文法的不適格文に対する統語的制約を用いた漸進的解析手法

加藤 芳秀　　松原 茂樹　　外山 勝彦　　稲垣 康善

名古屋大学大学院工学研究科

{yosihide,matu,toyama,inagaki}@inagaki.nuie.nagoya-u.ac.jp

1 はじめに

実用的な対話処理システムには, 文法的に不適格な文に対しても頑健に対処できることが要求されるが, これまでに作成されたシステムの多くは, 文法や辞書によって定められた適格な文を解析の対象としていた. これに対して, 文脈自由言語に対する統語解析アルゴリズムを文法的不適格文の解析のために拡張した手法がいくつか提案されている [1, 3, 6, 9, 10]. これらは, 文中に含まれる文法的誤りを修正することにより, 文法的不適格文に対する統語構造を作成するための枠組である.

ところで, 実時間対話処理システムを実現するためには, 自然言語文をその出現順序に従って順次解釈する枠組, すなわち, 漸進的解釈手法が不可欠となる [2]. 相手の発話を漸進的に解釈することにより, 割り込みや言い直しなどの生成も可能となり, より自然で円滑な対話の実現が期待できる. しかし, 上述した手法はいずれも一文単位での解析処理を基本とするため, 入力途中の段階でそれまでの入力に対する統語構造を作成することはできず, 漸進的な解析処理に適しているとは言い難い.

そこで本稿では, 文法的不適格文を漸進的に解析するための手法を提案する. 本手法では, 到達可能性, 及び連接可能性といった統語的制約を用いて, 文の入力途中の段階で随時, 誤り修正を実行する. これにより, 文法的不適格文に対して漸進性を損なうことなく統語構造を作成することができる. また, 誤り修正に対してコストを導入することにより, 入力文に対する修正の度合を表現する. 文法的に不適格な入力文は断片的な統語構造をもっていると考えられるが, これにより, 作成された統語構造がそれをどの程度保存しているかを評価できる. なお, 以下では, チャート法 [5] をベースとして議論を進める. これは, チャート法が不完全な文の構造の表現に適したデータ構造を備えていることによる.

本稿の構成は以下の通りである. 2節では, 漸進的なチャート解析手法について説明する. 3節では, 誤り修正手法について述べる. 4節では, 関連研究との比較を行う.

2 漸進的なチャート解析手法

本節では, チャート法に基づく漸進的な解析手法 [7] について簡単に説明する. この手法は, 単語が入力されるごとにそれまでの入力に対する統語構造を随時作成するため, 入力途中の段階での統語構造を用いた処理, 例えば, 意味解析や文脈解析, あるいは翻訳処理などを実行するための枠組として適している.

漸進的な解析処理において i 番目の語が入力されたときに用いられる手続きを以下に示す.

辞書引き 語 w_i の範疇が X ならば, 項 $[w_i]_X$ をもつ不活性弧をチャートの節点 $i-1$ と i の間に追加する.

文法規則の適用 チャートの節点 $i-1$ と i を結ぶ項 $[\ldots]_X$ をもつ弧に対して, 文法規則 $A \rightarrow XY\ldots Z$ が存在するならば, 項 $[[\ldots]_X[?]_Y\ldots[?]_Z]_A$ をもつ弧をチャートの節点 $i-1$ と i の間に追加する.

項の置き換え チャートの節点 0 と $i-1$ を結ぶ活性弧の項 σ の最左未決定項を $[?]_X$ とする. このとき, チャートの節点 $i-1$ と i を結ぶ弧の項 τ の範疇が X ならば, 項 σ の最左未決定項を項 τ で置き換えた項をもつ弧をチャートの節点 0 と i の間に追加する.

