自然言語処理 113－9
（1996．5．29）

範疇単一化文法を日本語へ適用する際の付加的な規則について

関洋平　飯島正　原田賢一
yohei@hara.cs.keio.ac.jp
慶應義塾大学理工学部

本論文は，Hans Uszkoreitの提案である範疇単一化文法を日本語に対して適用を行う際，日本語の品詞に対して設定する範疇と素性構造の提案および，実際にシステムを試作し，本提案の有効性を検証したことについての報告である。範疇単一化文法は範疇文法を単一化文法として定式化を行った文法枠組であり，語意情報主導のボトムアップな構文解析を行う。また，複合的な素性構造を用いることで，構文解析の枠組で意味解析の一部を取り扱えるようになり，基本概念の美しさから実現に比較的向いているという利点もある。この枠組を日本語に適用する際には，語順の自由度などの日本語の普遍的な特質を取り扱う必要がある。一般的には，複数の範疇を設定することで対応を行うが，本提案においては，前後の単語の情報を利用して範疇を変化させる付加的な規則を併用することで，辞書引きまでの無駄なバックトラック操作を削減することを試みている。

Additional rules in applying Categorial Unification Grammar to Japanese.

Yohei SEKI, Tadashi IIJIMA and Ken'ichi HARADA
Faculty of Science and Technology, Keio University

This paper describes an approach of applying Categorial Unification Grammar (CUG) which is the proposal of Hans Uszkoreit to Japanese. CUG is a grammatical framework that encodes categorial grammar principles in a complex-feature-based form. It enables a syntax mechanism to deal with a kind of semantic marker, and a conceptual elegance of it provides simple structure to implement. It is implemented using bottom-up parsing technique suitable for combinatory potential of lexical information. We propose a set of categories that corresponds to Japanese part of speech. In order to build Japanese CUG framework, it is necessary to realize Japanese universal characteristics, for example word order variation. In this proposal, we attempt to reduce backtrack operations until looking up a word from a dictionary by using additional top-down rules, instead of setting two or more categories in general. A prototype system is implemented in prolog, and a verification of effectiveness of this approach is also reported in this paper.

1 はじめに

本研究の目的は構文・意味レベルにおいて日本語を理解する計算機上のシステムを構築することにある。このような方向で近年取り組まれている文法の枠組の一つとして, 単一化に基づく文法 (略して**単一化文法**) と呼ばれる一連の文法形式がある。単一化文法は, 属性と属性値の組合せをリスト化したデータ構造である**素性構造**を用いることにより, 単語の接続情報および意味情報を単語に対応させて表現を行っている。すなわち, 文脈自由文法における句構造規則のようなルールおよび, 非終端記号に対応した属性と属性を計算する意味規則 (DCG における補強項に相当) などを, できるだけ辞書に埋め込む形で取り扱うことを目標としている文法枠組であるといえる。

本研究では, 係り受け関係の取扱いに**範疇文法** [1] の考え方を取り入れた単一化文法の枠組の一つであり, 日本語に対する適用例のあまりない**範疇単一化文法** [8](CUG) を日本語に対して適用することで, その有効性を検証する。範疇単一化文法は, 範疇文法を基本概念としていることから, より辞書情報に重点をおいたボトムアップな構文解析を行うことが可能になっている。同じ方向性をもった提案としては, 主辞駆動句構造文法 [6](HPSG) が有名である。 CUG は, HPSG が空所を取り扱うために slash 素性などの非局所素性を設定するというかたちで複数のスタックを用いるのに対して, 係り受け関係の解析に 1 つのスタックしか設定しない点や, 主辞が補語として取る引数を 1 つしか設定しない点などから, 計算機上で実現する上で, より洗練された枠組であるといえるが, 距離のある関係代名詞の係り受け解析などに柔軟性を欠くという問題点も指摘されている [5]。

