

痕跡処理のための逐次型一般化LRパーザSGLRの拡張

五百川 明 宮崎 正弘

新潟大学大学院自然科学研究科

1 はじめに

これまでに、構文解析において痕跡を扱うための手法がいくつか提案され [1, 2, 3, 4], 文法の記述性および解析効率における有効性が示されている. Prolog上に効率よく実装された, DCG 文法に対する高速な一般化 LR パーザとして SGLR[5] があるが, SGLR では痕跡処理は実現されていない. そこで, 本稿では SGLR を拡張することにより痕跡処理を実現する方法を示す.

文法記述において, 痕跡情報は個々の生成規則に個別に記述されるが, LR 法に基づく構文解析では LR 項の集合から構成される状態の間を遷移しながら解析が進むため, 個々の生成規則に対する, 構文上の個別な情報は扱えない. そのため, このままでは痕跡処理の実現は難しい. そこで, 必要に応じて各状態を分割し, 生成規則の個別性に対処することにより, LR 構文解析における痕跡処理実現を可能とした.

2 文法記述形式

XG, XGS など痕跡を考慮した文法記述形式がいくつか提案されているが, ここでは DCG の拡張である XGS を採用する. ただし, XGS におけるスラッシュ“.. /”の代わりに“/”を用いる. カテゴリ Nt/G は

「カテゴリ Nt を根とする解析木に, 痕跡と見なされるカテゴリ (痕跡カテゴリ) G が 1 つ含まれる」を意味する.

3 解析系の構成

3.1 LR 項の集合の構成

痕跡情報を痕跡スタックに保持し, 解析系が自動的に構文上の欠落部分に対して痕跡を仮定し補う場合, このスタックからポップされた情報が使用される. そのため, 痕跡情報としてどのような情報をどのようなタイミングで痕跡スタックにプッシュするかが重要となる.

LR 項ではドット“.”が, 生成規則の右辺のどの位置まで解析が進んだかを示しており, また同時に, ドットの直後にカテゴリが存在する場合にはそのカテゴリがこれから導出されるということを示している. カテゴリ Nt/G の導出が開始されるときに痕跡カテゴリ G の持つ情報を痕跡スタックにプッシュすれば良いのであるから, 状態遷移においてドットの直後にカテゴリ Nt/G を持つ LR 項が現れたときに痕跡スタックへのプッシュを行えば良い.

各状態は, 遷移元の状態を構成する LR 項の集合から関数 GOTO によって得られる LR 項の集合により構成される. したがって, ある状態において, これから導出される可能性のあるカテゴリは複数存在し得る. そのため, 先に述べたタイミングで痕跡スタックへのプッシュを行った場合, 痕跡とは関連しないカテゴリの導出において誤って痕跡情報が使用される可能性がある.

Gap Handling Extension of Sequential Generalized LR Parser "SGLR"

Akira IYOKAWA (iyo@tinlp.info.eng.niigata-u.ac.jp),
Masahiro MIYAZAKI (miyazaki@info.eng.niigata-u.ac.jp)
Niigata University

これを避けるために、カテゴリ Nt/G の導出に関連する LR 項の集合、つまり、ドットの直後にカテゴリ Nt/G を持つ LR 項から関数 CLOSURE によって得られる LR 項の集合を分離し、新たな状態を構成する。そして、この状態へ遷移するときに痕跡スタックへのプッシュを行う。

このように状態を分割することにより、移動および還元動作後の遷移先状態は複数存在し得ることになる。なお、生成規則の右辺の左端にカテゴリ Nt/G がある場合には、いま述べた方法は使用できない。このような場合には、状態を分割するのではなく、状態内部にカテゴリ Nt/G の導出に関連する LR 項の集合から構成される部分状態を形成することにより対処するなどの工夫が必要となる。