文法		辞書	
s	→ np vp	pron	→ I
np	→ pron	det	→ the
np	→ det n	n	→ train
np	→ gi pp	gi	→ going
np	→ n	p	→ by
vp	→ vt s	vt	→ think
vp	→ vi pp	vi	→ think
vp	→ be adj	be	→ is
pp	→ p np	adj	→ best

s : 文	np : 名詞句	vp :動詞句
pp :前置詞句		
pron: 代名詞	det: 冠詞	n : 名詞
vt : 他動詞	vi : 自動詞	gi :動名詞
p : 前置詞	be :be 動詞	adj:形容詞

図 1: 統語解析のための文法と辞書

この手法には, 従来の上昇型チャート解析手法 [5] と異なり, 活性弧に対して文法規則を適用する操作, 及び活性弧の項の最左未決定項を別の活性弧の項で置き換える操作が導入されている. これにより, 入力途中の段階で, それまでの入力に対する統語構造を漸進的に作成することが可能となる. 例えば, 図 1に示す文法と辞書を用いるとき, 英語文

(2.1) I think going by train is best.

における語 "think" が入力された段階までの解析処理を比較すると, 通常の上昇型チャート解析では, "I", 及び"think" にそれぞれ文法規則を適用することにより, 統語構造 $[[[I]_{pron}]_{np}[?]_{vp}]_s$, $[[think]_{vi}[?]_{pp}]_{vp}$, $[[think]_{vt}[?]_s]_{vp}$ を作成するが, 漸進的なチャート解析では "I think" に対する統語構造

(2.2) $[[[I]_{pron}]_{np}[[think]_{vt}[?]_s]_{vp}]_s$

(2.3) $[[[I]_{pron}]_{np}[[think]_{vi}[?]_{pp}]_{vp}]_s$

を作成することができる.

3 文法的不適格文に対する漸進的な解析手法

本稿で提案する手法は, 漸進的なチャート解析手法に対して誤り修正の操作を導入した枠組である. 文法的不適格文に対して, 語が入力されるたびに随時誤り修正を実行し漸進的に統語構造を作成する.

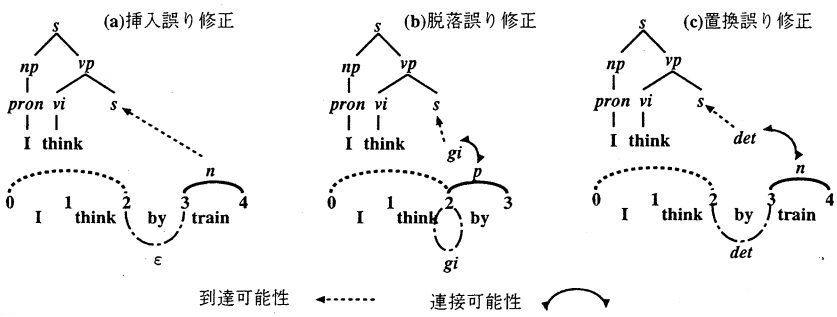

図 2: 誤り修正の例

3.1 誤りの種類とその修正

本手法で扱う文法的誤りは, 文献 [1, 3, 6, 9, 10] と同様, 余分な語の挿入 (挿入誤り), 必要な語の脱落 (脱落誤り), 別の語への置換 (置換誤り) の 3 種類とする. これらの誤りはそれぞれ,

- 語を読み飛ばす (挿入誤り修正)
- 語を挿入する (脱落誤り修正)
- 別の語に置き換える (置換誤り修正)

という操作により修正することができる. 例えば, 英語文
(3.1)* I think by train is best.
は英語文 (2.1) の "going" が脱落した文, "by" が余分に挿入された文, 範疇が冠詞である語が "by" で置き換えられた文のいずれかであると考えられ, それぞれ, "going" を挿入すること, "by" を読み飛ばすこと, "by" を範疇が冠詞である語に置き換えることによって誤り修正ができる. 実際,"think" が入力されたときに生成される節点 (節点 2) に対して項 $[going]_{gi}$ をもつ弧を追加し, これに対して文法規則を適用すると, 項
(3.2) $[[[going]_{gi}[?]_{pp}]_{np}[?]_{vp}]_s$
をもつ弧を節点 2 と 2 の間に作成できる. さらに, 項 (2.2) の最左未決定項 $[?]_s$ を項 (3.2) で置き換えることにより, 動名詞が脱落した "I think" に対する統語構造を表現する項
(3.3) $[[[I]_{pron}]_{np}[[think]_{vt}[[[going]_{gi}[?]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$
をもつ弧を作成できる. このように, 誤り修正を行うことにより, 文法的不適格文に対して統語構造を作成できる.

