日本語構文解析のための柔軟な制御機構を持つ文法記述手法

高橋博之　　宮崎正弘

新潟大学大学院自然科学研究科

1 はじめに

従来、日本語の構文解析のための文法記述手法としては、拡張 CFG による方法 [1] と係り受け規則による方法 [2] がある。これらの手法には以下に示す問題点がある。

自然に記述できない言語現象の存在
CFG では離れた単語間の関係を記述する場合に不自然な中間構造を必要とし、係り受け規則では文節構成のための規則を記述できない。

ルール適用の曖昧さの絞り込み機構が貧弱
補強項など、単一のルールの適用可能性を決める制約は記述できても、ルール A よりもルール B は優先して適用するというようなルール間の相互適用制約は記述できない。曖昧さの絞り込みは文法現象に応じて個別の手法があるが、それらを分離して記述することができない。

記述性・可読性の問題
ルールがフラットに記述され、構造を持たないためにルールが多くなると管理しにくくなる。構造の持つ素性が単純なリスト構造などのナイーブなデータ構造で表現されているため、素性を増やしていくと管理しにくくなる。

2 G3

我々は以上の問題を解決するために、一般化されたルール記述と柔軟な適用制御機構を持つ文法記述形式 G3[1] を提案する。G3 は拡張 CFG である DCG [3] をベースに係り受け風の規則を導入するなどの拡張を加えた文法記述手法であり、以下のような特長を持つ。

1. 句構造ルールと依存構造（係り受け）ルールの両方が利用可能
2. ルール適用の曖昧さの柔軟な絞り込み機構を持つ
3. オブジェクト指向プログラミングの手法を利用したメインテナンスの容易なデータ管理手法

[1] Generalized and Governed Grammer の略

3 ルール記述の拡張

G3 では CFG の記述方法を踏襲しつつ、遠隔ルールという記述法を導入し、日本語の係り受けを自然に記述できるようにしている。

3.1 遠隔ルールの導入

遠隔ルールは係り元と係り先の制約のみで、間に任意の要素を挟むことができるルールである。これに対して従来型の CFG 規則は近接ルールと呼ぶ。

遠隔ルールの記述表現としては、以下のように ... という表記を用いる

```
A -> B ... C
```

例えば、日本語の格後置詞句（例：「彼は」）から用言（例：「走る」）への係りは以下のようなルールで記述できる。これは日本語の文法記述として極めて自然である。

```
格 -> 格後置詞句 ... 用言
```

ここでルール左辺項は係り関係を表すオブジェクト[2]である。G3 では係り関係（依存関係）を関係オブジェクトという実体として表現することで、依存構造表現と句構造表現を統合している。

日本語の構文構造の例を図 1 に示す。構造間の包含関係を下向きの矢印で示す。ここでは、下位レベルでは句構造表現であり、上位レベルでは関係オブジェクトを利用した依存構造表現になっている。

3.2 ルールセット

従来のルールシステムの問題点の一つとしてルールが平面的に記述されているため、一群のルールをまとめた管理がしにくい点が挙げられる。たとえば、ある

[2] G3 では単語や句、係り関係など構文構造に関わる実体を一般にオブジェクトと呼ぶ

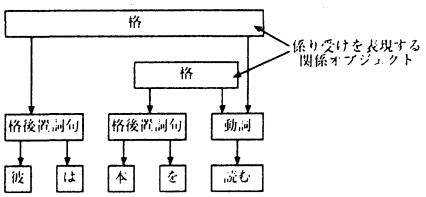


図 1: 構文構造の例

一群のルールは別の一群のルールより優先して適用するというような制約がうまく記述できない。

そこで G3 ではルールセットという概念を導入する。以下にルールセットを用いた記述の例を示す。ここでは文節の生成に関わるルールと、係り受けに関わるルールが別のルールセットに分類されている。

```
ruleset(文節) :: {
  格後置詞句 -> 名詞 格助詞
  格後置詞句 -> 格後置詞句 格助詞
}
ruleset(係り受け) :: {
  格 -> 格後置詞句 ... 用言
}
```

ここで「文節」ルールセットは「係り受け」ルールセットより先に適用されるという制約を与えれば、従来型の文節結合の後で係り受けを処理するという逐次型の解析を行うことができる。

また、ルールセットは入れ子にでき、ルールの階層的管理が可能である。

4 ルールの適用制御

G3 ではルール間の排他適用およびその優先制約はすべて明示的に記述しなくてはならない。明示的に記述されない限り、ルール間の排他適用制御は行われない。

例えば以下のようなルールの場合、CFG では x,y,z という記号列に対して (a) か (b) のどちらかしか適用できない。また、状況に応じてどちらを適用するか選択するということができない。

```
(a) s1 -> x y
(b) s2 -> y z
```

G3 ではこの記述自体には排他性はないものとして、両方同時に適用する[3]。どちらか片方のみ適用したい場合は、後述するルールセットメソッドなどの適用制御機構を利用して、適用ルールの優先制約（あるいは絞り込み手続き）を記述する必要がある。

