属性順位グラフ文法によるプログラム図の構文解析

指導教員　夜 久 竹 夫 教 授
6199M13　類瀬健二

概 要　木構造図一般を表現する中間言語として DXL(a Diagram eXchanged Language for tree-structured charts) が 1995 年に JIS X 0130 として制定されている．我々は Hichart を例に DXL に対応する木構造プログラム図に対する属性グラフ文法を考える. 宮崎らにより，edNCE グラフ文法を基底グラフ文法とした DXL 対応 Hichart のための属性グラフ文法 HCGG(HiChart Graph Grammar) が定式化されている．

本論文では，HCGG に対して等価な順位グラフ文法を定式化し、その順位関係を用いた構文解析アルゴリズムおよび計算量の評価について考える．

1　はじめに

プログラムの図表示は，視覚的プログラミングツールやソフトウェア事例データベースにおけるプログラム情報の視覚化には不可欠な機能である．

近年，様々なプログラム図の記述言語が報告されており，それぞれに特徴を持っており，それらの言語に基づいた多くの CASE ツールが開発されている．なかでも，1978 年に夜久らにより提案された Hichart(HIerarchical flowCHART description language)[1] は木フローチャートを基礎図式とする図形型言語であり，プログラムの階層構造，制御の流れおよびデータ構造を同時に表示できるという他のプログラム図言語とは異なる特徴も持っており，現在まで，Hicahrt 処理システムの開発が行われてきている [2],[3],[4],[5].

一方，木構造図データ交換言語 DXL(Diagram eXchange Language for tree structured charts)[6] は，多種多様木構造図に対する CASE ツール間でのデータを交換するための標準データ交換形式として設計され，1995 年に JIS X 0130 規格に制定された．各プログラム図言語に対して DXL との間の変換ツールを作成することにより，異なるプログラム図言語の CASE ツール間でデータを相互に変換し，利用ができるようになる．現在まで，DXL に対応したシステムの開発はあまり行なわれておらず，DXL に対応したアプリケーションの開発は急務である．

DXL に対応した Hichart プログラム図生成のための属性グラフ文法 HCGG(HiChart Graph Grammar) を定式化している [8]．HCGG は Rozenberg の edNCE グラフ文法 [7] を基底グラフ文法とし，木構造図の初歩的描画条件のための属性規則を付加した属性グラフ文法である．基底グラフ文法を edNCE グラフ文法に定式化したことにより，プログラム仕様書 Hiform など基底グラフ文法を edNCE グラフ文法による表やシミュレータ等の他の処理系と統一的に扱える可能性がある [9].

本論文では，HCGG に対して等価な順位グラフ文法を定式化し、その順位関係を用いた構文解析アルゴリズムについて議論する．

2　準備

2.1　edNCE グラフ文法 [7]

定 義　2.1[7]　Σ を 頂点のアルファベット，Γ を 辺のアルファベット とする．Σ と Γ 上のグラフとは 3 項組 $H = (V, E, \varphi)$ である．ただし，

1. V は 頂点の有限集合
2. $E \in \{(\mathrm{v},\gamma,\mathrm{w})|\mathrm{v},\mathrm{w}\in\mathrm{V},\mathrm{v}\neq\mathrm{w},\gamma\in\Gamma\}$ で，辺のラベルのアルファベット
3. φ は頂点をラベル付けする関数

□

定義 2.2[7]　$X_0 \rightarrow$(D,C) は edNCE 文法の生成規則である．

1. X_0:非終端な頂点アルファベット
2. D:グラフ
3. $C \subset \Sigma \times \Gamma \times \Gamma \times V_D \times \{\mathrm{in},\mathrm{out}\}$:
   接続関係(connection instructions)

□

定義 2.3[7]　edNCE グラフ文法 は次を満たす 6 項組 $GG = (\Sigma, \Delta, \Gamma, \Omega, R, S)$ である．

1. Σ:頂点アルファベットの有限集合
2. Δ:終端頂点アルファベットの有限集合

3. Γ:辺ラベルのアルファベットの有限集合
4. $\Omega \subseteq \Gamma$:終端辺ラベルの有限集合
5. R:生成規則の有限集合の有限集合
6. $S \in \Sigma - \Delta$:初期状態

□

2.2 属性 edNCE グラフ文法 [8]

定義 2.4[8] 属性 edNCE グラフ文法 は次の条件を満たす 3 項組 $G_N = \langle GG, A, F \rangle$ である.

