The 26th Annual Conference of the Japanese Society for Artificial Intelligence, 2012

4K1-OS-2-2

高階依存型理論を用いた自然言語の意味論構築に向けて

Toward semantics of natural language with higher-order dependent type theory

中野悠紀[*1] 戸次大介[*2]
Yuki Nakano Daisuke Bekki

[*1*2]お茶の水女子大学大学院 人間文化創成科学研究科 理学専攻 情報科学コース
Ochanomizu University,Graduate School of Humanities and Sciences, Faculty of Science,Department of Information Science

In recent years, the study of higher-order dependent type theory has been applied to ATP (Automated Theorem Proving) and programming languages. In application to natural languages, such as in Sundholm (1986) and Ranta (1994), various extensions have been made. In this paper, we point out that there are some problems left that can be solved by higher-order dependent type theory in semantics of natural language, and propose a technique to extend CG (Categorial Grammar) with higher-order dependent type theory.

1. はじめに

近年、プログラミング言語への応用やATP(Automated Theorem Proving) の構築など様々な高階依存型理論の研究が行われているが、自然言語への応用についてはSundholm(1986) やRanta(1994) など様々な拡張が行われている。しかしながら、自然言語の意味論については解決しなければならない問題が残っており、そのいくつかは高階依存型理論によって解決しうる問題である。本研究は高階依存型理論を用いたCG(Categorial Grammar) を提案し、それらの問題を解決することを試みる。

2. CG の論理的解釈

2.1 CG について

Categorial Grammar (CG) とは自然言語を理論的に記述する方法のひとつである。統語範疇を用いることで語彙項目を表現できる。たとえば、固有名詞 “John” の統語範疇は NP となり、自動詞 “runs” は $NP \to S$ となる。“John runs” という文は文法的に正しい文であるが、それは次のように文法的に正しい文であるということが証明できるからである。

$$\dfrac{\underset{NP}{\text{John}} \qquad \underset{NP \to S}{\text{runs}}}{S}$$

このように S が導出されれば、その文は文法的である。

2.2 統語素性

統語素性とは各統語範疇がそれぞれ異なる統語素性を持っており、たとえば統語範疇 NP は統語素性として「格」「人称・数」などを持つ。そしてそれぞれの値として「格」は「$nom+acc$」を、「人称・数」の場合は「3&s」などを持つ。たとえば、“John” が統語素性三人称単数主格を持つならば、その統語素性の値は「(3&s)&nom」となる。

2.3 提案

統語素性を持たない文法では、ほとんどの文は非文にならない。たとえば、動詞 “runs” の主語は統語素性「(3&s)&nom」を持つ NP でなければならないが、統語素性を考えなければ、統語素性「(1&s)&nom」を持つ “I” や統語素性「(3&s)&acc」を持つ “Him” などが “runs” の主語になることができ、本来非文であるはずの “I runs” や “Him runs” という文が文法的となってしまう。

しかし、CG と論理学や代数との対応に関する研究において、統語素性まで含めた CG が扱われたことはない。そこで、本研究では依存型を用いて統語素性を表現する手法を提案する。

3. Martin-Löf 型理論

初めに Martin-Löf の型理論スタイルで書かれた述語計算を導入する。

集合の判定
A が集合であるという判定を A : set と表し、A が集合ならば a が A の要素であることを a : A、a と b が A の等しい要素であることを $a = b$: A と表す。

関数の判定
A から B への関数を導入する判定を $f(x)$: $B(x : A)$ と表す。この判定は x : A、つまり変数 x が集合 A の要素であることを仮定しているので、仮定の判定と呼ぶ。

命題の判定
A が命題であるという判定は A : $prop$ と表す。また、A が集合で、仮定の判定に依存する命題関数は $B(x)$: $prop(x : A)$ と表す。

命題の主張
A は命題であり、真であるという判定を $A\ true$ と表す。
たとえば、“John smokes” が真であるときだけ “John stops smoking” が命題となることをこれらの判定を用いて表すと、$(John\ stops\ smoking)$: $prop((John\ smokes)\ true)$ となる。

Martin-Löf 型理論 (Martin-Löf 1975、1984) は依存型を持つ型理論である。

$$\begin{cases} \text{A:set} & (\text{A は集合である}) \\ \text{a:A} & (\text{a は A の要素である}) \\ \text{a=b:A} & (\text{a と b が A の等しい要素である}) \\ \text{A=B:set} & (\text{A と B は等しい集合である}) \end{cases}$$

