F_035

文脈自由文法の漸次学習のための準最適な規則集合探索の方式

Incremental Learning of Context Free Grammar by Searching for Semi-Optimum Rule Sets

杉田 雄大† 中村 克彦‡
Yudai Sugita Katsuhiko Nakamura

1 まえがき

最近，文脈自由文法に対する文法推論の研究が盛んに進められている．一般的な文脈自由文法の推論はもっとも多くの計算量を必要とするクラスに属する．このため，探索方式の改良による規則集合合成の高速化と共に暫時学習などの学習モデルの導入によって大きな規則集合を実際に生成できる方式が重要である．

われわれはブリッジ法と呼ばれる規則生成方式と，規則集合探索とを組み合わせた文脈自由文法の漸次学習方式をSynapseシステムに実装し，その改良を続けている．ブリッジ法による規則生成は，正例の文字列を上向きに構文解析して得られた不完全な導出木の欠けた部分を開始記号から下向きに探索し，これを補う規則を生成する．

この報告では，より複雑な文法をさらに効率よく求めるために，必ずしも最小ではない規則集合を高速に探索するための2つの方式を述べる．その一つは直列探索と呼ばれる規則集合探索方式であり，第二は生成される規則の形式に対して制限を与える方法である．直列探索によれば，一般に最小ではないが与えられた正例および負例を満足する規則集合をより高速に求めることができる．また，生成される規則の形式に制限をあたえ，探索空間を制限することによって，規則集合探索に要する時間を短縮できる．

2 ブリッジ法による規則生成

ブリッジ法による規則生成アルゴリズムは，正例の文字列を上向きに構文解析した結果，完全な導出木が生成されないとき，導出木の不足した部分を開始記号Sをもつ根から下向きに探索し，この部分を補うような規則を生成する．導出木の欠けた部分を"橋渡し"するように補うのでこの規則合成アルゴリズムをブリッジ法と呼ぶ．この方式は，以前のSynapseシステムで用いられていた帰納的CYKアルゴリズムに比べ，構文解析を行う回数が減少するため，一般に効率よく文法合成を行える．[1].

†東京電機大学大学院理工学研究科
‡東京電機大学理工学部

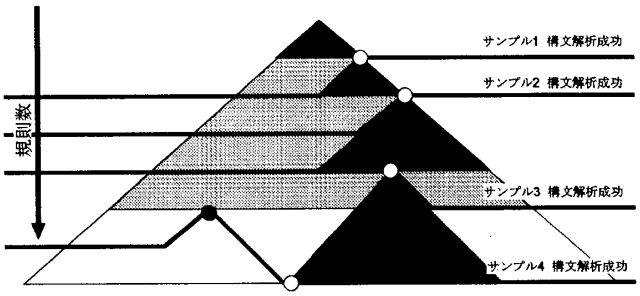

図 1: 大域探索と直列探索を表す探索木

3 規則集合の探索方式

Synapseシステムは，正負の記号列例の系列と初期規則集合を入力し，各正例に対して順に規則生成を行い，ある規則集合を発見したとき，これがすべての負例を導出しないことをチェックする．このようにして，すべての正例の記号列を導出し，どの負例の記号列をも導出しないような規則集合を出力する．このシステムには，次のような2つの探索方式が組み込まれている．

大域探索 反復深化によって最小の規則集合を求める．

直列探索 各正例に対して，これを導出する最小の部分規則集合を求める．

2つの方式の相違は，図1の探索木によって説明できる．正例の系列$w_1, w_2, \cdots, w_k$に対する探索木は次のように定義される．

1. 深さ0の節点である根は初期規則集合が対応する．
2. 深さ$j, (1 \leq j \leq k)$の節点には，正例w_jを導出するために追加された規則集合が対応する．根からこの節点までの経路に対応するすべての規則集合の和集合は正例$w_1, w_2, \cdots, w_j$のすべてを導出する．