1. $GG = (\Sigma, \Delta, \Gamma, \Omega, R, S)$
2. 属性集合A= $\bigcup_{X \in \Sigma} \mathcal{I}(X) \cup \mathcal{S}(X)$:継承属性 の集合 $\mathcal{I}(X)$ と 合成属性 の集合 $\mathcal{S}$(X)
3. 意味規則集合$F = \bigcup_{r \in R} F_r$: F_r は R の各生成規則 r=$X_0 \rightarrow$ (D,C) に対し, $\mathcal{S}(X_0) \bigcup_{X \in \Sigma - \Delta} \mathcal{I}$(X) の属性のみを全て定義する意味規則の集合

□

2.3 順位グラフ文法 [10]

定義 2.5[10] 二つの頂点 (v,w ∈ V) の ラベル組 とは lab_G(v,w) $=^{def}$ (φ(v),(v,γ,w),(w,γ,v),φ(w)) である. □

定義 2.6[10] 導出仕様書(derivation specification) とは, s=(P,$\tilde{X}_0$,$\tilde{D}$,$\tilde{b}$) であり, $\tilde{X}_0 \cong X_0$,$\tilde{D} \cong$D,$\tilde{b}$:$V_{\tilde{D}} \rightarrow V_D$ である. また 導出列(derivation sequence) とは d=($G_{i-1} \rightarrow_{S_i} G_i$|1≤i≤n) である. □

定義 2.7[10] $S_i \leq_D S_j$ でも $S_j \leq_D S_i$ でもない場合 S_i, S_j は 比較できない といい, $\times$ と書く. ただし $\leq_D$ は推移的閉方, 半順序で D の導出順序と呼ばれる. □

定義 2.8[10] v∈$V_{\tilde{D}_i}$ ならば S_D(v) $\stackrel{\text{def}}{=}$ S_i である. □

定義 2.9[10] v⋈w⇔S_D(v)⋈$_D$$S_D$(w) である. ただし v,w∈$V_{G_n}$,⋈∈{≒,<,>} である. □

定義 2.10[10] ⋈ ∈{≒,<,>,×} は 2 頂点間の順位関係 である. その ラベル間の順位関係 は $R_{\bowtie}$ であり, ⋈ はすべての lab_G(v,w) の集合である. □

定義 2.11[10] 拡張辺ラベルアルファベット(extended edge label alphabet) とは, $\Gamma_{\bowtie} =^{def} \Gamma \times$ {≒,<,>,×} である. □

定義 2.12[10] 順位グラフ文法は次の 6 項組である.
$GG_{\bowtie} = (\Sigma, \Delta, \Gamma_{\bowtie}, \Omega_{\bowtie}, R, S)$

□

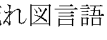
定義 2.13[10] Instantaneous Description(ID) とは (G,K,Ψ) の 3 項組である.

1. G:インスタンスグラフ
2. K:G 内の頂点の順序リスト
3. Ψ:そこまでの 導出仕様書 の構成の集合

□

定義 2.14[10] TOP(G,K) $=_{def}$ G|{v_j,$\cdots$,v_k}(1≤j≤k,v_j から v_k までは等しい順位) である. また TOP(G,K) は
シフト:(G,K,Ψ) $\longmapsto_S$ (G,Kw,Ψ),
還元:(G,K_1K_2,Ψ) $\longmapsto_R$ (G',K_1w,Ψ ∪ { s })
と呼ぶ 2 つの型の 動作(moves) がある. □

2.4 Hichart[1]

Hichart は 1978 年に夜久, 二木により発表された階層型流れ図言語である. Hichart は繰り返し記号  を初めて導入したプログラム図式言語で, プログラムを作成している要素間の制御の流れと階層的なつながりを疑似木構造グラフとして表現する. その後改良と拡張 [3] が続けられ, 現在ではプログラムの仕様を形作る 4 つの基本要素, (1) アルゴリズムの流れ, (2) データの流れ, (3) データ構造, (4) プログラム構造をすべて統一的に表示することができる.

2.5 DXL[6]

木構造図データ交換言語 DXL は,CASE ツールで作成した図のデータ形式を標準化してデータの再利用や流通を促進するために設計され, 1995 年に JIS X 0130 として制定された.