本論文の構成は以下の通りである。第 2 章で範疇文法と範疇単一化文法について簡単に説明を行う。第 3 章では日本語の品詞に対応した範疇を提案し, 第 4 章は, 第 3 章の提案によりどのような文法現象が取り扱えるか具体的に示す。第 5 章で, 付加的な規則を導入することについて説明を行い, 第 6 章でまとめと今後の課題を述べる。

2 範疇文法と範疇単一化文法

範疇文法では Joachim Lambek の提案による Lambek Calculus[4] の適用規則を基礎としている。適用規則をシーケント・カリキュラスの形式で書く [3] と,

/ 除去規則; $\frac{\Delta \Rightarrow B/A \quad \Gamma \Rightarrow A}{\Delta,\Gamma \Rightarrow B}$

\ 除去規則; $\frac{\Gamma \Rightarrow A \quad \Delta \Rightarrow A\backslash B}{\Gamma,\Delta \Rightarrow B}$

のようになる。構文構造などの文法情報は, 2 ～ 3 個の基本範疇 (名詞, 文など品詞の基本単位が設定される) を /,\ を用いて組み合わせた導出範疇に, 語彙のテンプレート (品詞) を対応させて表現を行う。形態素分割された単語に対応した語彙範疇 (基本範疇または導出範疇) に対して, 上記の除去規則を適用することでボトムアップな構文解析が行われる。範疇文法自体は, 実は文脈自由文法の別表記にすぎず, したがって文の生成能力も文脈自由文法と同等である。

範疇文法と単一化文法を融合させる最初の試みとして範疇単一化文法 (CUG) は提案された。具体的な融合の方法は, トップレベルの素性として value (値), direction(方向), argument(引数) を与えることで, /, \ を素性構造に埋め込むことにより実現している。これは, 範疇文法を単一化文法として定式化したものであるといえる ([8] では PATR 形式の dag 表現を用いて表されている)。以下に, 具体例として, 名詞「太郎」と後置詞「に」の素性構造を示す。基本範疇に対応させて素性構造を用いることで, 付加的な意味分類情報による格役割の特定や, 意味素性の伝搬などが取り扱えるようになる。

太郎 (n)

値	:	範疇	: n(名詞)
		分類	: 生物
		意味	: 太郎

に ($n\backslash pp_{\text{生物対象格}}$)

値	:	方向	:	'\'	
		引数	:	値	: 範疇 : n
					分類 : 生物
					意味 : X
		値	:	範疇	: pp(後置詞句)
				格	: に
				役割	: 生物対象格
				意味	: X

このような考え方に基づく融合の例としては, 他に, 単一化範疇文法 (UCG), 単一化に基づくコンビネータ範疇文法 (Unification-based CCG) がある。CUG はこのような提案を総称して用いられることもある。これとは逆に, 範疇文法の考えを基礎として, 素性構造を基本範疇に対応させた例としては, Type Logical Grammar や Typed Applicative Grammar[3] がある。後者の枠組では, プログラミング言語における基本データ型の概念を基本範疇に, 関数データ型の概念を導出範疇と対応させて捉えることで, 構文解析のステップを型推論とみなしており, 型に対応する意味関数の計算は別のステップで行われる。

3 日本語の範疇

日本語の基本範疇としては, 名詞 (N), 文 (S), 後置詞句 (PP) の三つを設けることにする。その三つを /, \ を用いて組み合わせることで, 以下のような品詞を表現することができる。