3.2 痕跡情報のポップ

解析過程のある状態において、現在の先読みに対して解析動作が定義されていない場合、つまり、解析が失敗した場合にのみ、そこに痕跡を仮定し補うのでは十分でない。現在の先読みに対して解析動作が定義されている場合でも、そこに痕跡を仮定し解析を続けることが可能な場合がある。したがって、痕跡スタックに痕跡情報が保持されている間は、入力文における各単語間に痕跡が存在する可能性を考慮する必要がある。

実際に、その痕跡の存在の仮定が適切であるかどうかは LR 構文解析表から即座に判定されるため、解析効率にはそれほど大きく影響しない。

3.3 痕跡に対する解析動作

仮定された痕跡が終端カテゴリである場合には、通常先読みと同様に扱われ、移動、還元動作が行われる。

また、非終端カテゴリである場合には、その導出の過程は実際には存在しない。そのため、その非終端カテゴリから関数 FIRST により得られる終端カテゴリを先読みとした還元動作は行われていない。

したがって、非終端カテゴリが痕跡として仮定された場合には、その非終端カテゴリの導出による状態遷移、および、その非終端カテゴリから関数 FIRST により得られる終端カテゴリを先読みとした還元動作、が再帰的に行われなければならない。

3.4 カテゴリ Nt/G の導出

状態遷移において、カテゴリ Nt/G の導出による遷移とカテゴリ Nt の導出による遷移とは区別されなくてはならない。

カテゴリ Nt/G が導出されたときには、その導出が開始されるときに痕跡スタックにプッシュされた痕跡情報は既にポップされていなくてはならないのであるから、カテゴリ Nt が導出されたときに既に痕跡情報がポップされていれば、このカテゴリ Nt の導出はカテゴリ Nt/G の導出であると判断することができる。

3.5 痕跡の対応付け

カテゴリ Nt/G の導出が開始されるときに痕跡カテゴリ G の持つ情報が痕跡スタックにプッシュされるが、痕跡に対応付けられるカテゴリの持つ情報を予め痕跡カテゴリ G に持たせることにより、それらのカテゴリの間の同一性を確保することができる。

DCG では、ある生成規則に記述されたカテゴリの間の情報の受け渡しには、引数として与えられる論理変数や補強項が利用される。SGLR では補強項は還元動作時に評価されるため、上述の場合、論理変数を利用して情報の受け渡しを行う。

このとき、状態とこれまでに導出されたカテゴリについての情報を保持している解析スタックと、カテゴリ Nt/G の導出開始を示す LR 項に対応する生成規則における論理変数が参照される。

4 動作例

文法1を用いて、例文"the child who has a toy smiles."を解析したときの様子を次に示す。

文法1からは付表1に示すLR構文解析表が得られ、また、状態遷移は付図1のようになる。

- 文法1 -

- | | |
|-----------------------|----------------------|
| 1. s → np, vp. | 7. det → [a]. |
| 2. np → det, n. | 8. det → [the]. |
| 3. vp → vi. | 9. n → [child]. |
| 4. vp → vt, np. | 10. n → [toy]. |
| 5. np → np, relC/np. | 11. vi → [smiles]. |
| 6. relC → relPron, s. | 12. vt → [has]. |
| | 13. relPron → [who]. |

以下に示す解析木において記号\$gapにより、カテゴリ np の痕跡の存在が示されている。

```
| ?- start.
Input: the child who has a toy smiles
Words: [the,child,who,has,a,toy,smiles]
Length: 7
= Result 1 =
|- s
  |- np
    |- np
      |- det -- < the >
      |- n -- < child >
    |- relC
      |- relPron -- < who >
      |- s
        |- np -- < $gap >
        |- vp
          |- vt -- < has >
          |- np
            |- det -- < a >
            |- n -- < toy >
        |- vp
          |- vi -- < smiles >
  
```

Argument Info: []
The Number of Results: 1
Run Time: 0 msec
yes
| ?-

5 おわりに

本稿では、SGLRにおいて痕跡処理を実現するための基礎となる部分について述べた。

今野らはXGSにおいて、スラッシュ記法のほかに"<"と">"による記法を導入し、関係節に対するロスの複合名詞句制約を実現している。また、徳永らは、DCGにスラッシュ記法と下位範疇化制約という2つの概念を導入することにより、自然言語にみられる移動現象を自然に記述する枠組みを提案している。

