アルゴリズム 26－6
（1992. 3. 23）

木構造図式の描画問題

海野浩　安斎公士　小倉耕一　夜久竹夫
大原簿記学校　関東学園大学経済学部　北海道東海大学　東京電機大学理工学部

木構造図式を与えられた美的条件を満たすように配置する問題は“美的描画問題”といわれる[2),3)]．木構造図式に対する，プログラム図式を指向した美的条件は，木に対する美的条件[2),11)]を変形することにより定式化された[17)]．その美的条件を満たす配置を与える手法も提案された[17)]．

本論文で我々ははじめに，従来の美的条件[17)]を満たす配置を与える手法[17)]を定式化し $O(n^3)$ 時間アルゴリズムを詳細に定める．次に上の美的条件[17)]に条件をひとつ加えて新たな美的条件を考えると，我々のアルゴリズムが新たな美的条件を満たす最も狭い配置を与えることを示す．その結果美的条件と計算量に関する新たな関係を得る．

Tidy Drawing of Tree Structured Diagrams

Hiroshi Unno　Koushi Anzai　Koichi Ogura　Takeo Yaku
Ohara Book Keeping School of Business　Kanto Gakuen Univ.　Hokkaido Tokai Univ.　Tokyo Denki Univ.

Layout problems of tree structured diagrams into the narrowest area under some eumorphous condition are called “tidy drawing problems”[2),3)]. For tree structured diagrams, program diagram oriented eumorphous conditions have been formalized[17)] by modifying the eumorphous condition for trees[2),11)]. And a method, which provids a layout satisfying the condition, was proposed[17)].

In this paper, we will formalize first the layout method[17)] which satisfies the former eumorphous condition[17)] and will introduce an $O(n^3)$ time algorithm. Furthermore, we introduce a new eumorphous condition, adding another constraint to the previous eumorphous condition[17)]. Finally, we will show that the $O(n^3)$ time algorithm provides one of the narrowest tree structured diagrams satisfying the new eumorphous condition.

1 まえがき

“木構造図式” は各頂点が次の四つの属性を持った木である：即ち四つの属性は (1) 頂点の幅, (2) 頂点の深さ, (3) 頂点の水平座標, (4) 頂点の垂直座標である. 木構造図式は同一でない大きさを持ち木構造状に連結された “セル” という, 2 次元平面上に配置された頂点からなる図式を表している.

一般にグラフを与えられた条件を満たすように配置する問題はグラフの配置問題といわれ, VLSI の配置問題がよく知られている. 我々が扱う “美的描画問題”[2),3)] はグラフの表示を指向した問題で, 木構造図式を与えられた美的条件を満たすように配置する問題である. 木構造図式から属性を取り去ると木になるが, 1970 年頃から木に対する美的描画問題が盛んに研究されてきた. まず, 2 分木に対する美的条件とその美的条件に対応する線形時間の近似描画アルゴリズムの例が示され[2),3)], 次にその美的条件のもとで最小領域に 2 分木を描画する問題が整数座標の場合 $\mathcal{NP}$ 完全であることが示された[6)]. 土田は Hichart 型のプログラム図式の描画問題[4)]をヒントとして, 一般の n 分木の描画問題を研究し, その結果, 最小領域に描画する問題が $O(n^4)$, $\mathcal{NP}$ 困難となるような二つの美的条件を得た[11)]. プログラム図式の描画問題は属性文法の立場からも研究されている[14),16)].

著者の一部らはプログラム図式の生成に関する研究[17)]の中で, 木に対する美的条件を拡張して, 木構造図式に対する美的条件を導入した. 我々の美的条件は親頂点は一番上の子頂点から j レベル ($j \geq 0$) だけ下のレベルに配置されるようになっている. 同時にその美的条件に対応した描画手法が一つ提案されている. 本論文で扱われるのはこの新しい美的条件である.

本研究の動機は以上のように定式化されつつある木構造図式の美的描画問題に木の描画問題の成果を適用して, 美的描画問題を体系化することである. 本論文の目的は, 最近提案された木構造図式の美的条件[17)]にもとづいて描画アルゴリズムを体系的に記述するとともに, 最小領域美的描画問題に関する美的条件と計算量との関係を体系的に与えることである.

2 準備

この節で我々は, 大きさが一定でない “セル” からなる木構造図式を考え, その図式を描画する際の従来の “美的” 条件[17)] を概観する. なお, 大きさ一定のセルからなる木構造型プログラム図式の描画に関しては, 木の描画問題に関する手法が適用可能である[2),3),11)]. はじめに次の用語を定める.

