グルーピング法によるGLRパーザの効率的な実装

植木 正裕　　白井 清昭　　徳永 健伸　　田中 穂積
東京工業大学大学院 情報理工学研究科

1 はじめに

自然言語の効率のよい構文解析法の1つとして、富田による一般化LR法(GLR法)がある。これはLR法を、一般の文脈自由文法(CFG)が扱えるように拡張したものである[4]。GLR法ではLR法と同様に、文法からLR表と呼ばれる解析動作表をあらかじめ作成しておき、解析スタックの状態と、次の入力記号(先読み記号)の組合せでLR表を引くことにより、次に行なうべき解析動作を決定する。LR法と異なり、GLR法はCFGを対象としているので、解析スタックの状態と先読み記号だけでは解析動作が一意に決定できない場合がある。このような場合、GLR法では、解析スタックを複数に分岐させることで、すべての解析動作を並行して行なう。GLR法では、解析スタックに対するmergeと、中間的な構文構造に対するpackという2つの操作により、解析の高速化を図っている。mergeとpack操作の詳細は文献[4]に譲るとして、ここで、それらの基本的な考え方を説明する。mergeは分岐した複数のスタックトップの状態が同じになる場合に、それぞれの分岐ごとに同一の解析動作を重複して施すのではなく、スタックトップを一本化し、その解析動作を一度だけ実行するための操作である。一方pack操作は、ある条件を満たす分岐を一本化するので、reduce時にスタックトップからポップする枝の数を削減する効果があり、それにより使用メモリの削減と解析速度の向上を図ることができる。Shannは実験によって、GLR法による文の解析速度にもっとも寄与する要因は、pack操作にあると述べている[2]。

しかし、複数個の先読み記号を扱う場合には、異なる先読み記号で同じ解析動作を重複実行する可能性があり、これはmergeとpackの2つの操作では回避できない。本論文では、新たに先読み記号のグルーピングと呼ばれる操作を導入し、スタックのmergeとpack操作だけでなく、このグルーピング操作もGLR法による解析の高速化に大きな役割を果たすことを示す。特に解析用の辞書が大規模になり、多品詞語の数が多くなるとともに、このグルーピングの効果が大きくなる。我々の実験によれば、先読み記号のグルーピングにより2倍程度の高速化を図ることができる。

2 形態素解析と構文解析の統合したMSLR法とグルーピング操作

従来、形態素解析と構文解析はそれぞれ別個のシステムにより解析が行なわれてきた。形態素解析システムは自然言語で書かれた文を入力とし、文を構成する形態素とともに、品詞の列を出力する。構文解析システムは、形態素解析システムが出力した品詞列を入力とし、それから解析結果として構文解析木を出力する。このとき、形態素解析システムは形態素レベルの知識(品詞の結合度etc.)を、構文解析システムは構文レベルの知識(文法や格フレーム辞書etc.)を利用して解析を行なうが、それぞれの解析過程で解析結果が複数得られることが多く、その中からもっとも正しいと思われる解を選択して解析を進める。このとき、もし形態素解析システムが誤った解を選択すると、それに続く構文解析システムの解も誤ったものになってしまう。

したがって、形態素レベルの知識と構文レベルの知識を同時に利用し、形態素レベルの曖昧性を許したまま、構文解析を行なう手法が望まれるが、その1つとしてMSLR法が提案され[3]、公開されている[5]。MSLR法は、富田によるGLR法をベースにしたもので、自然言語で書かれた文に対して、まず辞書引きによって各形態素の品詞をすべて抽出して、それらの品詞を先読み記号として構文解析を行なう。日本語のように単語間に区切りを置かない言語であれば、単語区切りも同時に決めなくてはならない。MSLR法では、文字と文字の区切りをステージと呼び、原則として、ステージを文頭からひとつずつ右に進めながら解析を行なう。

たとえば、「この手法は印象的だ。」という文に対するステージ5での辞書引き結果を図1に示す。辞書引きして抽出した単語区切りの候補(表記)に対して、表記とその単語の品詞を対にして登録する。これを以後「品詞つき表記」と呼ぶ。

一般に、辞書や文法のサイズが大きくなれば、各ステージでの品詞つき表記の数も増大し、それに応じてさまざまな解析動作を多数実行しなくてはならない。その際、解析動作の重複が生じることが多い。次節では、このような解析動作の重複を回避するためのグルーピングと呼ぶ手法を提案する。

