

## 木構造図式の美的描画について

Tidy Drawing of Tree Structured Diagrams

郷信義（東海大学電子計算センター）、岸本美紀（（株）富士通プログラム技研）  
小倉耕一（北海道東海大学電子計算センター）、土田賢省（日本電気（株））、  
夜久竹夫\*（東京電機大学理工学部）

Nobuyoshi Go<sup>1</sup>, Miki Kishimoto<sup>2</sup>, Koichi Ogura<sup>3</sup>, Kensei Tsuchida<sup>4</sup> and  
Takeo Yaku<sup>5</sup>.

<sup>1</sup>Academic Computing Center, Tokai Univ, Hiratsuka, Kanagawa 25912, Japan

<sup>2</sup>Fujitsu Program Laboratory Ltd, Nakaku-Yokohama, Kanagawa 23100, Japan

<sup>3</sup>Computing Center, Hokkaido Tokai Univ, Sapporo, Hokkaido 00500, Japan

<sup>4</sup>NEC Corp., Shibaura-Minato, Tokyo 10800, Japan

<sup>5</sup>Inform Sci Dept, Tokyo Denki Univ, Hatoyama, Saitama 35003, Japan

\*This author is partly supported by a grant from Ichimaru Corp.

**Abstract.** Layout problems of cells in tree structured diagrams into a narrow area under some eumorphous conditions are called "tidy drawing problems of trees"[2,3]. We have been dealt with these problems concerning to so-called "tree structured program diagrams" (see e.g. [1,11]). In this paper, we survey recent results in [10,14] standing for practical program diagrams. Furthermore, we introduce five eumorphous conditions for which tidy drawing problems have  $O(n)$ ,  $O(n^2)$ ,  $O(n^3)$  and  $O(n^4)$  time complexities, and is NP-hard, respectively.

### 1. はじめに

”木構造図式”は長方形の箱（セルという）が木構造状に連結された図式のこととて、データとアルゴリズムを表示するために多くの場面で用いられている。それらの中には、いくつかのプログラム図式（たとえば [1, 1978]）が含まれ

る。我々はその種のプログラム図式を設計する [1, 5, 8, 11] と共に、木構造型プログラム図式を表示する問題を研究してきた [4, 7, 9, 14]。

一方、木構造図式一般の表示問題はアルゴリズム論の立場からも盛んに研究されている。Wetherell-Shannon は 2 分木の描画における美的条件を定式化するとともに線形時間描画アルゴリズムの例を示した [2, 1979]。Reingold-Tilford は Wetherell-Shannon らの美的条件を満たしながら更に”狭い”領域に 2 分木を描画する問題を研究し、線形時間アルゴリズムを示した [3, 1981]。Supowit-Reingold は 最小領域 に 2 分木を美的に描画する問題が実数座標の場合線形計画法に還元可能なため多項式時間で計算可能で、整数座標の場合 NP 完全であることを示した [6, 1983]。Tsuchida は Hichart 型のプログラム図式の描画問題 [4, 1983] を発展させて、一般の  $n$  分木の描画問題を研究し美的条件をいくつか新たに導入した。その結果、最小領域に描画する問題が”美しく”なる順に  $O(n)$ ,  $O(n^2)$ ,  $O(n^3)$ ,  $O(n^4)$ , NP 困難となるような条件とアルゴリズムの階層を得た [10, 1987]。さらに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て西野 [13, 1988] [15, 1989] により、 $O(n)$  に対応する美的条件と属性文法の関係が調べられている。

本稿では、2 節の前半で [10] の結果を解説する。そこで解説される手法は H i c h a r t, P A D 等に適用可能である。2 節の後半では [10] の美的条件をプログラム図式の最近の描画モジュールに適合するよう変更した問題を扱い時間計算量との関係を示す。新たに得られた手法は H i c h a r t, P A D だけでなく S P D, Y A C I I などにも適用可能である。3 節では実際のプログラム図式の描画問題との関係について [14] の一部を解説する。

なお、ここで紹介される手法はプログラム図式 (H i c h a r t のほか P A D, S P D, Y A C I I など) だけでなく木構造型のデータ表示一般にも適用可能である。

