20－4

# 一般化LR法を用いた非文の処理

# Processing ill-formed sentences using generalized LR parsing algorithm

今井 宏樹*
Hiroki Imai
北陸先端科学技術大学院大学
Japan Advanced Institute of Science and Technology

田中 穂積
Hozumi Tanaka
徳永 健伸
Takenobu Tokunaga
東京工業大学 工学部 情報工学科
Department of Computer Science
Tokyo Institute of Technology

**Abstract**

In this paper, we describe a new method for analyzing ill-formed sentences using generalized LR parsing algorithm. In the proposed method, the parser analyzes an input sentence with the ordinary GLR parsing algorithm as long as the input is well-formed. The error handling algorithm is invoked only when an error is found. The error handling algorithm estimates the error type and corrects the error. Errors are categorized into four types which are ranked in order of liability. The algorithm includes the correction mechanisms for each of these error types, and tries them one by one in order of liability. Since our algorithm keeps track of the states of the analysis until an error happens, we would be able to avoid unnecessary recomputation. We conducted preliminary experiments and proved our method effective.

## 1 はじめに

従来の自然言語処理システムの多くは，文法に適合する文のみを解析するパーザを用いて構成されていた．しかし，それでは実用性に欠けるため，文法に適合しない非文の構文解析法の研究が進められている．これまでに，チャート法を基礎とする研究がおこなわれているが ([1], [2], [3])，解析効率の良い手法として知られる一般化 LR 法 (GLR) を用いた研究はほとんど行われていない．また，特に非文の解析が必要となる音声認識の分野においては，GLR が用いられることが多い．

GLR を用いて非文の処理をおこなった研究の例として斉藤らの手法がある [4]．斉藤らは音声入力の文を GLR によって解析する際に，音素単位で誤りを修正し解析を続行する手法を提案している．基本的な考え方は，各ステージにおいて誤りの可能性を考慮し，LR 表を参照して現在解析をしている状態で先読み記号以外にも動作が定義されている記号があればそれらも並列に実行するというものである．さらに，解析の効率化をはかるため，パーザに対して制約を加えたり，音声入力装置の誤り認識率のデータを基に解析候補に優先順位をつけて解析を行っている．しかし，この方法には，これらのヒューリスティクスは一般の非文の解析に利用できない，正しい入力文の解析にも誤りを含む文と同程度の時間がかかってしまう，などの問題がある．

以上のような背景から，本研究では，GLR を用いて非文を解析する新たな手法を検討した．

## 2 誤りの定義

自然言語処理の分野では，非文あるいは不適格文という語が表す意味の範疇は広い．本稿では，「非文を与えられた文法にそぐわない文」と考えて議論を進める．また，本研究では，文中

*今井 宏樹: 北陸先端科学技術大学院大学
〒923-12 石川県能美郡辰口町旭台 15
Tel: 0761-51-1111 (内線 1373)
E-mail: h-imai@jaist.ac.jp

の誤りが高々1箇所の場合に誤りの箇所を推定し修正を施す手法を検討した．

誤りの定義は Mellish らのものに準処し，以下の 4 種類の誤りに対して処理を行うこととした．

**置換誤り** 文中のある語が違う語に置換されている．

**挿入誤り** 文中に余分な語が含まれている．

**脱落誤り** 文中で必要な語が抜けている．

**未知語** 置換や挿入が起こっている語がシステムにとって未知のものである場合，それを未知語と呼ぶ．

## 3 アルゴリズム

本手法の基本的戦略は，可能な限り GLR (富田法) による解析を行い，途中で解析が失敗した時に誤りに対して処理を施すものである．

まず，GLR での解析中に，解析の履歴を記録する．これは，還元動作において，ポップすべきスタックはそのまま残し，新たにスタックを分岐して還元された記号をプッシュすることにより実現できる．そして，途中で解析が失敗した時，その情報を利用して処理を行う．誤りに対する処理は，以下の 2 つの処理を繰り返すことにより行われる．

