

# 図形オブジェクトの遠隔度に基づく階層集合の可視化モデル

酒井 恵光<sup>†</sup> 山口 和紀<sup>†</sup> 川合 慧<sup>†</sup>

本論文では、抽象オブジェクトの集合、集合の集合、集合の集合の集合などから階層的に構成される階層集合の可視化モデルを提案する。このモデルは、図形オブジェクトの縦一列の整列のような強い制約による表現ではなく、図形オブジェクトの「集まり」で集合を表現することで、より柔軟で適切な可視化を目的とするものである。図形オブジェクトの集まりは、「図形オブジェクトが互いに近い」という二項関係に基づく同値類として定式化した。「互いに近い」ことは、図形オブジェクト間のユークリッド距離を一般化した尺度である遠隔度によって定義した。同値類を再帰的に構成してできる「集まりの集まり」で「集合の集合」を表現することにより、階層集合の可視化のための制約を構成する。本論文では、抽象オブジェクトを図形オブジェクトに、抽象オブジェクト間の関係を図形オブジェクト間の幾何学的な制約に、それぞれ変換し、この制約を解いて図を生成する可視化の枠組みを用いている。この枠組みのもとで図形オブジェクトの集まりを作るための制約は図形オブジェクト間のその他の制約と組み合わせることが可能である。図形オブジェクトの同値関係を制約として扱う場合、その制約解消には、組合せの複雑さの問題があるため、最適化問題による近似を考案した。以上のモデルに基づき、可視化のプロトタイプシステムを実装し、モデルの有効性を示した。

## Visualization Model of Hierarchical Sets Based on Perception Distance ( $p$ -Distance) of Graphical Objects

EKO SAKAI,<sup>†</sup> KAZUNORI YAMAGUCHI<sup>†</sup> and SATORU KAWAI<sup>†</sup>

In this paper, we propose a visualization model for hierarchical sets in which an element may be a hierarchical set consisting of other elements and sets. In the model, a set of abstract objects is represented with a "group" of graphical objects closely placed in the picture. The "grouping" is done by the use of as weak constraints as possible, in contrast to in other popular and conventional methods which tend to use unnecessarily strong constraints such as linear alignment. The weakness of the constraints enables us to incorporate other constraints more freely into the visualization scheme, thus yielding more flexible visualization. The proposed model uses the framework of visualization in which abstract objects and relations among them are translated into graphical objects and geometrical constraints. A group of graphical objects is defined, in this model, as an equivalence class of the binary relation that "graphical objects  $A$  and  $B$  are closely placed." The binary relation is defined by perception distance ( $p$ -distance), which is a generalization of Euclidean distance. Graphical groups constitute a hierarchical set by applying the interpretation recursively with increasing magnitude of  $p$ -distance threshold. We develop a method to solve the constraint system in the model with a reasonable approximation. Various solutions can quickly be calculated by the use of non-linear programming. We show the effectiveness of the model with various examples produced by the prototype system.

### 1. はじめに

情報化社会の進歩とともに、コンピュータで扱うデータ量は莫大になってきている。それとともに、莫大な情報を効率良く、歪めずに人間に伝達する可視化の重要性も増してきている。そのような伝達手段の研究の一環として、本研究では、階層化された集合の可視化

のモデルを提案する。階層集合は、集合の要素が再び集合であることが許されている集合であり、クラスタ分析<sup>1)</sup>などで見られる。厳密な定義は3章で与える。

#### 1.1 情報の伝達手段としての可視化

何らかの構造を持つもの(システム)に関する説明を行う場合、自然言語や形式言語を用いたテキストベースの記述は、論理的厳密性の高い記述を行いやすい反面、記述が抽象的になり、一見ただけでは全体像を把握しにくくなりやすいという欠点がある。そこで、そのシステムの構造を表す図を用いて、説明の際

<sup>†</sup> 東京大学大学院総合文化研究科  
Graduate School of Arts and Sciences, The University of Tokyo

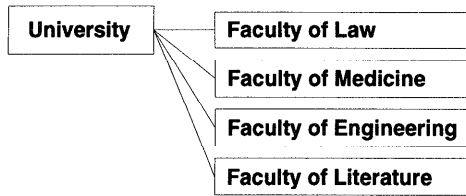


図1 可視化の例：説明図

Fig. 1 Visualization example: explanatory diagram.

