自 然 言 語 処 理　130－8
（1999．3．5）

複数の接続制約を取り扱う PGLR 法について

今井 宏樹　白井 清昭　田中 穂積
東京工業大学大学院 情報理工学研究科
{imai,kshirai,tanaka}@cs.titech.ac.jp

本論文では，解析木の生成確率を計算する PGLR 法に複数の接続制約を組み込む手法を提案する．本手法は，文脈自由文法 (CFG) に基づく大域的な制約と，記号間の接続表に基づく局所的制約を GLR 法の枠組の中で統合して扱う制約統合型アプローチの発展形である．複数の接続制約を同時に扱うことで，音声認識，形態素解析，構文解析を GLR 法の上で同時に行なうことが可能となる．ATR 対話コーパスを用いた実験では，本手法は従来の手法に対して，解析精度を落とさずに 1 文当たりの平均解析木数を約 20 万分の 1 に削減することができた．

A PGLR Parsing Method with Multi-level Connection Constraints

IMAI Hiroki　SHIRAI Kiyoaki　TANAKA Hozumi
Graduate School of Information Science and Engineering
Tokyo Institute of Technology
{imai,kshirai,tanaka}@cs.titech.ac.jp

This paper proposes a method of incorporating the probabilistic generalized LR (PGLR) parsing method with multi-level connection constraints. Our method is a further development of an integrated constraint approach, which incorporates global constraints such as CFGs and local constraints such as connection matrices between terminal symbols into a single GLR parsing framework. Incorporation of multi-level connection constraints enables to treat speech recognition, morphological analysis and syntactic parsing in the GLR framework. Experiments with the ATR dialog corpus show that our method maintained parsing accuracy with a reduction of about 1/200,000 in terms of the number of average parse trees.

1　はじめに

近年，大規模なテキストコーパスの利用が容易となり，自然言語処理において種々の統計的手法が研究されるようになった．統計情報を用いた構文解析もその中の一手法である．

一方で，従来の規則に基づいたアプローチと統計的アプローチはお互いに切り離せない関係にあることが指摘されている．この観点から 一般化 LR 法 (GLR) [9] の枠組の中に統計情報を組み込む試みがいくつかなされている [2, 3, 7, 11]．

田中らは，形態素の接続制約を LR 表に組み込む手法を提案した [8]．この手法は，接続表で表される局所的制約と，文脈自由文法 (CFG) が内包する大域的な制約を統合するものである．Li らは，音素レベルの接続制約を導入することにより，この手法を音声認識に応用した [5]．彼らの指摘によれば，局所的制約を CFG の枠組で全て表現しようとすると，規則数が組み合せ的に増大するため実用に向かず，LR 表の枠組で統合するのが良いとされている．このような手法は制約統合型アプローチと呼ばれ，形態素解析，構文解析，音声認識の各要素技術を独立に扱うカスケード型やインターリーブ型のアプローチと相対する関係にある．カスケード型，インターリーブ型のアプローチは各モジュールで全ての曖昧性を保持しながら処理を進めなければならないため，統合

された枠組の中で早期に不要な曖昧性を除去できる制約統合型アプローチの方が規則に基づくアプローチの中では望ましいと考えられる．

また，確率付き GLR 法に関する研究が行なわれてきた．それらは，自然言語処理や音声認識用の曖昧性を含んだ文法から生成される膨大な数の解析候補の中から尤もらしいものを選びだすことを目的としている．Sornlertlamvanich らは，その中でも PGLR モデルが実験的に最も良いことを示した [7]．

これらの背景を考慮すれば，制約統合型アプローチと PGLR モデルを組み合せると，自然言語処理や音声認識の分野により貢献できると期待される．そこで我々は，本論文で以下の 2 つを提案する．(1) 形態素解析，構文解析，音声認識を統合して扱うために複数レベルの接続制約を GLR の枠組の中へ同時に組み込む手法，(2) 複数レベルの制約が組み込まれた LR 表に PGLR モデルの確率値を割り当てる方法．また，これらの提案手法を用いてコーパスの構文解析実験を行ない，尤もらしい解析候補のみを効率よく取り出せることを示す．

なお，以後の手法の説明では，紙面の都合により LR 法，GLR 法に関する基礎的な説明は割愛する．LR 法に関しては文献 [1] を，GLR 法に関しては文献 [9, 10] をそれぞれ参照されたい．