大域探索によって最小の規則集合を求める場合には反復深化によってこの探索木のすべてが探索されるのに対し，直列探索においては図中の記号“○”で示す1つ

表 1: 合成された文法の合成時間と GR

		大域探索									直列探索		
		Minimum Set			Non-min.NT			Restricted			Restricted		
		R	Time	GR	R	Time	GR	R	Time	GR	R	Time	GR
(a)	A	4	0.07	32	4	0.05	13	5	0.16	47	5	0.05	47
	U	6	2.9	847	6	1.1	160	6	0.60	111	8	0.17	50
(b)	A	8	0.18	247	8	0.15	168	8	0.06	57	9	0.03	15
	U	8	0.19	247	8	0.13	168	8	0.07	57	8	0.08	36
(c)	A	10	210	20×10^4	10	3.5	417	10	3.2	368	14	0.24	19
(d)	A	–	–	–	10	2200	7.2×10^4	10	620	2.3×10^4	15	0.38	15
(e)	A	–	–	–	12	370	1.3×10^5	12	104	3.3×10^4	22	70	5.5×10^3
(f)	A	–	–	–	–	–	–	14	6.6×10^4	1.4×10^7	27	27	2.7×10^5

Minimum Set：非終端記号の数が最小な規則集合探索　　R： 生成規則数　　A：あいまいな文法学習

Non-min.NT：非終端記号の使用制限　　Time： 合成時間 (sec)　U：非あいまいな文法学習

Restricted：直接再帰の禁止，反転可能，リセットフリー　3つの制限を適用

の正例を導出する最小の部分規則集合を求め，それ以上の後戻りによる探索は行われない．したがって，直列探索では探索木の一つの経路が探索される．

4 規則の形式に対する制限

直列探索に加えて準最適な規則集合を高速に探索するもう一つの方法は，生成される規則の形式に次のような制限を与えることである．

非終端記号の使用制限 最小の規則集合を見つけるためには，ブリッジ法における規則生成手順において，規則に含まれる非終端記号の数を最小にする必要がある．$A \to BC$ なる規則が規則集合に存在し，新たに $B \to DE$ というような規則を追加する場合，B に大して既存の非終端記号を割り当てられるかどうか試みる．非終端記号の使用を制限した場合の規則合成はこの試みを省略し，単に新しい記号を非終端記号の集合に追加する．

直接再帰の禁止 $X \to X\gamma$ または，$X \to \beta X$ の形式ではない．

反転可能 $X \to \beta\gamma$ と $Y \to \beta\gamma$ の2つの規則を含まない．

リセットフリー $A \to X\gamma$ と $A \to Y\gamma$，または，$A \to \beta X$ と $A \to \beta Y$，の2つの規則を含まない．

5 文法合成の結果

Synapse システムは Hopcroft と Ullman による“オートマトン，言語理論，計算論 ”の文脈自由文法作成の練習問題となっている7つの文法を構成する課題に対して，例から直接文法を合成した．言語 (a) は平衡した括弧列の集合，(b) は回文の集合，(c) は正規表現によって表される文字列の集合，(d) は a とその2倍の数の b からなる文字列の集合，(e) は ww の形をもたない a, b の文字列の集合，(f) は $a^i b^j c^k$ で $i = j$ または $j = k$ である a, b, c からなる文字列の集合．表1は各言語について大域探索と直列探索によって規則集合を得るのに要した時間と合成された総規則数 (GR) である．

6 むすび

文脈自由文法の学習システム Synapse に組み込まれた直列探索と生成される規則の形式に制限を与える方式とその結果について述べた．これによると，どちらの方式も文法合成にかかる時間を短縮することができるが，得られる規則集合が大きくなることがある．言語によっては直列探索で得られる規則集合が非常に大きくなり，それに伴って長い計算時間を要する場合がある．今後はこれらの方式のさらに詳しい検証と Synapse システムが学習できる文法の拡張について検討と実験を進める予定である．

参考文献

[1] 中村 克彦:構文解析にもとづく規則生成と規則集合探索による文脈自由文法の漸次学習，人工知能学会論文誌，21巻4号 F，pp371-369,(2006)