## 2. セルの大きさ一定の場合の描画

ここでは長方形が木構造上に配置された図式を 木構造図式 といい、各長方形を セル という。この節では各セルの大きさが一定であるような木構造図式を考える。

この場合木構造図式の描画問題は下のように木の描画問題に帰着される。

本論文では、我々は子供頂点(セル)に上から下へと順序がついている木(木構造図式)を対象とする。

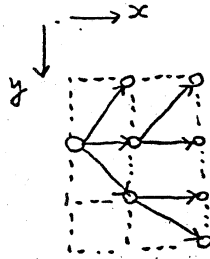
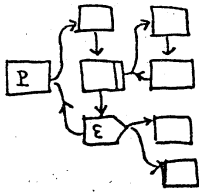


図 1. 木構造図式の例

図 2. 図 1 に対応する木の描画

はじめに、[10]で示された、描画の美的条件と対応するアルゴリズムの時間計算量を解説する。

本論文では[10]の描画の美的条件をセルの大きさ $1 \times 1$ のプログラム図式に適用するために、図2のようにx軸とy軸を入れ換える。即ち、ここでの木は、根を左側に持ち、左から右、上から下の方向に展開する。

定義 1. 木  $T$  の配置はつぎの関数  $\pi$  で表される。

$$\pi : T \text{ の頂点の集合} \rightarrow Z \times Z$$

ここで、頂点  $p$  の座標  $\pi(p) = (x, y)$  とすると、その  $x$  座標、 $y$  座標は、 $\pi_x(p)$ 、 $\pi_y(p)$  で表される。 $x$  軸は左から右へ、 $y$  軸は上から下の方向とする。□

木  $T$  の配置  $\pi(T)$  において、 $T$  の幅を次のように定義する：

$$\text{width}(\pi(T)) = \max \{ |\pi_x(p) - \pi_x(q)| ; p, q \text{ は } T \text{ の中の頂点} \}$$

頂点  $p$  のレベルは、 $p$  と根セルとの間の辺の数の数に 1 を加えた値で定義される。

頂点  $p$  において、関数  $\text{Index}$  を次のように定義する。

Index(p) = def=      0 : p が根セルであるとき  
                           i : p が p の親の i 番目の子供 であるとき

木の配置に関する 美的条件 は以下のように定められる。

条件 B 1. 木 T においてレベルが i である全ての頂点 p に対して、  
 $\pi_x(p) = i$     □

条件 B 2. 木 T において頂点 p が k 個 ( $k \geq 1$ ) の子供  $q_1, \dots, q_k$   
 ( $\text{Index}(q_i) = i, (1 \leq i \leq k)$ ) をもつならば、 $\pi_y(p) = \pi_y(q_1) +$   
 $[(\pi_y(q_k) - \pi_y(q_1)) / 2]$     □

条件 B 3. 木 T において頂点 p が  $k \geq 2$  個の子供  $q_1, \dots, q_k$  ( $\text{Index}$   
 $(q_i) = i, (1 \leq i \leq k)$ ) をもつならば、 $\pi_y(q_{i+1}) - \pi_y(q_i) \geq 1$  ( $1 \leq i \leq$   
 $k - 1$ )    □

条件 B 4. T を木とする。同じレベルで、 $\pi_y(p) < \pi_y(q)$  であるようなす  
 べての T の頂点 p, q において、 $\pi_y(p \text{ の末っ子}) < \pi_y(q \text{ の長男})$     □

条件 B 5. T1 と T2 が木 T の 同型な部分木 ならば、T1 と T2 は全く 同じ形  
 で配置されなければならない。    □

注: B 3 と B 4 からすべてのセル p, q に対して  $\pi(p) \neq \pi(q)$     □

条件 B #. 木 T において頂点 p が  $k \geq 3$  個の子供  $q_1, \dots, q_k$  ( $\text{Index}$   
 $(q_i) = i, (1 \leq i \leq k)$ ) をもつならば、  
 $\pi_y(q_{j+2}) - \pi_y(q_{j+1}) = \pi_y(q_{j+1}) - \pi_y(q_j)$  ( $1 \leq j \leq k - 2$ )    □

条件 B \$.  $\text{Index}(r_j) = \text{Index}(r_i) + 1$  であるような頂点  $r_i, r_j$  を根頂点  
 とするすべての木 T の部分木  $T_i, T_j$  に対して、  
 $\pi_y(T_j \text{ の一番上の頂点}) > \pi_y(r_i), \pi_y(r_j) > \pi_y(T_i \text{ の一番下の頂点})$     □