2 PGLR 法

PGLR モデルは，LR 表中の各 shift 動作，reduce 動作に対してその生起確率を与える形で定義される確率モデルである．GLR 法 [9] は，構文的曖昧性を含む一般の CFG も扱えるように LR 法 [1] を拡張したものであり，与えられた CFG からあらかじめ LR 表と呼ばれるプッシュダウンオートマトンを作成し，LR 表に記述された動作にしたがって解析を行なう．GLR 法では，初期状態から解析成功までに実行された一連の動作により生成される状態遷移系列が 1 つの解析木に相当する．したがって，状態遷移系列の生起確率が構文木の生成確率と等価になる．

Inui らは，このような考え方に基づいた PGLR モデル [3] を提案している．同様な確率モデルは Briscoe ら [2] により提案されているが，彼らの手法には確率の正規化に問題があり，Inui らの手法は正しい確率的定義を与えている．

ここでは PGLR の概要のみを例を挙げて説明する．Briscoe らのモデルを始めとする他の確率モデルとの比較については，理論的な考察を Inui ら [3] が，コーパスを用いた実験による比較を Sornlertlamvanich ら [7] がそれぞれ示しているので，そちらを参照されたい．

パーザのスタックの状態遷移列を T とする．T は式 1 のように表せる．

$$\sigma_0 \overset{l_1,a_1}{\Longrightarrow} \sigma_1 \Longrightarrow \cdots \overset{l_{n-1},a_{n-1}}{\Longrightarrow} \sigma_{n-1} \overset{l_n,a_n}{\Longrightarrow} \sigma_n \qquad (1)$$

ここで，σ_i, l_i, a_i はそれぞれスタックの状態，先読み記号，実行された動作を表している．これを用いて，状態遷移列 T の生成確率 $P(T)$ は

$$P(T) = P(\sigma_0, l_1, a_1, \sigma_1, \ldots, l_n, a_n, \sigma_n) \qquad (2)$$

と表せる．PGLR モデルでは，各解析ステップの実行される確率は，直前のスタックの状態のみに依存するという仮定を導入し，式 (2) を以下のように近似する．

$$P(T) \approx P(\sigma_0) \cdot \prod_{i=1}^{n} P(l_i, a_i, \sigma_i | \sigma_{i-1}) \qquad (3)$$

さらに，LR 表においては，

- l_i と a_i が決まれば σ_i は必ず一意に決まる．
- reduce 動作では先読み記号を消費せず，直前の動作と同じ先読み記号が用いられる．よって，reduce 動作では，先読み記号を予測する必要がない．

という 2 つの特徴があるため，式 (3) の各条件付き確率の推定は式 (4),(5) で行なうことができる．

$$P(l_i, a_i, \sigma_i) \approx P(l_i, a_i | s_{i-1}) \quad (s_{i-1} \in \boldsymbol{S_s}) \ (4)$$
$$P(l_i, a_i, \sigma_i) \approx P(a_i | s_{i-1}, l_i) \quad (s_{i-1} \in \boldsymbol{S_r}) \ (5)$$

ただし，$\boldsymbol{S_s}$ は shift 動作直後に遷移する状態の集合，$\boldsymbol{S_r}$ は reduce 動作直後に遷移する状態の集合をそれぞれ表す．

以下に，上記の定義を表 1 に示す文法を用いて具体的に説明する．

文法 G_1 から導出できる解析木は，図 1 に示される 2 種類がある．各解析木が角括弧内に書かれている回数だけそれぞれ学習データに出現しているとする．また，丸で囲まれた数字は各動作実行後のスタックトップの状態番号を表す．この時，PGLR モデルで計算される確率付き LR 表は表 2 のようになる．

表 1: サンプル文法 G_1

(1) $S \longrightarrow X\ u$
(2) $S \longrightarrow X\ v$
(3) $X \longrightarrow x$

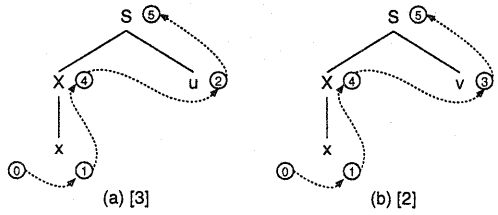

図 1: G_1 から導出可能な解析木

ここで，各動作の右に書かれている数字は学習データ全体で使用された回数を，下に書かれている数値はその動作の実行確率をそれぞれ表す．

表 2: PGLR モデルで生成された G_1 に対する LR 表

state	action				goto	
	u	v	x	\$	X	S
0 (S_s)			sh1 (5) 1		4	5
1 (S_s)	re3 (3) 0.6	re3 (2) 0.4				
2 (S_s)				rel (3) 1		
3 (S_s)				re2 (2) 1		
4 (S_r)	sh2 (3) 1	sh3 (2) 1				
5 (S_r)				acc (5) 1		