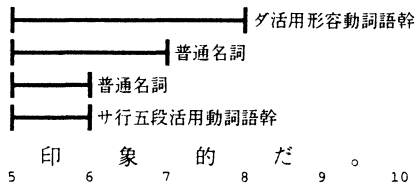

図 1: ステージ 5 を開始位置とした辞書引き結果

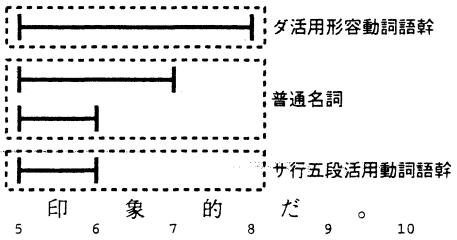

図 2: 品詞によるグルーピング結果

1. < 文 > → < 後置詞句 > < 文 >
2. < 文 > → < 動詞句 >
3. < 文 > → < 形容動詞句 >
4. < 動詞句 > → < 動詞語幹 > < 動詞語尾 >
5. < 形容動詞句 > → < 形容動詞語幹 > < 形容動詞語尾 >
6. < 後置詞句 > → < 名詞 > < 助詞 >
7. < 名詞 > → 普通名詞
8. < 動詞語幹 > → サ行五段活用動詞語幹
9. < 形容動詞語幹 > → ダ活用形容動詞語幹
10. < 助詞 > → 係助詞

図 3: サンプル文法 G

	普通名詞	ダ活用 形容動詞語幹	サ行五段活用 動詞語幹
5	re14	re14	re14
8	sh16	sh12	sh23

図 4: サンプル文法 G から生成した LR 表 (抜粋)

3 グルーピングによる解析動作の重複実行の回避

各ステージでの辞書引きにより得られた複数個の品詞つき表記に対して個々に解析動作を実行すると、実行すべき解析動作の数が増大し、解析動作の重複も頻繁に生じるようになる。それを避けるために我々は以下の 2 つのステップに分けてグルーピングを行なう。まずはじめに品詞によるグルーピングを施す。次に、解析動作によるグルーピングを施す。

以下では、図 3 に示す文法 G から生成した LR 表 (図 4) を用いて、「この手法は印象的だ。」を解析する場合を例に説明する。

3.1 品詞によるグルーピング

英語のように、単語と単語の間に空白を置く言語では、単語区切りの位置は空白によって決まる。したがって、辞書引きによって生じる曖昧性は多品詞語による曖昧性だけである。これに対して、日本語のように単語間に空白を置かない言語では、単語区切りの位置においても曖昧性が生じる。

このとき、単語の長さは異なるが品詞が一致するような単語区切りが複数存在する場合がある。たとえば、図 1 の 4 つの品詞つき表記のうち、「印:普通名詞」と「印象:普通名詞」では、品詞がともに普通名詞で一致する。GLR 法では、品詞を先読み記号として構文解析するということをすでに述べた。したがって、この場合には、同一品詞を持つ 2 つの異なる品詞つき表記に対して、同一の解析動作を重複して行なうことになってしまう。そこで、品詞つき表記を品詞ごとにグルーピングし、グルーピングした品詞を先読み記号として用いて構文解析することが必要になる。図 1 で得られた品詞つき表記に対しては、図 2 に示すように、普通名詞「印」と普通名詞「印象」をグルーピングして、1 つの**普通名詞**として扱う。このようにして、先読み記号は、**ダ活用形容動詞語幹**、**普通名詞**、**サ行五段活用動詞語幹**の 3 つになる。

3.2 解析動作によるグルーピング

同一品詞の品詞つき表記にグルーピングを施してから、グルーピングした品詞を先読み記号として LR 表を検索し、実行すべき解析動作を決定する。たとえば、図 5 まで解析が進んだ状態で、先ほどの 3 つの先読み記号、**ダ活用形容動詞語幹**、**普通名詞**、**サ行五段活用動詞語幹**で LR 表を検索すると、すべて re14 になっている (図 4 の LR 表の状態 5 の行)。このように一般に、各ステージで LR 表を検索して得た複数個の解析動作の中には、重複が含まれることが多い。そこで、先読み記号ごとに個別に reduce 動作を行なうことをやめ、reduce 動作の同一性による先読み記号のグルーピングを施すことにする。

「この手法は印象的だ。」の例では、図 6 に示すように、先読み記号となる**普通名詞**、**サ行五段活用動詞語幹**、**ダ活用形容動詞語幹**のいずれに対しても、G の規則 14 による reduce 動作を行ない、解析スタック上に助詞をプッシュすることになる。そこで、これら 3