定義 1 [17)]. 木構造は $T = (V, E, r, width, depth)$ である. ここで, (V, E) は木で V をセルの集合, E を辺の集合という. $r \in V$ は根セルである. $width$ は V から Z への写像で $width(p)$ はセル p の上下方向の長さ (幅という) を表す. $depth$ は V から Z への写像で $depth(p)$ は p の左右方向の長さ (深さという) を表す. □

定義 2 [17)]. 組 (T, π) を木構造図式という. ここで T は木構造で, $\pi : T$ のセルの集合 $\longrightarrow Z \times Z$ は T の配置といわれる. セル p の座標を $\pi(p) = (x, y)$ とするとき, その x 座標, y 座標は, $\pi_x(p), \pi_y(p)$ でそれぞれ表される. プログラム図式への適用を考慮して x 軸は左から右へ, y 軸は上から下へ向かうものとする. □

木構造図式は各頂点に 4 つの属性 $width(p)$, $depth(p)$, $\pi_x(p)$ と $\pi_y(p)$ が割り当てられた根付き木である. 我々は子供セルに上から下へと順序がついている順序木構造図式を対象とする. 木構

造図式は長方形が木構造上に配置された図式を意味する．本論文では，プログラム図式への応用を考慮して，木構造図式 $D=(T,\pi)$ の幅 $width(T,\pi)$ を次のように定める:

$$width(T,\pi)\equiv$$
$$\max\{\pi_y(p)+width(p)-\pi_y(q)-1 \mid \text{ただし,}p\text{ と }q\text{は }T\text{の中のセルで,}\pi_y(p)>\pi_y(q)\}.$$

セル p のレベル $level(p)$ は，p と根セルとの間の辺の個数で定義される．また，セル p に対して，関数 $Index$ を次のように定める:

$$Index(p)\equiv\begin{cases}0 & p\text{が根セルであるとき}\\ i & p\text{が }p\text{ の親の }i\text{ 番目の子供であるとき}.\end{cases}$$

木構造の配置に関するいくつかの条件を以下のように定める．
条件 **C0**[2),17)]．　木構造図式 (T,π) において，セル p と q のレベルが等しく，$\pi_y(p)<\pi_y(q)$ であるならば$\pi_y(q$の長男$)>\pi_y(p$ の末っ子$)+width(p$ の末っ子$)$，即ち線が交差しない．また，セル p に対して，

$$area(p,\pi)\equiv$$
$$\{(x,y)\mid \pi_x(p)\le x\le \pi_x(p)-depth(p)-1,\pi_y(p)\le y\le \pi_y(p)+width(p)-1\}.$$

としたとき，すべての p,q $(p\neq q)$ について

$$area(p,\pi)\cap area(q,\pi)=\phi.$$

即ちセル同士が重ならない．　□

次の条件 C1 はプログラム図式特有のものであり，論理上の階層レベルと見かけ上の階層レベルを一致させるための条件である．
条件 **C1**[2),17)]．　木構造図式 (T,π) において，セル p が子供セルを持つならば $depth(p)=1$．また，レベルが i である全てのセル p に対して，$\pi_x(p)=i$．　□
条件 **C2**[2),17)]．　木構造図式 (T,π) の部分木構造 T_1 と T_2 が位相同型ならば，T_1 と T_2 は平行移動に関して同型に配置される．　□

木の描画問題では親セルは子供セルの中心へ配置されるという条件が扱われているが，その代わりに我々はつぎの条件を考える．
条件 **C3(j)**[17)]．　木構造図式 (T,π) において，セル p が k 個の子供 $q_1,\cdots,q_k(Index(q_i)=i,1\le i\le k)$ をもつならば，$\pi_y(p)=\pi_y(q_1)+\min(j,\pi_y(q_k)-\pi_y(q_1))$．　□

木構造図式の集合から整数の集合への関数 $Intersect$ を次のように定める:

$Intersect(T,\pi)\equiv\max\{\pi_y(p)-\pi_y(q)+1;T_1$ と T_2 は，根セル同士が兄弟セルであり，$Index(T_2$ の根セル$)<Index(T_1$ の根セル$)$ であるような T の任意の部分木構造．p,q は T_2 と T_1 の任意のセルとする $\}$.

