Hichart プログラム図式の生成手法†

郷 信義†† 岸本美紀†††* 宮寺庸造‡
岡田直之‡‡ 土田賢省‡‡** 夜久竹夫‡

階層的プログラム図式の生成手法，特に表示に関する話題を報告する．Hichart[1),6),12)]は階層的プログラム図式言語の一種である．はじめに我々が実現した Hichart プログラム図式の生成系の概要を紹介する．次に我々は，その生成系の上で図式の描画問題に注目する．セルの大きさが一定の図式のみに適用可能な木の描画に関する美的条件[2),3),8)]を拡張して，セルの大きさが一定でないような木構造図式にも適用可能な美的条件を得る．得られた美的条件に対応するアルゴリズムとその計算量を考察する．なお，本論文で導入した美的条件は実際の生成系に組み込まれている．したがって我々の Hichart 生成系は定式化された美的条件を組み込んだ最初の生成系となる．また，本論文で得られる結果は，木構造型のプログラム図式の多くに適用可能である．

1. まえがき

1970年頃からプログラムの表示方法の一種として図式型プログラム言語が多く提案されてきた．それらには例えば，**NSD**, **Hichart**, **HCP**, **SPD**, **TSF**, **PAD**, **YAC II**, **50 SM** などがある[14)]．我々の対象とする **Hichart**[4)] は，木フローチャート[1)]を基礎図式とする図式言語（図式の記述仕様）に付けられた名前である．木フローチャートは，現在広く利用されている繰り返し記号  を最初に導入した図式である．その特徴は主に次の3点である．

1）Neumann のプログラムフローチャートの制御線をそのまま残した．

2）制御構造を表す箱記号として，繰り返しと分岐の2種類だけを追加した．

3）その制御記号を要とする扇型の疑似木構造型グラフにプログラムを表現した．

このため **Hichart** は他の多くの図式言語と異なり，流れ図としての性質を持ち，流れと制御構造を同時に階層的に表示することが可能となっている．

1980年頃からは，上記の多くのプログラム図式に対して処理系の実現が行われ，プログラム図式を対象とする図式生成系，図式エディタ，図式コンパイラなどの処理系が開発された[6),9),13),14)]．この種の処理系においてプログラム図式を美しくかつ効率的に描画することは重要な問題であるが，現在のところこの問題に関する体系的な研究はなされていない．

本研究の動機は描画問題をはじめとする体系的な理論を取り入れた図式処理系を実現することである．そのための最初のステップとして，本論文では次の2点を取り扱う．

1）グラフ理論的手法一般の取り入れやすい構造をした生成系の実現．

2）体系的理論の1つとして描画問題を定式化して，その数学的性質を明らかにする．

我々の生成系は **Pascal** ソースプログラムからプログラム図式を生成するシステムの一部として実現した．その開発経過は次のようになる．

1）当初は，セルといわれる長方形の大きさが一定（1(縦)×1(横)）であるような図式を対象とした[4)]．

2）次に，木の配置に関する結果[11)]を取り入れた図式配置の最適化アルゴリズムと，字句解析に基づくセルの大きさ決定モジュールを作成し，セルの縦方向の大きさが一定でなく横方向の大きさが一定であるような図式を対象とする生成系

† Generation of the Hichart Program Diagrams by NOBUYOSHI GO (Academic Computing Center, Tokai University), MIKI KISHIMOTO (FUJITSU Program Laboratory Corporation), YOUZOU MIYADERA (Department of Information Science, Faculty of Science and Engineering, Tokyo Denki University), NAOYUKI OKADA, KENSEI TSUCHIDA (NEC Corporation) and TAKEO YAKU (Department of Information Science, Faculty of Science and Engineering, Tokyo Denki University).
†† 東海大学電子計算センター
††† 富士通プログラム技研(株)
‡ 東京電機大学理工学部情報科学科
‡‡ 日本電気(株)
* 現在 富士通(株)
Present, FUJITSU LIMITED
** 現在 神奈川大学工学部工業経営学科
Present, Department of Industrial Engineering and Management, Faculty of Engineering, Kanagawa University

を実現した[7].

3) 最後に配置モジュール，印字モジュールおよびセルの大きさ決定モジュールすべての対象を横方向にも拡張して，セルの大きさが縦横両方向とも一定でないような図式を対象とする生成系を実現した[10].

2章で我々の **Hichart** 図式生成系のデータ構造と処理手法を紹介する．我々は，プログラム図式を階層型のグラフ（リスト）により表現することを試みた．その結果本論文で述べるようなグラフ理論的手法が自然に実現可能となった．さらに我々の生成系では，プログラム図式データを中心として，様々なサブシステムがその周辺に木構造的なモジュール構造をなしている．その結果以後の改良や機能の付加が容易となった．

さて，我々が扱う“木構造型図式”は長方形の箱（セル）が木構造状に2次元格子点上に配置された図式のことで，プログラム図式だけでなく，システムの組織図や階層を表示するためにも用いられている．この木構造を美的に描画する問題に似た問題として“木の描画問題”が知られている．木の描画問題はグラフアルゴリズム理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]．Reingold-Tilford は Wetherell-Shannon らの美的条件を満たしながらさらに“狭い”領域に2分木を描画する線形時間アルゴリズムを示した[3]．Supowit-Reingold は同じ美的条件のもとで最小領域に2分木を描画する問題が，実数座標の場合には線形計画法に還元可能なため多項式時間で計算可能であり，一方，整数座標の場合には NP 完全であることを示した[5]．土田は **Hichart** 型のプログラム図式の描画問題[4]にヒントを得て，一般の n 進木に対して美的条件をいくつか新たに導入した．さらにそれらの美的条件のもとで，最小領域に木を描画する問題が“美しさ”の条件がきつくなる順に $O(n)$, $O(n^2)$, $O(n^3)$, $O(n^4)$, NP 困難となることを示した[11]．そこで得られた木に対する美的条件をプログラム図式に対応させる試みは郷ほか[17],[18]により行われた．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により，いくつかの美的条件に対応する属性文法とその性質が調べられている[16],[19].