命題 ($prop$) と集合 (set) を区別しなければ、上記のような 4 つの判定を持ち、各型はそれぞれ、形成規則 (formation rule)、導入規則 (introduction rule)、除去規則 (elimination rule)、

連絡先: 中野悠紀，お茶の水女子大学大学院人間創成科学研究科理学専攻情報科学コース戸次研究室，東京都文京区大塚 2-1-1，nakano.yuki@is.ocha.ac.jp

等号規則 (equality rule) と呼ばれる 4 組の推論規則を持つ。Martin-Löf 型理論の推論規則は次のようになる。*1

Martin-löf 型理論 推論規則

Cartesian product:

$$\frac{A : set \quad B : set}{A \times B : set} \times F \qquad \frac{a : A \quad b : B}{(a,b) : A \times B} \times I$$

$$\frac{c : A \times B}{p(c) : A} \times E \qquad \frac{c : A \times B}{q(c) : B} \times E$$

$$\frac{c : A \times B \qquad \begin{array}{c}(x : A, y : B)\\ d(x,y) : C((x,y))\end{array}}{E(c,(x,y)d(x,y)) : C(c)} \times E$$

ただし

$$\begin{array}{lllll} p(c) & = & E(c,(x,y)x) : A & for & c : A \times B \\ q(c) & = & E(c,(x,y)y) : B & for & c : A \times B \end{array}$$

$(\Sigma x : A)B(x)$:

$$\frac{A : set \qquad \begin{array}{c}(x : A)\\ B(x) : set\end{array}}{(\Sigma x : A)B(x) : set} \Sigma F \qquad \frac{a : A \quad b : B(a)}{(a,b) : (\Sigma x : A)B(x)} \Sigma I$$

$$\frac{c : (\Sigma x : A)B(x)}{p(c) : A} \Sigma E \qquad \frac{c : (\Sigma x : A)B(x)}{q(c) : B(p(c))} \Sigma E$$

$$\frac{c : (\Sigma x : A)B(x) \qquad \begin{array}{c}(x : A, y : B(x))\\ d(x,y) : C((x,y))\end{array}}{E(c,(x,y)d(x,y)) : C(c)} \Sigma E$$

ただし

$$\begin{array}{lllll} p(c) & = & E(c,(x,y)x) : A & for & c : (\Sigma x : A)B(x) \\ q(c) & = & E(c,(x,y)y) : B(p(c)) & for & c : (\Sigma x : A)B(x) \end{array}$$

$(\Pi x : A)B(x)$:

$$\frac{A : set \qquad \begin{array}{c}(x : A)\\ B(x) : set\end{array}}{(\Pi x : A)B(x) : set} \Pi F \qquad \frac{\begin{array}{c}(x : A)\\ b(x) : B(x)\end{array}}{(\lambda x)b(x) : (\Pi x : A)B(x)} \Pi I$$

$$\frac{c : (\Pi x : A)B(x) \quad a : A}{ap(c,a) : B(a)} \Pi E$$

$A + B$:

$$\frac{A : set \quad B : set}{A + B : set} + F$$

$$\frac{a : A}{i(a) : A + B} + I \qquad \frac{b : B}{j(b) : A + B} + I$$

$$\frac{c : A + B \qquad \begin{array}{c}(x : A)\\ d(x) : C(i(x))\end{array} \qquad \begin{array}{c}(y : B)\\ e(y) : C(j(y))\end{array}}{D(c,(x)d(x),(y)e(y)) : C(c)} + E$$

Absurdity:

$$\frac{}{\perp : set} \perp F \qquad \frac{c : \perp}{case_{\emptyset}(c) : C(c)} \perp E$$

Negation:

$\sim A = A \supset \perp : prop \ for \ A : prop$

Martin-Löf 型理論を用いて論理記号を解釈すると次のようになる。

$$\begin{array}{lllll} A\&B & = & (\Sigma x : A)B : prop & for & A : prop, B : prop \\ (\exists x : A)B(x) & = & (\Sigma x : A)B(x) : prop & for & A : set, B(x) : prop\ (x : A) \\ A \supset B & = & (\Pi x : A)B : prop & for & A : prop, B : prop \\ (\forall x : A)B(x) & = & (\Pi x : A)B(x) : prop & for & A : set, B(x) : prop\ (x : A) \\ A \vee B & = & A + B : prop & for & A : prop, B : prop \end{array}$$