3.2 誤りの同定方法

前節では, 誤り修正により, 文法的不適格文に対して統語構造を作成する例を示した. しかし, 上述した 3 つの操作を適切に用いなければ, 統語構造を作成することはできない. その上, 統語構造を作成できない不要な操作を実行すると解析の効率が低下する. すなわち, 可能な誤り修正の種類と適用箇所を限定して, 効率的に統語構造を作成する必要がある. そこで本手法では, これまでの解析結果, 及び先読みした語のもつ情報を用いて誤りの同定を行い, 誤り修正を随時実行することにより, 漸進的に統語構造を作成する.

まず, 誤りの同定において用いられる到達可能性, 及び連接可能性について説明する. いずれも範疇間の関係を表す. まず, 範疇 X が範疇 Y に到達可能であるとは, Y を根とし, X をその左端の子とする解析木が存在することを意味する. 到達可能性は文法規則からあらかじめ計算しておくことができる. 到達可能性を用いることにより, 不要な文法規則の適用を回避することができる. 一方, 範疇 X が範疇 Y に連接可能であるとは, X が Y の右隣に位置できることを意味する. 連接可能性を用いることにより, ある範疇の隣に位置する範疇を推定できる. 以下では, X を範疇, σ を項とするとき, 記法 $X \rightsquigarrow \sigma$ で X が σ の最左未決定項の範疇に到達可能であることを表し, 記法 $X \frown \sigma$ で X が σ の範疇に連接可能であることを表す.

i 番目の語が入力されたときに行う誤り修正の操作を以下に示す.

挿入誤り修正 チャートの節点 0 と $i-2$ の間の活性弧の項 σ, 及び i 番目の語 w_i の範疇 X に対して, $X \rightsquigarrow \sigma$ ならば, 項 ε の弧をチャートの節点 $i-2$ と $i-1$ の間に追加する.

脱落誤り修正 チャートの節点 0 と $i-1$ の間の活性弧の項 σ, 及び節点 $i-1$ と i の間の弧の項 τ に対して, 範疇 X が $X \rightsquigarrow \sigma$ かつ $X \frown \tau$ ならば, 項 $[*]_X$ の弧をチャートの節点 $i-1$ と $i-1$ の間に追加する.

置換誤り修正 チャートの節点 0 と $i-2$ の間の活性弧の項 σ, 及び節点 $i-1$ と i の間の弧の項 τ に対して, 範疇 X が $X \rightsquigarrow \sigma$ かつ $X \frown \tau$ ならば, 項 $[*]_X$ の弧をチャートの節点 $i-2$ と $i-1$ の間に追加する.

到達可能性は, 不要な文法規則の適用を回避するものであるが, これを誤り修正に対して用いることにより, 不要な操作を回避できる. また連接可能性を導入することにより, 脱落誤り修正, 及び置換誤り修正において考慮すべき範疇をさらに絞り込むことができる. 例えば, 英語文 (3.1) において, "by" が入力された段階で, "by" の直前に脱落誤り修正を実行するとき, 通常は, 図 1の文法に出現する 9 つの範疇すべてについて挿入を実行しなければならない. これに対して, 到達可能性を用いることにより, "I think" に対する統語構造 (2.2), 及び (2.3) の最左未決定項の範疇に到達可能な範疇, すなわち, det, gi, n, p, $pron$ の語のみが挿入可能な範疇となる. さらに, 連接可能性を用いることにより, 上の 5 つの範疇のうち, "by" の範疇 p に連接可能な範疇は gi のみであり, 結局, 範疇 gi の語の挿入のみ実行すればよい. 実際, 脱落誤り修正として適切なものは gi を挿入する操作のみである. このように, 到達可能性, 及び連接可能性を用いることにより, 効率的に誤り修正を実行できる. これらの性質は, 1 語先読みすることにより利用できるため, 漸進的な処理においては有用である.