我々は3章でプログラム図式を指向した木構造図式に対する美的条件を導入し，4章でそれらの美的条件に対応するアルゴリズムと計算量について考察する．

まず3章で木構造図式の美的条件を扱う．グラフアルゴリズム理論で研究された n 進木に対する美的条件[11]は，大きさ一定のセルからなるプログラム図式にのみ適用可能である．そこで我々はその条件[11]を大きさの一定でないセルからなる木構造に対応するよう拡張して定式化する．さらに従来の美的条件[5],[11]に含まれる，親頂点が子頂点のちょうど中央に配置されるという条件は，**PAD**, **Hichart** など一部のプログラム図式の特殊な表示場面（例えば清書）でしか適用できない．したがってプログラム図式の美的条件としては，親頂点は一番上の子頂点から k レベル（$k \geqq 0$）だけ下に配置されるようになっている方がよい場合がある．そこで我々は，そのような条件も定式化した．その結果得られる美的条件は **PAD**, **Hichart** の編集時の表示を含めて，**SPD**, **YAC II** などの表示にも適用可能である．n 進木に関しては，部分木間の横方向の重なりの大きさが配置アルゴリズムの計算量に影響を与えることが知られている[11]．そこで我々は，木構造について重なりを許す美的条件 $E_*(j)$ と重なりを許さない美的条件 $E_0(j)$ を考え，それらを定式化する．

4章では3章で導入された美的条件 $E_0(j)$ と $E_*(j)$ に対応する2つのアルゴリズムを考える．それらは従来のアルゴリズムを3章での美的条件の拡張と変更に対応するよう変更したものである．さらに，我々が新たに導入した美的条件と計算量の関係について述べる．

本論文で新たに得られた手法は **Hichart**, **PAD** だけでなく **SPD**, **YAC II** や木構造型のデータ表示の生成系一般にも適用可能である．

2. プログラム図式生成系

この章では本稿で対象とするプログラム図式処理系の中で用いられているプログラム図式の内部表現とその処理系の内部構造について紹介する．3章以降において描画問題はその内部表現と処理系を前提として考察される．

はじめに **Hichart** の言語仕様について解説する．**Hichart** は**図1**に示すような制御記号からなるグラフである．

Hichart のプログラム図式は木フローチャートともいわれ，次のような図式である．以下の**図2**と**図3**で **Hichart** プログラム図式を説明する．

次にこの処理系における内部表現について述べる．**Hichart** 処理系では **Hichart** プログラム図式を共通

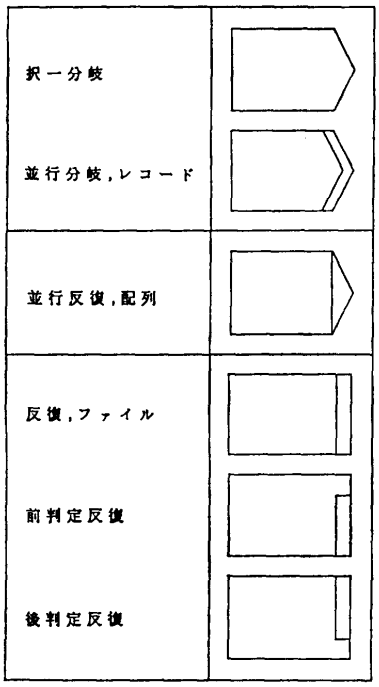

図 1 Hichart の制御記号
Fig. 1 Hichart control symbols.

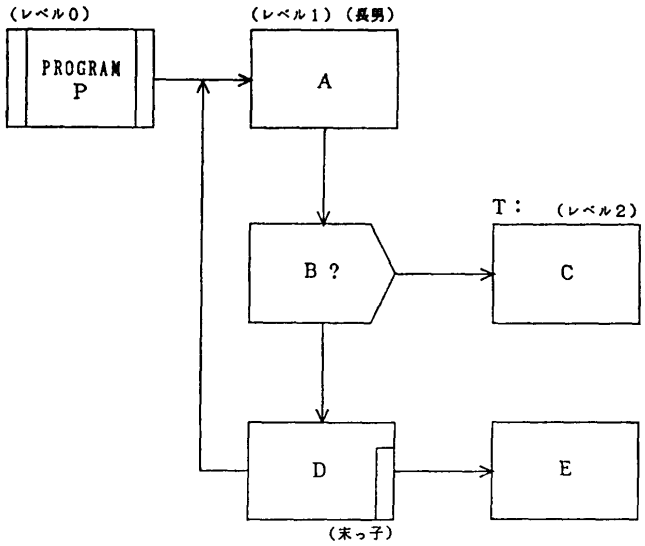

図 2 Hichart 図式の例
Fig. 2 An example of Hichart program diagram.