*1 等号規則 (equality rule) は省略している。

3.1 問題点

依存型を用いて統語素性を表現するには 2 つの問題点がある。

問題 1

例えば、Montague 流の “every” の語彙項目は CG（から方向性を省略した体系) においては以下のようになる。

$\vdash \lambda Q.\lambda P.\forall x(P(x) \rightarrow Q(x)) : (NP \rightarrow S) \rightarrow ((NP \rightarrow S) \rightarrow S)$

これは Martin-Löf の体系では以下のように表される。

$(\lambda Q)(\lambda P)((\Pi x : NP)((\Pi y : P(x))Q(x))) : (\Pi Q : (\Pi x : NP)S)((\Pi P : (\Pi x : NP)S)S)$

$NP : set$ とすると、“every” の導出は次のようになる。

$$\cfrac{\cfrac{\boldsymbol{(\Pi x : NP)((\Pi y : P(x))Q(x)) : S}}{(\lambda P)((\Pi x : NP)((\Pi y : P(x))Q(x))) : ((\Pi P : (\Pi x : NP)S)S)} \Pi I}{(\lambda Q)(\lambda P)((\Pi x : NP)((\Pi y : P(x))Q(x))) : (\Pi Q : (\Pi x : NP)S)((\Pi P : (\Pi x : NP)S)S)} \Pi I$$

一番上の部分で $S \stackrel{\mathrm{def}}{\equiv} set$ でなければ ΠF 規則が適用できないので、$S \stackrel{\mathrm{def}}{\equiv} set$ と仮定すると証明図 1 のように導出できる。

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\overset{3}{P : (\Pi x : NP)set} \quad \overset{1}{x : NP}}{P(x) : set} \Pi E \quad \cfrac{\overset{2}{Q : (\Pi x : NP)set} \quad \overset{1}{x : NP}}{Q(x) : set} \Pi E}{(\Pi y : P(x))Q(x) : set} \Pi F \quad NP : set}{\boldsymbol{(\Pi x : NP)((\Pi y : P(x))Q(x)) : set}} \Pi F, 1}{(\lambda P)((\Pi x : NP)((\Pi y : P(x))Q(x))) : ((\Pi P : (\Pi x : NP)set)set)} \Pi I, 2}{(\lambda Q)(\lambda P)((\Pi x : NP)((\Pi y : P(x))Q(x))) : (\Pi Q : (\Pi x : NP)set)((\Pi P : (\Pi x : NP)set)set)} \Pi I, 3$$

証明図 1:Martin-Löf を用いた “every” の導出

しかし、$S \stackrel{\mathrm{def}}{\equiv} set$ と仮定して$(\Pi x : NP)S : set$を導出すると、

$$\frac{NP : set \qquad \begin{array}{c}(x : NP)\\ \vdots \\ \boldsymbol{S : set}\end{array}}{(\Pi x : NP)S : set} \Pi F, 1$$

となる必要がある。ここで、$\stackrel{\mathrm{def}}{\equiv} set$ なので $set : set$ という判定が成り立つ必要があるが、これは Martin-Löf 型理論において妥当な判定ではない。

問題 2

また、たとえば自動詞 “runs” が主語に三人称単数かつ主格の名詞句しか取らないことを示したいとするならば、“runs” から主語が「$(3\&s)\&nom$」であることを指定できなければならない。つまり “runs” が (意味表示) 側に「$(3\&s)\&nom$」という情報を持ち、「$John, runs \vdash run(John) : S$」等は証明可能で「$I, runs \vdash run(I) : S$」等は証明できない体系を考える必要がある。また、この 3 や s や nom は統語範疇 NP の語彙項目のみが持つ統語素性なので統語範疇 NP に依存しているはずである。したがって、2 つ目の問題点として “runs” 側から「$(3\&s)\&nom$」を指定でき、3 や s や nom が NP に依存しているような依存関係を “runs” の型の側で表現する方法を必要とする。

4. λ-cube を用いた解決手法

4.1 λ -cube

様々な型理論のうちある種のものは λ-cube (Barendregt(1992)) によって上手く分類することができる.