- 誤り箇所の推定
- 推定された箇所の修正

以下，それぞれの処理について説明を行う．なお，以下の説明では，単語を辞書引きした時にそのカテゴリ (終端記号) も正しく引かれると仮定する．

### 3.1 誤り箇所の推定

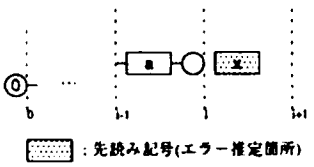
誤り箇所の推定においては，新たに

> 解析失敗までに還元された終端記号は
> 誤りの可能性が低い

というヒューリスティクスを導入した．これは，「一度還元されている部分は構文的に正しく解析されている可能性が高い」という考えに基づいている．パーザはこの知識と解析の履歴を利用して，以下の順位でステージ単位で誤り箇所を推定する．

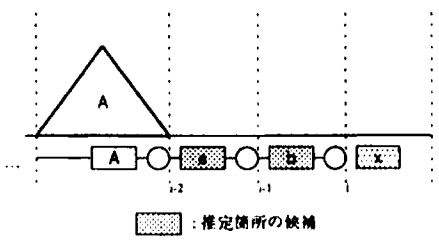
1. 誤りの起こった先読み記号

   i : 解析の失敗したステージ



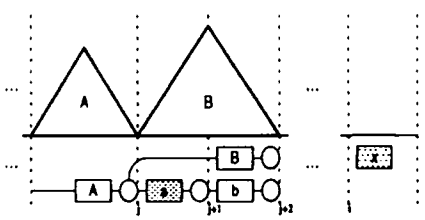

2. まだ還元されていない終端記号

   i : 解析失敗時のステージ

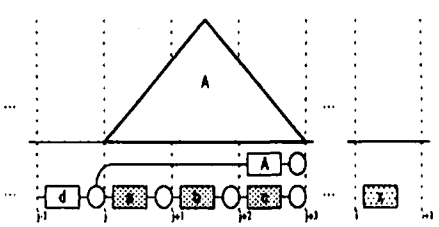



3. 2 つの隣接する部分木の間

   j : エラー推定されたステージ



4. 部分木の内部



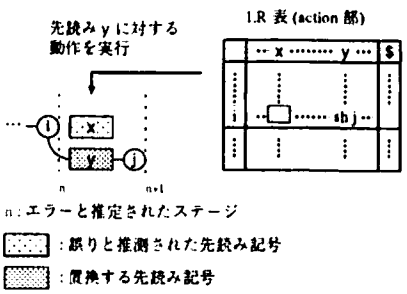
なお，各項目において複数箇所の候補がある場合，右にあるステージから順に決定する．

### 3.2 指定された箇所の修正

まず，エラーと推定されたステージの解析を始める状態 (すなわち，そのステージから始まる単語を先読みとする状態) までパーザの状態を戻す．その状態から，LR 表を参照し，誤りの原因となった記号に対して，置換，挿入，脱落の各誤りに対する処理をそれぞれ実行する．未知語は，形式的には置換，挿入のどちらかに含まれるので，それ自体に特別の処理は必要ない．なお，以下の図中の LR 表で，□とある欄には動作が定義されていないことを示す．

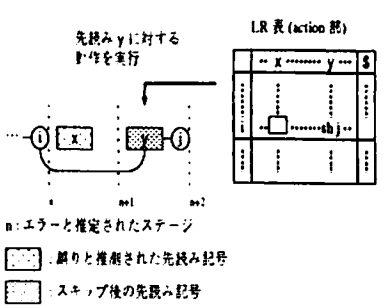
**置換誤りに対する処理**

誤りと推定された先読み記号，及び $ 以外の先読み記号に対して動作可能ならば，その動作を実行する．



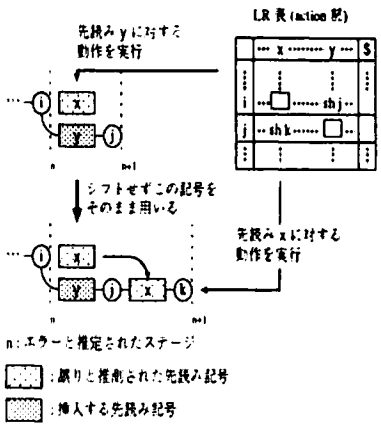

**挿入誤りに対する処理**

その状態のままさらにもう 1 語先読みをする．




**脱落誤りに対する処理**

誤りと推定された先読み記号の直前に，その状態で動作可能な記号を補う．




そして，修正されたスタックを通常の GLR のアルゴリズムで解析を進め，最終的に受理されればその修正は正しい．途中で失敗すれば修正は正しくないとする．以上の手続きを可能性のあるすべての箇所について行うことにより，文法的に正しい修正をすべて得ることができる．

