文脈自由文法の有限オートマトン近似に基づく漸進的構文解析の高速化

湊 恵一† 加藤 芳秀† 松原 茂樹‡ 稲垣 康善†

†名古屋大学大学院工学研究科 ‡名古屋大学情報連携基盤センター

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

1 はじめに

実時間音声言語処理システムでは，ユーザの発話をその出現と共に順次理解する必要がある．そのような処理を実現するためには，発話の断片が入力されるごとに，順次，解析処理を実行する，漸進的な構文解析が必要である [1, 2, 3, 4]．漸進的構文解析手法としてこれまでに，漸進的チャート解析が提案されている [3]．漸進的チャート解析では，単語が入力されるたびに，それまでに入力された文の断片に対する構文木を生成する．これにより，文全体が入力されていなくても，その時点までの構文構造を明らかにすることができる．

しかし，この手法は，解析処理時間についてはこれまであまり考慮されておらず，現状の方式で，ユーザの発話を実時間で処理することは必ずしも容易ではない．

そこで本稿では，発話を同時進行的に処理する構文解析の実現を目的として，漸進的構文解析のための有限状態変換器 (Finite State Transducer; FST)[5] の構成手法，および，それを用いた構文木生成手法を提案する．漸進的チャート解析では，単語が入力されるたびに文法規則を組み合わせて構文木を生成するが，このような構文木をあらかじめ作成し，FST に記述しておくことにより，構文木を即時に出力でき，高速に漸進的構文解析を実行できる．

本手法では，文脈自由文法 (Context Free Grammar; CFG) を近似変換して FST を構成する．したがって，構成した FST で解析可能な文の集合は，もとの CFG で解析できる文の集合のサブセットであり，より多くの文を解析できるように，FST を構成する必要がある．そこで，生成頻度の高い構文木が記述された弧は，解析においてたどられる可能性が高いという，FST の弧と構文木の関係に着目して，そのような弧を優先的に作成する．その優先度は，構文木付きテキストコーパスから獲得した構文木生成確率をもとに計算する．

2 漸進的な有限状態構文解析

漸進的チャート解析は，CFG に基づく構文解析手法である．新たな単語が入力されるたびに，文法規則の適用を繰り返すことにより，その単語に対する可能な構文木を生成し，それをそれまでの入力に対して生成された構文木と結合することにより，漸進的に構文木を生成することができる．日本語文「今日 / 予約 / を / する」に対する漸進的チャート解析の過程を図 1に示す．図中の番号は，処理の順序を示しており，実線矢印は規則の適用を，点線矢印は構文木の結合を表す．例えば，品詞[1]が n である単語「予約」に対しては，図のように，“NP → n” から “S → VP” までの 4 つの文法規則を順に適用することにより (処理 (3) ～ (6))，その単語に対する構文木を作る．そして，「今日」の品詞である adv に対する構文木と結合することにより (処理 (7))，「今日 / 予約」に対する構文木を生成する．しかし，この手法では，単語が入力されるたびに文法規則を組み合わせて，新たに構文木を生成するため，解析処理に時間がかかる．使用する文法が同一であれば，同一の品詞に対して作成すべき構文木は，あらかじめ決まる．したがって，それを事前に計算しておき，解析時に利用すれば，効率的な構文解析が可能である．

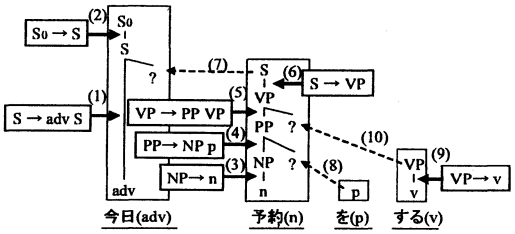

図 1: 漸進的チャート解析の解析過程

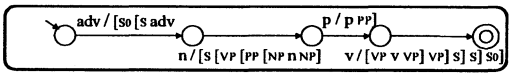

図 2: 漸進的構文解析に用いる FST の例

本研究では，このような漸進的構文解析を実現するために FST を利用する．FST は，有限状態オートマトンに出力ラベルを付加した状態遷移機械であり，状態を遷移するたびに，弧に割り当てられた記号を出力する [5]．漸進的構文解析を実現するために，弧の入力ラベルには入力単語を，出力ラベルにはその単語に対して出力すべ

