Nicely Drawing Tree-Structured Diagrams on the Euclidian Plane

°Youzou Miyadera[1] Koushi Anzai[2] Hideaki Konya[1] Kensei Tsuchida[3] Takeo Yaku[4]
[1]Tokyo Denki Univ. [2]Kanto Gakuen Univ. [3]Toyo Univ. [4]Nihon Univ.

7E−5

1 Introduction

A *tree-structure* is $T = (V, E, r, width, depth, Int_x, Int_y)$, where (V, E) is an *ordered tree* (a *tree* denotes an ordered tree), V is a set of *cells*, and E is a set of *edges*. The *root* cell is $r \in V$. The map $width : V \to R$ is the *width function* of the cells. The vertical length is represented by $width(p)$, which is called the *width of the cell p*. The map $depth : V \to R$ is the *depth function* of the cells. The horizontal length is represented by $depth(p)$, which is called the *depth of the cell p*. A real value Int_x (Int_y) is *the minimum interval of cells* with respect to the x-coordinate (y-coordinate). Int_x and Int_y are spaces where to draw connection lines among cells.

A *placement* of a tree-structure $T = (V, E, r, width, depth, Int_x, Int_y)$ is defined by the function $\pi : V \to R^2$.

$\pi_x(p)$ and $\pi_y(p)$ denote the x-coordinate and the y-coordinate of p respectively, where $\pi(p) = (x, y)$. We suppose that the x-coordinate directs from left to right and the y-coordinate directs downward because of the application to program diagrams, and we suppose that the topleft point of a cell is assigned by its coordinate.

Let T be a tree-structure and π be a placement of T. Then, $D = (T, \pi)$ is called a *tree-structured diagram*.

We define the location of a cell as follow. The *width* of a tree-structured diagram $D = (T, \pi)$ is defined by : $width(T, \pi) \equiv \max\{ \pi_y(q) + width(q) - \pi_y(p) \mid p$ and q are cells in T and $\pi_y(p) \le \pi_y(q)\}$.

The *level* of a cell p is defined by the number of edges between the cell p and the root cell. For a cell p, the function *Index* is defined by :

$$Index(p) \equiv \begin{cases} 0: & \text{if } p \text{ is the root cell} \\ i: & \text{if } p \text{ is the } i\text{-th child of the parent of } p. \end{cases}$$

We have implemented a system for generating **Hichart** [3] tree-structured diagrams which is output on the character displays such as used with PC and Host machines. Therefore, the eumorphous conditions are also applied on the integral lattice. Because of the recent widespread use of Unix workstations and bit-map displays, however, we are forced to develop the **Hichart** system on bit-map displays, on which the location is indicated by using values on the Euclidian plane.

In this paper, we modify the eumorphous conditions [1, 4] and introduce new conditions on the Euclidian plane. Next, we provide $O(n)$-time algorithm that corresponds to the new conditions. Results of this paper will be applied to drawings of structured program diagrams including **Hichart**, **PAD**, **SPD**, **TSF**, **tree-styled browser** among others.

2 Eumorphous Conditions for Program Diagrams

For a tree-structure $T = (V, E, r, width, depth, Int_x, Int_y)$, we introduce constraints concerning the placement of T.

Constraint $B1_R$ (cf.[4]). For a tree-structured diagram (T, π), if the level of a cell p equals the level of a cell q, and $\pi_y(p) < \pi_y(q)$, then $\pi_y(\textit{the oldest child of } q) \ge \pi_y(\textit{the youngest child of } p) + width(\textit{the youngest child of } p) + Int_y$.

For a tree-structured diagram (T, π) and a cell p, let $\textit{pseudo-area}(p, \pi)$ be a set defined by: $\textit{pseudo-area}(p, \pi) \equiv \{ (x, y) \mid \pi_x(p) - Int_x/2 \le x \le \pi_x(p) + depth(p) + Int_x/2, \pi_y(p) - Int_y/2 \le y \le \pi_y(p) + width(p) + Int_y/2 \}$.

Constraint $B2_R$ (cf.[4]).
$\textit{pseudo-area}(p, \pi) \cap \textit{pseudo-area}(q, \pi) = \phi$ for all p and $q (p \ne q)$.