	品詞分類	範疇
文	文	S
動詞	動詞 (格三つ, あげる)	PP\(PP\(PP\S))
	動詞 (格二つ, 会う)	PP\(PP\S)
	動詞 (格一つ, 歩く)	PP\S
助詞	必須格助詞 (が, を, に)	N\PP
	係助詞 (は)	N\(S/(PP\S)),N\(S/S)
	任意格助詞 (時間格に, 場所格で)	N\((PP\(PP\(PP\S))) /(PP\(PP\(PP\S))))
	終助詞 (ね, か)	S\S
	接続助詞 (が)	S\(S/S)
助動詞	時制助動詞 (た, ている)	(PP\(PP\(PP\S))) \ (PP\(PP\(PP\S)))
	使役助動詞 (せる, させる)	(PP\(PP\S)) \ (PP\(PP\(PP\S)))
	受動助動詞 (れる, られる)	(PP\(PP\S)) \(PP\(PP\S))
	断定助動詞 (です, だ, である)	N\(PP\S)
	名詞	N
名詞	連体詞 (この, その, あの)	N/N
	接尾辞 (時, 君)	N\N
	時間を表す名詞 (昨日,3 時, 夕方)	(PP\(PP\(PP\S))) /(PP\(PP\(PP\S)))
その他	接続詞	S/S
	副詞	(PP\(PP\(PP\S))) /(PP\(PP\(PP\S)))
	感動詞	S

この他に用言については語幹と活用語尾を分離し, 語幹には基本範疇として V を与え, 活用の種類を接続情報の素性として持たせるようにしている。具体的には, 形容動詞語幹「嫌い」に範疇 $PP\backslash V_{\text{形容動詞}}$ を, 活用語尾「な」に範疇 $(PP\backslash V_{\text{形容動詞}})\backslash(N/N)$ を, 断定助動詞「だ」に範疇 $V\backslash(PP\backslash S)$ を与えることにより, 「太郎はあの子が嫌いだ」「太郎が嫌いな子がいる」というように, 形容動詞の語幹「嫌い」に「な」や「だ」が接続することで, どの格役割を持つ品詞がどの位置に来るかという区別ができるようになる。このような分離を行うさい, 態を変化させるような助動詞の取扱いについて第 4 章で説明を行う。

また, 同様にサ変名詞「愛」に活用語尾「している」が接続すると, 「愛する主体」と「愛される対象」が格後置詞句として堀起こされる。「している」が「勉強」に接続した場合は, 格後置詞句は「勉強する主体」の一つの役割しか堀起こさない。この区別は, サ変名詞に格数素性を持たせて, 活用語尾「している」との間で共起関係を調べることにより行っている。

4 日本語に対する解析例

本節では, CUG による日本語の文法現象への対応の具体例を示す。CUG の枠組においては, 態を変化させるような助動詞や, 距離をおいて係り受けを行うとりたて詞のような品詞に対して, 特殊な範疇を与えることにより対応することができる。また, 本研究で採用している範疇文法は, 引数を前後どちらに取るかを区別しているため, 係り受けの非交差制約を自然な形で表現できる。

1. 助動詞「せ」「させ」による使役化

助動詞「せ」「させ」には, 動詞に接続して文の述部の意味を「その行為の強制」に変化させ, 構文的には動詞がとる後置詞句に, 「強制者」の役割をもつ格後置詞句を一つ追加する。例えば, 「花子が料理を食べる」という文の動詞語幹「食べ」に接続して「太郎が花子に「花子が料理を食べる」という行為を強制する」というように意味を変化させる。これを範疇によって表すと, 以下のようになる。

- 「させ」の範疇

$$(PP_{sem_x}\backslash(PP_{nom,が,sem_y}\backslash V_{sem_z}))\backslash(PP_{sem_x}\backslash(PP_{acc,に,sem_y}\backslash(PP_{nom,が,sem_w}\backslash V_{force_{agt:w,pas:y,obj:z}})))$$

上記の範疇を与えて, 試作したシステムで解析した結果を以下に示す。

```
| ?- start([太郎, が, 花子, に, 料理, を, 食べ, させ, る,. ]).
解析結果:
 構文木: [[[太郎, が],[[花子, に],[[料理, を],[[食べ, させ], る]]]],.]
 素性構造: [範疇:ss, 意味:[関係:causative, 動作主格:taro, 生物対象格:hanako, 内容:[関係:eat, 動作主格:hanako, 無生物対象格:cooking]]]
yes
```