λ-cube とは、8 つの型理論 ($\lambda \rightarrow, \lambda 2, \lambda\underline{\omega}, \lambda\omega, \lambda P, \lambda P2, \lambda P\underline{\omega}, \lambda C = \lambda P\omega$) の総称であり、図 1 のような立方体で表される。

$$
\dfrac{\dfrac{\begin{matrix}\vdash NP:\Box \quad \vdash S:\Box\\ \vdots\end{matrix} \qquad \dfrac{\begin{matrix}\vdash NP:\Box \quad \vdash S:\Box\\ \vdots\\ \pi 1\end{matrix} \qquad \begin{matrix}\vdash NP:\Box \quad \vdash S:\Box\\ \vdots\end{matrix}}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)x:NP\vdash(\Pi y:P(x).Q(x)):S}\,(S,S)}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)\vdash(\Pi x:NP.(\Pi y:P(x).Q(x))):S}\,(\Box,S) \qquad \begin{matrix}\vdash NP:\Box \quad \vdash S:\Box\\ \vdots\end{matrix}}{Q:(\Pi\alpha:NP.S)\vdash(\lambda P:(\Pi\alpha:NP.S).(\Pi x:NP.(\Pi y:P(x).Q(x)))) \ : \ ((\Pi P:(\Pi\alpha:NP.S).S))}\,ABS
$$

$$
\dfrac{\cdots \qquad \begin{matrix}\vdash NP:\Box \quad \vdash S:\Box\\ \vdots\end{matrix}}{\vdash(\lambda Q:(\Pi\alpha:NP.S).(\lambda P:(\Pi\alpha:NP.S).(\Pi x:NP.(\Pi y:P(x).Q(x))))) \ : \ (\Pi Q:(\Pi\alpha:NP.S).(\Pi P:(\Pi\alpha:NP.S).S))}\,ABS
$$

証明図 2:λ-cube を用いた語彙項目 “every” の導出

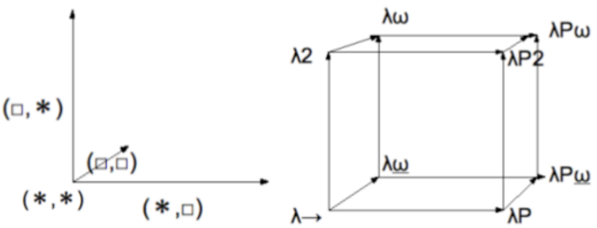

図 1:λ-cube

λ-cube の項 A は次のように定める。

定義 4.1.1 (λ-cube の項)

$$A ::= x|*|\Box|A_1,A_2|\lambda x:A_1.A_2|\Pi x:A_1.A_2$$

ここで、x は変数、$*,\Box$ はそれぞれ定数記号であり、$*$ は型全体の集まりである種を表す定数記号であり、$\Box$ は種全体の集まりを表す定数記号である。

また λ-cube の文脈と判定は次のように定める。

定義 4.1.2 (λ-cube の文脈)

$$x_1:A_1, x_2:A_2, ..., x_n:A_n$$

(ただし $n \geq 0, 1 \leq i \leq n$ において x_i は変数、A_i は型である)

定義 4.1.3 (λ-cube 判定)

$$\Gamma \vdash t:A$$

(ただし Γ は文脈,t は項,A は型である)

メタ変数 s, s_1, s_2 が $*,\Box$ 上を動くとし、Γ は文脈を、A, B, B', C は型を、f, a, b は項を、x は変数を表すとすると、λ-cube の推論規則は次のようになる。

λ-cube の推論規則

axiom $\langle\rangle \vdash A:s$

s(start rule) $\dfrac{\Gamma\vdash A:s}{\Gamma,x:A\vdash x:A}\,s$

w(weakening rule) $\dfrac{\Gamma\vdash A:B \quad \Gamma\vdash C:s}{\Gamma,x:C\vdash A:B}\,w, x\notin\Gamma$

APP(application rule) $\dfrac{\Gamma\vdash f:(\Pi x:A.B) \quad \Gamma\vdash a:A}{\Gamma\vdash fa:B[x:=a]}\,APP$

ABS(abstraction rule) $\dfrac{\Gamma,x:A\vdash b:B \quad \Gamma\vdash(\Pi x:A.B):s}{\Gamma\vdash(\lambda x:A.b):(\Pi x:A.B)}\,ABS$

