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

中野悠紀　(指導教員：戸次大介)

1 はじめに

高階依存型理論の研究は Martin-Löf 型理論にさかのぼるが、自然言語への応用については Ranta(1994) などがあり、その後も様々な拡張が行われている。自然言語の意味論には高階依存型理論によって解決しうる問題がいくつかあるが、本研究は高階依存型理論を用いた CG(Categorial Grammar) を提案する。

2 Martin-Löf 型理論

Martin-Löf 型理論 (Martin-Löf 1975、1984) は依存型を持つ型理論である。命題 ($prop$) と集合 (set) を区別しなければ、次の 4 つの判定を持つ。

$$\begin{cases} \text{A:set} \\ \text{a:A} \\ \text{a=b:A} \\ \text{A=B:set} \end{cases}$$

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)、等号規則 (equality rule) と呼ばれる 4 組の推論規則を持つ。Martin-Löf 型理論を用いて論理記号を解釈すると次のようになる。

$$A\&B = (\Sigma x : A)B : prop \quad \text{for} \quad A : prop, B : prop$$

$$(\exists x : A)B(x) = (\Sigma x : A)B(x) : prop \quad \text{for} \quad A : set, B(x) : prop\ (x : A)$$

$$A \supset B = (\Pi x : A)B : prop \quad \text{for} \quad A : prop, B : prop$$

$$(\forall x : A)B(x) = (\Pi x : A)B(x) : prop \quad \text{for} \quad A : set, B(x) : prop\ (x : A)$$

$$A \vee B = A + B : prop \quad \text{for} \quad A : prop, B : prop$$

3 CG の論理的解釈

3.1 提案

CG(Steedman 1996) において統語素性 (Syntactic feature) は統語範疇をより詳しく指定する上で非常に重要である。統語素性はそれぞれの統語範疇で異なり、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「$x\&s$」(x は人称の値であり、例えば三人称ならば 3) を持つ。例えば、"John" が統語素性に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。また、統語範疇 NP は NP の統語素性しか持たないので、統語範疇 NP と NP の統語素性の間には依存関係があると考えられる。

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。例えば、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。

3.2 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。

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

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。

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

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

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

図 1:Martin-Löf を用いた "every" の導出

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

$$\dfrac{NP : set \quad \begin{matrix}(x : NP)\\ \vdots \\ \boldsymbol{S : set}\end{matrix}}{(\Pi x : NP)S : set}\,\Pi F,1$$

$S \stackrel{\text{def}}{\equiv} set$ 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、という問題がある。

問題 2

また、"runs" が主語に三人称単数かつ主格しか取らないことを示したいとするならば、"runs" は「$(3\&s)\&nom$」という統語素性に依存しているので、"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" の型 (意味表示) 側に「$(3\&s)\&nom$」という情報を持ち、「$John, runs \vdash run(John) : S$」は証明可能で「$I, runs \vdash run(I) : S$」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 Barendregt の λ -cube

λ-cube とは、8 つの型理論 ($\lambda \to, \lambda 2, \lambda\underline{\omega}, \lambda\omega, \lambda P, \lambda P2, \lambda P\underline{\omega}, \lambda C = \lambda P\omega$) の総称であり、図 2 の様な立方体で表される。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、システムの違いは次の (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。

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

s_1, s_2 は $*$,□ 上を動き、$*$ とは型全体の集まりである種を表し、□ とは種全体の集まりを表す。

8 つのシステムでそれぞれ (s_1, s_2) は図 3 のようにな

$$
\cfrac{\vdash NP:\Box \quad \vdash S:\Box \qquad \cfrac{\cfrac{\cfrac{\vdash NP:\Box \quad \vdash S:\Box \;\vdots\; \pi1 \qquad \vdash NP:\Box \quad \vdash S:\Box}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)x:NP\vdash(\Pi y:P(x).Q(x)):S}\;(S,S)}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)\vdash(\Pi x:NP.(\Pi y:P(x).Q(x))):S}\;(\Box,S) \qquad \vdash NP:\Box \quad \vdash S:\Box}{Q:(\Pi\alpha:NP.S)\vdash(\lambda P:(\Pi\alpha:NP.S).(\Pi x:NP.(\Pi y:P(x).Q(x)))) \;:\; ((\Pi P:(\Pi\alpha:NP.S).S))}\;ABS \qquad \vdash NP:\Box \quad \vdash S:\Box}{\vdash(\lambda Q:(\Pi\alpha:NP.S).(\lambda P:(\Pi\alpha:NP.S).(\Pi x:NP.(\Pi y:P(x).Q(x))))) \;:\; (\Pi Q:(\Pi\alpha:NP.S).(\Pi P:(\Pi\alpha:NP.S).S))}\;ABS
$$

図 4:λ-cube を用いた語彙項目 “every” の導出

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

図 5:$run(p(\langle John,\langle\langle a,b\rangle,c\rangle\rangle)):S$ の導出

る。このうち λP は Martin-Löf 型理論に対応する。

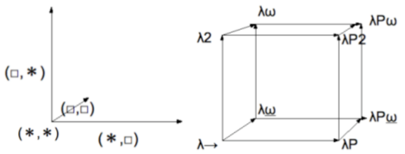

図 2:λ-cube

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

図 3:各システムにおける (s_1, s_2) 規則

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 $\pi1$ のような証明木が現れるので、

$$
\cfrac{\vdash NP:\Box \quad \vdash S:\Box \;\vdots \qquad \cfrac{\vdash NP:\Box \quad \vdash S:\Box \;\vdots \quad Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)\vdash P:(\Pi\alpha:NP.*)}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S),x:NP\vdash P:(\Pi\alpha:NP.*)}\;w}{Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S),x:NP\vdash P(x):*}\;APP
$$

$\pi1$

$Q:(\Pi\alpha:NP.S),P:(\Pi\alpha:NP.S)\vdash P:(\Pi\alpha:NP.*)$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、図 4 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 axiom に追加する必要がある。また、Martin-Löf は λP なので図 2 より (S,S),$(S,\Box)$ しか使うことができないので、$set : set$ 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、図 2 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 Categorial Grammar 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,□) (NP,□) (N,□) ($CONJ$,□)... 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,□) の s_1 には $*$ か □ 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 □ 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。

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、

$\vdash NP:\Box, \vdash S:\Box, \vdash F:\Box, \vdash John:NP, \vdash a:3, \vdash b:s, \vdash c:nom$

$\vdash 3:(\Pi z:NP.F), \vdash s:(\Pi z:NP.F), \vdash nom:(\Pi z:NP.F)$

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。

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

三人称単数、主格の “John” は次のように導出される。

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

“runs” は、次のように表す。

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

すると図 5 のように$run(p(\langle John,\langle\langle a,b\rangle,c\rangle\rangle)):S$となる。さらに、

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

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 Categorial Grammar 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” を定め、“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.