2. 助動詞「れ」「られ」による受動化

助動詞「れ」「られ」は, 「太郎が花子に料理を食べさせ」というような文章に接続して, 行為主体者と行為受動者の意味役割を入れ替えるという機能をもつ。範疇によって表すと, 以下のようになる。

- 「られ」の範疇

$$(PP_{acc,sem_x}\backslash(PP_{nom,が,sem_w}\backslash V_{下一段動詞,sem_{force_{agt:w,pas:x,obj:[eat:[agt:x,obj:y]]}}}))\backslash(PP_{acc,sem_w}\backslash(PP_{nom,が,sem_x}\backslash V_{下一段動詞,sem_{force_{agt:w,pas:x,obj:[eat:[agt:x,obj:y]]}}})))$$

前出の例文に「られ」を接続して解析した結果を以下に示す。

```
| ?- start([太郎, が, 花子, に, 料理, を, 食べ, させ, られ, る],. ]).
解析結果:
 構文木: [[[太郎, が],[[花子, に],[[[[料理, を],[食べ, させ]], られ], る]]],. ]
 素性構造: [範疇:ss, 意味:[関係:causative, 動作主格:hanako, 生物対象格:taro, 内容:[関係:eat, 動作主格:taro, 無生物対象格:cooking]]]
yes
```

ただし, 被害の受身(間接受身)のような役割を果たす「れ」「られ」は, 使役化の「せ」「させ」と同じような範疇を与える。追加される格後置詞句の役割は「被害者」となる。

3. 助詞「は」による主題化

以下の二つの例文による「は」の役割の違いを扱うために, 以下の範疇を「は」に対して与える。

- 例文

  太郎**は**あの子**を**愛している。

  太郎**は**あの子**が**愛している。

- 「は」の範疇

  N \ (S/(PP \ S))

  ※ N と PP は **情報を共有**

これにより, 上の例文の「は」が, 動詞句「あの子を愛している」において欠落している主格の役割を果たしており, 下の例文の「は」は, 動詞句「あの子が愛している」において欠落している目的格の役割を果たしていることが扱えるようになる。試作システムにおける解析結果は以下のようになった。

```
| ?- start([太郎, は, あの, 子, を, 愛, してい, る,. ]).
解析結果:
 構文木: [[[太郎, は],[[[あの, 子], を],[[愛, してい], る]]],.]
 素性構造: [範疇:ss, 意味:[関係:love, 様相:進行, 動作主格:taro, 生物対象格:[child, 属性:[型:融合, 距離:遠]]]]
yes
| ?- start([太郎, は, あの, 子, が, 愛, してい, る,. ]).
解析結果:
 構文木: [[[太郎, は],[[[あの, 子], が],[[愛, してい], る]]],.]
 素性構造: [範疇:ss, 意味:[関係:love, 様相:進行, 動作主格:[child, 属性:[型:融合, 距離:遠]], 生物対象格:taro]]
yes
```

同じような範疇を与える品詞としては, 「係助詞」「とりたて詞」と呼ばれるもの以外に,

アクセントのおかれた「が」(総記の役割を果たすと思われる) などがある。ここで示した範疇は, 名詞を引数として取った後は, 関数と引数の方向を逆転させることで, 情報の伝搬を行いやすくしている。すなわち, タイプレージング [4] を静的に行わせているともいえる。「が」の場合は, N\PP と N\(S/(PP\S)) の両方の範疇を持つので, タイプレージングルールを動的に適用する方が, 効率が良いとも思われる。このように動詞が求める格役割を補完するような場合は, 上記の範疇を用いるが, そうでない場合 (「時間」や「場所」を主題化している場合) は, 上記の品詞に対して N\(S/S) という範疇を与える。