英語文 (3.1) に対する各誤り修正の実行例を図 2に示す. 図 2(a) は, 語 "train" が入力された段階での挿入誤り修正の

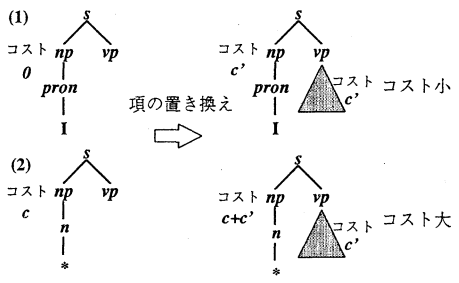

図 3: 枝刈りにおけるコストの比較

実行例である．“I think” に対する弧 (2.2) の最左未決定項の範疇 s に，語 “train” の範疇 n は到達可能であるため，語 “by” は読み飛ばされる．図 2(b) は，語 “by” が入力された段階での脱落誤り修正の実行例である．“I think” に対する弧 (2.2) の最左未決定項の範疇 s に到達可能で，かつ語 “by” の範疇 p に連接可能な範疇 gi の語が挿入される．図 2(c) は，語 “train” が入力された段階での置換誤り修正の実行例である．“I think” に対する弧 (2.2) の最左未決定項の範疇 s に到達可能で，かつ語 “train” の範疇 n に連接可能な範疇 det の語によって語 “by” が置き換えられる．このように，入力途中の段階で随時，誤り修正を実行する．

3.3 誤り修正に対するコスト

一般に，少ない誤り修正により作成された統語構造ほど，もとの入力がもっている断片的な統語構造をより多く保存している可能性が高いと考えられる．本手法では，誤り修正に対してコストを導入することにより，入力文に対する修正の度合を表現する．これにより，上記の可能性の程度を評価する．あらかじめ，挿入誤り修正，脱落誤り修正，置換誤り修正に対してそれぞれコストを定めておく．統語構造のコストを，その統語構造の作成に実行された誤り修正のコストの和と定め，統語構造の有用性をコストにより評価する．統語構造のコストは，誤り修正により統語構造を作成した時点でその誤り修正のコストを逐次加算することにより計算する．

一方，解析の途中で作成される統語構造が増大すると，解析の効率が低下する．この問題を解決するために，コストを用いて弧の枝刈りを行う．チャートの節点 0 と i の間を結ぶ弧のうち，その項の未決定部分の範疇が一致する弧をすべて選び出し，コストが最小でない弧を刈る．例えば，英語文 (3.1) において，語 “I” が入力された段階で表 3の弧 (1) がチャートに作成される．語 “think” が入力された段階で置換誤り修正により，節点 0 と 1 の間に項 $[*]_n$ をもつ弧が作成され，その結果，表 3の弧 (2) が作成される．これらの弧は未決定部分の範疇が vp と一致する．このとき，弧 (1) と比べて弧 (2) は置換誤り修正を実行した分だけ，コストが高いため枝刈りされる．仮に，弧 (1) と弧 (2) に対して項の置き換えを実行する場合，弧 (2) をもとに作成した弧のコストは，常に弧 (1) をもとに作成した弧のコストを上回る．

3.4 文法的不適格文の解析例

英語文 (3.1)* に対する漸進的な解析処理の過程を表 1に示す．各行がチャートの弧に対応しており，# 付の数は弧の作成順序，loc は弧の場所を表す．また，弧の生成の過程を図 4に示す．なお，簡単のため，一回の誤り修正に対するコストはいずれも 1 としている．

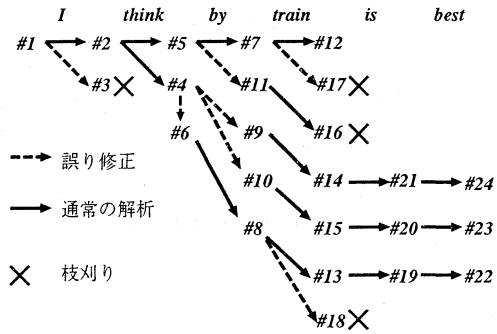

図 4: 弧の作成プロセス

脱落誤り修正により，語 “by” が入力された段階で，動名詞が脱落した “I think” に対する統語構造 #6 が作成される．語 “train” が入力された段階で，挿入誤り修正により，“by” が余分に挿入された “I think by” に対する統語構造 #9 が作成され，置換誤り修正により，範疇 det の語が “by” に置き換わった “I think by” に対する統語構造 #10 が作成される．また，誤り修正により統語構造 #3, #11, #17, #18 が作成されるが，これらはすべて枝刈りされる．このように，文法的不適格文に対して入力途中の段階でそれまでの入力に対する統語構造を作成できる．