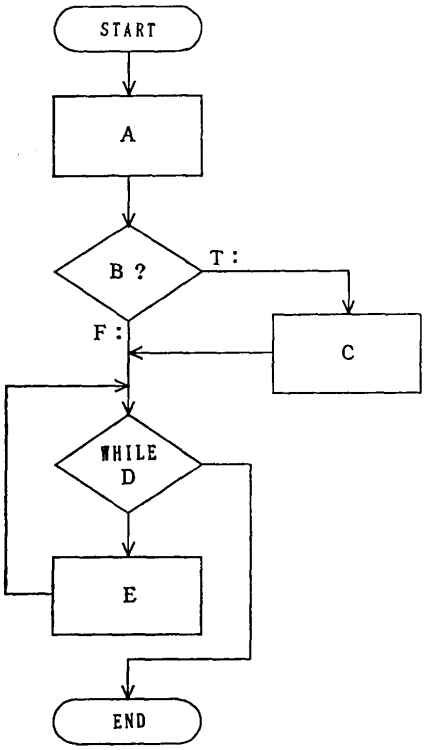

図 3 図2に対応する Neumann 流れ図
Fig. 3 A Neumann flowchart corresponding to Fig. 2.

の内部表現で表し，統一的に扱う．この内部表現はHコードと呼ばれる．Hコードは1つのセルの情報を一レコードで表す．通常，グラフは頂点と辺によって表されるが，Hコードではセルの情報の中にセル同士の接続関係の情報を含めることにする．Hコードの構成を図4に示す．図5に **Hichart** 図式のHコード表現を示す．

我々が対象としている **Hichart** によって記述されるプログラムを扱うプログラム開発環境に ETA_AIDE[15] がある．ETA_AIDE は仕様書，プログラ

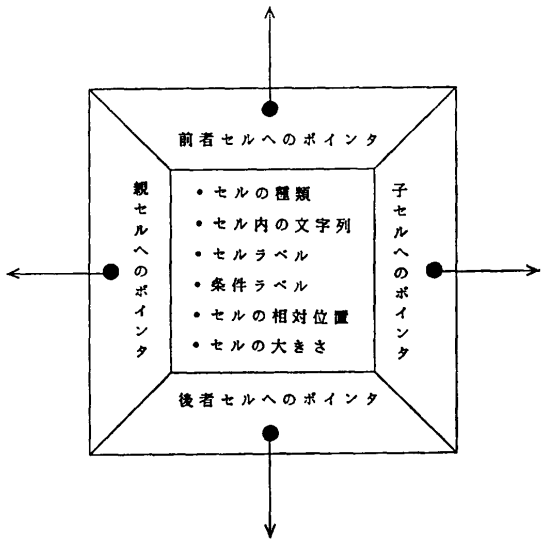

図 4 内部表現Hコードの構成
Fig. 4 Structure of internal representation H-code.

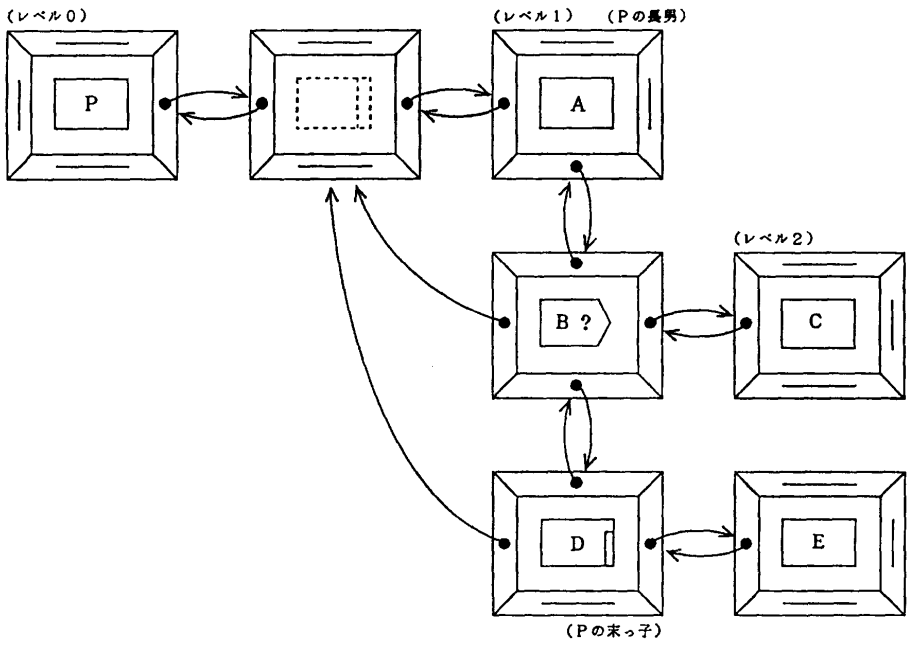

図 5 内部表現Hコードによる Hichart プログラム図式の表現例 (概念図)
Fig. 5 An example of Hichart program diagram represented by internal representation H-code (an outline).

ム図式，ソースプログラム，関係データベースそれぞれを管理するシステムとそれらの間のフィルタからなる．

この ETA_AIDE の中で中核となるのが **Hichart** プログラム図式の処理系 Eta/H* で，それは次の 3 つのツールから構成される．

① Eta/HED (Hichart flowchart EDitor)
② Eta/HfromP (Hichart from Pascal translator)
③ Eta/HtoP (Hichart to Pascal translator)

Hコードを中心として，これら 3 つのツールからなる処理系の構造を**図 6** に示す．

生成系と描画問題はこのうち①エディタ HED と②Pascal オートフローチャータ HfromP に関係がある．以下で生成系の機能と内部構造を示す．生成系は次の 3 つのモジュールによって構成されている．

① HCellOpt (Hcode CELL size OPTimizer)：セルの中に記述する文字列を整理，セルを描く際の大きさを決定してHコードリスト内 (図 4 参照) に書き込む (約 500 Pascal ステップ)．
② HListLO (Hcode LIST Layout Optimizer)：**Hichart** 流れ図を描く際の各セルの位置を決