Constraint $B3_R$ (cf.[1]). For a tree-structured diagram (T, π), if T_1 and T_2 are topologically isomorphic *sub-tree-structures*, then T_1 and T_2 are placed in the same form with respect to a parallel movement.

Constraint $B4^1_R$ (cf.[4]). In a tree-structured diagram (T, π), $\pi_x(p) = i$ is satisfied for any cell p in T with the level i, where i is an integral value.

Constraint $B4^2_R$. For a tree-structured diagram (T, π), if a cell p has k children $q_1, \ldots, q_k$ ($Index(q_i) = i, 1 \le i \le k$), then $\pi_x(q_i)$ satisfies the following :

$$\pi_x(p) + depth(p) + Int_x = \pi_x(q_i), \quad (1 \le i \le k).$$

Constraint $B5_R$ (cf.[4]). For a tree-structured diagram (T, π), if a cell p has k children $q_1, \ldots, q_k (Index(q_i) = i, 1 \le i \le k)$, then $\pi_y(p)$ satisfies the following : $\pi_y(p) + width(p)/2 = \pi_y(q_1) + (\pi_y(q_k) + width(q_k) - \pi_y(q_1))/2$.

Constraint $B5^+_R(a)$. For a given number a which is a non-negative real value, in a tree-structured diagram (T, π), let a cell p have k children $q_1, \ldots, q_k$ ($Index(q_i) = i$, $1 \le i \le k$), and $c = (\pi_y(q_k) + width(q_k) - \pi_y(q_1) - width(p))/2$. Then, $\pi_y(p)$ satisfies the following:

$$\pi_y(p) = \begin{cases} \pi_y(q_1): & c < 0 \\ \pi_y(q_1) + a: & 0 \le a \le c \\ \pi_y(q_1) + c: & 0 \le c < a. \end{cases}$$

A function *Intersect* of the set of tree-structured diagrams to the real numbers is defined by : $Intersect(T, \pi) \equiv max_{(p \in T_1, q \in T_2)}\{\pi_y(q) - (\pi_y(p) + width(p))\}$, where T_1 and T_2 are arbitrary sub-tree-structures for which the root cells are brothers and $Index$(the root of T_2)> $Index$(the root of T_1). p and q are arbitrary cells in T_1 and T_2, respectively.

Constraint $B6_R(b)$ (cf.[4]) . For a given non-negative real value b, a placement π satisfies : $Intersect(T, \pi) \leq -b$.

Constraint $B7_R$. Let a cell p be in a tree-structured diagram (T, π), and $p_0, p_1, \ldots, p_m$ be the cells on the path from the root cell $r = p_0$ of T to $p_m = p$. Then, $\pi_x(p) = \Sigma_{(0 \leq i \leq m-1)}(depth(p_i) + Int_x)$, and there exists an integral value M satisfying $(depth(p_i) + Int_x) \geq M$ $(0 \leq i \leq m-1)$.

Notation 1 The eumorphous conditions for a tree-structured diagram denote the combinations of the above constraints :
$E_1(a) = B1_R \wedge B2_R \wedge B3_R \wedge B4^2_R \wedge B5^+_R(a) \wedge B7_R$,
$E_2(a) = B1_R \wedge B2_R \wedge B3_R \wedge B4^2_R \wedge B5^+_R(a) \wedge B6_R(-Int_y) \wedge B7_R$.

These constraints are efficient for the attractive printing of several kinds of program diagram languages, and these conditions are applicable not only to **PAD** and **Hichart** diagrams but also to **SPD** and **TSF** diagrams.

3 Drawing Methods

In this section, we consider a method of obtaining placements under the eumorphous conditions introduced in the preceding section. In this paper, we express **Procedure** by *Pascal − Like procedure*. We present the following procedure *Fine* which satisfies the eumorphous condition $E_1(a)$ $(a \geq 0)$, and provides a placement narrower than or equal to the placement $\pi_{E_2(a)}$ by a procedure developed by the authors satisfying the condition $E_2(a)$. This procedure is a modification of *Init* [4] for extension to the Euclidian plane.