### 3.3 再計算の回避

本研究では，修正処理において，解析の履歴情報を利用することにより，不要な再計算を避け高速化をはかる手法を考案した．つまり，誤り修正後のスタックと履歴に残っているスタックの位置と状態が等しければそれらをマージすることによりそれ以後の処理を軽減している．

## 4 実験

次に，この手法を基にパーザを実装し，実験を行った．実験では，規則数 7 の非常に単純な文法 I，規則数 180 の中規模な文法 II の 2 種類の英語の文法を用いた．文法の概要を図 4.1 に示す．

| | 文法 I | 文法 II |
|---|---|---|
| 規則数 | 7 | 180 |
| 非終端記号の数 | 4 | 34 |
| 終端記号の数 | 4 | 26 |
| 最長規則長 | 2 | 5 |
| 平均規則長 | 1.86 | 2.17 |

図 4.1：実験に使用した文法

それぞれの文法について，単語数3～50のPPアタッチメントのあいまい性のある文を用いて正しい文と非文を入力し，解析速度を比較した．それぞれの文法に対する結果を図4.2，4.3に示す．

正しい文と非文の解析時間の比は，文法Iで正しい文の約5倍，文法IIでは置換誤りを含む文で約5倍，それ以外の誤りを含む文で約100倍，という結果を得た．

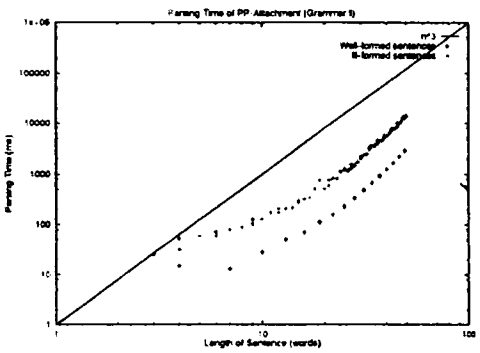


図4.2：文法Iによる実験結果

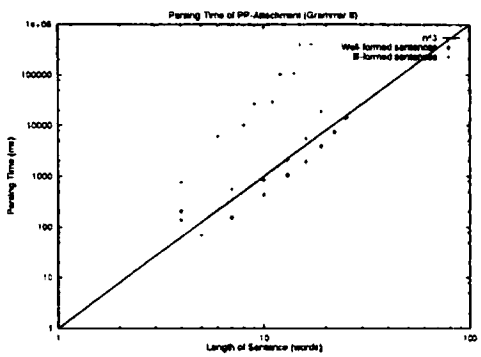


図4.3：文法IIによる実験結果

## 5 まとめ

本稿では，一般化LR法を用いた非文の解析法を提案した．これは，可能な限り解析を続け，途中で失敗した場合に誤りに対する処理を施すものである．また，解析失敗までの解析データを利用し，誤り修正後の処理で再計算を避ける工夫を行った．さらに，本アルゴリズムを実装して実験を行い，アルゴリズムの有効性を検討した．

我々の手法は，誤りの修正処理に関しては，斉藤らのアルゴリズム[4]とあまり変わりはない．しかし，誤り箇所を推定するアルゴリズムは本研究で新たに提案するものである．本手法の利点は，以下の2点である．

- 正しい入力文に対して通常通り解析が行える
- 解析失敗までの履歴情報を利用して誤りに対する計算量を軽減できる

今後の課題として，複数の誤り箇所が存在する場合の処理の検討，非文処理の計算量解析，アルゴリズムの効率化，などが考えられる．現在，複数の誤りを扱うアルゴリズムを検討中である．

## 参考文献

[1] C. S. Mellish. Some chart-based techniques for parsing ill-formed input. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 102–109, 1989.

[2] T. Kato. Yet another chart-based technique for parsing ill-formed input. Technical report, SIGNL 83-10, Information processing society of Japan, 1991.

[3] S. Meknavin, M. Okumura, and H. Tanaka. A chart-based method of ID/LP parsing with generalized discrimination networks. In *Proceedings of the International Conference on Computational Linguistics*, pp. 401–407 (Vol.1), 1992.

[4] H. Saito and M. Tomita. Parsing noisy sentences. In *Proceedings of the International Conference on Computational Linguistics*, pp. 561–566, 1988.