の理解の助けとするということが多く行われる。

たとえば、ある大学が、法、医、工、文の各学部から構成されていることを説明する際、組織の構造を表す図1のような図が用いられる。

このような説明のための手段として使用される図は、説明しようとする内容を適切に表現している必要がある。なんらかの内容（ここではシステムの構造）を表す抽象的な記述を入力とし、それをもとにその内容を表す図を出力する過程を、ここでは可視化（visualization）とよぶ。そして、可視化の入力となる抽象的な記述を、ここでは抽象データとよぶ。

### 1.2 図における関係の表現

抽象データ中に現れるオブジェクト（抽象オブジェクト）間のある関係は、通常、図上においてそれらの抽象オブジェクトに対応する図形オブジェクトの間関係（規則性のある配置、線による接続など）によって表現される。図上で関係付けられる図形オブジェクトは1つのまとまりを構成し、そのまとまりは抽象データにおいて関係付けられる抽象オブジェクトのまとまりに対応する（図2）。

### 1.3 図形オブジェクトの集まり

本論文では、抽象データ中の関係を図上で表現するため、図形オブジェクトの近接性による集まりを用いる方法について論ずる。これは、ある関係によって結ばれる抽象オブジェクトがあるときに、その抽象オブジェクトに対応する図形オブジェクトを図上で近くに配置することで抽象データ中のその関係を表す方法である。たとえば、図3のように図形オブジェクトを近接させることで、抽象オブジェクトの学問領域の分類が表される。

既存の可視化システムでは、抽象データ中の関係を図上で表すために、図形オブジェクトを縦一列や横一列に配置するような、図形オブジェクトの整列を用いることが多く行われている。図形オブジェクトの集まりによって表される関係は、基本的には図形オブジェクトの整列を用いても表すことができる。しかし、図形オブジェクトが整列していることから生じる「整理された」という印象を与えたくない場合には、図形オ

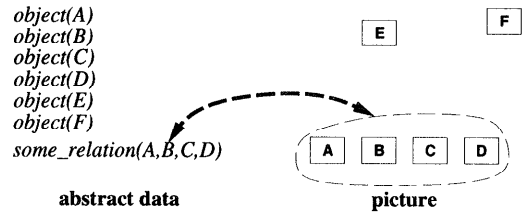


図2 抽象データでの関係と図上での関係

Fig. 2 Object groups in the abstract data and the picture.

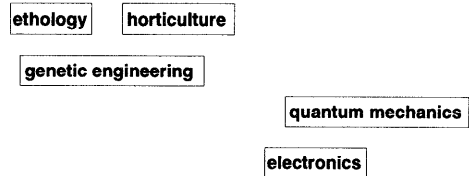


図3 近接による集まりを用いた図の例

Fig. 3 Representation of abstract relation by graphical group.

ブジェクトが整列していない集まりによる関係の表現が必要となる。また、図形オブジェクトを整列させるには図形オブジェクトの配置により強い制約が加えられるため、抽象データの構造によっては可視化の際に整列のための制約が対立して適切な図が描けなくなる可能性がある。すなわち、整列による配置は不必要で過剰な制約を導入していることになる。これを避けるには、図形オブジェクトの配置に関する制約がより少ない配置構造によって関係を表すことが要求される。図形オブジェクトの集まりによる関係の表現はこのような要求に応えるために必要である。

