順位グラフ文法の並列構文解析*

3J－3

源内剛[†]　　塩谷勇[‡]

産能大学[§]

1　はじめに

順位グラフ文法は，頂点のラベル間に順位関係をもつグラフ文法である．これまでに，制限された順位グラフ文法に対する逐次構文解析アルゴリズムが提案されており，頂点数 n の入力グラフに対して $O(n)$ 時間で解析できることが示されている [2]．それらの制限のうち，順位関係に関する反射的な非終端記号をもたないという制限は，逐次構文解析においてハンドルを線形時間で見つけるために必要な条件である．

一般の順位グラフ文法の構文解析は，ハンドルを還元する際にうめ込み規則に従って辺を張りかえる必要があるため，構文解析全体を $O(n)$ 時間以下に抑えることは困難である．加えて反射的関係を含むため，ハンドルを見つけるために $O(n^2)$ 時間を必要とする．

本稿では，$O(\log n)$ 時間でハンドルを見つける CREW-PRAM(Concurrent Read Exclusive Write - Parallel Random Access Machine) 上の並列アルゴリズムを提案する．

2　グラフとグラフ文法

本稿で対象とするグラフは次に定義される頂点と辺がラベル付けされた有向グラフである．

Σ_V を頂点のラベルのアルファベット，Σ_E を辺のラベルのアルファベットとする．Σ_V, Σ_E 上のグラフ (Graph) G は３つ組 $G = (V_G,\ E_G,\ L_G)$ で定義される．ここで，

- $V_G \neq \phi$ は G の頂点の集合
- $E_G \subseteq \{(v,\ x,\ w) \in V_G \times \Sigma_E \times V_G \mid v \neq w\}$ は G の辺の集合
- $L_G : V_G \to \Sigma_V$ は G の頂点のラベル付け関数である．

頂点 v から w の辺のラベルの集合を $E_G(v, w) \stackrel{\mathrm{def}}{=} \{x \in \Sigma_E \mid (v,\ x,\ w) \in E_G\}$ と定義する．グラフ H がグラフ G の全部分グラフであるとき $H \subset G$ と記す．$V \subseteq V_G$ である頂点集合 V に対して，$V_H = V$ のとき，かつそのときに限り $G \mid V$ と記し，これを G の全部分グラフ $H \subset G$ への制限と呼ぶ．$G \setminus H$ の全部分グラフ H' に対して，H' の全ての頂点が H のある頂点に隣接しているとき，かつそのときに限り $R_G(H)$ と記す．２つのグラフ G と H の頂点間にラベルと隣接性を保存するような全単射写像 $f : V_G \to V_H$ が存在するとき，G と H は同型であるといい $G \cong H$ と記す．２つの頂点 $v, w \in V_G$ のラベル組 $lab_G(v, w)$ を $lab_G(v, w) \stackrel{\mathrm{def}}{=} (L_G(v),\ E_G(v,\ w),\ E_G(w,\ v),\ L_G(w))$ と定義する．

グラフ文法 (Graph Grammar) GG は５つ組 $GG = (\Sigma_V,\ \Sigma_T,\ \Sigma_E,\ P,\ S)$ で定義される．ここで，

- Σ_V は頂点のラベルの有限集合
- $\Sigma_T \subseteq \Sigma_V$ は終端ラベルの有限集合
- Σ_E は辺のラベルの有限集合
- P は生成規則の有限集合
- $S \in \Sigma_V \setminus \Sigma_T$ は開始記号である．

開始グラフは S とラベル付けされた１つの頂点からなるグラフである．$\Sigma_N \stackrel{\mathrm{def}}{=} \Sigma_V \setminus \Sigma_T$ は非終端ラベルの集合である．以下では，終端ラベル及び非終端ラベルでラベル付けされた頂点を，それぞれ終端頂点，非終端頂点と呼ぶことにする．生成規則 $p = (L,\ R,\ I,\ O) \in P$ はΣ_V, Σ_E 上の２つのグラフ L, R (左辺と右辺) とうめ込み規則 $I, O \subseteq V_R \times \Sigma_V \times \Sigma_E \times \Sigma_E$ からなる．ここで I は入る辺，O は出る辺をいう．すべての規則 $p \in P$ において，左辺 L が１つの非終端頂点のみからなるグラフ文法を文脈自由グラフ文法という．本稿では文脈自由グラフ文法のみについて議論する．