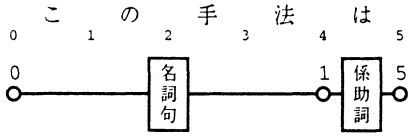

図 5: ステージ 5 まで解析が終了したときの解析スタックの状態

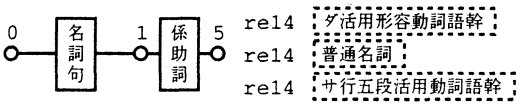

図 6: グルーピング後の各先読み記号に対する LR 表検索結果

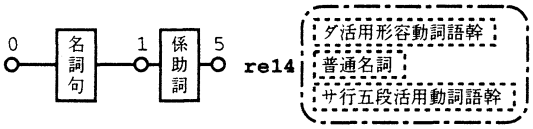

図 7: reduce 動作によるグルーピング結果

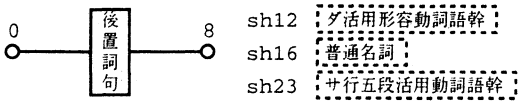

図 8: 各先読み記号に対して最終的に実行すべき shift 動作

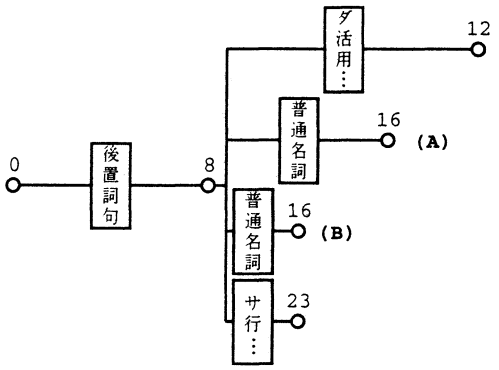

図 9: グルーピングを解除してすべての品詞つき表記を shift した様子

つの reduce 動作をグルーピングして (図 7 の一点鎖線で囲まれたグルーピング参照)、1 回だけ reduce 動作を実行する。

reduce 動作を実行すると、スタックトップの状態が変化し、次に実行する解析動作を再び LR 表から得ることになる。そのため、reduce 動作による先読み記号のグルーピングは、各スタックトップに対して、LR 表から解析動作を得るごとに繰り返し行ない、reduce 動作によるグルーピングをやり直す必要がある。

同一の品詞に対しては解析動作は常に一致するので、3.1 で述べた品詞によるグルーピングは、解析動作のグルーピングを行なうことにより結果的に達成されるので不必要であると思われるかもしれない。しかし、品詞によるグルーピングは辞書引き直後に一度だけ行なえばよいので、LR 表から解析動作を得るたびごとに行なう解析動作のグルーピングの計算量を減らすことができる。このことから、提案する手法では、辞書引き後の品詞によるグルーピングをまずはじめに行なってから、LR 表検索後の解析動作によるグルーピングを行なっている。

3.3 グルーピング操作の実装上の注意

解析の各ステージで品詞によるグルーピングと解析動作によるグルーピングを行ない、解析を進める。あるステージで行なうべき解析動作が最終的に shift 動作になり、残りの解析文の解析に進む。この最後に行なう shift 動作では、品詞つき表記が異なれば、異なる品詞つき表記をスタックにプッシュしなければならない。

例では、図 7 の rel4 の実行により**助詞**が解析スタックにプッシュされ、さらに文法 G の規則 6 により**後置詞句**に reduce されて図 8 の状態に至る。図 1 でグルーピングされた長さの異なる 2 つの普通名詞「印」と「印象」を shift の段階では区別しなくてはならない。そこで、グルーピングしておいた品詞つき表記をすべて展開して、それぞれを解析スタックにプッシュする。2 つの普通名詞「印」と「印象」を解析スタックにプッシュした後の様子を図 9 に示す。図 9 の (A) の分岐は普通名詞「印象」をプッシュしたものであり、(B) の分岐は普通名詞「印」をプッシュしたものである。

4 実験と考察

4.1 実験方法

実験に用いる文法は、CFG 形式で記述した日本語文法で、ルール数は約 900 である [5]。構文解析でも広く用いられている句構造文法とはやや異なる文節文法を用いている。これは、文節内の構文構造を解析し、文節単位のまとまりを作る規則と、各文節間の修飾関係に関する規則からなる。また、辞書としては EDR 日