### 1.4 階層集合の可視化

通常、図上の図形オブジェクトの集まりによって表されるのは、一般には抽象データ中の抽象オブジェクトの何らかの集合である。たとえば、図3の2つの集まりは、抽象データ中の {ethology, genetic engineering, horticulture} および {electronics, quantum mechanics} という2つの集合を表している。このように、図形オブジェクトの集まりによる関係の表現は、一般に集合を可視化する方法となる。

抽象オブジェクトを要素とする集合がいくつかある場合、さらにそれらの集合を要素として持つような集合が定義される場合がある。これは図の上では集まりの集まりとして表現できる。たとえば、図3の例にさらにいくつかの学問領域を加えた例を考えると図4のように集まりの集まりが作られる。ここで、集まりの集まりは図5のような階層構造を反映して作られて

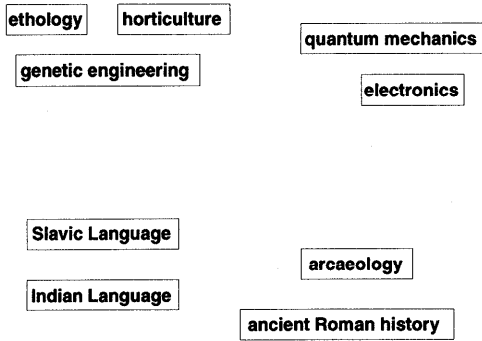


図4 階層集合を表す図の例

Fig. 4 Example of a picture representing a hierarchical set.

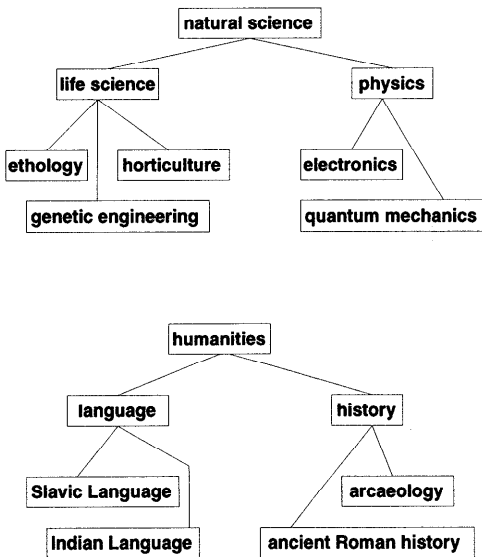


図5 階層構造の例

Fig. 5 Example of a hierarchical structure.

いる。

このように、図形オブジェクトの集まりによって集合を表す方法は、階層構造を持つ集合を表す場合に対して自然に拡張することができる。本研究では、階層的な構造を持つ集合（階層集合）を図形オブジェクトの集まりによって表す一般的な方法を与える。

階層構造を持つ集合は、さまざまな自然科学的、工学的、社会科学のシステムにおいて現れる。階層集合の可視化技術の応用可能性としては、このような構造を持つさまざまなシステムの説明のためのプレゼンテーションツールでの利用がまず考えられる。未整理の知識を関連付けてゆくブレインストーミングでは、関連付けられた知識に対応する図形オブジェクトの集まりを用いた図が必要とされる。データマイニングのシステムにおいても、関連付けられたデータの表示方

法は重要であり、この分野での応用も期待される。階層集合の可視化技術は、このような多様な分野における基盤技術としての応用可能性を持つ。

### 1.5 距離の一般化

図形オブジェクトの集まりによって抽象データ中の関係を表すことの妥当性の根拠の1つとして、心理学の分野でゲシュタルト要因（Factor of Gestalt）<sup>2)</sup>として知られる経験的事実がある。これは、図上で互いに近くにあるものや、形の似ているもの、大きさの似ているもの、色の似ているもの、同じ向きのものなどは結び付きやすい、というものである。

図上での図形オブジェクトの集まりは、図形オブジェクトの図上での近接によって構成される。図形オブジェクトが空間的に近い、ということは図形オブジェクトの間に定義される距離の値が十分小さい、という言葉で定義される。