木  $T$  と配置  $\pi$  に対して関数  $\text{Intersect}$  を次のように定める：

$\text{Intersect}(\pi, T) = \text{def} = \max \{ \pi y(p) - \pi y(q) + 1 ; T_1 \text{ と } T_2 \text{ は、根頂点同士が兄弟頂点であり、} \text{Index}(T_2 \text{ の根頂点}) = \text{Index}(T_1 \text{ の根頂点}) + 1 \text{ であるような } T \text{ の任意の部分木。 } p, q \text{ は } T_1 \text{ と } T_2 \text{ の任意の頂点} \}$

条件  $B7(k)$ .  $k \geq 0$  に対して配置  $\pi$  が  $\text{Intersect}(\pi, T) \leq k$  を満たす。  $\square$

記法 1. [10] 上記の条件を組み合わせてレイアウトの美的条件  $E_0, E_1, E_k, E_*, E\#$  を以下のように定める。

$E_0 = B1 \text{ .and. } \dots \text{ .and. } B5 \text{ .and. } B\$ \text{ .and. } B7(0)$

$E_1 = B1 \text{ .and. } \dots \text{ .and. } B5 \text{ .and. } B\$ \text{ .and. } B7(1)$

$E_k = B1 \text{ .and. } \dots \text{ .and. } B5 \text{ .and. } B\$ \text{ .and. } B7(k), \quad (k > 1)$

$E_* = B1 \text{ .and. } \dots \text{ .and. } B5 \text{ .and. } B\$$

$E\# = B1 \text{ .and. } \dots \text{ .and. } B5 \text{ .and. } B\#$   $\square$

注： $E_0 = > E_1 = > E_k = > E_*$  である。

これらの美的条件の時間計算量は頂点の個数  $n$  に関して次のようになる。

定理 1 [10]. 上で定められた美的条件  $E_0, E_1, E_k (k > 1), E_*$  のそれぞれを満たす最小幅配置を見つける、計算時間が  $E_0: O(n), E_1: O(n^2), E_k: O(n^3), E_*: O(n^4)$  となるようなアルゴリズムが存在する。  $\square$

定理 2 [10]. あたえられた配置が上の条件  $E\#$  を満たしかつ最小幅かどうか判定する問題は NP 困難である。  $\square$

プログラム図式の表示においては条件  $B2$  は必ずしも適当ではない。即ち、上から下へ VDT 上に図式を表示する場合、親モジュールより先に子モジュールの図式が表示されてしまう。そこで、条件  $B2$  のかわりに、つぎの条件を導入する。

条件 B 2 ( j ) . 木 T の全ての頂点 p に対して、 p が k 個 (  $k \geq 1$  ) の子供  $q_1, \dots, q_k$  (  $\text{Index}(q_i) = i, (1 \leq i \leq k)$  ) をもつならば、  

$$\pi y(p) = \pi y(q_1) + \min \{j, \pi y(q_k) - \pi y(q_1)\}$$
 □

記法 2 . 美的条件  $E_0, E_1, E_k, E_*, E\#$  において B 2 のかわりに B 2 ( j ) (  $j \geq 0$  ) をおいて新たに美的条件を考え、それらをそれぞれ  $E_0(j), E_1(j), E_k(j), E_*(j), E\#(j)$  とかく。 □

このとき [ 1 0 ] の結果を応用することにより次の 2 つの定理がなりたつ。

定理 3 . 各  $j \geq 0$  に対して上で定められた美的条件  $E_0(j), E_1(j), E_k(j), E_*(j)$  のそれぞれを満たす最小領域配置を見つける、計算時間がそれぞれ  $O(n), O(n^2), O(n^3), O(n^4)$  となるようなアルゴリズムが存在する。 □

定理 4 . 各  $j \geq 0$  に対して、与えられた配置が上の条件  $E\#(j)$  を満たしかつ最小幅かどうかを判定する問題は NP 困難である。 □

### 3. 大きさ不定のセルからなる図式の描画

本節では、前節の美的条件とアルゴリズムを拡張してセルの大きさが一定でない木構造図式の描画問題に適用する方法について述べる。

セルの大きさが一定でないような木構造図式に適合するように前節の美的条件を拡張したものを以下に示す。まず、拡張するにあたりセルの座標を次のように定める。

座標の単位は、標準セル(大きさ  $1 \times 1$ )の描ける長さとし、セル p の座標が  $(x, y)$  のとき、セル p の左側に x 個、上に y 個の標準セルを描くことができるものとする。木構造図式の配置  $\pi$  は木の配置と同様に定められる。

