3I1-R-9-6

部分方向性組み合わせ論理における日本語のかき混ぜ文の正規化制約

Normal Forms for Japanese Scrambling in Subdirectional Combinatory Logic

尾崎博子[*1] 戸次大介[*2]
Hiroko Ozaki 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 natural language processing with Combinatory Categorial Grammar (CCG), we have a problem called "spurious ambiguity" in which many derivations may appear for one semantic representation. To solve this, we can use normal form parsing which is an efficient and sound technique for parsing. Subdirectional Combinatory Logic (SDCL) was proposed by [Bekki 10] to establish a relation between CCG and logical systems. SDCL is an extension of Combinatory Logic (CL) that distinguishes two directions of functional applications. We can describe various languages and their phenomena by means of subsets of the directed combinators. The aim of this paper is to propose a method to apply normal form parsing for Japanese scrambling in terms of SDCL.

1. はじめに

部分方向性組み合わせ論理（SDCL）[Bekki 10] は組み合わせ論理（CL）の一種で、CL におけるコンビネータそれぞれに対し「方向性」に関する複数の変種が存在する。SDCL は各自然言語に対応する組み合わせ範疇文法（CCG）[Steedman 00] をクラスとして持ち、言語間の違いを方向性のあるいくつかのコンビネータの有無によって表すことができると考えられている。

CCG を用いた構文解析においては、簡単な文に対してさえ複数の構文木が導出されることがある。しかし処理効率の観点からは 1 つの意味に対しては導出を一意に決定したい。この上で正規化解析は構文解析のための効率的で完全性のある手法である。これは範疇文法が論理の証明論と対応しており、論理学の手法を応用できることの一つの利点であると考えられる。

しかし、各言語について独自のコンビネータの集合があるとすると、各コンビネータに対し正規形を考えなければいけない。現時点では文法一般についての考察だけではなく個別言語についての事例を見ていくことも重要である。

そこで本研究では、日本語のかき混ぜ文（scrambling）について [Bekki 10] が採用するコンビネータ **C** を用いた分析に対して、正規化解析の手法を適用する方法を提案する。

2. 部分方向性組み合わせ論理

SDCL では、組み合わせ論理の関数適用の 2 つの方向性を区別するため、Lambek 計算と同様に 2 種類の含意（/ と \）が存在する。たとえば、組み合わせ論理におけるコンビネータ $\mathbf{K} : A \to (B \to A)$ には以下の 4 種類が対応する。

$$\mathbf{K}_{//} : (A/B)/A \quad \mathbf{K}_{\backslash/} : (A\backslash B)/A$$
$$\mathbf{K}_{/\backslash} : (A/B)\backslash A \quad \mathbf{K}_{\backslash\backslash} : (A\backslash B)\backslash A$$

(他のコンビネータについても同様)

連絡先: 尾崎博子，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，ozaki.hiroko@is.ocha.ac.jp

このようにコンビネータの型に方向性を持たせることで、自然言語の文における語順を記述できるように組み合わせ論理を拡張した論理を部分方向性組み合わせ論理という。

このシステムの正当性については [尾崎・戸次 11] にて 1) Subject-reduction、2) 合流性、3) 停止性、等の計算論的性質を証明済みである。

3. 構文、型規則、略記法、簡約規則

基本型の集合、コンビネータの集合が与えられたとき、部分方向性組み合わせ論理の構文、型規則、略記法、簡約規則を以下のように定める。

構文 型 $\tau ::= \gamma \mid \tau/\tau \mid \tau\backslash\tau$ (γ は基本型)
項 $\Lambda ::= x \mid c \mid \Lambda^{\triangleright}\Lambda \mid \Lambda^{\triangleleft}\Lambda$ (c はコンビネータ)

型規則
$$(>)\frac{\Gamma \vdash M : A/B \qquad \Delta \vdash N : B}{\Gamma, \Delta \vdash M^{\triangleright}N : A}$$
$$(<)\frac{\Delta \vdash N : B \qquad \Gamma \vdash M : A\backslash B}{\Delta, \Gamma \vdash M^{\triangleleft}N : A}$$