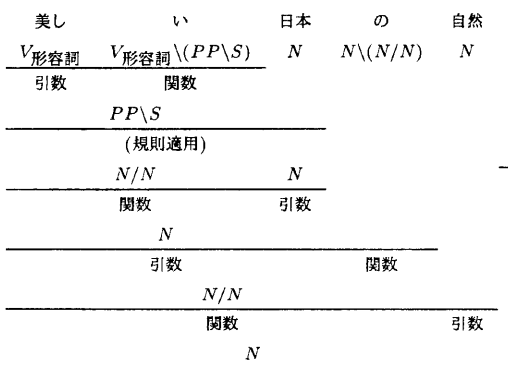
4. 係り受けの非交差制約

以下では, 「美しい日本の自然」と「日本の美しい自然」という二つの文に対して, 本研究で設定した範疇を用いた解析結果を示すことで, 係り受けの非交差制約現象へ対応できることを紹介する。

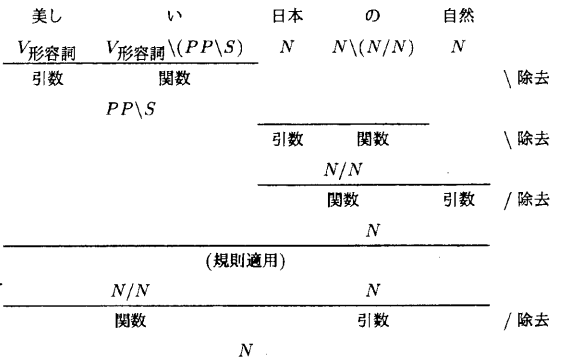
(a) 美しい日本の自然

「美しい日本の自然」という文の意味は2通り考えられる。一つは, 「美しい」が「日本」を修飾する場合, もう一つは「日本の自然」を修飾する場合である。本研究で設計した範疇を用いて構文解析を行った例を以下に示す。

解析結果 (a.1)
「美しい」が「日本」を修飾する場合



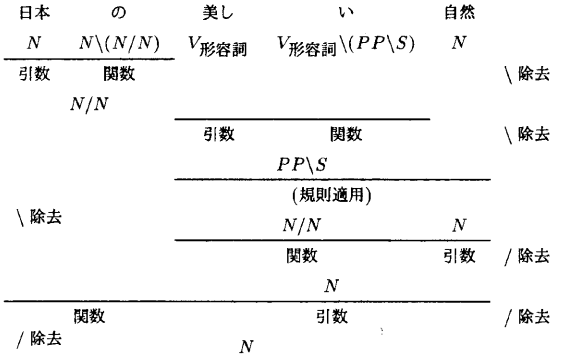
解析結果 (a.2)
「美しい」が「日本の自然」を修飾する場合



以上に示したように, 「美しい」が「日本」あるいは「日本の自然」を修飾することを自然な形で解析できる。本研究で試作しているシステムは, 複数の解析結果がある場合, バックトラックして全ての解析結果を検出する。

(b) 日本の美しい自然

「日本の美しい自然」における「美しい」は, 「日本」を修飾することができない。これは, 日本語が前から後ろへ向けて修飾する言語であることによる。本研究で提案した範疇で「日本の美しい自然」を解析すると, 以下のようになり, この説明にあった解析を行っていることが理解できる。



5 付加的な規則の導入

日本語の単語に対して範疇を設定する際には, 一つの単語に対して複数の範疇を設定する場合もある。ある日本語の品詞に対して, 共通の複数の範疇を設定する必要がある場合, 全て辞書に登録し

ていては辞書が繁雑になるし，構文解析を実行する上で，辞書引きの段階までバックトラックする手間がかかる。本研究では，このことを避けるため，いくつかの付加的な規則を設けている。

付加的な規則とは，引数としてとる範疇の情報によって関数の範疇を変化させるトップダウンルールのことであり，範疇文法の特徴であるより語彙情報主導な枠組を求めるという観点からは逆行している。しかし，上記の利点からいくつかの現象に絞って採用することにする。