[1]本研究では，品詞は範疇に含まれないものとし，品詞を終端記号，範疇を非終端記号として扱っている．

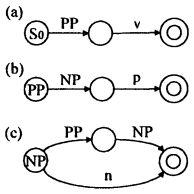

図 3: RTN の例

$S_0 \rightarrow PP\ v$
$PP \rightarrow NP\ p$
$NP \rightarrow PP\ NP$
$NP \rightarrow n$

図 4: CFG の例

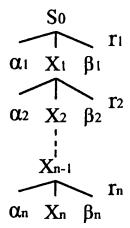

図 5: 文法規則の適用

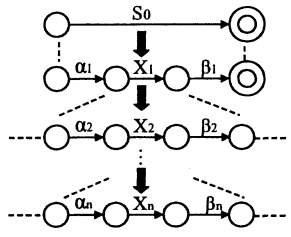

図 6: 文法規則の適用に対応するネットワークの置き換え

き構文木を割り当てる．解析では，入力に従って FST 上を遷移し，出力ラベルに付与された構文木を用いて漸進的に構文木を生成する．図 2に，図 1の解析過程に対応した FST の動きを示す．入力文「今日 / 予約 / を / する」の品詞系列 "adv n p v" に対して，順次，出力した木を単に結合することにより，その時点までの入力に対する構文木が生成できる．

3 FST の構成手法

本節では，漸進的な有限状態構文解析を実現する FST を構成する手法について述べる．本手法では，CFG を FST に変換することによりそれを実現する．CFG を再帰遷移ネットワーク (Recursive Transition Network; RTN) で表現し，ネットワークの弧を別のネットワークで置き換える操作を繰り返すことにより FST を作成する．これは CFG を近似したものであり，もとの CFG で解析可能な文であるからといって，必ずしも FST で解析できるわけではないが，本手法では，統計情報を用いて置き換え操作の優先度を計算することにより，カバレッジの高い FST を作成する．

3.1 CFG から FST への変換

RTN は，CFG をネットワークで表現したものであり，受理する言語はそれと等価である．RTN では，各非終端記号に対して，それぞれ対応するネットワークが一つ存在する．ネットワーク中の非終端記号をラベルにもつ弧は，その非終端記号に対応するネットワークにより再帰的に定義される．例えば，図 3 に示す RTN は，図 4 に示す CFG を表現する．S_0,PP,NP は非終端記号であり，n,p,v は終端記号である．S_0,PP,NP には，ネットワーク (a),(b),(c) がそれぞれ対応する．

RTN 中の非終端記号をラベルにもつ弧を，その非終端記号に対応するネットワークで置き換える操作を繰り返すことにより，一つの有限オートマトンを構成できる．一般に，この操作は無限に適用できるが，有限回の置き換えを実行すると，もとの RTN が受理する言語のサブセットを受理するようなオートマトンを構成できる．

上で述べた方法で構成したオートマトンに対して，適切な構文木を出力ラベルとして与えることにより，漸進的構文解析を実現する FST を構成できる．以下では，そのための手法について考える．

ネットワーク中の弧を，そのラベルに対応するネットワークで置き換える操作は，CFG における文法規則のトップダウンの適用とみなすことができる．すなわち，図 6のようなネットワークの置き換えは，図 5のような文法規則の適用と対応している．したがって，図 6中の X_n を入力ラベルにもつ弧を遷移することは，入力に対して図 5のような構文木を認識することに相当する．すなわち，このような構文木を，X_n を入力ラベルにもつ弧に対する出力ラベルとして与える．

本手法では，このように，弧のネットワークによる置き換えと文法規則のトップダウンな適用とを対応させ，FST 中の弧の出力ラベルを与える．

まず，本手法での構文木の表記法について述べる．本手法では，括弧を用いて構文木を次のように表現する．

$$[_A \alpha_1 \ldots \alpha_i \ldots \alpha_n\ _A] \tag{1}$$

ここで，A は非終端記号，$\alpha_i (1 \leq i \leq n)$ は括弧で表現した構文木である．これは，根が A で，その構成要素が $\alpha_1, \ldots, \alpha_i, \ldots, \alpha_n$ である構文木を示す．なお，以下では $[_A$ を左括弧付範疇，$_A]$ を右括弧付範疇とよぶ．