セル p の幅が s である ( $\text{width}(p) = s$ ) とはセル p 中の y 軸方向に s 個の標準セルを描けることを意味する。セル p の深さが t である ( $\text{depth}(p) = t$ ) とは、

セル  $p$  の中の  $x$  軸方向に  $t$  個の標準セルを描けることを意味する。

木構造図式  $T$  の幅  $\text{width}(\pi, T)$  を次のように定義する。

$\text{width}(\pi, T) = \max \{ \pi y(p) + \text{width}(p) - \pi y(q) - 1 ; p, q \text{ は } T \text{ 中のセルで, } \pi y(p) > \pi y(q) \}$

セル  $p$  のレベルは、 $p$  と根セルの間の辺の数に 1 を加えた値で定義される。

仮定. 本論文では、セル  $p$  が子供を持つならば  $\text{depth}(p)=1$  と仮定する。セル  $p$  があらず操作の論理的レベルと見かけ上のレベルを一致させるためである。□

条件 B 2' . 木構造図式  $T$  において、セル  $p$  が  $k \geq 1$  個の子供  $q_1, \dots, q_k$  ( $\text{Index}(q_i) = i, 1 \leq i \leq k$ ) をもつならば  $\pi y(p)$  は次をみたす:  $g = \pi y(q_k) + \text{width}(q_k) - \pi y(q_1) - 1$  とするとき

$$\pi y(p) + \lfloor \text{width}(p)/2 \rfloor = \pi y(q_1) + \lfloor g/2 \rfloor \quad \square$$

条件 B 3' . 木構造図式  $T$  において、セル  $p$  が  $k \geq 2$  個の子供  $q_1, \dots, q_k$  ( $\text{Index}(q_i) = i, 1 \leq i \leq k$ ) をもつならばすべての  $i' (1 \leq i' \leq k-1)$  について  $\pi y(q_{i'+1}) - \pi y(q_{i'}) - \text{width}(q_{i'}) + 1 \geq 1$  □

条件 B 4' . 木構造図式  $T$  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  $\pi y(q \text{ の長男}) > \pi y(p \text{ の末っ子}) + \text{width}(p \text{ の末っ子})$  □

条件 B 6' . 木構造図式  $T$  において、セル  $p$  と配置  $\pi$  に対して、 $\text{area}(p, \pi) = \text{def} = \{ (x, y) ; \pi x(p) \leq x \leq \pi x(p) + \text{depth}(p) - 1, \pi y(p) \leq y \leq \pi y(p) + \text{width}(p) - 1 \}$

としたとき、すべての  $p, q (p \neq q)$  について  $\text{area}(p, \pi) \cap \text{area}(q, \pi) = \emptyset$ . □

条件 B 8' . 木構造図式  $T$  において、 $\text{Index}(r_i) + 1 = \text{Index}(r_j)$  であるようなセル  $r_i, r_j$  を根セルとするすべての部分木構造図式  $T_i, T_j$  に対して、

$\pi y(T_i \text{ の一番下のセル}) + \text{width}(T_i \text{ の一番下のセル}) < \pi y(r_j) + \lfloor \text{width}(r_j) / 2 \rfloor$  かつ  $\pi y(r_i) + \lfloor \text{width}(r_i) / 2 \rfloor < \pi y(T_j \text{ の一番上のセル})$  □

2節の美的条件 $E_0$ 、 $E_1$ 、 $E_k$ 、 $E_*$ 、 $E\#$ 、 $E_0(j)$ 、 $E_1(j)$ 、 $E_k(j)$ 、 $E_*(j)$ 、 $E\#(j)$ において条件B2、B3、B4、B $\$$ のかわりにB2'、B3'、B4'、B6'、B $\$$ 'をいれてあらたに美的条件 $E_0'$ 、 $E_1'$ 、 $E_k'$ 、 $E_*'$ 、 $E\#'$ 、 $E_0'(j)$ 、 $E_1'(j)$ 、 $E_k'(j)$ 、 $E_*'(j)$ 、 $E\#'(j)$ を考えることが可能である。我々はプログラム図式の描画モジュール(HfromP (Hichart from Pascal converter) という)に $E_0'$ 、 $E_*'$ 、 $E_0'(j)$ 、 $E_*'(j)$ を組み込んで利用中である。