4.1 無効化

ルール間の適用排他の一番単純なパターンとして、一方が適用できたらもう片方は適用しないというケースがある。

例えば、「学校/へ/は/」のように付属語（助詞）が複数連鎖する場合、以下のようなルールを用いて、中間構造を作りながら構造を作っていく。

```
格後置詞句 -> 名詞 助詞
格後置詞句 -> 格後置詞句 助詞
```

このような場合、中間構造「学校へ」を放置しておくと、「学校へは」と「学校へ」の両方に格係りルールを適用してしまうという問題がある[4]。

このようにあるルールが適用されたらそれ以降のルール適用を排除したい場合はそのオブジェクトを無効化する。無効化は以下のようにルールの要素を [ ] で囲って示す。

```
格後置詞句 -> 名詞 助詞
格後置詞句 -> [格後置詞句] 助詞
```

この無効化により、中間構造の格後置詞句にそれ以上他のルールが適用されることを防ぐことができる。

4.2 順序制御

日本語の文節結合処理と、係り受けのように、複数の処理段階を逐次的に行う場合には、ルールセット間の優先順序を設定することで、特定のルールを優先して適用することができる。

順序制御は以下のように、ルールセット名を優先順に並べて指定する。

```
order(文節, 係り受け)
```

4.3 ルールセットメソッド

ルールの適用可能性をより精密に絞り込むためには最終的には文型パターン情報へのマッチのような手続き的処理が必要になる。このような場合にルールセットメソッドを用いる。ルールセットメソッドの記述例を以下に示す。

[3]遠隔ルールの導入ですでに構文構造が単純な木構造ではなくなっているため、このような適用を可能にしてもパージングアルゴリズムには影響しない

[4]前述のように明示しない限りルールの排他は行われないため、「学校へは」が生成されてもそれだけでは「学校へ」へのルール適用は排除されない

```
ruleset(格) :: {
  格 -> 格後置詞句 ... 用言
  method { 格の係りの絞り込み処理 }
}
```

ここで method { ... } とある部分がルールセットメソッドである。ルールセットメソッドは Prolog で記述される。パーザは該当ルールセットに属するルールの適用可能個所のリストをルールセットメソッドに送る。ルールセットメソッドは適用個所の絞り込みを行い、そのうち実際に適用する個所のリストを返す。

格の係りの場合、ある格後置詞句が複数の用言に係りうる場合に曖昧さの絞り込み処理が必要になる(「A-> B...C」で B が共有不能)。しかし、パーザにはそのような知識はないので、格係りルールの適用可能個所をすべて格ルールセットのルールセットメソッドに送る。ルールセットメソッドは適用の曖昧さがあるかどうかチェックし、あったら絞り込み処理を行う。

ルールセットメソッドが曖昧さを確実に一つに絞り込めない場合は多義を発生させることもできる。これは Prolog のバックトラックを利用して、複数回の成功という形で実装する。

5 オブジェクト指向のデータ管理

構文構造は各種の素性を持つ。素性は解析手法を改良していく段階で後から追加されることがしばしばある。しかし、素性構造を単純なリスト構造などのナイーブな形式で保持し、その全体に対してユニフィケーションを行うような従来の手法では、素性の追加が文法の随所に悪影響を与えてしまう可能性がある。

G3 ではオブジェクト指向プログラミングの手法である「カプセル化」を導入することで、これらの問題に対処する。すなわち、全ての素性は各構造オブジェクトの中に隠蔽され、「メソッド」と呼ばれる手続きを経由してしかアクセスできない。

メソッド記述の例を示す。以下の例はオブジェクト「動詞」の持つメソッド「他動詞」の記述例である。G3 では DCG と同様に、メソッドや補強項などの記述言語として Prolog を使用する。ここで、ME はそのオブジェクト自身である[5]。ME::品詞コード (H) は動詞の持つ述語「品詞コード ()」の呼び出しであり、ここでは自身の品詞コードを取得している。

```
動詞:他動詞 (ME) :- ME::品詞コード (H),
                    H=200.
```

このメソッドは以下のように、ルールの補強項などで使用される。C や V がそれぞれのオブジェクトの実体であり、それぞれのメソッド「格」「他動詞」を呼び出している(「を」格で他動詞ならば適用)。

```
格 -> 格後置詞句 (C) ... 動詞 (V)
        {C::格 (を),V::他動詞}
```

この実装では、他動詞かどうかのチェックとしてその品詞コードが 200 であるかどうかを調べているが、使用する辞書によっては他動詞かどうかが、独立したフラグである可能性もある。実際のデータ構造とその操作がメソッドによって隠蔽されているため、使用する辞書が変更されても、関連するメソッドの変更によって容易に対処できる。

6 G3 による解析例

「彼は本を読んで学ぶ」という文を例に G3 での解析手順を示す。

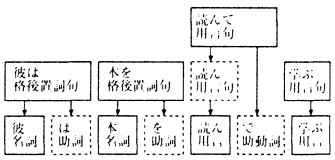
この文は形態素解析の結果、以下のようなオブジェクトの列となる。

日本語での構文解析では通常文節をまとめる処理が先に行われ、次に係り受け処理が行われる。そこでルール全体を大きく「文節」ルールセットと「係り受け」ルールセットに分け、順序制御により「文節」ルールセットを優先して適用する。