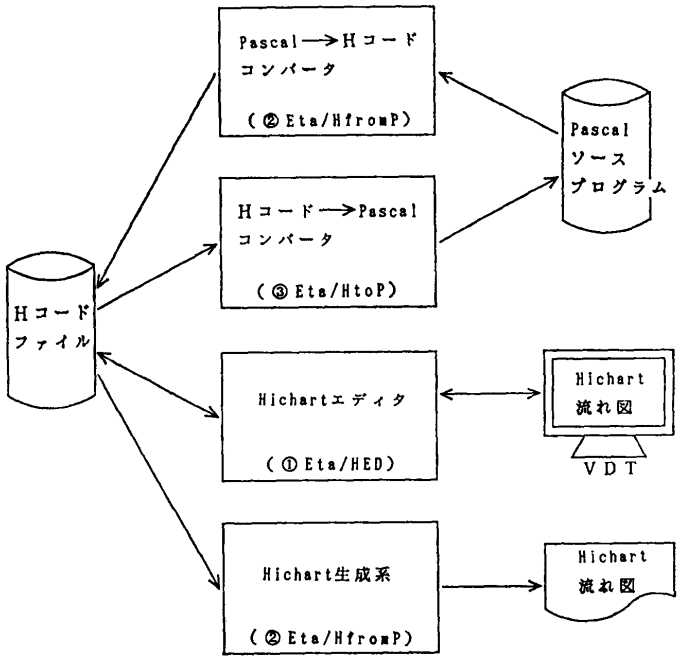

図 6 Hichart プログラム図式処理系の構造
Fig. 6 Structure of the processing system of Hichart program diagram.

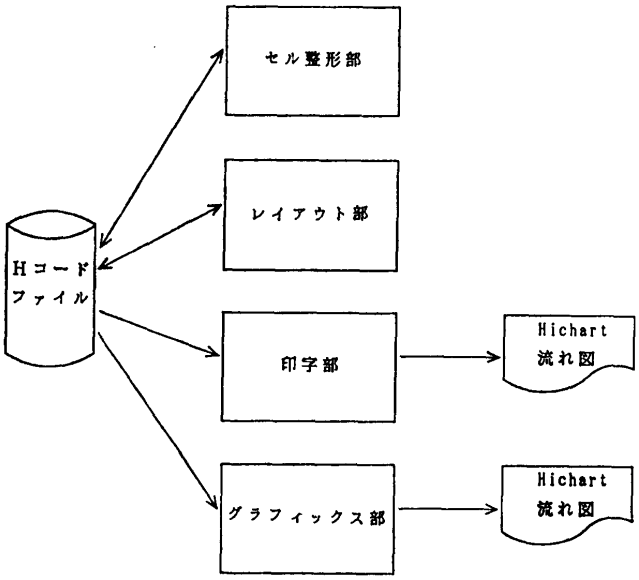

図 7 Hichart プログラム図式生成系のシステム流れ図
Fig. 7 A system flowchart of the generating system of Hichart program diagram.

定し，Hコードリストに記入する（約 1,000 Pascal ステップ）．

③ HListToChar (Hcode LIST TO CHARacter format converter)：生成されたHコードリストをもとに，**Hichart** 流れ図を印字装置に出力する（約 2,000 Pascal ステップ）．

生成系全体の構造を図 7 に示す．3章と4章で示される手法はこのうちの②HListLO に利用されている．

3. プログラム図式の描画条件

この章で我々は，大きさが一定でない“セル”からなる木構造型プログラム図式を考え，その図式を描画する際の図式の“美的”条件を考察する．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．

はじめに，描画問題を木構造型プログラム図式一般に適用するために次の用語を定める．

定義 1. 木構造は

$T=(V, E, r, \text{width}, \text{depth})$

である．ここで，(V, E) は順序木（とくに断らない限り本論文では木と書いたら順序木を表すこととする），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．写像 width：$V\to Z$ はセルの幅関数，width (p) はセル p の縦の長さ（幅という）を表す．写像 depth：$V\to Z$ はセルの深さ関数，depth (p) はセル p の横の長さ（(階層レベルの) 深さという）を表す． □

定義 2. 木構造 T の配置は次の関数 π で表される．

π：T のセルの集合$\to Z\times Z$

ここで，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． □

組 (T, π) を木構造図式という．木構造は各頂点に2つの属性 width (p) と depth (p) が割り当てられた根付き木である．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．セルは各長方形を意味し，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，プログラム図式を対象とするため図式の x 方向の長さを（階層レベルの）深さとし，y 方向の長さを（プログラムの）幅ということにする．さらに，我々は子供頂点（セル）に上から下へと順序がついている順序木（順序木構造図式）を対象とする．セルの座標を次のように考える．

座標の単位は，標準セル（大きさ 1×1）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 (T, π) の幅 width (T, π) を次のように定義する．

$$\text{width}(T,\pi) =_{\text{def}} \max\{\pi_y(p)+\text{width}(p)-\pi_y(q)-1\}$$

ただし，p，q は T の中のセルで，$\pi_y(p)>\pi_y(q)$．

セル p のレベルは，p と根セルとの間の辺の個数で定義される．セル p に対して，関数 Index を次のように定義する．

Index $(p)=\text{def}=0$：p が根セルであるとき

i：p が p の親の i 番目の子供

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B1[2]．（セルの間の線が交差しない）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$)$