略記法
$$\tau\diagup\!\!\diagdown\sigma \stackrel{def}{\equiv} \tau/\sigma \text{ または } \tau\backslash\sigma$$
$$M^{\Phi}N \stackrel{def}{\equiv} M^{\triangleright}N \text{ または } M^{\triangleleft}N$$
$$(<>) \stackrel{def}{\equiv} (<) \text{ または } (>)$$

簡約規則
$$\mathbf{B}^{\Phi}_{/} f^{\triangleright} g^{\triangleright} x \underset{CL}{\to} f^{\triangleright}(g^{\triangleright}x)$$
$$\mathbf{B}^{\Phi}_{\backslash} f^{\triangleleft} g^{\triangleleft} x \underset{CL}{\to} f^{\triangleleft}(g^{\triangleleft}x)$$
$$\mathbf{S}^{\Phi}_{/} f^{\triangleright} g^{\triangleright} x \underset{CL}{\to} (f^{\triangleleft}x)^{\triangleright}(g^{\triangleright}x)$$
$$\mathbf{S}^{\Phi}_{\backslash} f^{\triangleleft} g^{\triangleleft} x \underset{CL}{\to} (f^{\triangleright}x)^{\triangleleft}(g^{\triangleleft}x)$$
$$\mathbf{K}^{\Phi}_{/} x^{\triangleright} y \underset{CL}{\to} x$$
$$\mathbf{K}^{\Phi}_{\backslash} x^{\triangleleft} y \underset{CL}{\to} x$$

4. 関連研究

CCG のように、関数合成規則を持つ範疇文法では、文の単語数に対して指数的に構文解析結果が出る可能性がある。1 つの意味に対して複数の異なる構文解析結果が生じる、という spurious ambiguity の問題である。例えば、 "John likes

Mary.” という文には以下のような複数の異なる構文解析結果が得られる。

(1) John likes Mary

1. $\dfrac{\dfrac{\dfrac{\text{John}}{T/(T\backslash NP)}\ \dfrac{\text{likes}}{S\backslash NP/NP}}{S/NP}(>B)\ \ \dfrac{\text{Mary.}}{NP}}{S}(>)$

2. $\dfrac{\dfrac{\text{John}}{T/(T\backslash NP)}\ \dfrac{\dfrac{\text{likes}}{S\backslash NP/NP}\ \dfrac{\text{Mary.}}{NP}}{S\backslash NP}(>)}{S}(>)$

この例のように、全ての構文解析結果を計算することは非効率的である。そのため、一つの意味に対して唯一の解析結果（正規形）を見つけることが求められる。正規形を見つけることができれば、他の余分な結果を随時枝刈りしていくことができ、文の単語数が増加しても処理効率を上げることができると考えられる。そのため、効率的かつ正確に正規化解析を実行する手段を取り入れる必要がある。

[Eisner 96] では関数合成と関数適用規則に対し、正規化制約を定め、唯一の正規形を見つけることに成功した。また、安全性や完全性についても証明されている。しかし、型繰り上げ規則の適用は、辞書内に限られていた。

[Hockenmaier 10] では [Eisner 96] では示されていなかった一般合成規則と型繰り上げ規則についても対処できるように正規化制約の拡張を行い、その効果についても検証されている。

5. CCG における日本語のかき混ぜ文

標準的な語順は言語によって異なる。たとえば英語では SVO が標準的な語順である。日本語では SOV が標準的な語順だが、ある程度自由に語順を変更できる「かき混ぜ操作」があることが知られている。かき混ぜ文は以下のような日本語特有の組み合わせ規則により得られると仮定する。

$$\dfrac{X\backslash Y\backslash Z : f}{X\backslash Z\backslash Y : \mathbf{C}f(= \lambda y.\lambda z.fzy)}(<\mathbf{C})$$