$s = (p,\ \tilde{L},\ \tilde{R},\ \tilde{b})$ を GG の導出指定という．ここで，

- $p = (L,\ R,\ I,\ O) \in P$
- $\tilde{L} \cong L$
- $\tilde{R} \cong R$
- $\tilde{b} : V_{\tilde{R}} \to V_R$ である．

導出指定 $s = (p,\ \tilde{L},\ \tilde{R},\ \tilde{b})$ において，$\tilde{L} \subset G$ かつ $V_G \cap V_{\tilde{R}} = \phi$ のとき s はグラフ G に適用可能であるという．

$G \xrightarrow[s]{} G'$ は次の条件をすべて満足するとき導出ステップという．

1. s が G に適用可能
2. $\tilde{R} \subset G'$
3. $G \mid (V_G \setminus V_{\tilde{L}}) = G' \mid (V_{G'} \setminus V_{\tilde{R}})$
4. $\tilde{R}$ と残りのグラフの間の辺が次のとおり：すべての $v \in V_{\tilde{R}}$ と $R_G(\tilde{L})$ のある頂点 w に対して，
   $E_{G'}(w,\ v) = \{x' \in \Sigma_E \mid \exists x \in E_G(w,\ \tilde{L}) : (\tilde{b}(v),\ L_G(w),\ x,\ x') \in I\}$
   $E_{G'}(v,\ w) = \{x' \in \Sigma_E \mid \exists x \in E_G(\tilde{L},\ w) : (\tilde{b}(v),\ L_G(w),\ x,\ x') \in O\}$

G_0 が開始グラフで，すべての導出ステップにおいて新しい頂点のみが作られるとき，すなわち $V_{\tilde{R}_i} \cap V_{\tilde{R}_j} \neq \phi \Rightarrow i = j$ のとき，導出ステップの列 $D = (G_{i-1} \xrightarrow[s_i]{} G_i \mid 1 \leq i \leq n)$, $n \in N$, $s_i = (p_i,\ \tilde{L}_i,\ \tilde{R}_i,\ \tilde{b}_i)$ を導出列という．G_n は GG によって導出可能であるという．すべての導出可能なグラフのクラスを $S(GG)$ と記す．$S(GG)$ のうち，終端頂点のみからなるすべてのグラフのクラスを GG の言語といい $\mathcal{L}(GG)$ と記す．

3　順位グラフ文法

$D = (G_{i-1} \xrightarrow[s_i]{} G_i \mid 1 \leq i \leq n)$, $n \in N$, $s_i = (p_i,\ \tilde{L}_i,\ \tilde{R}_i,\ \tilde{b}_i)$ を導出列とする．$\tilde{L}_j \subset \tilde{R}_i$, $1 \leq i, j \leq n$

*A Parallel Parsing for Precedence Graph Grammar

[†]Takeshi GENNAI

[‡]Isamu SHIOYA

[§]SANNO College, 1573 Kamikasuya, Isehara, Kanagawa, 259-11 Japan

のとき s_i は s_j に先行するという．この関係の反射的かつ推移的閉包を $\leq_D$ と記す．$\leq_D$ は半順序関係であり，D の導出順序と呼ばれる．$s_i \leq_D s_j$ でも $s_j \leq_D s_i$ でもないとき s_i と s_j は比較できないという．$v \in V_{\tilde{R}_i}$，$1 \leq i \leq n$ のとき $s_D(v) \stackrel{\mathrm{def}}{=} s_i$ とする．導出順序は G_n における頂点間の順序を次のように定める：隣接している頂点 v, $w \in V_{G_n}$ と $\Theta \in \{\doteq, \lessdot, \gtrdot\}$ に対して，

$$v \ \Theta \ w \Leftrightarrow s_D(v) \ \Theta_D \ s_D(w)$$

$s_D(v)$ と $s_D(w)$ が比較できないとき，$v \times w$ と記す．頂点のラベル間の順位関係 R_Θ, $\Theta \in \{\doteq, \lessdot, \gtrdot, \times\}$ は導出可能なグラフ G が存在して，$v, w \in V_G$，$v \ \Theta \ w$ となるすべての $lab_G(v, w)$ の集合である．