ゲシュタルト要因でいう、近くにあるものは結び付き、ということ、空間的距離が十分小さいものは結び付き、と言い換えることができる。ここで、たとえば図形オブジェクトの大きさについて、大きさの違いを表す尺度（面積、幅、高さなどについての差など）を考え、これを図形オブジェクト間に定義される一種の距離と見なせば、ゲシュタルト要因における図形オブジェクトの大きさに関する言及は、「図形オブジェクトの間の大きさによって定義される距離が十分小さいものは結び付き」と言い換えることができる。同様に、色や向きなどの、図形オブジェクトの他の属性についても、距離に相当する尺度を適切に定義することができるならば、「図形オブジェクトの間の距離が十分近ければ、それらは結び付き」と一般化することができる。したがって、図形オブジェクトの空間的な集まりによって抽象オブジェクトの間の関係を表す方法は、図形オブジェクトの他のさまざまな属性に関する「集まり」による関係の表現に応用しうる。

### 1.6 本論文の構成

本論文では、2章で研究の背景について、3章で関連研究について述べ、4章、5章で抽象データの構造と「集まり」の図構造の定義を与える。6章ではこれらの定義に基づいて階層集合の可視化のモデルを与える。7章で実行例を示し、8章では実行例をふまえ、モデルの有効性を示す。9章は結論である。

## 2. 研究の背景

ここでは、制約に基づく可視化の枠組みと、可視化における集まりによる関係の表現の必要性について述べる。

## 2.1 制約に基づく可視化

本研究では、可視化の枠組みとして、Kamadaらが提案した可視化写像 (visual mapping)<sup>3)</sup>に基づく枠組みを用いる。この枠組みでは、まず抽象データ中の抽象オブジェクト間のある関係を図上の図形オブジェクトの間のどのような関係で表すかを、可視化規則 (mapping rule) として用意しておく。これはたとえば、抽象データ中の関係「 $R_1(X, Y)$ 」は  $X, Y$  に対応する図形オブジェクトを「横に並べる」というような規則である。そして、抽象データが入力として与えられると、その抽象データに可視化規則を適用し、図形オブジェクトおよび図形オブジェクトの間の関係を生成する。これを図構造データとよぶ。たとえば、可視化規則と抽象データが次のようになっているとする。

可視化規則：

$$\begin{aligned} object(X) &\rightarrow labelled\_box(X) \\ R_1(X, Y) &\rightarrow horizontal\_listing(X, Y) \end{aligned}$$

抽象データ：

$$\begin{aligned} obj\_A &= object("A") \\ obj\_B &= object("B") \\ R_1(obj\_A, obj\_B) \end{aligned}$$

このとき、図構造データは次のようになる。

$$\begin{aligned} obj\_A &= labelled\_box("A") \\ obj\_B &= labelled\_box("B") \\ horizontal\_listing(obj\_A, obj\_B) \end{aligned}$$

次に、この図構造データに現れる図形オブジェクト間の関係を図形オブジェクトの属性値間の制約条件に変換し (変換規則はあらかじめ与えておく)、この制約条件を解いて、これらの関係をすべて満たすような図形オブジェクトの属性値の組を求める。こうして得られた属性値に基づき、図形オブジェクトを描くことで、元となった抽象データの内容を表す図を描く。こうして、抽象データの内容を可視化した図が得られる。

この枠組みでは、抽象データ間の関係に従った配置の調整が、制約条件を解く部分に限定されているため、可視化写像では、関係間の相互作用を扱わなくてよい。

このような制約に基づく可視化の枠組みの有効性はすでに示されているが<sup>3)~5)</sup>、この枠組みの中で、抽象データ中のどのような関係を図上のどのような関係で表すのが適切かに関してはそれほど深く研究されてはならず、比較的単純で実装も容易な配置関係しか利用されていない。