範疇をプログラミング言語におけるデータ型として捉えると，複数の範疇を設定することは，型多相 (Type Polymophism) の考え方に対応しており，引数の型によって関数の役割が変化するような付加的な規則は，汎用関数 (generic function) の役割を果たしているともいえる。

1. 動詞と形容詞による連体修飾

   日本語の動詞と形容詞は連体形と終止形が同じ活用をもつという特徴がある。これにより格後置詞句が欠落した文が，後方に名詞を補って連体修飾を行う，ということが可能になっている。この現象を取り扱うために，以下の規則を導入する。

   (a) $PP\backslash S \quad N \rightarrow N/N \quad N$

   ※PP $\backslash S$ の PP と $N/$ N の N は **情報を共有**

   これにより「私は望遠鏡を持つ少年をみる。」のような文章が与えられた際，望遠鏡を持っているのは「私」ではなくて「少年」であることが取り扱えるようになる。試作システムにおける解析結果は以下のようになった。

```
| ?- start([私, は, 望遠鏡, を, 持, つ, 少年, を, 見る,。]).
解析結果:
 構文木: [[[私, は],[[[[[望遠鏡, を],[持, つ]], 少年], を],[見, る]]],。]
 素性構造: [範疇:ss, 意味:[関係:see, 動作主格:i, 生物対象格:[boy, 属性:[関係:have, 動作主格:boy, 無生物対象格:telescope]]]]
yes
```

2. 語順の自由度

   語順の自由度を取り扱う方法としては，引数の集合を取ることが出来るように枠組を拡張するという方法がある ([2][7] など)。この考え方は，HPSG においても応用されている。しかし，これは範疇文法の根本的な利点を損なう感触も受ける。ここでは規則により取り扱う方法について検証を行う。以下の 7 つの規則を導入することにより，格後置詞句の入れ替えが可能になる。

   • 必須格後置詞句 (が, に, を格) の入替え規則

   (a) $PP_{が} \; PP_{を}\backslash(PP_{が}\backslash S) \rightarrow PP_{が} \; PP_{が}\backslash(PP_{を}\backslash S)$

   (b) $PP_{が} \; PP_{に}\backslash(PP_{が}\backslash S) \rightarrow PP_{が} \; PP_{が}\backslash(PP_{に}\backslash S)$

   (c) $PP_{に} \; PP_{を}\backslash(PP_{に}\backslash S) \rightarrow PP_{に} \; PP_{に}\backslash(PP_{を}\backslash S)$

   (d) $PP_{が} \; PP_{を}\backslash(PP_{に}\backslash(PP_{が}\backslash S))$
   $\rightarrow PP_{が} \; PP_{が}\backslash(PP_{を}\backslash(PP_{に}\backslash S))$

   • 試作システムにおける解析結果

```
| ?- start([花子, が, 太郎, に, 会う,。]).
解析結果:
 構文木: [[[花子, が],[[太郎, に],[会, う]]],.]
 素性構造: [範疇:ss, 意味:[関係:meet, 動作主格:hanako, 生物対象格:taro]]
yes
| ?- start([太郎, に, 花子, が, 会う,。]).
解析結果:
 構文木: [[[太郎, に],[[花子, が],[会, う]]],.]
 素性構造: [範疇:ss, 意味:[関係:meet, 動作主格:hanako, 生物対象格:taro]]
yes
```

   • 任意格後置詞句 (時間格に, 場所格で) の入替え規則

(a)		$(PP\backslash(PP\backslash(PP\backslash S)))$ $/(PP\backslash(PP\backslash(PP\backslash S)))$	$PP\backslash(PP\backslash S)$
	$\rightarrow$	$(PP\backslash(PP\backslash S))$ $/(PP\backslash(PP\backslash S))$	$PP\backslash(PP\backslash S)$
(b)		$(PP\backslash(PP\backslash(PP\backslash S)))$ $/(PP\backslash(PP\backslash(PP\backslash S)))$	$PP\backslash$ S
	$\rightarrow$	$(PP\backslash S)/(PP\backslash S)$	$PP\backslash S$
(c)		$(PP\backslash(PP\backslash(PP\backslash S)))$ $/(PP\backslash(PP\backslash(PP\backslash S)))$	S
	$\rightarrow$	S/S	S