表 2 からわかるように，S_s に属する状態 1 では状態全ての動作で確率が正規化されているのに対し，S_r に属する状態 4 では各先読み記号ごとに確率が正規化されているため各動作に割り当てられた実行確率はそれぞれ 1 となっている．また，図 1 の各構文木の生成確率はそれぞれ以下のように計算される．

$$P(\mathrm{tree}(a)) = 1 \times 0.6 \times 1 \times 1 = 0.6 \quad (6)$$

$$P(\mathrm{tree}(b)) = 1 \times 0.4 \times 1 \times 1 = 0.4 \quad (7)$$

3 複数レベルの接続制約を扱うための GLR 法の拡張

本節では，綾部らが提案した手法 [12] の不備を補い，改良したアルゴリズムを示す．まず，3.1 節で本手法の基礎となる CFG の層と呼ばれる概念を説明する．3.2, 3.3 節では LR 表の生成アルゴリズム，解析アルゴリズムをそれぞれ示す．最後に，3.4 節で例を用いて複数の制約を LR 表に同時に組み込むことの利点を述べる．

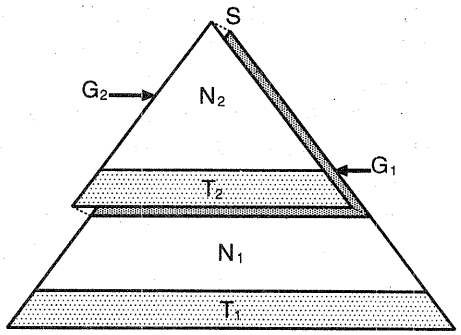

図 2: CFG の層

3.1 文脈自由文法の層

与えられた CFG から生成される LR 表に複数の制約を同時に組み込むための条件として，その CFG が層を持つことが必要である．

CFG $\boldsymbol{G_1} = (\boldsymbol{N_1}, \boldsymbol{T_1}, \boldsymbol{S}, \boldsymbol{P_1})$ $(\boldsymbol{S} \in \boldsymbol{N_1}, \boldsymbol{N_1} \cap \boldsymbol{T_1} = \emptyset)$ が与えられた時，以下の条件を満たすような CFG $\boldsymbol{G_2} = (\boldsymbol{N_2}, \boldsymbol{T_1}, \boldsymbol{S}, \boldsymbol{P_2})$ $(\boldsymbol{S} \in \boldsymbol{N_2}, \boldsymbol{N_2} \cap \boldsymbol{T_2} = \emptyset)$ が存在する場合，$\boldsymbol{G_1}$ は $\boldsymbol{G_2}$ を層に持つと定義する．

$$\boldsymbol{N_2} \subset \boldsymbol{N_1} \quad (8)$$

$$\boldsymbol{T_2} \subset \boldsymbol{N_1} \quad (9)$$

$$\boldsymbol{P_2} \subset \boldsymbol{P_1} \quad (10)$$

$$\forall\alpha\forall\beta, \alpha \in \boldsymbol{T_2}^* \wedge \beta \in \boldsymbol{T_1}^* \wedge \alpha \Rightarrow \beta \quad (11)$$

この関係は図 2 のように表すことができる．$\boldsymbol{G_1}$ の中に $\boldsymbol{G_2}$ が包含され，$\boldsymbol{T_1}$, $\boldsymbol{N_1}$, $\boldsymbol{T_2}$, $\boldsymbol{N_2}$ が層をなして，任意の $\boldsymbol{T_2}$ レベルの記号列は必ず $\boldsymbol{T_1}$ レベルの記号列に展開される．

なお，この関係はさらに再帰的に定義可能であることに注意されたい．上の例では簡単のため 2 層の状態を示したが，$\boldsymbol{G_2}$ に対しても同様の関係を持つ $\boldsymbol{G_3}$ を定義することで 3 層の CFG を定義できる．さらに同様の手順を繰り返すことにより，任意の数の層を持つ CFG を定義可能である．