次の条件は部分木同士の水平方向の重なりに関するものである．n 進木の描画問題ではこの重なりの値により計算量が異なることが知られている[11)]．
条件 **C4(k)**[11),17)]．　$k\ge 0$ に対して，配置πが $Intersect(T,\pi)\le k$.を満たす．　□

以上の条件を組み合わせて“美的条件”を導入する．

記法 1　木構造図式の美的条件 $E_0(j)$ と $E_*(j)$ を以下のように定める:
$E_0(j)\equiv C0\wedge C1\wedge C2\wedge C3(j)\wedge C4(0)$.
$E_*(j)\equiv C0\wedge C1\wedge C2\wedge C3(j)$.　□

図 1 に条件 $E_0(1)$ を満たす木構造図式の例を示す．

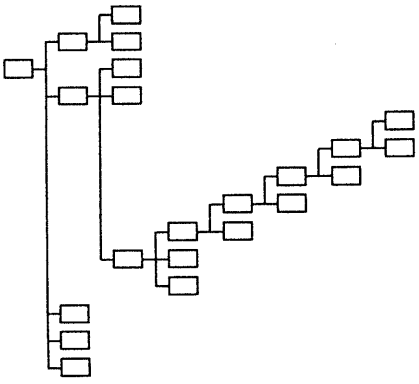

図 1: $E_0(1)$ を満たす木構造図式の例

3 描画アルゴリズム

この節では前節で紹介された美的条件 $E_*(j)$ に注目し，それを満たすような配置をあたえるアルゴリズムについて考える．プログラム生成手法[17]の中で述べられた方法を定式化し，詳細に定める．木構造 T を美的条件 $E_0(j),(j \geq 0)$ を満たすように最小幅に配置することを T の配置初期化といいその配置 π_{init} を初期配置という．配置初期化アルゴリズム $Init$ は容易に構成可能である:

アルゴリズム **Init**：配置初期化

[形式] $Init(T, j, \pi_{init})$

[入力] $\begin{cases} T & = & (V, E, r, width, depth), \\ j & : & E_*(j) \text{ の } j \end{cases}$

[出力] π_{init} : $E_0(j)$ を満たす T の最小幅配置

[方法] 深さ優先探索を行なう．深さ優先探索で最も早く訪れるセル (最も上の枝の葉) を p とする．p のレベルを l としたとき $\pi(p) = (l, 1)$ とする．

上のアルゴリズム $Init$ の時間計算量は明らかに $O(n)$ である．

美的条件 $E_0(j),(j \geq 0)$ を満たすように初期配置された木構造 T の配置 π_{init} に対して，条件 $E_*(j),(j \geq 0)$ を満たしながら更に狭い領域に再配置することを配置改良と呼ぶことにする．はじめに配置改良のためにいくつかの用語を導入する．

定義 3 木構造 $T = (V, E, q, width, depth)$, 配置 π と $j \in Z$ に対して，集合 $U_j = \{v \mid v \in V, \pi_x(v) = j\}$ を考える．このとき，

$$U_j = \{v_1, \cdots, v_k \mid \pi_y(v_i) < \pi_y(v_{i+1})(1 \leq i < k)\}.$$

と書けるとき，v_i の上方移動可能距離 $Move\_v_i\_\pi$ を次のように定義する．

v_i が葉でないとき．

$$Move\_v_i\_\pi \equiv \begin{cases} 1 & (i = 1) \\ \pi_y(v_i) - \pi_y(v_{i-1}) - width(v_{i-1}) & (2 \leq i \leq k). \end{cases}$$

v_i が葉のとき．

新しいセル u_j を考え，$U_j^+ = \{v \mid v \in V \cup \{u_j\}, \pi_x(v) = j\}$ を考える．このとき $U_j^+ = \{v_1, \cdots, v_k\}, \pi_y(v_m) < \pi_y(v_{m+1})$ $(1 \leq m < k)$ と書けるとする．ただし $v_l = u_j, \pi(u_j) = (j, \pi_y(v_i)), width(u_j) = width(v_i), depth(u_j) = 1$ と定義しておく．

このとき

$$Move\_u_j\_\pi \equiv \begin{cases} 1 & (u_j = v_1) \\ \pi_y(u_j) - \pi_y(v_{l-1}) - width(v_{l-1}) & (u_j \neq v_1). \end{cases}$$