置き換え操作は以下の通りである．FST の弧の集合を E とし，弧 $e = (q_s, X, o_l X o_r, q_e) \in E$ を RTN 中の X に対応するネットワーク M_X で置き換えることにより，E' に更新する．ただし，弧 e は，状態 q_s から状態 q_e へと遷移し，入力ラベル X，出力ラベル $o_l X o_r$ を持つ．また，o_l は左括弧付範疇の系列，o_r は右括弧付範疇の系列を示す．M_X 中の弧の集合を E_X とする．E_X の要素は 3 項組 (始点の状態，ラベル，終点の状態) である．i_X, f_X を M_X の初期状態，最終状態とし，状態 q_1, q_2 はそれぞれ $q_1 \neq i_X, q_2 \neq f_X$ であるとする．

$$\begin{aligned} E' \quad = \quad & (E - \{e\}) \\ & \cup \{(eq_1, A, A, eq_2) \mid (q_1, A, q_2) \in E_X\} \\ & \cup \{(q_s, A, o_l[_X A, eq_2) \mid (i_X, A, q_2) \in E_X\} \\ & \cup \{(eq_1, A, A_X]o_r, q_e) \mid (q_1, A, f_X) \in E_X\} \\ & \cup \{(q_s, A, o_l[_X A_X]o_r, q_e) \mid (i_X, A, f_X) \in E_X\} \end{aligned} \tag{2}$$

置き換え操作の例を図 7に示す．図 7では，NP を入力ラベルとする弧を図 3(c) のネットワークで置き換えている．その結果，NP の代わりに，系列 "PP NP"，または "n" での遷移が可能となる．さらに，新たに作られた弧に対して出力ラベルを与える．置き換える弧の入力ラベルは NP，出力ラベルは "$[_{S_0}[_{PP}$ NP" であるから，式 (2) では，X =NP，$o_l = [_{S_0}[_{PP}$，$o_r = \varepsilon$ である．したがって，以下のように出力ラベルを与えればよい．

- PP をラベルに持つ弧は，弧の始点が図 3(c) の初期状態であるので，"$[_{S_0}[_{PP}[_{NP}$PP" を与える

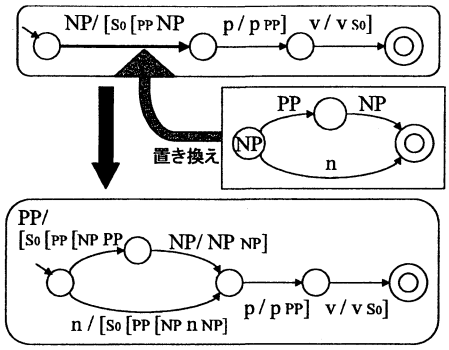

図 7: 置き換え操作の例

- NP をラベルに持つ弧は，弧の終点が図 3(c) の最終状態であるので，“NP $_{\text{NP}}$]” をそれぞれ与える．
- n をラベルに持つ弧は，弧の始点および終点がそれぞれ図 3(c) の初期状態および最終状態であるので，“[$_{\text{S}_0}$[$_{\text{PP}}$[$_{\text{NP}}$ n $_{\text{NP}}$]” を与える．

得られた FST では，「私 (n) は (p) 行く (v)」の単語ごとの入力に対して，

“[$_{\text{S}_0}$[$_{\text{PP}}$[$_{\text{NP}}$ n $_{\text{NP}}$]”，
“[$_{\text{S}_0}$[$_{\text{PP}}$[$_{\text{NP}}$ n $_{\text{NP}}$] p $_{\text{PP}}$]”，
“[$_{\text{S}_0}$[$_{\text{PP}}$[$_{\text{NP}}$ n $_{\text{NP}}$] p $_{\text{PP}}$] v $_{\text{S}_0}$]”

が順に出力される．

3.2 統計情報を用いた置き換える弧の選択

3.1節の手法により，漸進的構文解析を実現する FST を構成できる．ここで考えなければならない問題は，ネットワーク置き換え操作の適用が有限回であるために，構成された FST はもとの RTN を近似したものとなり，もとの RTN により解析できる文が必ずしも構成された FST により解析できるとはかぎらない点である．ネットワーク置き換え操作には任意性があるが，より多くの文を解析できる FST を構成する上で，優先的に置き換えを適用すべき弧とそうでない弧が存在すると考えられる．本節では，弧に対する置き換え優先度を導入し，優先度の高い弧から順に置き換え操作を実行する．