かき混ぜ文の例としては以下のようなものがある。

(2) a. 太郎が 花子に 次郎を 紹介した。

$$\dfrac{\dfrac{\text{太郎が}}{NP_{ga}}\ \dfrac{\dfrac{\text{花子に}}{NP_{ni}}\ \dfrac{\dfrac{\text{次郎を}}{NP_o}\ \dfrac{\text{紹介した。}}{S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o}(<)}{S\backslash NP_{ga}\backslash NP_{ni}}(<)}{S\backslash NP_{ga}}(<)}{S}(<)$$

b. 太郎が 次郎を 花子に 紹介した。

$$\dfrac{\dfrac{\text{太郎が}}{NP_{ga}}\ \dfrac{\dfrac{\text{次郎を}}{NP_o}\ \dfrac{\dfrac{\text{花子に}}{NP_{ni}}\ \dfrac{\text{紹介した。}}{S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o}(<\mathbf{C})}{S\backslash NP_{ga}\backslash NP_o}(<)}{S\backslash NP_{ga}}(<)}{S}(<)$$

c. 次郎を 太郎が 花子に 紹介した。

$$\dfrac{\dfrac{\text{次郎を}}{NP_o}\ \dfrac{\dfrac{\text{太郎が}}{NP_{ga}}\ \dfrac{\dfrac{\text{花子に}}{NP_{ni}}\ \dfrac{\text{紹介した}}{S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o}(<\mathbf{C})}{S\backslash NP_{ga}\backslash NP_o}(<)}{S\backslash NP_o\backslash NP_{ga}}(<\mathbf{C})}{S}(<)$$

この文において動詞 “紹介した” の統語範疇は $S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o$ であり、標準的な語順は “NP-ga NP-ni NP-o Verb” (2a) である。しかし、“NP-ga NP-o NP-ni Verb” (2b) や “NP-o NP-ga NP-ni Verb” (2c) のように語順が変化した例でも意味的に等価で許容可能な文を得ることができる。

6. 提案手法

[Bekki 10] において、かき混ぜ操作のコンビネータ $\mathbf{C}$ は日本語のために導入され、以下のように定義される:

$$\begin{array}{ll}\mathbf{C}_{/} & A/C/B\backslash(A/B/C) : \lambda P.\lambda y.\lambda x.\lambda z.Pzy\\ \mathbf{C}_{\backslash} & A\backslash C\backslash B/(A\backslash B\backslash C) : \lambda P.\lambda y.\lambda z.Pzy\end{array}$$

しかし、この分析には処理の観点からは大きな問題がある。

まず、コンビネータ $\mathbf{C}$ を以下のように繰り返し適用すると、無限に解析結果が生じてしまう、という問題がある。そのため、コンビネータ $\mathbf{C}$ を 2 回連続して適用している場合は正規形ではない：

(3)

$$\dfrac{\dfrac{\dfrac{\text{太郎が}}{NP_{ga}}\ \dfrac{\text{花子に}}{NP_{ni}}\ \dfrac{\text{次郎を}}{NP_o}\ \dfrac{\dfrac{\text{紹介した。}}{S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o : \lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)}}{S\backslash NP_{ga}\backslash NP_o\backslash NP_{ni} : \mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))}(<\mathbf{C})}{S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o : \mathbf{C}(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)))}(<\mathbf{C})$$

しかし、単に連続するコンビネータ $\mathbf{C}$ を除去するだけでは十分でない。たとえば、 “昨日” のような副詞を 2 つのコンビネータ $\mathbf{C}$ の間に挟んだ以下のような形も正規形ではない：