「文節」ルールセットの関連するルールを以下に示す。

```
ruleset(文節) :: {
  格後置詞句 -> 名詞 [助詞]
  格後置詞句 -> [格後置詞句] [助詞]
  用言句 -> 用言
  用言句 -> [用言句] [助動詞]
}
```

これらのルールの適用後の構造を以下に示す。中間構造の用言句や、今後の解析に関係しない助詞などは、邪魔にならないように無効化してある(無効化されたオブジェクトは破線で示す)。



次に、係り受け処理が行われる。関連する規則を以下に示す。

[5] C++ での this にあたる

```
ruleset(係り受け) :: {
  ruleset(格) :: {
    格 -> 格後置詞句 ... 用言
    method { 格係りの絞り込み処理 }
  }
  ruleset(節) {
    節 -> 用言句 ... 用言句
    method { 節係りの絞り込み処理 }
  }
  method { 係り受け一般の絞り込み処理 }
}
```

ここでルールが適用可能な個所は、格に関する「彼は... 読む」「彼は... 学ぶ」「本を... 読む」「本を... 学ぶ」と、節に関する「読んで... 学ぶ」の5個所である。

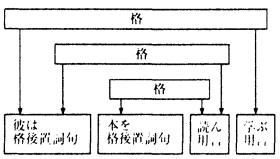
「係り受け」ルールセットのルールセットメソッドでは係り受け一般に適用できる制約を適用する。ここでは係り受けの交差禁止制約を用いる。

「彼は... 読む」と「本を... 学ぶ」は両方適用すると交差禁止制約違反である。しかし、このレベルではどちらを適用したらいいのかはわからないので、多義を発生させる。すなわち、「彼は... 読む」「彼は... 学ぶ」「本を... 読む」「読んで... 学ぶ」(パターン A) と、「彼は... 学ぶ」「本を... 読む」「本を... 学ぶ」「読んで... 学ぶ」(パターン B) の二つである。以下、それぞれの場合に分けて説明する。

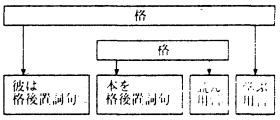
パターン A の場合 パーザはルールの適用可能個所をサブルールセットごとに分類して、それぞれのルールセットのルールセットメソッドに送る。すなわち、「彼は... 読む」「彼は... 学ぶ」「本を... 読む」が「格」ルールセットのルールセットメソッドに送られ、「読んで... 学ぶ」が「節」ルールセットのルールセットメソッドに送られる。

一般に日本語の係り受けでは係り先は一つであるという制約が用いられる。これを適用すれば、「彼は... 読む」と「彼は... 学ぶ」は互いに排他である。しかし、「は」の場合は複数の動詞の主語になることがしばしばある (現にこの文の場合がそうである)。そこで、「は」の係り先は絞り込まずに両方残すという選択肢も考えられる。

その場合の構造は以下のようになる (関連するオブジェクトのみ示している)。



このように G3 では従来のパーザでは不可能な構造出力も可能である。しかし、従来型の単純な木構造を期待しているアプリケーションにはこのような出力結果は適さない。そのような場合には、係り先を一つに絞り込む必要がある。ここでは「は」は遠くに係る傾向があるという経験則などを用いることになる。その場合は「彼は... 学ぶ」が採用され、以下のような構造になる。



なお、「節」ルールセットでは適用可能個所が「読んで... 学ぶ」の一カ所しかなく、これはこのまま適用される。

パターン B の場合 「格」ルールセットでは「本を... 読む」と「本を... 学ぶ」が排他である。この場合は意味情報などを利用すれば「本を読む」の方が適切であると判定されるだろう[6]。結果はパターン A の後者と同じになる。

「節」ルールセットの動作はパターン A と同様である。

7 おわりに

G3 に基づいて構文解析を行うパーザは現在は試験実装の段階で、処理効率が悪い。今後の課題としてパーザの実行効率の改善と、ツールとして利用可能な日本語文法の記述と配付が挙げられる。

試験実装パーザ、および日本語文法は以下の Web ページで公開する予定である。

http://www.nlp.ie.niigata-u.ac.jp/nlp/g3/

参考文献

[1] 沼崎浩明, 宮崎正弘. 話者の対象認識過程に基づく日本語助詞「が」と「は」の意味分類とパーザへの実装. 言語処理学会誌, Vol. 2, No. 4, pp. 67–81, 1995.

[2] 黒橋禎夫, 長尾眞. 並列構造の検出に基づく長い日本語文の構文解析. 言語処理学会誌, Vol. 1, No. 1, pp. 35–57, 1994.

[3] 田中穂積. 自然言語解析の基礎. 産業図書, 1989.

[6] 格に関する係りの可能性がこの時点ですべて列挙されているので、動詞「読む」に係る要素が一つもないのは不自然という情報も利用できる