2.6 HCGG

定義 2.15[8] DXL 対応 Hichart に対する属性 edNCE グラフ文法を HCGG(HiChart Graph Grammar) と呼ぶ.

□

3 HCGG に対する順位グラフ文法

この章では, DXL 対応 Hichart についての順位グラフ文法を定式化する.

	[module_list]	"Profile"	[explanation]	[module]	"Identifier is"
[module _profile]					
"M_Packet"		$R_{\lessdot}$			
[profile _module _list]					
[profile]	$R_{\doteq}$			$R_{\lessdot}$	$R_{\lessdot}$
[module _list]					
"Profile"	$R_{\gtrdot}$		$R_{\doteq}$	$R_{\times}$	$R_{\times}$
[explanation]					
[module]	$R_{\doteq}$			$R_{\lessdot}$	$R_{\lessdot}$
"Identifier _is"	$R_{\gtrdot}$		$R_{\lessdot}$	$R_{\times}$	$R_{\times}$

表 1: HCPGG の順位関係表

3.1 HCPGG

我々は HCGG に対して，順位関係に矛盾が生じないように生成規則を修正し，Hichart 対応順位グラフ文法(HiChart Precedence Graph Grammar) を定式化する．

定義 3.1 HCPGG は 3 項組 $\langle GG_{\bowtie}, A, F\rangle$ によって構成され，生成規則 69 個，意味規則 728 個からなり，スタートグラフは"[module_packet]"とする． □

定理 3.2 L(HCGG) を生成する順位グラフ文法が存在する．

証明 HCPGG を考えると HCPGG の頂点アルファベット間において，順位関係を矛盾なく定義できる □

定義 3.3 表 1 は，HCPGG の順位関係表の一部であり，606 個の順位関係から構成され，2 頂点アルファベット間はグラフにおいて直接連結される． □

4 HCPGG に対する構文解析

この章では，HCPGG に対する構文解析アルゴリズム，およびそのアルゴリズムの計算時間について考察する．

4.1 HCPGG に対する構文解析アルゴリズム

HCPGG の構文解析アルゴリズムを，構文解析を行なう"HCPGG_analysis"と属性評価を行なう"attribute_evaluation"の 2 つから構成した．

4.1.1 HCGPP_analysis

"HCPGG_analysis"では入力グラフを与え，構文木を作成する．

Algorithm 1

```
HCPGG_analysis(Graph G){
  while(G != start graph ){
    precedence_analysis(G);
  }
  generate_parse_tree(derivation sequence);
}
```

アルゴリズム"HCPGG_analysis"では入力したグラフがスタートグラフになるまで"precedence_analysis"を繰り返し，"generate_parse_tree"で構文木を作成する．また，"generate_parse_tree"は 492 行で構成する．

アルゴリズム"precedence_analysis"では，シフトと還元を行ない，入力したグラフに対して，このグラフのハンドルに対する生成規則の左辺に置きかえる．

Algorithm 2

```
Graph precedence_analysis(Graph G){
  K = shift_order_list(G)
  production = find_production(K);
  G = replace_graph(G,K,production);
  K = rewriting_order_list(K,production);
}
```

アルゴリズム "precedence_analysis"は 4 ステップからなる．最初のステップは，"shift_order_list"(315 行) で，入力したグラフに対して順序ハンドルの探索を行ない，頂点の順序リストを作成する．第 2 のステップでは，その順序リストから，順位ハンドルを右辺とする生成規則を"find_production"(218 行) によって発見する．第 3 のステップでは，グラフの書き換える作業を"replace_graph"(2774 行) で行なう．第 4 のステップでは，順序リストから，生成規則の右辺に相当する頂点を取り除き，左辺に相当する頂点を付け加える作業を"rewriting_order_list"(250 行) が行なう．

4.1.2 attribute_evaluation

"attribute_evaluation"では，構文木に対して属性評価を行なう．

Algorithm 3

```
attribute_evaluation(parse tree){
  S = attribute_parse_tree(parse tree);
  attribute_calcurate(S);
}
```