Procedure *Fine*: the fine placement
[Calling sequence] *Fine* $(T_p, a, \pi, x_p, maxdepth)$
[Input] T_p : a tree structure
$T_p = (V, E, p, width, depth, Int_x, Int_y)$.
a : a given real number of $E_1(a)$.
x_p : x-coordinate of the placement π of the cell p.
[Output] π : the placement for T_p satisfying $E_2(a)$.
$maxdepth$: x-coordinate which is the maximum depth of the moved sub-tree-structure T_p.
[Global Variables]
$free_y$: the bounded value of y-coordinate of the placement π for the cell p.
$CheckGap[\,]$: A table contains the lower bound of y-coordinate with respect to x-coordinate $M \times i$, where M denotes $min\{depth(p) \mid p \in V\} + Int_x$.
[Local Variables]
x_p : x-coordinate of the placement π of the cell p. (When the procedure *Fine* is called recursively by itself, the value of x_p is saved on the program stack.)
$maxdepth_p$: a temporary variable for analysing the value of $maxdepth$ for the cell p.
[Constants]
Int_x, Int_y: the interval between cells for x-coordinate and y-coordinate, respectively.
[Method]
begin
  $maxdepth := x_p + depth(p) + Int_x$;
  if (p is a leaf) **then**
    begin
      $\pi(p) := (x_p, free_y)$;
      $\pi(p).Gap := free_y - max_{(\lceil x_p \rceil \leq i \leq \lceil x_p + depth(p) \rceil)} CheckGap[i]$;
      $free_y := free_y + width(p) + Int_y$;
      for $i := \lceil x_p \rceil$ **to** $\lceil x_p + depth(p) \rceil$ **do** $CheckGap[i] := free_y$;
    end
  else
    begin
      $k := Index$(the last child of p);
      Let q_i $(1 \leq i \leq k)$ be a child of p;
      for $q := q_1$ **to** q_k **do**
        begin
          $Fine(T_q, a, \pi, x_p + depth(p) + Int_x, maxdepth_p)$;
          $maxdepth := max\{maxdepth, maxdepth_p\}$;
        end;
      $c := ((\pi_y(q_k) + width(q_k) - \pi_y(q_1) - width(p))/2$;
      case
        $c < 0 : \pi(p) := (x_p, \pi_y(q_1))$;
        $0 \leq a \leq c : \pi(p) := (x_p, \pi_y(q_1) + a)$;
        $0 \leq c < a : \pi(p) := (x_p, \pi_y(q_1) + c)$;
      end
      $\pi(p).Gap := min\{\pi_y(p)$
        $-max_{(\lceil x_p \rceil \leq i \leq \lceil x_p + depth(p) \rceil)}\{CheckGap[i]\}, \pi(q_1).Gap\}$;
      for $i := \lceil x_p \rceil$ **to** $\lceil x_p + depth(p) \rceil$ **do**
          $CheckGap[i] := \pi_y(p) + width(p) + Int_y$;
      if $\pi(p).Gap > 0$ **then**
        begin
          for $i := \lceil x_p \rceil$ **to** $\lceil maxdepth \rceil$ **do**
            $CheckGap[i] := CheckGap[i] - \pi(p).Gap$;
          $\pi(p).Gap := 0$;
        end
      $free_y := max\{free_y, CheckGap[\lceil x_p \rceil]\}$;
    end
  return(π , $maxdepth$)
end. { *Fine* }

We obtain the following theorem about the time complexity of constructing for a placement of a tree-structure.

Theorem 1. *Procedure Fine provides data for a placement π with the narrowest width, which satisfies the condition $E_1(a)$, and runs in $O(n)$-time.*

4 Conclusion

We introduced new eumorphous conditions for drawing tree-structured diagrams on the Euclidian plane and provided $O(n)$-time algorithm under the conditions. Then we had a new relationship between eumorphous conditions and the time complexity. Unsolved problems include expansion of the variation of eumorphous conditions and examination of its relationship with time complexity.

References

[1] Supowit, K.J. and Reingold, E.M., The Complexity of Drawing Trees Nicely, *Acta Inform.*, Vol.18, (1983), 377-392.

[2] Tsuchida, K., The Complexity of Tidy Drawings of Trees, Topology and Computer Science(S.Suzuki ed.), Kinokuniya, Tokyo(1987), 487-520.

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

[4] Ogura, K., Go, N., Kishimoto, M., Miyadera, Y., Okada, N., Tsuchida, K., Unno, H. and Yaku, T., Generation of the Hichart Program Diagrams, *J. Inform. Process.*, Vol.15, No.2(1992), 293-300.