CON(conversion rule) $\dfrac{\Gamma\vdash A:B \quad \Gamma\vdash B':s \quad B=_\beta B'}{\Gamma\vdash A:B'}\,CON$

識別規則 (s_1,s_2)rule $\dfrac{\Gamma\vdash A:s_1 \quad \Gamma,x:A\vdash B:s_2}{\Gamma\vdash(\Pi x:A.B):s_2}\,(s_1,s_2)$

型、項、判定と推論規則の大部分は 8 つのシステムで共通であり、システムの違いは識別規則である (s_1, s_2) 規則において (s_1, s_2) に現れうる型や種にある。s_1, s_2 は $*,\Box$ 上を動く。8 つのシステムでそれぞれ (s_1, s_2) は表 1 のようになる。このうち λP は Martin-Löf 型理論に対応する。

System	(s_1,s_2)			
$\lambda\rightarrow$	$(*,*)$			
$\lambda 2$	$(*,*)$	$(\Box,*)$		
λP	$(*,*)$	$(*,\Box)$		
$\lambda P2$	$(*,*)$	$(\Box,*)$	$(*,\Box)$	
$\lambda\underline{\omega}$	$(*,*)$			$(\Box,\Box)$
$\lambda\omega$	$(*,*)$	$(\Box,*)$		$(\Box,\Box)$
$\lambda P\underline{\omega}$	$(*,*)$		$(*,\Box)$	$(\Box,\Box)$
$\lambda P\omega=\lambda C$	$(*,*)$	$(\Box,*)$	$(*,\Box)$	$(\Box,\Box)$

表 1:識別規則の対応表

4.2 問題 1 の解決手法

“every” の語彙項目を λ-cube の体系で考えると、

$$\vdash(\lambda Q:(\Pi\alpha:NP.S).(\lambda P:(\Pi\alpha:NP.S).(\Pi x:NP.(\Pi y:P(x).Q(x)))))$$
$$: \ (\Pi Q:(\Pi\alpha:NP.S).(\Pi P:(\Pi\alpha:NP.S).S))$$

となる。“every” の導出の過程で、次の $\pi 1$ のような証明木が現れるので、

$$
\dfrac{\begin{matrix}\vdash NP:\Box \quad \vdash S:\Box\\ \vdots\end{matrix} \qquad \dfrac{\begin{matrix}\vdash NP:\Box \quad \vdash S:\Box\\ \vdots\end{matrix} \qquad \boldsymbol{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)\vdash P:(\Pi\alpha:NP.*)}}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S),x:NP\vdash P:(\Pi\alpha:NP.*)}\,w}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S),x:NP\vdash P(x):*}\,APP
$$

$\pi 1$

$$Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)\vdash P:(\Pi\alpha:NP.*)$$

を示すには $S \overset{\text{def}}{\equiv} *$ でなければならない。$S \overset{\text{def}}{\equiv} *$ と仮定し、(s_1, s_2) 規則に $(S, S), (\Box, S), (\Box, \Box)$ を使うことで、証明図 2 のように語彙項目 “every” を導出できる。

考察

この導出においては $S \overset{\text{def}}{\equiv} *$ であり、$\vdash NP:\Box$ と $\vdash S:\Box$ を公理に追加する必要がある。また、Martin-Löf は λP なので表 1 より (S,S),$(S,\Box)$ しか使うことができないため、$set:set$ という判定が必要という結論になった。この導出で使う (s_1, s_2) 規則は $(S, S), (\Box, S), (\Box, \Box)$ であるので、表 1 より CCG の計算には $\lambda\omega$ が必要ということになる。つまり、[2] における Martin-Löf の定式化では CG の統語素性を表現することはできない。

また、ここまでは統語範疇 S と NP のみを考えていたが、他の統語範疇も踏まえるならば、統語範疇の数だけ $(S,\Box)$ $(NP,\Box)$ $(N,\Box)$ $(CONJ,\Box)$... というような規則が必要になると考えられる。$\lambda\omega$ では $(s_1,\Box)$ の s_1 には $*$ か $\Box$ しか入らなかったが、s_1 に複数の種類 (S や NP など) の種か $\Box$ が現れるシステムが必要であり、したがって $\lambda\omega$ を拡張する必要がある。

$$\dfrac{\vdash (\lambda r : (\Sigma z : NP.(\Sigma y : (\Sigma x : 3.s).nom)).run(p(r))) : (\Pi r : (\Sigma z : NP.(\Sigma y : (\Sigma x : 3.s).nom)).S) \quad \vdash \langle John, \langle\langle a, b\rangle, c\rangle\rangle : (\Sigma z : NP.(\Sigma y : (\Sigma x : 3.s).nom))}{\vdash run(p(\langle John, \langle\langle a, b\rangle, c\rangle\rangle)) : S}\ APP$$