3.2 LR 表生成アルゴリズム

ここでは，既存のアルゴリズムとの相違点を中心に要約して手順を示す．

1. G_2 に対して，既存の LR 表生成アルゴリズムを用いてクロージャ展開を行ない，GOTO グラフを生成する．
2. 規則集合 $\{P_1 - P_2\}$ を用いて，T_2 から T_1 を導出する部分のクロージャ展開を行ない，GOTO グラフを拡張する．
   その際，T_2 レベルの先読みを持ち，かつドットが右辺の最右端に到達しているアイテム $X \rightarrow \alpha\cdot; v_2$ $(v_2 \in T_2)$ が存在する状態に対して，先読み記号 v_2 を展開するアイテム $v_2 \rightarrow \cdot\beta; v_1$ $(v_1 \in T_1)$ を追加することに注意．
3. 既存の LR 表生成アルゴリズムを用いて GOTO グラフから LR 表を作成する．
4. 3. でできた LR 表から，制約伝播アルゴリズム [5] を使用して，与えられた T_1, T_2 の各レベルの接続制約を満たさない動作を除去し，LR 表を圧縮する．

3.3 解析アルゴリズム

基本的な解析アルゴリズムは今までの GLR 法のものと同じであるが，複数の制約を LR 表に組み込むために T_2 に属する記号を先読みとする動作を定義したため，その部分の処理のためにアルゴリズムに少々変更が必要となる．具体的な変更点は以下の通りである．

- $X \rightarrow \alpha (X \in T_2)$ なる規則を reduce する時には，スタックをポップした後に通常行なう X をスタックに積む goto 動作を行なわず，代わりに X を入力スタックに押し戻して X を先読みとする動作を引続き実行する．
- T_2 に属する記号 Y を shift する時には，Y をスタックに積んだ後にさらにもう 1 語先読みを行ない[1]，解析動作を継続する．

3.4 LR 表生成・解析の例

この節では，表 3 に示す動詞を構成する文法を例に提案したアルゴリズムを説明する．

文法 G_2 は，終端記号が「か」「け」「る」のような文字となっており[2]，文字列を導出するための非終端記号が五段動詞語幹，一段動詞語幹，五段動詞語尾，一段動詞語尾，の 4 種の細品詞となっている．この例では，開始記号から文字列を導出する CFG (表 3 中の G_{21}) は開始記号から細品詞列を導出する CFG (表 3 中の G_{22}) を包含し，2 つの層を形成している．

また，G_2 に対する T_1 と T_2 の接続制約がそれぞれ表 4，表 5 で表されているとする．この接続制約を LR 表に組み込む場合を考える．

まず，文字レベルの制約 (T_1) のみを組み込んだ LR 表を Li の LR 表生成アルゴリズム [5] を用いて作成すると，表 6 のようになる．しかしながら，この LR 表に細品詞の接続制約をそのまま組み込むことは不可能である．以下に例を挙げて説明する．

図 3 は G_2 から導出可能な解析木である．(c), (d) の解析木は，細品詞のレベルでの接続に矛盾がある．表 5 の制約を LR 表に組み込むことができれば，この 2 つの木の生成を抑制できる．しかし，(a) と (c) の解析の流れを比較すると，「動詞語幹」まで部分木が構成された時点でのパーザの状態がいずれも同じ状態 4 になってしまうため，五段動詞語幹，一段動

表 3: サンプル文法 G_2

G_{21}

G_{22}

(1) 動詞 ⟶ 動詞語幹 動詞語尾
(2) 動詞語幹 ⟶ 五段動詞語幹
(3) 動詞語幹 ⟶ 一段動詞語幹
(4) 動詞語尾 ⟶ 五段動詞語尾
(5) 動詞語尾 ⟶ 一段動詞語尾
(6) 五段動詞語幹 ⟶ か
(7) 一段動詞語幹 ⟶ か
(8) 五段動詞語尾 ⟶ け
(9) 一段動詞語尾 ⟶ け る

表 4: 文字レベル (T_1) の接続表

	か	け	る	$
か	0	1	0	0
け	0	0	1	1
る	0	0	0	1

表 5: 細品詞レベル (T_2) の接続表

	五段動詞語幹	一段動詞語幹	五段動詞語尾	一段動詞語尾	$
五段動詞語幹	0	0	1	0	0
一段動詞語幹	0	0	0	1	0
五段動詞語尾	0	0	0	0	1
一段動詞語尾	0	0	0	0	1

[1] この時の先読み記号は T_1 に属していることに注意．
[2] 本節での説明のために終端記号を文字としている．単語や音素を終端記号の単位としても構わない．