に対して

$Move\_v_i\_\pi \equiv \min\{Move\_u_j\_\pi \mid \pi_x(v_i) \leq j \leq \pi_x(v_i) + depth(v_i) - 1\}$ とする. □

定義 4 木構造 $T = (V, E, q, width, depth)$ と配置 πに対して $i = \min\{\pi_x(v) \mid v \in V\}, k = \max\{\pi_x(v) \mid v \in V\}$ とおく. 各 $j(i \leq j \leq k)$ に対して, $U_j = \{q \mid q \in V, \pi_x(q) = j\}$ とする. このとき集合

$$S = \bigcup_{i \leq j \leq k} \{z \in U_j \mid v \in U_j \text{に対して}, \pi_y(z) \leq \pi_y(v)\}$$

を (T, π) の上辺をなすセルの集合という. □

定義 5 集合 $\{z_1, \cdots, z_k\}$ $(k \geq 1)$ を, 木構造図式 (T, π) の上辺をなすセルの集合とし, 各 i $(1 \leq i \leq k)$ に対して m_i を z_i の上方移動可能距離とする. このとき次のように定義される $Move\_T\_\pi$ を (T, π) の上方移動可能距離という, ここで

$$Move\_T\_\pi \equiv \min\{m_i \mid 1 \leq i \leq k\}.$$ □

セル q を根とする部分木構造 T_q の配置改良は以下の方法で決まる:

アルゴリズム **SubOpt** : 部分木構造の配置改良

[形式] $SubOpt(T_q, \pi_{in}, j, \pi_{out})$

[入力]
$$\begin{cases} T_q & = & (V, E, q, width, depth) \\ \pi_{in} & : & \text{最適化前の配置} \\ j & : & \text{整数}, E_*(j) \text{ の } j \end{cases}$$

[出力] π_{out} : $E_*(j)$ を満たす π_{in} より幅が狭いか等しい T_q の配置

[方法]

```
begin
    if q は葉 then
        begin
            (* Shift により (T_q, π_in) に上方移動を行う *)
            Shift(T_q, π_in, j, π_out)
        end
    else (* q は子を持つ *)
        begin
            Sons(q) = {p_1, ..., p_k} (k ≥ 1) を q の子供セルの集合とする.
            (* q の子供を根とする部分木構造 T_{p_i}
               (1 ≤ i ≤ k) の配置改良を行う *)
            for i = 1 to k do
                begin
                    (* 部分木構造 T_{p_i} の配置改良を行う *)
                    SubOpt(T_{p_i}, π_in, j, π_out);
                    π_in := π_out
                end ;
            π''_x(q) := π_{in_x}(q);
            π''(q) を条件 C3(j) により設定する;
            π''(s) := π_in(s) (s ≠ q);
            (* Shift により (T_q, π'') に上方移動を行う *)
            Shift(T_q, π'', j, π');
            π_out := π'
        end { else }
```

end. { $SubOpt$ }

[時間複雑度] $SubOpt$ 自身を n 回, 各 $SubOpt$ で $Shift$ を 1 回呼ぶ. $Shift$ の複雑度については以下で述べる.

次のアルゴリズム $Shift$ は部分木構造を上方へ平行移動する. $Move\_T_q\text{-}\pi_{in}$ は, (T_q, π_{in}) が他の部分木構造の配置と重ならないで, かつ最も上の配置となるまでの y軸方向の平行移動可能距離である.

アルゴリズム **Shift** : 部分木構造の上方移動

[形式] $Shift(T_q, \pi_{in}, j, \pi_{out})$

[入力] $\begin{cases} T_q & = & (V, E, q, width, depth) \\ \pi_{in} & : & T_q\text{の配置} \\ j & : & \text{整数}, E_*(j)\text{ の }j \end{cases}$

[出力] π_{out} : T_qの配置:再配置

[方法]

begin

if $\pi_{in}(q) = (l, 1)$ **then**

begin

$\pi_{out} := \pi_{in}$;

return

end ;

if セル qは長男でない **then**

begin

- $\{z_1, \cdots, z_k\}$ $(k \geq 1)$ を (T_q, π_{in}) の上辺をなすセルの集合とするとき, 各 i $(1 \leq i \leq k)$ に対して (z_i, π_{in}) の上方移動可能距離 m_iを求め, それらの最小値を (T_q, π_{in}) の上方移動可能距離 $Move_T_q\text{-}\pi_{in}$とする (時間複雑度:各 i に対して $O(n)$, 全体で $O(n^2)$);
- 部分木構造 T_qの配置π_{in}を y軸負方向へ値 $Move_T_q\text{-}\pi_{in}$だけ平行移動させπ_{out}とする (時間複雑度:$O(n)$).

end

else $\pi_{out} := \pi_{in}$

end. $\{Shift\}$

[時間複雑度] $O(n^2)$

以上により木構造 Tの配置はつぎのアルゴリズム $LayOut$ で得られる.