(4)

$$\dfrac{\dfrac{\dfrac{\text{太郎が}}{NP_{ga}}\ \dfrac{\text{次郎を}}{NP_o}\ \dfrac{\text{花子に}}{NP_{ni}}\ \dfrac{\dfrac{\text{昨日}}{T/T : \lambda P.\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e)\wedge Pyzxe}\ \dfrac{\dfrac{\text{紹介した。}}{S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o : \lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)}}{S\backslash NP_{ga}\backslash NP_o\backslash NP_{ni} : \mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))}(<\mathbf{C})}{S\backslash NP_{ga}\backslash NP_o\backslash NP_{ni} : (\lambda P.\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e)\wedge Pyzxe)(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)))}(>)}{S\backslash NP_{ga}\backslash NP_o\backslash NP_{ni} : \mathbf{C}((\lambda P.\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e)\wedge Pyzxe)(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))))}(<\mathbf{C})$$

ところが、これは単純に 2 つのコンビネータ $\mathbf{C}$ が一つの節に現れてはいけない、ということでない。1 つ以上の引数が 2 つのコンビネータ $\mathbf{C}$ の間に入っている場合は、正規形になるのである。以下にその例を示す：

(5)

$$\dfrac{\dfrac{\dfrac{\text{次郎を}}{NP_o}\ \dfrac{\text{太郎が}}{NP_{ga}}\ \dfrac{\dfrac{\text{花子に}}{NP_{ni} : \lambda P.P(h)}\ \dfrac{\dfrac{\text{紹介した。}}{S\backslash NP_{ga}\backslash NP_{ni}\backslash NP_o : \lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)}}{S\backslash NP_{ga}\backslash NP_o\backslash NP_{ni} : \mathbf{C}\backslash\ \lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)}(<\mathbf{C})}{S\backslash NP_{ga}\backslash NP_o : (\lambda P.P(h))(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)))}(<)}{S\backslash NP_o\backslash NP_{ga} : \mathbf{C}((\lambda P.P(h))(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))))}(<\mathbf{C})$$

この問題を解決するために、以下の手法を提案する。

$\mathbf{C}$ の正規形

$\mathbf{C}(X(\mathbf{C}Y))$ という形の意味表示が XY と等価（$\alpha\beta$-同値）なら $\mathbf{C}(X(\mathbf{C}Y))$ は正規形ではない、とする

先に示した例文を用いてこの手法を検証する。

(3)

$\mathbf{C}(X(\mathbf{C}Y))$
$= \quad \mathbf{C}(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)))$
$\rightarrow \quad \mathbf{C}(\lambda y.\lambda z.(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))zy)$
$\rightarrow \quad \mathbf{C}(\lambda y.\lambda z.\lambda x.\lambda e.introduce(e,x,y,z))$
$\rightarrow \quad \lambda y.\lambda z.(\lambda y.\lambda z.\lambda x.\lambda e.introduce(e,x,y,z))zy$
$\rightarrow \quad \lambda y.\lambda z.\lambda x.\lambda e.introduce(e,x,z,y)$
$= \quad \lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)$

XY
$= \quad \lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)$

(4)

$\mathbf{C}(X(\mathbf{C}Y))$
$= \quad \mathbf{C}((\lambda P.\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge Pyzxe)(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z)))$
$\rightarrow \quad \mathbf{C}((\lambda P.\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge Pyzxe)(\lambda y.\lambda z.(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))zy))$
$\rightarrow \quad \mathbf{C}((\lambda P.\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge Pyzxe)(\lambda y.\lambda z.\lambda x.\lambda e.introduce(e,x,y,z)))$
$\rightarrow \quad \mathbf{C}(\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge introduce(e,x,y,z))$
$\rightarrow \quad \lambda y.\lambda z.(\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge introduce(e,x,y,z))zy$
$\rightarrow \quad \lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge introduce(e,x,z,y)$

XY
$= \quad (\lambda P.\lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge Pyzxe)(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))$
$\rightarrow \quad \lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge (\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))yzxe$
$\rightarrow \quad \lambda y.\lambda z.\lambda x.\lambda e.yesterday(e) \wedge introduce(e,x,z,y)$

(5)