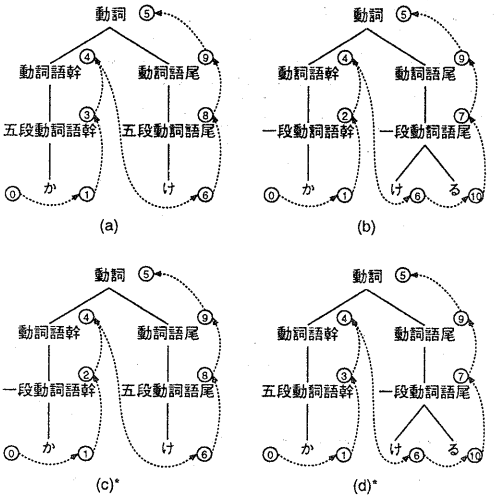

図 3: G_2 から導出可能な解析木

詞語幹のどちらから木が組み上げられたのか区別がつかなくなってしまう．また，(a) と (d) の解析を比較した場合，動詞語幹の部分木構成直後の先読み記号は文字の「け」であり，細品詞ではない．よって，五段動詞語尾と一段動詞語尾のいずれの細品詞が後続するのかは，この時点で判断できない．この 2 点により，今までの LR 表，LR パーザの枠組みでは，一段動詞語幹と五段動詞語尾，もしくは五段動詞語幹と一段動詞語尾がこの順に接続不可能である，という情報を組み込むことができない．

しかしながら，本手法を用いれば，文法の中間層のレベルの接続制約を組み込むことができる．図 4 は G_2 から本手法で LR 表を生成する過程の GOTO グラフを示している．網掛け部は手順 2. で拡張された部分である．この時点で，細品詞 ($\boldsymbol{T_2}$) レベルの制約が組み込まれ，I_0 のアイテムから不必要な先読みが削除される．表 7 はこの GOTO グラフから生成される LR 表である．この LR 表と 3.3 節に示した解析アルゴリズムを用いると，図 3 の (a) から (d) の各構文木を生成する解析の流れは図 5 のようになる．この例では，$\boldsymbol{T_2}$ に属する五段動詞語幹，一段動詞語幹に対して shift 動作が定義されて動作実行後の状態が 1 と 2 にそれぞれ分かれたことによって，木 (c), (d) の解析における先読み「け」に対する動作を区別することが可能となっている．そして，それぞれ文字「け」を shift した状態においてどちらも後続する動作がないため，(c), (d) の解析に失敗し，解析木の生成が抑制される．

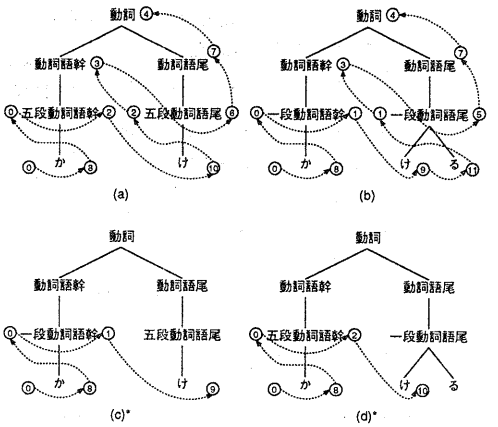

図 5: 2 つの接続制約を組み込んだ LR 表による解析例

本節で示した例では，文字レベル，細品詞レベルの 2 つのレベルの接続制約が同時に LR 表に含まれている．この場合，すでに書かれたテキストを対象とした形態素解析への応用となるが，音素を終端記号とする単語辞書規則を導入して音素レベル，細品詞レベルの接続制約を同時に用いれば音声認識への応用も可能である．

4 複数の接続制約を含む LR 表への PGLR 確率モデルの適用

この節では，2 節で示した PGLR 確率モデルの定義をほとんど変更することなく，複数の接続制約を含んだ LR 表にも応用可能であることを示す．

3 節で示した手法で作成される LR 表が従来の LR 表と大きく異なる点は，文法全体 ($\boldsymbol{G_1}$) から見た時には非終端記号となる $\boldsymbol{T_2}$ に属する記号を先読みとする動作が定義されていることである．PGLR モデルは全ての shift 動作，reduce 動作に対して生起確率を与える確率モデルのため，$\boldsymbol{T_2}$ に属する記号を先読みとする動作にも確率を与えなければならない．しかしながら，$\boldsymbol{T_1}$ に属する記号を先読みとする動作と $\boldsymbol{T_2}$ に属する記号を先読みとする動作が，LR 表中の同じ状態に混在する状況は，どんな LR 表にも現れる．したがって，確率を与える際の正規化をどのように行なうかが問題となる．