## 2.2 図形オブジェクトの整列による関係の表現

ここでは、可視化における「集まりによる関係の表現」の必要性について考察する。

既存の可視化システムで抽象データ中の関係を図上で表すために多く用いられる図形オブジェクト間の配置方法として、図形オブジェクトを適当な順に一行に (多くは縦一列あるいは横一列に) 並べる、という方法がある。図形オブジェクトの整列は、図形オブジェクトが近接しているだけの場合に比べて図形オブジェクトの結び付きが強くなる傾向がある。また、図形オブジェクトの整列は図形オブジェクトの集まりを作る場合に比べて、図形オブジェクトの配置される領域の面積が小さくとどまることが多い。しかし、図形オブジェクトの整列による「整理された」という印象を与えたくないような場合には、抽象データ中の関係を図形オブジェクトの整列によって表すのは適切ではない。また、図形オブジェクトの整列によって図形オブジェクトの集まりを代用する場合には、以下のような点が問題になる。

### 配置順序の矛盾

ある一連の処理において、process 1~process 5がこの順に実行され、process 1が data Cと、process 2が data Aと、process 3が data Bと、process 4が data Dと、process 5が data Eとそれぞれ関連付けられ、さらに、data A, data B, data Cがまた別の意味で関連しているとする。処理 process 1~process 5の実行順序に関する関係 *procedure* を図形オブジェクトの縦一列の整列 *vertical\_listing* で表し、プロセスとデータの関連付け *relation<sub>1</sub>* を横一列の整列 *horizontal\_listing* で、また、data A, data B, data Cの間の関係 *relation<sub>2</sub>* を図形オブジェクトの集まり *group* で表すように記述された可視化規則を用いてこのような内容を持つ抽象データを可視化すると図6のような図で表すことができる。

ここで、data A, data B, data Cに関する関係 *relation<sub>2</sub>* の図上での表現を、図形オブジェクトの集まり *group* ではなく縦一列の整列 *verti-*

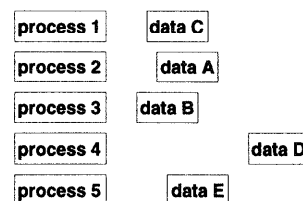


図6 プロセスとデータの関連を表す図の例

Fig. 6 Example of a picture representing relations among processes and data.

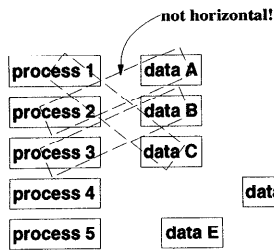


図7 Vertical listing による配置指定でおこる矛盾の例  
Fig. 7 Example of conflict of layout by vertical listing.

*cal\_listing*で表した場合を考える。抽象データ中の  $relation_1(dataA, dataB, dataC)$  という記述が可視化規則の  $relation_1(\dots) \rightarrow vertical\_listing(\dots)$  によって  $vertical\_listing(dataA, dataB, dataC)$  という図形オブジェクト間の関係になり、その結果、いわば「偶然に」抽象データ中に data A, data B, data C の関係がこの順に書かれていた、という理由で data A, data B, data C に対応する図形オブジェクトをこの順に縦一列に並べようとするため適切な図が描けなくなる(図7)。

このような配置の矛盾を避けるには次のような戦略が考えられる。

- (1) 抽象データ中で矛盾が起きないような順序での記述を行う。
- (2) 配置の順序まで指定する *vertical\_listing*ではなく、縦一列ということだけ指定する *vertical*を用いる。
- (3) 抽象データから図構造データを作る際に、配置順序をチェックして矛盾が起らないようにする。