アルゴリズム **LayOut** : 木構造の配置

[形式] $LayOut(T, j, \pi)$

[入力] $\begin{cases} T & = & (V, E, r, width, depth), \\ j & : & \text{整数}, E_*(j)\text{ の }j \end{cases}$

[出力] π : $E_*(j)$ を満たす Tの配置

[手法]

begin

$Init(T, j, \pi_{init})$;

$SubOpt(T, \pi_{init}, j, \pi)$

end.

[時間複雑度] 定理 2参照

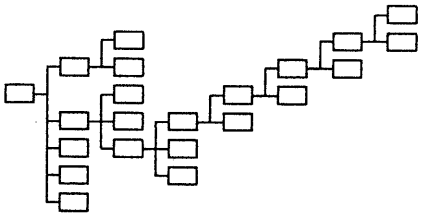

図 2:図 1 に対するアルゴリズム $LayOut$ の出力

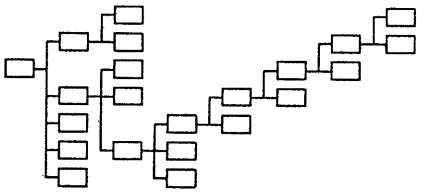

図 3:図 1 に対する $E_*(1)$ のもとでの最小幅配置

定理 1 アルゴリズム $LayOut$ は条件 $E_*(j)$ を満たすアルゴリズム $Init$ より狭いか等しい幅の木構造図式を出力する.

定理 2 木構造図式のセルの個数を n としたとき, アルゴリズム $LayOut$ の計算量は $O(n^3)$ である.

証明. セルを深さ優先に訪れ, 各セルで $SubOpt$ を 1 回呼ぶ. □

4 最適性

この節では, 3節で導入されたアルゴリズム $LayOut$ の出力が最小幅配置となるような美的条件を示す. はじめに $LayOut$ が必ずしも条件 $E_*(j)$ に対する最小幅配置を与えるとは限らないことを示す.

定理 3 アルゴリズム $LayOut$ は美的条件 $E_*(j)$ を満たす最小幅の木構造図式を与えるとは限らない.

証明. 図 2, 図 3 の二つの例により示される. □

次に新たな美的条件を考えるため, 以下の条件 C5 を導入する: はじめに木構造図式 $C = (U, F, s, width', depth', \pi')$ と $D = (V, E, r, width, depth, \pi)$ の距離を

$$Dist(C, D) \equiv \min_{x \in Z} \{ \mid \pi'_y(p) - \pi_y(q) \mid ; p \in U, q \in V, \pi'_x(p) = \pi_x(q) = x \}$$

と定める.

条件 C5. 木構造図式 Dの根 vの子供を $v_1, \cdots, v_n$ とする. このとき, $D_{v_i}\ (2 \leq i \leq n)$ に対して, $^{\exists} j < n$, $Dist(D_{v_i}, D_{v_j}) = 1$. □

上の条件 C5 にもとづき美的条件 $E_*(j)$ を強めて, 次の美的条件 $F_*(j)$ を導入する:

記法 2 $F_*(j) \equiv E_*(j) \wedge C5$. □

このとき美的条件 $F_*(j)$ に対して次が成り立つ.

定理 4 与えられた $j \geq 0$ に対してアルゴリズム $LayOut$ は美的条件 $F_*(j)$ を満たす最小幅配置を出力する.