まず，どのような弧を優先して置き換えるべきかについて図 5 および図 6 を用いて考える．図 5 に示す構文木を解析時に生成するためには，それに対応する図 6 のような置き換えを実行しなければならない．図 5 に示す構文木の生起確率が高い場合，これに対応する図 6 の置き換えを実行すれば，より多くの文を解析できる FST を構成できる．逆に，図 5 に示す構文木の生起確率が低ければ，図 6 の置き換えを実行しても，その結果得られた弧は，文の解析能力に大きく寄与しない．そこで本手法では，図 6 の X_n をラベルにもつ弧の置き換え優先度の計算に，図 5 の構文木の生起確率を用いる．

本手法では，文法規則 $r_1, \ldots, r_n$ を順にトップダウンに適用した結果として得られる構文木の生起確率を次のように定義する．

$$P(T) = \prod_{i=1}^{n} P(r_i \mid r_{i-N+1}, \ldots, r_{i-1}) \qquad (3)$$

式 (3) が示すように，文法規則 r_i の適用確率は文脈 $r_{i-N+1}, \ldots, r_{i-1}$ に依存するとする．

しかし，このようにして求めた適用確率は，データのスパースネスの問題により，信頼性の高い値であるとは言い難い．そこで，線形補間法によりスムージングした適用確率を用いることとする．

3.3 置き換え操作の制限

本手法では，優先度の高い順に弧の置き換え操作を実行する．置き換え操作の適用を繰り返すにつれて，FST の規模も大きくなる．本手法では，使用できるメモリの容量に応じて，作成可能な弧の数に制限を設定し，それを越えない範囲で置き換え操作を繰り返す．

3.4 弧の除去

前節までに述べた FST 作成手法では，非終端記号を入力ラベルに持つ弧はそのまま FST 中に残される．しかし，このような弧は解析時に使用されることはなく，これらの弧を除去しても解析性能は変わらない．さらに，これらの弧を除去することによって，解析上，不要となる弧が生まれる．このような弧を FST から除去することにより，FST のサイズは小さくなり．その分だけ新たにネットワークの置き換えが可能となり，より多くの文を解析できるようになる．

すなわち，本手法では，以下のような弧を除去する．

1. 弧の数が制限を超えたとき，優先度の低い弧を FST から除去する．
2. 弧の除去の結果，成功経路を構成しなくなった弧を FST から除去する．

ここで，成功経路とは，FST の初期状態から最終状態までを結ぶ弧の系列である．2. において，弧が成功経路を構成しなくなったかどうかの判定は，次の通りである．状態 p から出て状態 q へ入る弧 e を除去するとき，p から出る弧が e の他に存在しなければ，p から他の状態へ出る遷移ができなくなるため，p へ入る全ての弧は成功経路を構成しない．同様に，q へ入る弧が e の他に存在しなければ，初期状態から q への経路が消滅するため，q から出る全ての弧は成功経路を構成しない．

3.5 FST 構成手順

3.1 節から 3.4 節で説明した，FST 構成の手順は以下の通りである．

1. 非終端記号を入力ラベルとする弧がなければ，置き換え操作を終了する．
2. 最も優先度が高い弧をネットワークで置き換える．
3. FST の弧の数が閾値を越えていなければ，1 へ．
4. 最も優先度が低い弧を除去する．その結果，成功経路を構成しない弧が現れたら，連続的に除去する．3 へ．

表 1: 本手法と漸進的チャート解析の解析結果

手法	解析時間 (秒)	文正解率 (%)
漸進的チャート解析	2.82	33.4
本手法	0.05	88.7

4 FST を利用した構文木生成

3 節で述べた手法により作成した FST を用いて，漸進的に構文木を生成する手法は以下の通りである．FST に単語を順に入力し，状態を非決定的に遷移することにより，構文木の出力を得る．そのとき，それ以前に入力された単語列に対する出力構文木を示す記号列に，遷移した弧に記述された出力ラベルを連接して，新しい構文木を生成する．

5 解析実験

5.1 実験の概要

実験では，ATR 音声言語データベース [8] に対して作成された構文木付きテキストコーパス[2]を用いた．学習データとして 9,081 文をランダムに抽出し，そこから文法規則とその適用確率を獲得した．使用した文法規則は 698 種類であり，品詞は 337 種類，範疇は 153 種類である．また，テストデータとして 1,874 文 (平均形態素数 9.4) を用い，それに付与されている構文木を正解とした．実験では，Pentium4 2GHz CPU，及び，2GB メモリを搭載した Linux ワークステーションを用いた．作成可能な FST の弧の数は，メモリ容量の制限から，15,000,000 を上限として設定した．