   • 試作システムにおける解析結果

```
| ?- start([太郎, が, 花子, に, 三, 時, に, 会う,。]).
解析結果:
```

```
構文木: [[[太郎,が],[[花子,に],[[[三,時],に],[会,
う]]]],.]
素性構造: [範疇:ss,意味:[関係:meet,時間格:
[three,o_clock],動作主格:taro,生物対象格:hanako]]
yes
| ?- start([太郎,が,三,時,に,花子,に,会う,。]).
解析結果:
構文木: [[[太郎,が],[[[三,時],に],[[花子,に],[会,
う]]]],.]
素性構造: [範疇:ss,意味:[関係:meet,時間格:
[three,o_clock],動作主格:taro,生物対象格:hanako]]
yes
```

3. 活用語尾の接続

活用語幹と活用語尾を分離したことで, 両者を接続する際に, 活用語幹が取る格の数に応じて, 活用語尾の範疇を変化させる規則を導入することで, 辞書の繁雑さを避け, 無駄な構文的曖昧さを省くことることができる。以下の規則を導入することで, 格数の変化に応じて活用語尾が接続できるようになる。語幹の格の数の情報は素性に埋め込む。

- 活用語幹の格数に応じた活用語尾の接続規則

(a) $PP\backslash(PP\backslash V$(格数2)) → $PP\backslash(PP\backslash V$(格数2))
$(PP\backslash(PP\backslash(PP\backslash V)))$ $\backslash(PP\backslash(PP\backslash(PP\backslash S)))$
$(PP\backslash(PP\backslash V))$ $\backslash(PP\backslash(PP\backslash S))$

(b) $PP\backslash V$(格数1) → $PP\backslash V$(格数1)
$(PP\backslash(PP\backslash(PP\backslash V)))$ $\backslash(PP\backslash(PP\backslash(PP\backslash S)))$
$(PP\backslash V)\backslash(PP\backslash S)$

4. 遊離数量詞

日本語における数量詞は, 「3キロ走る」のように述語を直接計量する場合と, 「3匹の子豚」「子豚3匹」「子豚が3匹」のように主語または目的語を計量する場合がある。数量詞はさまざまな場所に移動して計量を行うことがあるが, ここでは, 上記4つの係り受けについて, 範疇を動的に変化させる規則を導入して取り扱うことを試みた。規則を適用する際には, 計量可能な単位の情報を数量詞の素性に埋め込むことで, 数量詞が対象となる単語を計量できるかどうか単一化操作を用いてチェックを行う。

- 遊離数量詞の範疇変化規則

(a) N(数量詞) (PP\S)/(PP\S) → PP\S(計量述語) PP\S
3キロ 走る

(b) N(計量対象) N → N(数量詞) N\N(数量句)
子豚 3匹

(c) PP(計量対象) PP → N(数量詞) PP\PP(数量句)
子豚が 3匹

5. 同じ品詞が連続する場合

日本語の助動詞には, 連接の順序に制約が課されていることが指摘されている。実際に, 「させたくなかっただろう」は日本語として正しいが, 「させたくだろうなかった」は誤りである, という例が挙げられる。また, 終助詞 (か・よ・ね) においては「そうかよ」「そうかね」「そうよね」のように言うことはできるが, 「そうよか」「そうねか」「そうねよ」と言うことはできないという現象が見られる。更に, 格助詞の連続について考えてみると, 「からも」とは言えても「もから」とは言えないという制約がある。