これらの考え方を取り込みつつ、解析系全体としての評価を行うことが今後の課題である。

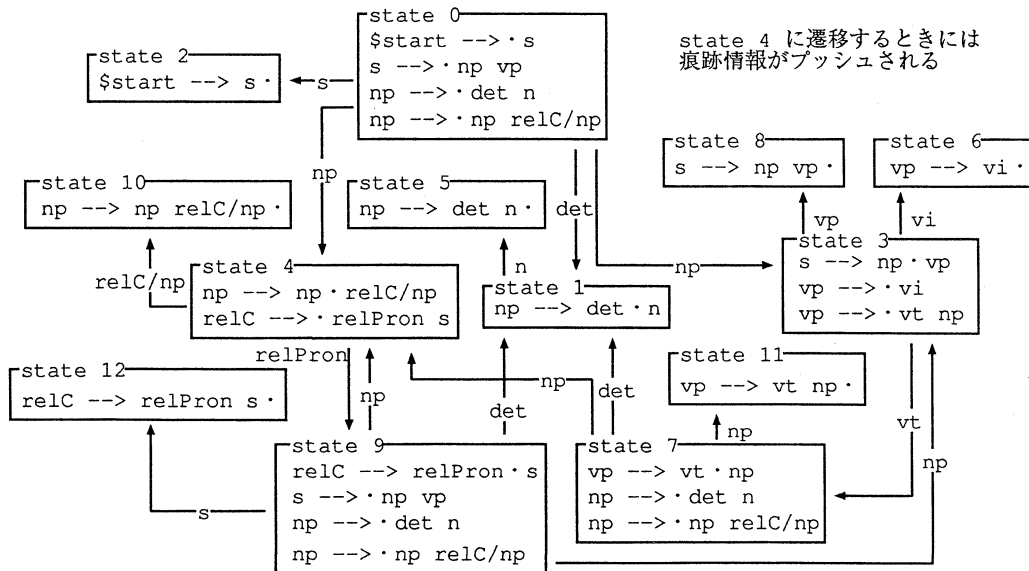
参考文献

- [1] 今野聡, 田中穂積: 左外置を考慮したボトムアップ構文解析システム, コンピュータソフトウェア, Vol.3, No.2, pp.19-29 (1986).
- [2] 林達也: 拡張CFGとその構文解析法 YAPX について, 情報処理学会論文誌, Vol.29, No.5, pp.480-487 (1988).
- [3] 徳永健伸, 岩山真, 田中穂積: 論理文法におけるギャップの扱い, 情報処理学会研究報告, 90-NL-76-3 (1990).
- [4] 春野雅彦, 松本裕治, 長尾真: 痕跡を扱うためのチャート法の拡張, 情報処理学会研究報告, 92-NL-89-5 (1992).
- [5] 沼崎浩明, 田中穂積: SGLR:逐次型一般化LRパーザのPrologによる実現, 情報処理学会論文誌, Vol.32, No.3, pp.396-403 (1991).
- [6] 田中穂積: 自然言語解析の基礎, 産業図書 (1989).
- [7] A.V. エイホ, J.D. ウルマン著, 土居範久訳: コンパイラ, 培風館 (1986).

付表 1: LR 構文解析表

状態	ACTION						GOTO				
	det	n	vi	vt	relPron	\$	s	np	vp	relC	relC/np
0	sh1						2	3, 4(push np)			
1		sh5									
2						acc					
3			sh6	sh7					8		
4					sh9						10
5			re2	re2	re2	re2					
6			re3	re3	re3	re3					
7	sh1							11, 4(push np)			
8			re1	re1	re1	re1					
9	sh1						12	3, 4(push np)			
10			re5	re5	re5	re5					
11			re4	re4	re4	re4					
12			re6	re6	re6	re6					

sh: 移動動作, re: 還元動作, acc: 受理, push: 痕跡情報をプッシュ



付図 1: 状態遷移