4 関連研究との比較

文脈自由文法を対象とした統語解析アルゴリズムを，文法的不適格文の処理のために拡張した手法はこれまでにいくつか提案されている．本節では，解析の漸進性という観点に着目してこれらの手法との比較を与える．

Mellish[9] や加藤 [3] はチャート法に基づく手法を提案している．これらは上昇型チャート解析により一文全体を解析し，解析に失敗すると下降型解析を用いて誤り修正を実行する．しかし，通常のチャート解析をベースとするため，入力途中の段階でそれまでの入力に対する統語構造を作成できない．また，誤り修正は，文全体が入力された後で解析の失敗が明らかになってから起動される．

今井ら [1] や斎藤 [10] は GLR 法に基づく枠組を提案している．今井らの手法は GLR 法により入力文を解析し，解析に失敗すると後戻りして誤り修正を行う．誤りを即座に修正できないため，漸進的な解析に適さない．一方，斎藤の手法は，入力文の解析と同時進行的に誤り修正を行う．斎藤の手法は，本稿で提案した手法と同様に，解析と同時進行的に誤り修正を行う．しかし，通常のチャート法と同様，GLR 法も入力途中の段階でそれまでの入力に対する統語構造を作成できないため，漸進的な解析に適さない．

Lee ら [6] はアーリー法に基づく手法を提案している．入力の解析と同時進行的に誤り修正を行い，解析結果として項の集合を更新する．しかし，アーリー法は解析終了時に全体の統語構造を作成するため，漸進的な解析に適さない．

5 おわりに

本稿では，解析処理と同時進行的に誤り修正を実行する漸進的な文法的不適格文解析手法を提案した．漸進的なチャート解析の解析途中の段階で随時，誤り修正を実行することに

表 1: 文法的不適格文 "I think by train is best." に対する漸進的な解析過程

入力語	チャート				枝刈りの
word	#	*loc*	*term*	cost	有無
	1	0-0	$[?]_s$	0	
I	2	0-1	$[[[I]_{pron}]_{np}[?]_{vp}]_s$	0	
think	3	0-1	$[[[*]_n]_{np}[?]_{vp}]_s$	1	×
	4	0-2	$[[[I]_{pron}]_{np}[[think]_{vt}[?]_s]_{vp}]_s$	0	
	5	0-2	$[[[I]_{pron}]_{np}[[think]_{vi}[?]_{pp}]_{vp}]_s$	0	
by	6	0-2	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[?]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	7	0-3	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[?]_{np}]_{pp}]_{vp}]_s$	0	
	8	0-3	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[?]_{np}]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
train	9	0-3	$[[[I]_{pron}]_{np}[[think]_{vt}[?]_s]_{vp}]_s$	1	
	10	0-3	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{det}[?]_n]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	11	0-3	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[*]_{det}[?]_n]_{np}]_{pp}]_{vp}]_s$	1	
	12	0-4	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[train]_n]_{np}]_{pp}]_{vp}]_s$	0	
	13	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[train]_n]_{np}]_{pp}]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	14	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[train]_n]_{np}[?]_{vp}]_s]_{vp}]_s$	1	
	15	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[by]_{det}[train]_n]_{np}[?]_{vp}]_s]]_{vp}]_s$	1	
	16	0-4	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[*]_{det}[train]_n]_{np}]_{pp}]_{vp}]_s$	1	×
is	17	0-4	$[[[I]_{pron}]_{np}[[think]_{vi}[[by]_p[[[*]_{pron}]_{np}]_{pp}]_{vp}]_s$	1	×
	18	0-4	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[*]_{pron}]_{np}]_{pp}]_{np}[?]_{vp}]_s]]_{vp}]_s$	2	×
	19	0-5	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[train]_n]_{np}]_{pp}]_{np}[[is]_{be}[?]_{adj}]_{vp}]_s]]_{vp}]_s$	1	
	20	0-5	$[[[I]_{pron}]_{np}[[think]_{vt}[[[train]_n]_{np}[[is]_{be}[?]_{adj}]_{vp}]_s]]_{vp}]_s$	1	
	21	0-5	$[[[I]_{pron}]_{np}[[think]_{vt}[[[by]_{det}[train]_n]_{np}[[is]_{be}[?]_{adj}]_{vp}]_s]_{vp}]_s$	1	
best	22	0-6	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{gi}[[by]_p[[train]_n]_{np}]_{pp}]_{np}[[is]_{be}[best]_{adj}]_{vp}]_s]_{vp}]_s$	1	
	23	0-6	$[[[I]_{pron}]_{np}[[think]_{vt}[[[train]_n]_{np}[[is]_{be}[best]_{adj}]_{vp}]_s]_{vp}]_s$	1	
	24	0-6	$[[[I]_{pron}]_{np}[[think]_{vt}[[[*]_{det}[train]_n]_{np}[[is]_{be}[best]_{adj}]_{vp}]_s]_{vp}]_s$	1	