GG の任意のラベル組 t において2つ以上の順位関係が存在するとき GG は順位衝突 (*precedence conflict*) を持つという．

定義 3.1 (順位グラフ文法) *次の条件をすべて満足するグラフ文法 GG を順位グラフ文法 (Precedence Graph Grammar, PGG) と呼ぶ．*

1. *合流性を保つ*
2. *対称性を保つ*
3. *UI(uniquely invertible) である*
4. *反射的な非終端記号を持たない*
5. *順位衝突を持たない*

∎

これらの条件の定義については [2][3] を参照．

4　順位グラフ文法の並列構文解析

ここでは，与えられた PGG に対する構文解析アルゴリズムを示す．

定義 4.1 (ハンドル) *G をある導出可能なグラフとする．次の条件を満足するとき，全部分グラフ $H \subset G$ を G のハンドル (handle) という．*

1. *任意の隣接した頂点 v, $w \in V_H$ に対して，$lab_G(v, w) \in R_{\doteq}$*
2. *任意の隣接した頂点 $v' \in V_{R_G(H)}$, $w' \in V_H$ に対して，$lab_G(v', w') \in R_{\lessdot}$*

∎

ハンドルを求めるアルゴリズムでは，まず各頂点のラベル間の順位の大小関係を並列に求める．次に求まった順位をキーに並列にソートし，最後にソートされた列を並列に探索する．具体的な手順は次の通り．

```
procedure Parallel_Get_Handle (var handle : subgraph);
begin
  for 1 ≤ i, j ≤ n pardo M[i, j] ← 0;
  for g ∈ G, h ∈ G pardo begin
    if lab(g) ⋖ lab(h) then M[g, h] ← 2
    else if lab(g) ≐ lab(h) then M[g, h] ← 1
  end;
  for 1 ≤ i ≤ n pardo M[i, i] ← 1;
  for 1 ≤ i, j < k ≤ n pardo begin
    if M[i, j] < M[i, k] then begin
      S[i, j, k] ← 1; S[i, k, j] ← 0
    end else begin
      S[i, j, k] ← 0; S[i, k, j] ← 1
    end
  end
  for 1 ≤ i, j ≤ n pardo S[i, j, j] ← 1;
  for 1 ≤ i, j, k ≤ n pardo begin
    K[i, j, k] ← 0;
    for 0 ≤ m ≤ ⌈log n⌉ − 1 do begin
      K[i, j, k] ← K[i, j, k] + 2^m;
      if (k − 1) ∧ K[i, j, k] = 0 then
        S[i, j, k] ← S[i, j, k] + S[i, j, k + 2^m]
    end
  end
  for 1 ≤ i, j ≤ n pardo ORDER[i, j] ← S[i, j, 1];
  for 1 ≤ i, j ≤ n pardo begin
    C[i, ORDER[i, j]].v ← j;
    C[i, ORDER[i, j]].r ← M[i, j];
  end;
  for 1 ≤ i ≤ n pardo begin
    for 1 ≤ j ≤ prod_max do
      if C[i, j].r = 1 and j < C[i, j].v then
        handle[j] ← C[i, j].v
  end
end {Parallel_Get_Handle};
```

このアルゴリズムは CREW-PRAM 上で $O(\log n)$ 時間で実行できる．

5　むすび

本稿では構文解析の際にハンドルを n^3 プロセッサ $O(\log n)$ 時間で見つける並列アルゴリズムを提案した．これにより，ハンドル獲得の処理を並列に行なうことで，順位グラフ文法に対して反射的な非終端記号をもたないという制限を加えることなく構文解析ができる．

参考文献

[1] Aho, A. V. and Ullman, J. D. : "The Theory of Parsing, Translation, and Compiling, Volume I : Parsing", Prentice-Hall, Englewood Cliffs (1972).

[2] Franck, R. : "A Class of Linealy Parsable Graph Grammars", Acta Informatica, Vol. 10, pp. 175-201, Springer-Verlag (1978).

[3] Kaul, M. : "Computing the Minimum Error Distance of Graphs in $O(n^3)$ time with Precedence Graph Grammars". *in* Syntactic and Structural Pattern Recognition edited by G. Ferraté et al., Springer-Verlag (1988).