品詞つき表記の異なり数	2,872,039
品詞の異なり数	2,760,025
1 品詞あたりの品詞つき表記数	1.04

表 1: 品詞によるグルーピングの効果

reduce 動作の延べ数	10,478,354
reduce 動作の異なり数	4,253,140
1 動作あたりの先読み記号数	2.46

表 2: reduce 動作によるグルーピングの効果

本語単語辞書 (約 2 0 万語) [1] を用いた。実験に用いた解析システムは、第 2 節に挙げた MSLR 法を実装したシステムで、第 1 節で説明した merge 操作と pack 操作、第 3 節のグルーピングの手法が組み込まれている。入力文は、EDR コーパスからランダムに抽出した文、約 13,000 文である。

まず、品詞によるグルーピングの効果を調べるために、解析の各ステージにおいて、辞書引きによって得られた品詞つき表記の数 (図 1 の例では 4 個) と、異なり品詞数 (図 1 の例では 3 個) を数えた。これによって、解析の各ステージで何個の品詞つき表記が 1 つの品詞としてグルーピングされているかを推定することができる。入力文全体での合計値をまとめたものを表 1 に示す。この数が大きいほどグルーピングの効果が大きいと考えられる。

また、解析動作によるグルーピングの効果を調べるために、各解析ステージにおいて、すべての先読み記号に対して LR 表から得られる reduce 動作の延べ数と異なり数をそれぞれ求めた。これらの値から、各ステージにおいて、何個の先読み記号が 1 つの reduce 動作の先読み記号としてグルーピングできるのかを推定することができる。入力文全体でのそれぞれの数値を表 2 に示す。この数が大きいほどグルーピングの効果が大きいと考えられる。

4.2 実験結果と考察

品詞によるグルーピングでは、1 品詞あたりの品詞つき表記数は 1.04 と、それほど大きな値とはならなかった。それに対して、解析動作によるグルーピングでは、1 動作あたりの先読み記号数が 2.46 と大きな値になっている。グルーピングを行なわなければ、入力文全体で約 1000 万回もの reduce 動作が実行されるが、グルーピングを行なった場合には、その半分以下の約 400 万回ですむことになる。これは、解析時間の短縮に大きな効果が期待できるということを意味する。

解析動作によるグルーピングの効果を解析時間で計ったものを表 3 に示す。ここで、1 文あたりの解析時間は 1 文から得られるすべての解析結果 (構文木) を計算するのに要する時間である。解析動作のグルーピング

	実験 A	実験 B
解析動作のグルーピング	なし	あり
平均解析時間	0.46 sec/文	0.24 sec/文

表 3: 解析時間による比較

を行なっている実験 B では、解析動作のグルーピングを行なわない実験 A と比べて、1 文あたりの解析時間は約半分に短縮されている。これは、解析動作のグルーピングによって、実際に実行される解析動作の数が約半分になっている表 2 の結果からも十分に予想された結果である。

以上の結果から、本論文で提案した先読み記号のグルーピングの有効性が実験的に確認された。

5 おわりに

第 4 節で示す実験で、我々が提案した解析動作による先読み記号のグルーピングが解析時間の短縮に大きな役割を果たすことが確認された。ある状態で行なう GLR 法の解析動作が、たとえ先読み記号が異なっても同じになることが多い言語であれば、いかなる言語であっても、グルーピングの効果が大きいことが予想される。今後、英語などの、単語と単語との間に空白を置く言語についても、本論文で述べたグルーピングの効果を検証したい。

参考文献

[1] EDR. 電子化辞書仕様説明書 第 2 版. 日本電子化辞書研究所, 3 1995.

[2] Patrick Shann. Experiments with GLR and chart parsing. In *Generalized LR Parsing*, pp. 17–34. Kluwer Academic Publishers, 1991.

[3] Hozumi Tanaka, Takenobu Tokunaga, and Michio Aizawa. Integration of morphological and syntactic analysis based on LR parsing algorithm. 自然言語処理, Vol. 2, No. 2, pp. 59–74, 4 1995.

[4] Masaru Tomita and See-Kiong Ng. The generalized LR parsing algorightm. In *Generalized LR Parsing*, pp. 1–16. Kluwer Academic Publishers, 1991.

[5] 東京工業大学田中・徳永研究室. Morphological and syntactic LR parser — MSLR parser. available from `http://tanaka-www.cs.titech.ac.jp/pub/mslr/`, 1998.