$\mathbf{C}(X(\mathbf{C}Y))$
$= \quad \mathbf{C}((\lambda P.P(h))(\mathbf{C}(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))))$
$\rightarrow \quad \mathbf{C}((\lambda P.P(h))(\lambda y.\lambda z.(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))zy))$
$\rightarrow \quad \mathbf{C}((\lambda P.P(h))(\lambda y.\lambda z.\lambda x.\lambda e.introduce(e,x,y,z)))$
$\rightarrow \quad \mathbf{C}((\lambda y.\lambda z.\lambda x.\lambda e.introduce(e,x,y,z))(h))$
$\rightarrow \quad \mathbf{C}(\lambda z.\lambda x.\lambda e.introduce(e,x,h,z))$
$\rightarrow \quad \lambda y.\lambda z.(\lambda z.\lambda x.\lambda e.introduce(e,x,h,z))zy$
$\rightarrow \quad \lambda y.\lambda z.\lambda e.introduce(e,y,h,z)$

XY
$= \quad (\lambda P.P(h))(\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))$
$\rightarrow \quad (\lambda z.\lambda y.\lambda x.\lambda e.introduce(e,x,y,z))(h)$
$\rightarrow \quad \lambda y.\lambda x.\lambda e.introduce(e,x,y,h)$

このように、(3) と (4) において、意味表示を比較すると、$\mathbf{C}(X(\mathbf{C}Y))$ と XY が一致し、これは余分な導出であることが分かる。よってこれらを枝刈りすればよいことが分かる。一方、(4) においては $\mathbf{C}(X(\mathbf{C}Y))$ と XY は一致せず、$\mathbf{C}(X(\mathbf{C}Y))$ が正規形であることが分かり、いずれも望ましい結果である。

7. まとめと今後の課題

部分方向性組み合わせ論理（SDCL) は組み合わせ論理（CL）の各コンビネータに対し、「方向性」の異なるいくつかのコンビネータが存在し、その選択により個々の自然言語や言語現象を記述することができる論理体系である。本研究では、日本語のかき混ぜ文を導出する $\mathbf{C}$ コンビネータに関して正規形を提案した。

構文解析において、最初の問題はコンビネータ $\mathbf{C}$ が無限に繰り返し適用可能であることであった。しかし連続するコンビネータ $\mathbf{C}$ を除去するだけでは、2つのコンビネータ $\mathbf{C}$ の間に副詞を挟んだ場合にも意味的に等価な導出が現れてしまうため、この方法では十分ではない。また、1つの節内に2つのコンビネータ $\mathbf{C}$ が現れてはいけないという制約では、正規形であるものも除去してしまう。そこで、$\mathbf{C}(X(\mathbf{C}Y))$ の意味表示が XY の意味表示と $\alpha\beta$-同値である場合に枝刈りを行う、という手法を提案した。

今後の課題としては、SDCL を用いた解析器の実装、および日本語の他の言語現象についての分析を考えている。

参考文献

[Bekki 10] Bekki, D.: Combinatory Categorial Grammar as a Substructural Logic — Preliminary Remarks —, in *the Proceedings of LENLS 7, JSAI International Symposia on AI 2010*, pp. 70–83 (2010)

[Eisner 96] Eisner, J.: Efficient Normal-Form Parsing for Combiantory Categorial Grammar, in *ACL '96 Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pp. 79–86 (1996)

[Hockenmaier 10] Hockenmaier, J. and Yonatan, B.: Normal-form parsing for Combinatory Categorial Grammars with generalized composition and type-raising, in *the Proceeding of COLING 2010*, pp. 465–473 (2010)

[Steedman 00] Steedman, M. J.: *The Syntactic Process (Language, Speech, and Communication)*, The MIT Press (2000)

[尾崎・戸次 11] 尾崎博子, 戸次大介. 2011. 「部分方向性組み合わせ論理の計算論的性質とその証明」, Technical Report of Department of Information Science, Ochanomizu University, OCHA-IS 10-2, February 7th, 2011