単語を組み合わせた際に, 組み合わせる前の元の単語の意味を合計した以上の意味が生じている場合には, 連続する単語の組合せを一単語として登録してしまうことが考えられる。そうでない場合は, 元の品詞と同じ品詞を返すような範疇 (助動詞, 終助詞など) が連続する場合は, 除去操作を適用する際に, 連接制約素性を用いて, チェックすることにより対応する。 動詞や格助詞が連接した場合は, 連接制約素性が接続条件を満たした場合, 片方の範疇がもう片方の範疇を補語として取ることができるように, 高階化するルールを与えてやることで対応を行う。

- 同じ品詞が連続する場合の範疇高階化規則

(a) 動詞の範疇高階化規則 (前方主体)

PP\S (PP\S)/(PP\S) → PP\(PP\S) PP\(PP\S)
燃え 始め

(b) 格助詞の範疇高階化規則 (後方主体)

N\PP N\PP → N\(S/S) (N\PP)\(N\(S/S))
から も

6 おわりに

範疇単一化文法に付加的な規則を併用して用いることで、以下のような日本語の現象に対応できる。

	日本語の文法現象	アルゴリズム	実現
構文的特徴	係り受けの曖昧性	◎	◎
	語順の自由度	◎	△
	入れ子構造と並置構造	◎	△
	同じ品詞の連続	△	×
意味的特徴	遊離数量詞	△	×
	副詞の呼応	◎	△
	「自分」の照応先特定	△	×

対応済み…◎, 一部対応…△, 未対応…×

前章までで言及していない項目がいくつかある。入れ子構造と並置構造は「～が～が」「～に～に」のように共通の助詞を持った後置詞句が連続する文の取扱いについて検討している。副詞の呼応現象は,「少しも～ない」のように距離のある依存関係を持つ現象であるが,「～」の動詞の部分に「ない」が否定のアスペクト素性を追加するような範疇および素性構造を与え,「少しも」が後方に否定のアスペクト素性をもつ範疇を必要とするかたちにすることで対応している。再帰代名詞の「自分」の照応については, 範疇よりも素性の制約をベースにして補完を行っている。

これより, 範疇単一化文法の日本語への有効性を示すことができた。今後の課題としては, 日本語に頻出する格後置詞句などの省略現象への付加的な規則を用いた対応, 複数の文にまたがる依存関係の抽出, 日本語を網羅する大規模な辞書の構築, Lambek Calculus の合成規則や上昇規則を導入することによる, 構文解析アルゴリズムの最適化などが考えられる。

参考文献

[1] Wojciech Buszkowski,Witold Marciszewski and Johan van Benthem: "Categorial Grammar," Linguistic and Literary Studies in Eastern Europee Vol.25, John Benjaminss Pub(1988).

[2] Beryl Hoffman: "A CCG APPROACH TO FREE WORD ORDER LANGUAGES," Proc. of 31th Annual Meeting of ACL(1993), pp.300-302.

[3] 飯島正, 関洋平, 柳原正秀他: "範疇文法のための関数的な計算機構," 情報処理学会研究報告 96-NL-111-6(1996), pp.33-40.

[4] Joachim Lambek: "The Mathematics of sentense structure," American Mathematical Monthly 65(1958), also in [1].

[5] Carl J. Pollard: "Categorial Grammar and Phrase Structure Grammar:An Excursion on the Syntax-Semantics Frontier," in Richard T.Oehrle,Emmon Bach and Deirdre Wheeler: "Categorial Grammars and Natural Language Structures," Dordrecht Reidel Pub(1988), pp.391-415.

[6] Carl Pollard and Ivan A.sag: "Head-Driven Phrase Structure Grammar," The University of Chicago Press,CSLI(1994).

[7] Satoshi Tojo: "Free-ordered CUG on Chemical Abstract Machine," Proc. of COLING 1994(1994), pp.870-874.

[8] Hans Uszkoreit: "Categorial Unification Grammars," Proc. of COLING 1986(1986), pp.187-194.