$+\text{width}(p\text{の末っ子})$ □

条件B2. (セル同士が重ならない) 木構造図式 (T,π) において，セル p に対して，$\text{area}(p,\pi)=\text{def}=\{(x,y)\mid \pi_x(p)\leqq x\leqq \pi_x(p)-\text{depth}(p)-1,\ \pi_y(p)\leqq y\leqq \pi_y(p)+\text{width}(p)-1\}$ としたとき，すべての p,q $(p\neq q)$ について

$\text{area}(p,\pi)\cap\text{area}(q,\pi)=\phi.$ □

条件B3[2]. 木構造図式 (T,π) において，T_1 と T_2 が同型な部分木構造ならば，T_1 と T_2 は平行移動に関して同じ形で配置されなければならない. □

次の条件B4は木の美的条件[2]としては一般的であるが，木構造図式として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B4. 木構造図式 (T,π) において，レベルが i であるすべてのセル p に対して，$\pi_x(p)=i$. □

次の条件B5は2進木に関してすでに導入されていたものである[2]. この条件は **PAD**, **Hichart** など一部の図式の清書の際に有効である.

条件B5. 木構造図式 (T,π) において，セル p が $k\geqq 1$ 個の子供 $q_1,\cdots,q_k$ $(\text{Index}(q_i)=i, 1\leqq i\leqq k)$ をもつならば $\pi_y(p)$ は次を満たす：

$g=\pi_y(q_k)+\text{width}(q_k)-\pi_y(q_1)-1$ とするとき

$\pi_y(p)+[\text{width}(p)/2]=\pi_y(q_1)+[g/2]$ □

プログラム図式の表示においては条件B5は必ずしも適当ではない. すなわち，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. そこで我々は新たに，条件B5のかわりに，次の条件も導入する.

条件B5(j). 木構造図式 (T,π) において，セル p が k 個の子供 $q_1,\cdots,q_k$ $(\text{Index}(q_i)=i, 1\leqq i\leqq k)$ 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：

$\text{Intersect}(T,\pi)=\text{def}=\max\{\pi_y(p)-\pi_y(q)+1;\ T_1$ 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 はそれぞれ T_1 と T_2 の任意のセルとする$\}$

次の条件は部分木構造同士の横方向の重なりに関するものである. 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B¥(k). $k\geqq 0$ に対して，配置 π が Intersect $(T,\pi)\leqq k$ を満たす. □

以上の条件を組み合わせて木構造図式に対する，"美的条件" を導入する. はじめの2つの条件は木の美的条件[2),11)]のうちB5に相当する部分を変更せずに対象を木構造図式に拡張したものである. プログラム図式に対しての利用に当たっては **PAD** と **Hichart** の図式清書に用いることが可能である.

記法 1. 上記の条件を組み合わせて木構造図式の美的条件 E_0, E_* を以下のように定める：

$E_0=\text{B1}\wedge\cdots\wedge\text{B5}\wedge\text{B¥}(0)$

$E_*=\text{B1}\wedge\cdots\wedge\text{B5}$ □

次の条件は一般の木構造図式の表示一般に向くよう上の条件のうち，B5をB5(j) に変更したものである.

記法 2. 美的条件 E_0, E_*, においてB5のかわりにB5(j) をおいて新たに美的条件を考え，それらをそれぞれ $E_0(j)$, $E_*(j)$ とかく. すなわち

$E_0(j)=\text{B1}\wedge\text{B2}\wedge\cdots\wedge\text{B5}(j)\wedge\text{B¥}(0)$

$E_*(j)=\text{B1}\wedge\text{B2}\wedge\cdots\wedge\text{B5}(j)$ □

この条件はプログラム図式では扇型に記述される **PAD**, **Hichart** などへの適用だけでなく，扇型に記述されない図式でも各セルの階層レベルと x 座標の値が一致する **SPD**, **TSF** など多くのプログラム図式の描画に適用可能である. さらに上の条件を満たさない **HCP** などの描画の一部にも適用可能である.

図8 に美的条件 E_* を適用した **Hichart** プログラム図式の例を示す.

4. 描画方法

この章では前章で得られた美的条件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. はじめに木の描画問題から直接得られる事柄を述べる.

木の描画問題においては，ある頂点の子供頂点同士の間隔を一定とするとその描画問題が NP 困難であることが知られている[5]. したがって木構造についても，次の定理1が成り立つ. すなわち

条件B#. 木構造図式 (T,π) においてセル p が $k\geqq 3$ 個の子供 $q_1,\cdots,q_k$ $(\text{Index}(q_i)=i, 1\leqq i\leqq k)$ をもつならば，

$\pi_y(q_{j+2})-\pi_y(q_{j+1})=\pi_y(q_{j+1})-\pi_y(q_j)$

$(1\leqq j\leqq k-2)$ □

とし，$\text{E\#}=\text{def}=\text{B1}\wedge\text{B2}\wedge\cdots\wedge\text{B5}\wedge\text{B\#}$ としたとき