5.2 実験結果

本手法の性能を評価するために，漸進的チャート解析との比較を行った．ただし，式 (3) の N の値を 5 として計算し，形態素あたりの処理時間を最大 10 秒に制限した．

形態素あたりの解析時間，および，文正解率を表 1に示す．本手法による解析時間は平均 0.05 秒であり，漸進的チャート解析に比べ，解析速度が大幅に向上した．日本語の発話速度は，通常，形態素あたり 0.25 秒程度であるという報告があり [6, 7]，これは，本手法による漸進的構文解析が，話者発話に追随可能な解析速度を備えていることを意味している．一方，解析の正解率は 88.7% であり，話し言葉の構文解析としての本手法の利用可能性を確認した．なお，比較対象である漸進的チャート解析での正解率が低いのは，解析時間の制限によるものである．

次に，弧の置き換え優先度を導入する効果を測定するために，置き換え優先度を使用しなかったとき，および，式 (3) の N の値が 1 ～ 5 のそれぞれの場合について，正解率を測定した．結果を図 8に示す．優先度を用いることにより正解率が大幅に向上しており，置き換え戦略の重要性を示している．また，文脈を考慮して文法規則の適用確率を求めることにより，正解率が向上することを確認した．

[2]http://tanaka-www.cs.titech.ac.jp/pub/sldb-tree/

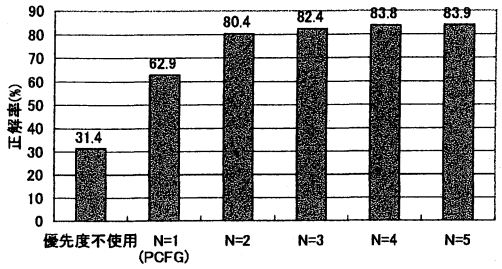

図 8: FST を用いた解析における正解率

また，非終端記号をラベルに持つ弧を除去する効果を調査した．N=5 のとき，弧を除去しないときの正解率は 83.9% であったのに対し，除去したときは 88.7% であった．

6 おわりに

本稿では，CFG から FST への近似変換手法を提案し，それに基づく漸進的構文解析について述べた．本手法では，漸進的構文解析の途中で出力する構文木をあらかじめ作成し，それを FST 上に記述する．解析では，入力にしたがって FST 上を状態遷移することにより，漸進的に構文木を生成する．実験の結果，高い正解率を達成可能な漸進的構文解析を，話者発話に追随可能な速度で実現できることを確認した．

参考文献

[1] 加藤 芳秀，松原 茂樹，外山 勝彦，稲垣 康善，“確率文脈自由文法に基づく漸進的構文解析”，電気学会論文誌，Vol. 122-C, No.12, pp.2109-2119 (2002)

[2] 加藤 芳秀，松原 茂樹，外山 勝彦，稲垣 康善，“主辞情報付き文脈自由文法に基づく漸進的な依存構造解析”，電子情報通信学会論文誌，Vol.86-D-II, No.1, pp.86-97 (2003).

[3] S. Matsubara, S. Asai, K. Toyama and Y. Inagaki, “Chart-based Parsing and Transfer in Incremental Spoken Language Translation System”, Proc. of NLPRS-97, pp.521-524 (1997).

[4] T. Murase, S. Matsubara, Y. Kato and Y. Inagaki, “Incremental CFG Parsing with Statistical Lexical Dependencies”, Proc. of NLPRS-2001, pp.353-358 (2001).

[5] E. Roche and Y. Schabes, “Finite-State Language Processing”, The MIT Press (1997).

[6] 篠崎 隆宏，斎藤 洋平，堀 智織，古井 貞熙，“話し言葉音声の認識を目指して”，情報処理学会研究報告，SLP34-22, pp.125-130 (2000).

[7] 高木 亮，松原 茂樹，稲垣 康善，“同時通訳コーパスを用いた通訳者発声タイミングの分析”，言語処理学会第 8 回年次大会発表論文集，pp.383-386 (2002).

[8] 浦谷則好，竹沢寿幸，松尾秀彦，森田千帆，“音声言語データベースの構成”，テクニカルレポート TR-IT-0056, ATR 音声翻訳通信研究所 (1994).