しかし，以下の事実により，$\boldsymbol{T_2}$ に属する記号をを先読みとする動作は全て状態と先読みで正規化すればよいことがわかる．

表 6: G_2 から通常のアルゴリズムで生成される LR 表

	action				goto						
	T_1				T_2				N_2		
	か	け	る	$	五段動詞語幹	一段動詞語幹	五段動詞語尾	一段動詞語尾	動詞語幹	動詞語尾	動詞
0	sh1				3	2			4		5
1		re6/re7									
2		re3									
3		re2									
4		sh6					8	7		9	
5				acc							
6			sh10	re8							
7				re5							
8				re4							
9				re1							
10				re9							

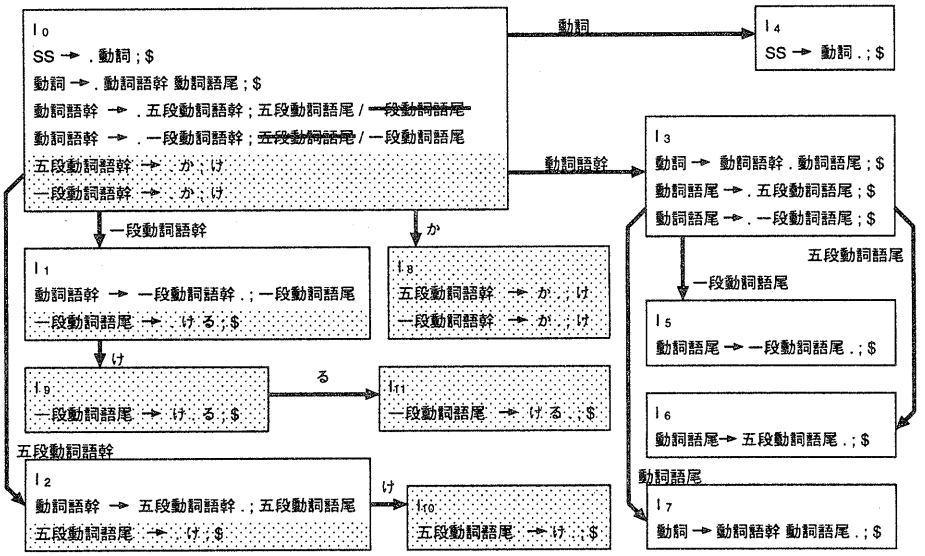

図 4: 複数の接続制約を扱う LR 表生成アルゴリズムで作成した，G_2 に対する GOTO グラフ

表 7: 図 4 の GOTO グラフから生成される G_2 に対する LR 表

	action								goto		
	N_1			T_2					N_2		
	か	け	る	五段動詞語幹	一段動詞語幹	五段動詞語尾	一段動詞語尾	$	動詞語幹	動詞語尾	動詞
0	sh8			sh2	sh1				3		4
1		sh9					re3				
2		sh10				re2					
3						sh6	sh5			7	
4								acc			
5								re5			
6								re4			
7								re1			
8		re6/re7									
9			sh11								
10								re8			
11								re9			

事実 1: T_2 を先読みとする動作が定義されている状態は，初期状態を除いて全て S_r に属する．

証明: もし S_s に含まれる状態に T_2 を先読みとする動作が存在すると仮定すると，T_2 は T_1 に属していることになるが，これは式 (9) と G_1 の定義から導かれる $T_1 \cap T_2 = \emptyset$ に反する． □

初期状態は例外であるが，T_1 を先読みとする動作は状態のみで正規化し，T_2 を先読みとする動作は状態と先読みで正規化する (すなわち，T_2 を先読みとする動作の確率値は全て 1 とする) ことでよい．

5 評価実験

5.1 評価方法

ATR の音声対話データベース (SLDB，以下 ATR コーパスと呼ぶ) [6] を使用した実験を行ない，本手法の有効性を評価した．ATR コーパスは 618 旅行対話を収録した約 21,000 文に対して形態素情報と構文木が付与されている．