より, それまでの入力に対する統語構造を作成できることを例を以って示した.

本手法では, 入力を漸進的に解析する過程において作成した統語構造をすべて保持するため, 結果として統語構造が爆発的に増大する恐れがある. 本稿では, この問題に対して有用でない統語構造を枝刈りする方法を提案した. より効率的な解析のために, 統語構造の作成を制御する枠組を導入することは, 今後の課題である.

著者らはこれまでに, 文法的不適格文に対して誤り修正を行うことにより翻訳正解率が向上することを確認している [4]. 現在, 本稿で提案した手法を漸進的な英日話し言葉翻訳システム [8] に適用することを検討している.

参考文献

[1] 今井 宏樹, Theeramunkong, T., 奥村 学, 田中 穂積 : 一般化 LR 構文解析法による文中の複数箇所の誤りの検出と修正, 言語処理学会第 2 回年次大会, 153-156 (1996).

[2] Inagaki, Y. and Matsubara, S.: Models for Incremental Interpretation of Natural Language, *Proc. of 2nd Symposium on Natural Language Processing*, 51-60 (1995).

[3] 加藤 恒昭 : 一般化弧を用いた A* 探索による非文の解析, 情処学論, Vol.36, No.10, 2343-2352 (1995).

[4] 加藤 芳秀, 松原 茂樹, 浅井 悟, 外山 勝彦, 稲垣 康善 : 話し言葉における文法的不適格文に対する漸進的翻訳手法, 情報処理学会第 55 回全国大会 (2), 43-44 (1997).

[5] Kay, M : Algorithm Schemata and Data Structures in Syntactic Processing, *Technical Report CSL-80-12*, Xerox PARC (1980).

[6] Lee, K.J., Kweon, J.K., Seo, J. and Kim, G.C.: A Robust Parser Based on Syntactic Information, *Proc. of the 7th Conf. of European Chapter of the Association for Computational Liguistics*, 223-228 (1995).

[7] Matsubara, S., Asai, S., Toyama, K. and Inagaki, Y.: Chart-based Parsing and Transfer in Incremental Spoken Language Translation, *Proc. of the 4th Natural Lnaguage Processing Pacific Rim Symposium*, 521-524 (1997).

[8] 松原 茂樹, 浅井 悟, 外山 勝彦, 稲垣 康善 : 不適格表現を活用した漸進的な英日話し言葉翻訳手法, 電学論, Vol.118-C, No.1, 71-78 (1998).

[9] Mellish, C.S.: Some Chart-Based Techniques for Parsing Ill-Formed Input, *Proc. of 27th Conf. of Association for Computational Linguistics*, 102-109 (1989).

[10] 斎藤 博昭 : 一般 LR 構文解析法におけるエラー処理, 情処学論, Vol.37, No.8, 1506-1513 (1996).