定理 1. 木構造図式 (T,π) において配置 π 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

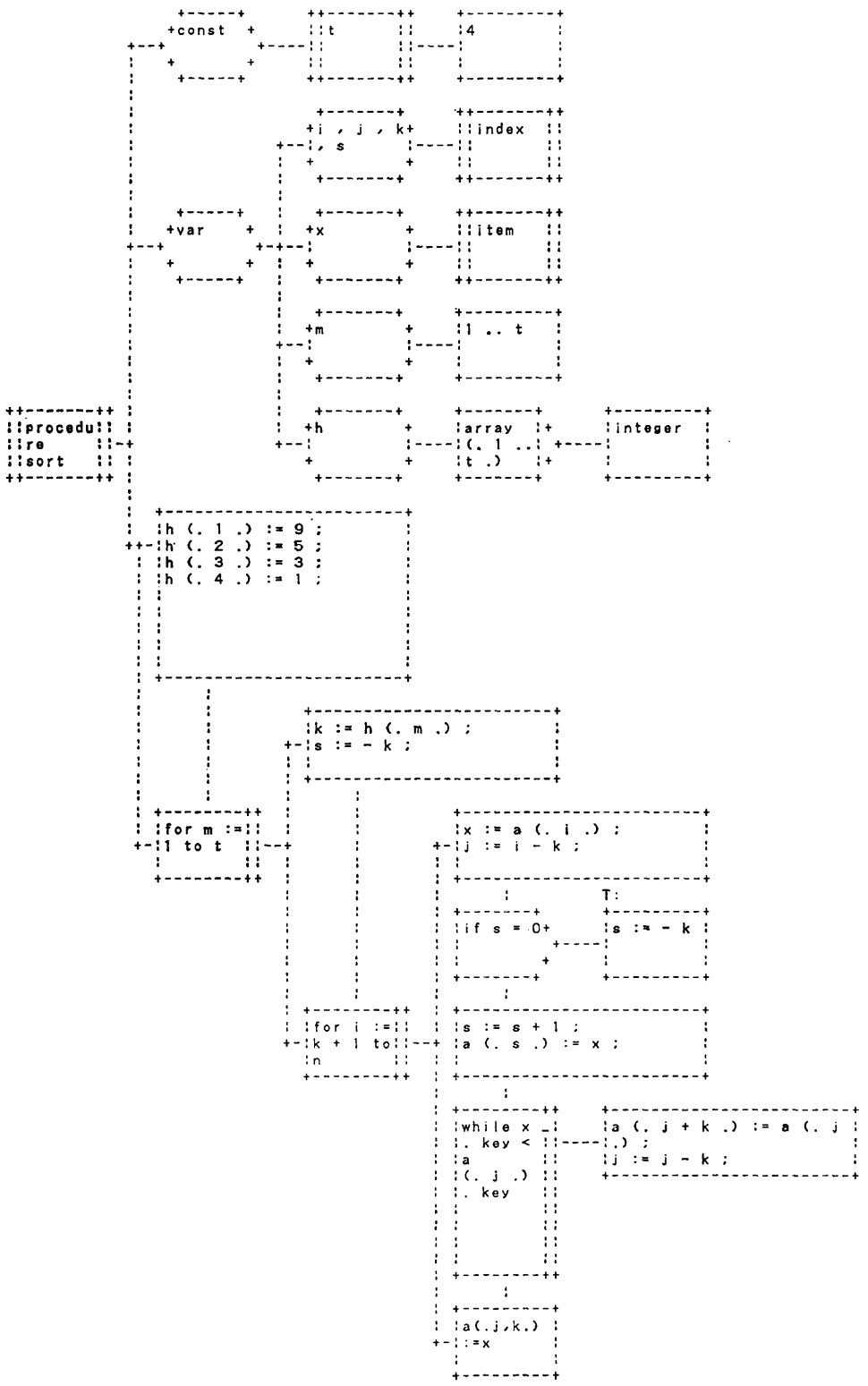

図 8 美的条件 E_* を適用した Hichart プログラム図式の例 (shell sort を行う手続きの例)
Fig. 8 An example of Hichart program diagram applying to condition E_*.

は NP 困難である. □

また木の場合次のような条件を付け加えた場合の計算量が既にわかっている. したがって木構造図式についても同様の結果が得られる. 次の条件の原型はアルゴリズムの作成を容易にするために考えられたものである[11].

条件B$. 木構造図式 (T, π) において, Index (r_i) $+1=$Index (r_j) であるようなセル r_i, r_j を親セルとするすべての部分木構造 T_i, T_j に対して,

π_y (T_iの一番下のセル)
$+$width (T_i の一番下のセル)の最大値$<\pi_y(r_j)$
$+[$width $(r_j)/2]$

かつ

$\pi_y(r_i)+[$width $(r_i)/2]<\pi_y$ (T_j の一番上のセル)

□

記法 3. 上記の条件を組み合わせてレイアウトの美的条件 $E_{*\$}$ を以下のように定める.

$E_{*\$}=def=$B1$\wedge\cdots\wedge$B5$\wedge$B$ □

定理 2[11]. 木構造 T に関してすべてのセル v が width $(v)=$depth $(v)=1$ を満たすとする. このとき上で定められた美的条件 $E_{*\$}$ を満たす最小幅配置を出力する, 計算時間が $O(n^4)$ となるようなアルゴリズムが存在する. □

次に, セルの大きさが一定でない木構造に対する条件 $E_0(j)$, $E_*(j)$, $(j\geqq 0)$ を満たす描画手法について考える.

木構造Tを美的条件 $E_0(j)$, $(j\geqq 0)$ 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, π_{init})

[入力] $T=(V, E, r,$ width, depth$)$, j

[出力] π_{init}: $E_0(j)$ を満たすTの最小幅配置

[方法] ポストオーダの順に条件B1, B2, B3, B4, B5(j) を満たすよう各セルを配置する.

補題 1. 上のアルゴリズム Init により得られる π_{init} は $E_0(j)$ を満たす最小幅の配置である. また, Init の時間計算量は明らかに $O(n)$ である. □

証明. B1, B2, B4, B5(j) は明らかに満たされる. ポストオーダで配置されるため, B3も満たされている. 最小幅についても明らかである. □

美的条件 $E_0(j)$, $(j\geqq 0)$ 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, さらに条件 $E_*(j)$, $(j\geqq 0)$ を満たすように再配置することを配置美化と呼ぶことにする. 以下で導入する配置美化のためのアルゴリズム Layout は上の Init と次に示す SubOpt を組み合わせる. なお, セルの大きさが一定であるような木構造を対象とした場合の条件 E_0, E_* については別のアルゴリズム[11]が知られている.