また，田中らがこのコーパス用に開発した日本語句構造文法 [13] を使用した．この文法は 441 種の細品詞を終端記号とする 859 規則からなる．我々は，コーパスから獲得した 5,222 単語の辞書規則をこの文法に追加し，単語レベル，細品詞レベルの 2 層を持つ文法を作成した．文法の緒元を表 8 に示す．

表 8: 実験に使用した文法

規則集合	P_2	$\{P_1 - P_2\}$	P_1
規則数	859	5,222	6,081
終端記号 (単語) 数	—	4,791	4,791
平均規則長	1.39	1.00	1.05

まず，コーパスから 1 形態素文 (「はい」などの相づちが含まれる) を実験データから除去し，残りの文を句構造文法を用いて GLR パーザで解析した．受理できた文のうち，コーパスの形態素・構文情報から半自動的にもしくは人手で正解構文木を付与できた 9,794 文を今回の実験に使用した．この評価データの文の長さ分布を図 6 に示す．

次に，単語レベルの制約のみを組み込んだ LR 表と単語レベル，細品詞レベルの制約を組み込んだ 2 種類の LR 表を作成した．そして，これらの LR 表に対して PGLR モデルによる確率値を付与した．

評価方法には，文正解率と生成解析木数を用いた．各文に付与された正解構文木が解析結果の上位 1 位，10 位に含まれた文をそれぞれ正解として評価した．

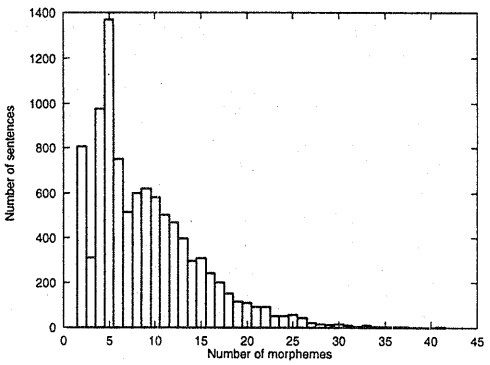

図 6: 評価データの文の長さ分布

さらに，他の確率モデルと比較するため，PCFG モデル [11] についても，PGLR モデルと同様，単語接続表のみ，単語と細品詞の 2 つの接続表を含んだ LR 表をそれぞれ作成し，評価した．

また，評価データの規模が小さいため，データを 10 セットにランダムに分割し，クロスバリデーションにより評価した．データ不足を補うため，各手法においてフロアリングによるスムージングを行ない，フロア値を変化させて，もっとも良い結果を採用した．LR 表の各動作へのフロア値の与え方として，以下の 3 種類を試みた．

(1) LR 表中の全ての動作に一律に一定のフロア値を与える．

(2) 競合を起こしている部分のみ 0 頻度の動作を削除し，残りの全ての動作に一律に一定のフロア値を与える．

(3) 0 頻度の動作は全て削除し，フロアリングは行なわない．すなわち，スムージングを行なわない．

5.2 結果と考察

表 9 に正解率の実験結果を示す．1-con, 2-con は LR 表に組み込まれた制約の数を表し，括弧内の数字はフロアリング手法を表している．PCFG では，学習時に使われた LR 表の動作でなく規則を数えるため，手法 (2) は適用できない．各手法において，2 レベルの制約を組み込んだ LR 表が，1 レベルのものより良い正解率を出していることがわかる．確率モデル間の比較では，PGLR の方が PCFG より良い結果となり，Sornlertlamvanich らの報告 [7] と一致した．

表 9: 各言語モデルの被服率と正解率 (単位: %)

モデル	被覆率	Top 1 正解率	Top 10 正解率
1-con PGLR (1)	100.0	76.7	94.4
2-con PGLR (1)	100.0	83.7	98.7
1-con PGLR (2)	99.8	77.1	94.0
2-con PGLR (2)	99.8	83.7	97.6
1-con PGLR (3)	60.1	91.3	99.1
2-con PGLR (3)	59.6	95.2	99.4
1-con PCFG (1)	100.0	58.5	90.8
2-con PCFG (1)	100.0	81.0	98.4
1-con PCFG (3)	91.2	60.3	91.6
2-con PCFG (3)	90.8	82.6	98.4

表 10: 各手法ごとの最大および平均構文解析木数

手法	最大	平均
1-con + (1)	2.06×10^{9}†	8.17×10^{6}†
2-con + (1)	1.11×10^{5}	45.8
1-con + (2)	3.25×10^{6}	1.04×10^{3}
2-con + (2)	9.63×10^{3}	4.23
1-con + (3)	264	2.94
2-con + (3)	24	1.35

