 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.

[11] 木下恵梨子, 中村絢子, 峯島宏次, 戸次大介. 依存型意味論とオントロジーを用いた論理的多義とコアーション現象の分析にむけて. 言語処理学会第 22 回年次大会発表論文集, pp. 429–432, 2016.