セル q を根とする部分木構造 T_q の美的配置は以下の方法で決まる.

アルゴリズム SubOpt: 部分木構造の上方移動

[形式] SubOpt $(T, \pi, j, p, \pi_{\text{in}}, \pi_{\text{out}},$ Mark$)$

[入力] $T=(V, E, r,$ width, depth$)$: 木構造
π : Tの配置
p : Tのセル, T_p はpを根とするTの部分木構造とする
π_{in}: T_p の配置
j : $E_*(j)$ の j

[出力] π_{out} : T_p の配置, 再配置
Mark: Tのセル上のマーク

[方法] (概略)

```
begin
  (* This algorithm runs in postorder and *)
  (* determines locations of the cells in  *)
  (* postorder.                            *)
  Initially πout := πin;
  Let p1, ..., pk be the sons of p;
  for i=1 to k do
    begin
      if pi is unmarked leaf then
        begin
          move upward pi satisfying
          B1, B2, B3, B4 and B5(j) with respect
          to p;
          Mark "moved" pi;
          Update πout; πin := πout
        end;
      if pi is unmarked interior cell then
        begin
          SubOpt (Tpi, πin, j, pi, πin, πout, Mark);
          Mark pi "moved";
          Update πout; πin := πout
        end
    end;
  Evaluate the maximum distance d of the upward
  movement of p satisfying B1, B2, B3, B4 and
```

B5(j) with respect to T_p and p, where p is the set of cells located above T_p;
Shift upward all cells in T_p by d;
Mark p "moved";
Update π_{out}; $\pi_{in} := \pi_{out}$
end.

補題 2. 上のアルゴリズム SubOpt により得られる T_p の配置 π_{out} は $E_*(j)$ を満たす配置であり，π_{out} の幅は配置 π_{in} より狭いか等しい. □

証明. B1, B2, B4, B5(j) は明らかに満たされる．ポストオーダで配置されるため，B3 も満たされている．幅についても明らかである. □

以上により木構造Tの美的配置は次のアルゴリズム Layout で得られる.

アルゴリズム Layout：木構造の配置最適化
[形式] Layout (T, j, π, width_T_π)
[入力] $T=(V, E, r$, width, depth): 木構造
j : $E_*(j)$ の j
[出力] π : $E_*(j)$ を満たすTの美的配置
width_T_π : (T, π) の幅
[手法]
begin
Init (T, j, π_{init});
Initially, all cells in T are unmarked;
SubOpt ($T, \pi_{init}, j, r, \pi_{init}, \pi$, Mark)
end.

命題 1. 上のアルゴリズム LayOut は $E_*(j)$ を満たす π_{init} より狭いか等しい幅を持つ配置πを出力する. □

証明. アルゴリズムの正当性は次のように示される．B1, B2, B4, B5(j) は明らかである．B3 についてはアルゴリズムがポストオーダに動くことから明らかである. □

このレイアウト問題の計算量は $O(n^4)$ である[11]．条件 E_* については木に対するアルゴリズムと同様のアルゴリズムが考えられる.

5. むすび

我々が作成したプログラム図式生成系では，プログラム図式が階層的なリスト構造として表現されたことにより，描画問題のようなグラフ理論的問題が容易に取り扱えることが2章で示された．我々が導入したような生成系の構造は構造的プログラム図式一般をリスト表現する際に有効である．その対象には木構造型プログラム図式だけでなくモジュール型の **NSD** なども含まれる.

従来のグラフ（n進木）に対する美的条件はセルの大きさが一定の木構造図式に対してしか適用可能でなかったが，われわれはセルの大きさが一定でない木構造図式に対する美的条件を3章で定式化することができた．さらに親セルの位置についても新たな条件を導入した．その結果得られた美的条件は左側中央から扇型に記述される **Hichart, PAD** だけでなく左上から右下へ記述される形式の **SPD, YAC II, TSF** などにも適用可能である．ただし，等しい階層レベルを持つセルを左から右側へ並べる場合のある **HCP** には部分的にだけ適用可能で，モジュール構造をした **NSD, Chapin** チャートなどには適用不可能である.

4章では3章で導入された美的条件に対応する描画手法について考察した．その結果3章の美的条件が木の美的条件と比較され，それらの一部が美しさの程度が上がる順に $O(n)$, $O(n^4)$, NP 困難な時間計算量の上限を持つことが示された．実際の利用においては $O(n)$ の計算量を持つ条件はプログラムの開発途中での利用が適当で，$O(n^4)$ の計算量を持つ条件はプログラムの完成時の文書化に適当である．最後の NP 困難な条件は非常に美的と思われるが，あまり実用的ではないことが示された．さらに本論文の主たる結果として我々が新たに導入した美的条件に対応するアルゴリズム LayOut を導入した．なお，アルゴリズム LayOut が美的条件を満たす図式を出力することは示されたがその図式が最小幅かどうかという問題は未解決である.

我々はプログラム図式の描画モジュールに E_0, E_*, $E_0(j)$, $E_*(j)$ と対応するアルゴリズムを組み込んでプログラム教育・開発に利用中である.