アルゴリズム"attribute_evaluation"では，"attribute_parse_tree"(115 行) において属性計算を行なう

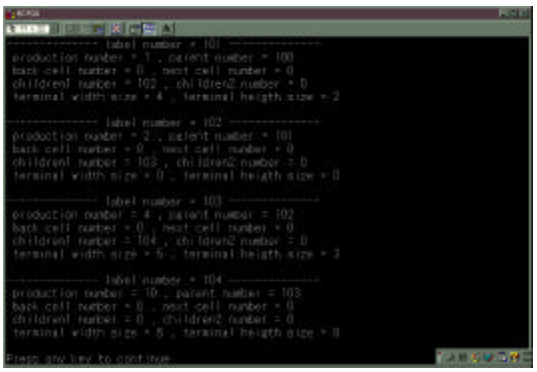


図 1: HCPGG parser

順番 (S) を求める．次に”attrbibute_calcurate”(1267 行) で任意の生成規則の継承または合成属性の計算を行なう．

4.2 HCPGG による構文解析アルゴリズムの計算時間

定理 4.1 ”HCPGG_analysis”は線形時間で実行可能である．

証明 HCPGG において，グラフ内の辺の数が m の場合，そのグラフに対する導出列の長さは高々m の定数倍となる．”precedence_analysis”は，導出列の長さの回数だけ行なわれる．従って高々k_1m 回行なわれる．”shift_order_list”は”precedence_analysis”が行なわれている間に，順位ハンドルとなる頂点の移動を m+1 回，等しい順位ハンドルの発見を高々k_1*k_2*m 回行なっている．”find_production”，”replace_graph”，”rewriting_order_list”の計算時間は O(1) である．また，”generate_parse_tree”の計算時間は導出列の長さの定数倍より O(m) である． □

定理 4.2 ”attribute_ebaluation”は線形時間で実行可能である．

証明 構文木の各々の頂点を高々2 回しか通過しない． □

5 HCPGG parser

この章では，HCPGG の parser の実行例を表示する．図 1 は実際に HCPGG parser を実行したものである．

6 終わりに

本論文では，HCGG と等価な順位グラフ文法の存在を確認し HCPGG を構成した．また HCPGG に対する構文解析アルゴリズムを構築および計算量の評価，また Parser の開発 (約 7000 行) を行なった．

今後は Hicahrt 処理システムのエディタ等の開発を行っていく必要がある．

参考文献

[1] T.Yaku, K.Futatsugi, A.Adachi and E.Moriya; HICHART–A Hierachical Flowchart Description Language–; *Proc. IEEE COMPSAC 11*, 157-163(1987)

[2] 西野哲郎; 属性グラフ文法とその Hichart 型プログラム図式に対するエディタへの応用; コンピュータソフトウェア, Vol,5, No.2, 81-92 (1988)

[3] Y.Adachi, K.Anzai, K.Tsuchida and T.Yaku; Hierachical Program Diagram Editor Based on Attribute Graph Grammar; *Proc. IEEE COMPSAC 20*, 205-213(1996)

[4] Y.Adachi, Y.Miyadera, K.Sugita, K.Tsuchida and T.Yaku; A Visual Programming Environment Based on Graph Grammars and Tidy Drawing; *Proc. ICSE 20*-Ⅱ,74-79(1998)

[5] 安達由洋, 大井裕一, 大澤優, 二木厚吉, 夜久竹夫; DXL 対応 Hichart プログラム図に対する属性グラフ文法; **日本大学文理学部自然科学研究所研究紀要第 33 号**, 149-164(1998)

[6] 木構造図用データ交換言語 DXL; JIS X 0130-1995; 日本工業標準審議会, 1-36 (1995)

[7] Grzegoz Rozenberg; Handbook of Graph Grammars and Computing by Graph Transformation; World Scientific(1996)

[8] 宮崎征宏, 類瀬健二, 土田賢省, 夜久竹夫; プログラム図に対する描画を考慮した NCE 属性グラフ文法; 電子情報通信学会技術研究報告 Vol.100 No.52,1-8(2000)

[9] T.Arita, K.Tomiyama, K. Tsuchida, Y. Yaku et al; Syntactic Processing of Diagrams by Graph Grammars; *Proc. IFIP WCC ICS2000* ,145-151(2000).

[10] Manfred Kaul; Practical Applications of Precedence Graph Grammars,Graph Grammars and Their Application to Computer Science, LNCS 291,326-342(December-1986),Virginia:Springer-Verlag