証明図 3:$run(p(\langle John, \langle\langle a, b\rangle, c\rangle\rangle)) : S$ の導出

4.3 問題 2 の解決へ向けて

$\lambda\omega$ が複数の種を持つよう新しく拡張したシステムでは、統語範疇 S や統語範疇 NP という種の他に、統語素性の種 F を持つとする。三人称を表す型を 3、単数を表す型を s、主格を表す型を nom とする。この時 “John runs” が文法的となることの導出を考える。つまり、

$$\vdash NP : \Box, \qquad \vdash S : \Box, \qquad \vdash F : \Box$$
$$\vdash John : NP,$$
$$\vdash a : 3, \qquad \vdash b : s, \qquad \vdash c : nom,$$
$$\vdash 3 : (\Pi z : NP.F), \quad \vdash s : (\Pi z : NP.F), \quad \vdash nom : (\Pi z : NP.F)$$

であるとき$run(John) : S$となることを示す。

まず、統語素性の型である$(\Pi z : NP.F)$が種であることは次のように証明できる。

$$\dfrac{\vdash NP : \Box \qquad \dfrac{\vdash F : \Box \quad \vdash NP : \Box}{x : NP \vdash F : \Box}\ w}{\vdash (\Pi z : NP.F) : \Box}\ (\Box, \Box)$$

三人称単数、主格の “John” は次のように導出される。

$$\dfrac{John : NP \qquad \dfrac{\dfrac{a : 3 \quad b : s}{\langle a, b\rangle : (\Sigma x : 3.s)}\ \Sigma I \qquad c : nom}{\langle\langle a, b\rangle, c\rangle : (\Sigma y : (\Sigma x : 3.s).nom)}\ \Sigma I}{\langle John, \langle\langle a, b\rangle, c\rangle\rangle : (\Sigma z : NP.(\Sigma y : (\Sigma x : 3.s).nom))}\ \Sigma I$$

“runs” は、次のように表す。

$$\vdash (\lambda r : (\Sigma z : NP.(\Sigma y : (\Sigma x : 3.s).nom)).run(p(r))) : (\Pi r : (\Sigma z : NP.(\Sigma y : (\Sigma x : 3.s).nom)).S)$$

すると証明図 3 のように$run(p(\langle John, \langle\langle a, b\rangle, c\rangle\rangle)) : S$となる。さらに、

$$\dfrac{\langle John, \langle\langle a, b\rangle, c\rangle\rangle : (\Sigma z : NP.(\Sigma y : (\Sigma x : 3.S).nom))}{p(\langle John, \langle\langle a, b\rangle, c\rangle\rangle) : NP}\ \Sigma E$$

となる。しかし、$p(\langle John, \langle\langle a, b\rangle, c\rangle\rangle) : NP$が$John$と等価であることを示さなければ、$run(John) : S$は証明できない。

5. まとめと今後の課題

本研究では高階依存型理論を用いて CG を拡張する上での問題点を指摘し、λ-cube を用いた解決策を示した。

今後の課題としては、$\lambda\omega$ を拡張したシステムの体系を考案すること、そのシステム上で$p(\langle John, \langle\langle a, b\rangle, c\rangle\rangle) : NP$が$John$と等価であることを示すこと、また “every” の語彙項目を定義し、“Every student runs” のような文が文法的な文として導出されるかなどを検証することが挙げられる。

参考文献

[1] Barendregt,H. 1992, *Lambda calculi with types.* In S. Abramsky, D. M. Gabbay, and T.S.E. Maibaum (Eds.), *Handbook of Logic in Computer Science, Volume 2 Background: Computational Structures.* pp.117–309,Oxford Science Publications.

[2] Ranta,A. 1994, *Type-Theoretical Grammar.* Oxford University Press.

[3] 龍田真, 1992, 「型理論」, 近代科学社.

[4] 戸次大介, 2010, 「日本語文法の形式理論—活用体系・統語構造・意味合成—」, 日本語研究叢書 24, くろしお出版.