今後の課題としては，3章で述べた美的条件のバリエーションを増やすことが必要である．さらに4章のアルゴリズム LayOut が最小幅の配置を出力するかどうか判定し，もし最小幅の配置を出力しないのであれば改良して計算量を評価することが必要である．また，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，4章の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統合的に動かす必要がある．したがって，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を会話的に動かす必要があるかど

うかを検討すると共に会話的に動かす必要がある場合はその計算量を評価する必要がある.

謝辞 この生成系は長期間にわたって開発された.処理系の開発を行ってくださった高橋美智子氏(元東海大学),長田芳一氏(東海大学),小倉耕一氏(東海大学,現,北海道東海大学),竹内一浩氏(東海大学,現,日本電気(株)),近藤理彦氏(早稲田大学,現,日本アイビーエム(株)),番場浩氏(東海大学,現,北海道東海大学),今井一昭氏(東海大学,現,(株)CSK),海野浩氏(東京電機大学,現,大原簿記学校)に感謝します.さらにこの処理系開発にあたってアドバイスをいただいた二木厚吉氏(電子技術総合研究所),安斉公士氏(関東学園大学),西野哲朗氏(東京電機大学),杉田公生氏(東海大学)にも感謝します.

なお,筆者らのうち,宮寺と夜久は東京電機大学総合研究所学術研究援助および(株)ガロアによる委託研究費を受けている.

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,電子通信学会オートマトンと言語研究会資料,AL 78-47, pp. 61-66 (1978).
2) Wetherell, C. and Shannon, A.: Tidy Drawing of Trees, *IEEE Trans.*, Vol. SE-5, pp. 514-520 (1979).
3) Reingold, E. M. and Tilford, J.S.: Tidy Drawing of Trees, *IEEE Trans.*, 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,第 27 回情報処理学会全国大会論文集, pp. 549-550 (1983).
5) Supowit, K. J. and Reingold, E. M.: The Complexity of Drawing Trees Nicely, *Acta Inf.*, 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, No. 5, 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, p. 17 (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, No. 10, pp. 1019-1026 (1986).
10) Miyadera, Y, Imai, K., Kuwabara, H., Unno, H. and Yaku, T.: ETA 87—An Extension of a Hichart Flowchart Processing System, 第 35 回情報処理学会全国大会論文集, pp. 1201-1202 (1987).
11) Tsuchida, K.: The Complexity of Tidy Drawings of Trees, *Topology and Computer Science* (Suzuki, S. ed.), pp. 487-520, Kinokuniya, Tokyo (1987).
12) Yaku, T., Futatsugi, K., Adachi, A. and Moriya, E.: HICHART—A Hierarchical Flowchart Description Language—, *Proc. IEEE COMPSAC*, Vol. 11, pp. 157-163 (1987).
13) Sugai, M., Uchiyama, A., Hagiwara, N., Rekimoto, J., Koyamada, M. and Shigou, O.: A Coherent Software Development System with Interface Dictionaries, *ibid*, Vol. 11, pp. 428-432 (1987).
14) 原田賢一編:構造エディタ, p. 198, 共立出版,東京 (1987).
15) 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, 構造エディタ(原田賢一編),第Ⅲ部 6章,共立出版,東京 (1987).
16) 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).
17) 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).
18) 郷 信義,岸本美紀,小倉耕一,土田賢省,夜久竹夫:木構造図式の美的描画について,京都大学数理解析研究所講究録, No. 695, pp. 55-64 (1989).
19) Nishino, T.: Attribute Graph Grammars with Applications to Hichart Program Chart Editors, *Advances in Software Science and Technology*, Vol. 1, pp. 426-433 (1989).

(平成元年 10 月 20 日受付)
(平成 2 年 7 月 10 日採録)

郷 信義(正会員)

1958 年生. 1981 年東海大学理学部情報数理学科卒業. 同年同大学研究生. 1982 年同大学電子計算センター技術職員として勤務,現在に至る. CAI コースウェアの開発等に従事. 日本ソフトウェア科学会, CAI 学会各会員.

岸本 美紀

1961年生．1984年東海大学理学部情報数理学科卒業．現在，富士通(株)に勤務．

宮寺 庸造（正会員）

昭和37年生．昭和59年東京電機大学理工学部数理学科卒業．昭和61年同大学院修士課程理工学研究科数理学専攻修了．同年より同大学情報科学科助手として勤務，現在に至る．以来，言語処理系に関する研究開発およびソフトウェア開発環境の研究開発に従事．プログラム言語の抽象化，プログラム検証に興味を持つ．日本ソフトウェア科学会，CAI 学会各会員．

岡田 直之（正会員）

昭和36年生．昭和59年東海大学理学部情報数理学科卒業．同年日本電気(株)入社．現在，基本ソフトウェア開発本部勤務．コンパイラの開発に従事．

土田 賢省（正会員）

昭和33年生．昭和57年早稲田大学理工学部数学科卒業．昭和59年同大学院理工学研究科博士前期課程修了．同年日本電気(株)入社．同年から平成2年3月まで，ソフトウェア生産技術開発本部に勤務．その間，プログラム自動生成，ソフトウェア生産への AI 応用の研究開発に従事．平成2年4月より神奈川大学工学部工業経営学科助手．

夜久 竹夫（正会員）

1947年生．1970年自由学園大学部理学科卒業．1972年早稲田大学大学院数学専攻修士課程修了．1976年同大学院博士課程単位取得退学．理学博士．1976年東海大学理学部情報数理学科専任講師．1979年同助教授．1985年東京電機大学理工学部数理学科助教授．1986年同情報科学科助教授，現在に至る．セルオートマトン，図形型プログラム言語，グラフアルゴリズム等の研究に従事．日本数学会，電子情報通信学会，ACM 各会員．