証明. Tに対するアルゴリズム $LayOut$ の出力を $D = (T, \pi)$ とする. Dは定理 1 より美的条件 $E_*(j)$ を満たす. またアルゴリズム $Shift$ により Dは条件 C5 も満たしている. なぜなら, もし美的条件 $E_*(j)$ を満たす Dより狭い木構造図式 (T, π') が存在したとすると (T, π') は C5 を満たさない. □

5 おわりに

木に対しては扇型に配置される美的条件とその美的条件に対応する時間計算量の研究が行われ，既に得られていた成果[11]はセルの大きさが一定で扇型に配置された木構造図式に対して適用可能であった．セルの大きさが一定でない木構造図式に対しては，プログラム図式を指向した美的条件 (2節の $E_*(j)$) と非定式的な描画手法だけが知られていた[17]．本論文で我々は後者の美的条件 $E_*(j)$ を満たす配置を与える $O(n^3)$ 時間アルゴリズムを得た．さらに我々は従来の美的条件 $E_*(j)$ を損なうことなく新たな条件を加えた美的条件 $F_*(j)$ を導入した．この時，我々の $O(n^3)$ アルゴリズムが新たな美的条件 $F_*(j)$ を満たす最小の配置を与えることを証明した．その結果美的条件と計算量についての新たな関係を得た．即ち，木の描画問題の立場から考えると従来の結果は $O(n^4)$ 時間[11]と $\mathcal{NP}$完全性[3]に対応する美的条件であり，今回新たに $O(n^3)$ 時間に対応する美的条件が知られたことになる．本来の美的条件 $E_*(j)$ に対応する計算量を評価する課題は未解決である．

参考文献

1) 夜久竹夫, 二木厚吉: フローチャートの木構造型記法, 信学技報, Vol. AL78-47, pp. 61-66, (1978).

2) C.Wetherell and A.Shannon: Tidy drawing of trees, *IEEE Transac.* Vol.SE-5,pp.514-520,(1979).

3) E.M.Reingold and J.S.Tilford: Tidy drawing of trees, *IEEE Transac.*, Vol.SE-7, pp.223 - 228, (1981).

4) 郷信義, 岸本美紀, 高橋美智子, 長田芳一, 夜久竹夫: 木フローチャート記述言語 Hichart の処理系実現について, 情報処理学会第 27 回全国大会講演論文集,pp.549-550,(1983).

5) K.J.Supowit and E.M.Reingold: The complexity of drawing trees nicely, *Acta Informatica*, Vol.18, pp.377-392,(1983).

6) 岡田謙一, 北川節: PAD によるソフトウェア開発システム, 情報処理学会論文誌, Vol.26, pp.898 - 904, (1985).

7) 郷信義, 小倉耕一, 竹内一浩, 土田賢省, 近藤理彦, 岸本美紀: "階層的フローチャート言語 Hichart に対するオートフローチャータの実現", 日本ソフトウェア科学会, 構造エディタに関するワークショップ予稿,(1986).

8) 夜久竹夫, 二木厚吉, 足立暁生, 番場浩: 階層的流れ図言語 Hichart の情報処理記号, 早稲田大学情報科学研究教育センター紀要,Vol.3,pp.92-107,(1986).

9) 大原茂之: 木構造化チャートによるプログラム開発･保守技法, 情報処理学会論文誌, Vol.27, pp. 1019-1026,(1986).

10) Y.Miyadera,K.Imai,H.Kuwabara,H.Unno and T.Yaku: "ETA87 - An Extension of a Hichart flowchart processing system", 情報処理学会第 35 回全国大会講演論文集,pp.1201-1202,(1987).

11) K.Tsuchida: "The Complexity of Tidy Drawings of Trees", *"Topology and Computer Science"* (S.Suzuki Ed.), pp.487-520, Kinokuniya, Tokyo, (1987).

12) T.Yaku,K.Futatsugi,A.Adachi and E.Moriya: Hichart -A hierarchical flowchart description language-, *Proc IEEE COMPSAC*,Vol.11,pp.157-163,(1987).

13) 夜久竹夫, 杉田公生, 二木厚吉, 守屋悦朗:
"Hichart とプログラム開発環境 ETA_AIDE", 原田賢一編 "構造エディタ", 第 III 部 6 章, 共立出版, 東京,(1987).

14) 西野哲朗: 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,pp.81-92,(1988).

15) 郷信義, 小倉耕一, 土田賢省, 夜久竹夫, 岸本美紀: Hichart 流れ図の描画アルゴリズム, 日本ソフトウェア科学会第 5 回大会論文集,pp.181-184,(1988).

16) T.Nishino: Attribute graph grammars with applications to Hichart program chart editors, *Advances in Software Science and Technology*,Vol.1, pp.426-433,(1989).

17) 郷信義, 岡田直之, 岸本美紀, 土田賢省, 宮寺庸造, 夜久竹夫: Hichart プログラム図式の生成手法, 情報処理学会論文誌,Vol.31,pp.1463-1473,(1990).