なお、実際の描画モジュールでは、1ページあたりの領域が縦横とも有限であるために、各セルがページ境界にまたがらないように図式が描画される必要がある。このため、レイアウトモジュールとセルの大きさの決定モジュールを並行して動かす必要がある。

#### 4. おわりに

はじめに2節の前半でセルの大きさが一定であるような木構造図式の描画における美的条件と計算時間の関係について[10]の結果を紹介した。2節の後半では[10]の結果をより広い範囲のプログラム図式の描画問題に拡張した。更に3節で、セルの大きさ不定のプログラム図式の描画に応用している状況を紹介した。今後の課題としては、3節で述べた美的条件に対応する計算量を評価するとともに、美的条件のバリエーションを増やすことが必要である。また、3節の最後に述べたように本稿で紹介した描画モジュールとセルの大きさ決定モジュールとの相互作用についても研究する必要がある。

以下の図3に描画モジュールにより出力されたプログラム図式を示す。

#### 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、電子通信学会オートマトンと言語研究会資料 AL78-47 (1978), 61 - 66.
2. C. Wetherell and A. Shannon, Tidy drawing of trees, IEEE Transac. SE



- 5 (1979), 514 - 520.
3. E.M. Reingold and J.S. Tilford, Tidy drawing of trees, IEEE Transac. SE-7 (1981), 223 - 228.
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 H i c h a r t の処理系実現、情報処理学会第27回全国大会講演論文集 (1983), 549 - 550.
5. 夜久竹夫・二木厚吉・足立暁生、階層的フローチャート記述言語 H i c h a r t によるプログラムの記述、" 計算機科学の基礎理論報告書" (1984)、484 - 494、昭和58年度文部省科学研究費による研究組織 (L A シンポジウム1983)。
6. K. J. Supowit and E. M. Reingold, The complexity of drawing trees n icely, Acta Informatica 18 (1983), 377 - 392.
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 H I C H A R T 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、階層的流れ図言語 H i c h a r t 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 3 (1986)、92 - 107.
9. 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 (1987)、1201-1202.
10. K. Tsuchida, The Complexity of Tidy Drawings of Trees, in "Topology and Computer Science" (S.Suzuki Ed.), Kinokuniya, Tokyo, 1987, 487-520.
11. T. Yaku, K. Futatsugi, A. Adachi and E. Moriya, HICHART - A hierarchical flowchart description language -, Proc IEEE COMPSAC 11 (1987), 157-163.
12. 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" , 共立出版 , 1987, 第三部 6章.
13. 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 5 (1988), 81-92.

14. 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 (1988)、181-184.

15. T. Nishino, Attribute graph grammars with applications to Hichart program chart editors, Advances in Software Science and Technology 1 (1989), to appear.

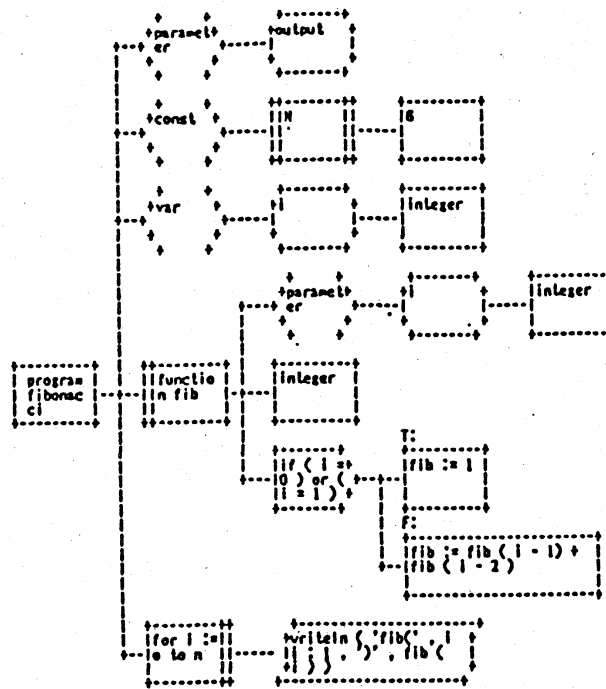


図3. HfromPの出力例