次に，表 10 は各フロアリング手法で得られる解析木の数を示している[3]．†で示した値は解析木数が 4 バイト整数型で扱える最大値を超えた 69 文を除いて計算した．細品詞の制約を LR 表に組み込むことにより，平均解析木数が大幅に削減されることがわかる．特に，LR 表の動作の削除が行なわれない手法 (1) に対する削減率が約 20 万分の 1 と顕著である．

PGLR モデルは尤もらしい解析候補により良い選好情報を与え，複数レベルの制約を扱う GLR 法により，制約を満たさない候補の生成が抑制される．この 2 つの利点を組み合せた本手法は，少ない解析コストで高い性能をあげていると言える．

6 まとめ

本論文では，複数レベルの接続制約を GLR 法の枠組に統合する手法，拡張 LR 表に PGLR モデルの確率値を割り当てる手法の 2 つを提案した．実験では，本手法により，単語と細品詞の接続制約を満たさない解析候補の生成が抑制され，かつ正しい解析木を効率よく取り出せることが示された．この結果から，制約統合型アプローチは，単語や細品詞を導出する文法に音素レベルの層を追加することにより，自然言語処理だけでなく，音声認識の分野にも有効であると期待される．

音声認識に対しても本手法が有効であるかを調査するため，現在 ATR コーパス用の辞書規則を異音レベルに展開した辞書規則を作成し，HMM-LR[4] に本手法を統合した音声認識システムを構築中である．このシステムを用いた音声認識実験の結果は，機を改めて発表する予定である．

[3]解析木数は，確率モデルの違いではなく，フロアリングの手法の違いによる LR 表から削除される動作数のみに依存することに注意．

謝辞

実験用のコーパス (SLDB) を提供下さった ATR の竹澤寿幸氏，および ATR コーパス用日本語文法を提供下さったランゲージウェアの衛藤純司氏に感謝致します．

参考文献

[1] A.V. Aho, R. Sethi, and J.D. Ullman. *Compilers: Principles, Techniques, and Tools.* Addison Wesley, 1986.

[2] T. Briscoe and J. Carroll. Generalized probabilistic LR parsing of natural language (corpora) with unification-based grammars. *Computational Linguistics*, Vol. 19, No. 1, pp. 25–59, 1993.

[3] K. Inui, V. Sornlertlamvanich, H. Tanaka, and T. Tokunaga. A new formalization of probabilistic GLR parsing. In *Proc. of 5th International Workshop on Parsing Technologies*, 1997.

[4] K. Kita, T. Morimoto, K. Ohkura, and S. Sagayama. Continuously spoken sentence recognition by HMM-LR. In *Proc. of ICSLP92*, pp. 305–308, 1992.

[5] H. Li and H. Tanaka. A method for integrating the connection constraints into an LR table. In *Proc. of NLPRS95*, pp. 703–708, 1995.

[6] A. Nakamura, S. Matsunaga, T. Shimizu, M. Tonomura, and Y. Sagisaka. Japanese speech databases for robust speech recognition. In *Proc. of ICSLP96*, Vol. 4, pp. 2199–2202, 1996.

[7] V. Sornlertlamvanich, K. Inui, K. Shirai, H. Tanaka, and T. Tokunaga. Empirical evaluation of probabilistic GLR parsing. In *Proc. of NLPRS97*, pp. 169–174, 1997.

[8] H. Tanaka, T. Tokunaga, and M. Aizawa. Integration of morphological and synatactic analysis based on LR parsing algorithm. In *Proc. of 3rd International Workshop on Parsing Technologies*, pp. 101–109, 1993.

[9] M. Tomita. *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems.* Kluwer Academic Publishers, 1986.

[10] M. Tomita, editor. *Generalized LR Parsing.* Kluwer Academic Publishers, 1991.

[11] J.H. Wright and E.N. Wrigley. GLR parsing with probability. In *[10]*, chapter 8, pp. 113–128. Kluwer Academic Publishers, 1991.

[12] 綾部寿樹, 徳永健伸, 田中穂積. 複数の接続表の制約の LR 表への組み込み – LR 表工学 (2) –. 情処研報,NL-117-10, pp. 67–74, 1997.

[13] 田中穂積, 竹澤寿幸, 衛藤純司. MSLR 法を考慮した音声認識用日本語文法 – LR 表工学 (3) –. 情処研報,SLP-15-25, pp. 145–150, 1997.