(1)の方法の場合、図の構造を意識して抽象データを記述する必要があり、与えられた抽象的な記述から自動的に図を描くという可視化の趣旨にそわない。図6の例でいうと、data C, data A, data Bの順に図形オブジェクトが並ぶのは、process 1~process 5を表す図形オブジェクトと一緒に平面上に図形オブジェクトを配置する際の都合で起こることであって、図によって表そうとしている抽象データの内容として関係があるわけではない。

(2)の方法では、順序を指定せずに配置する *vertical*という配置指定の内容が問題になる。図上の図形オブジェクトは、たとえ一列に並んでいても、それらが離れて配置されている場合は必ずしも結び付かない。そこで、順序を指定しない配置 *vertical*の実現には、

- 図形オブジェクトの  $x$ 座標が一致する。
- 図形オブジェクトが集まっている。

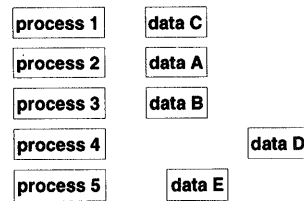


図8 意図しない含意の例  
Fig. 8 Example of unexpected implication.

という条件を満たすような配置を行うことになり、結局、図形オブジェクトの集まりを作る必要が生じる。

(3)の方法では、図形オブジェクトの配置に関する調整の作業を制約解決の過程に押し込めるといって、ここで採用している可視化の枠組みの利点を失うことになる。

#### 存在しない関係の排除

仮に順序なしの配置 *vertical*によって、図7のような矛盾が生じない図が得られたとする(図8)。このとき、process 1~process 5の間に、「この順に実行される」という順序関係がある場合は、同様の配置がされている data C, data A, data B の間に、順序関係があると誤解する可能性がある。しかし、data C, data A, data Bがこの順に縦一列に並んでいるのは、偶然に図構造データ中に現れた関係を満たす配置がそうであったためであり、data C, data A, data Bの間にそのような順序関係は(ここで想定している抽象データの中には)存在しない。そこで誤解を避けるためには図6のように data C, data A, data Bが縦一列に並ばない配置の方が適切である。

Marks<sup>4)</sup>も主張しているように、可視化においては、不適切な含意が生じないようにすべきである。これを避けるには、順序が表されないような配置が可能となる制約を生成することが望ましい。それは、意図しない含意をより確実に避けるために、たとえば  $not\_vertical(\{A, B, C, D, E\})$  というような、意図しない含意が起らないという制約を明示的に付加することで可能となる。

#### 2.3 図形オブジェクトの「集まり」

図形オブジェクトが整列して配置されていなくても、それらが集まって配置されているならば、その集まりによって、各図形オブジェクトに対応する抽象オブジェクト間のまとまりの関係が表されていると見ることができる。すなわち、抽象オブジェクト間のまとまりの関係だけを可視化する場合は、対応する図形オブジェクトを集まった形で配置すればよく、それを実現する弱い制約を用いるべきである。そして、たとえば、図形オブジェクトを整列させることによってより有効な

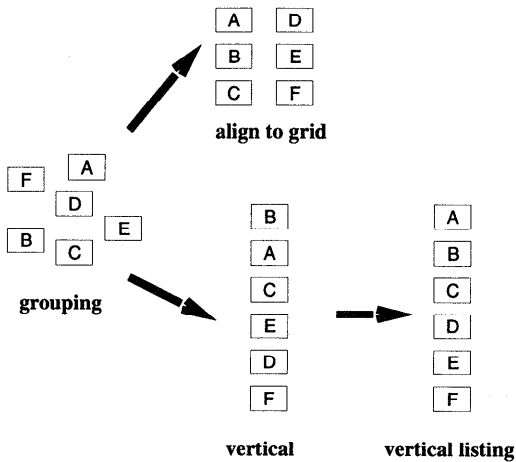


図9 図形オブジェクトの「集まり」とより強い制約

Fig.9 Specialization of group of graphical objects.

図となると判断される場合、たとえば

- (1) 図形オブジェクトどうしの結び付きをより強調したい。
- (2) 描画領域の面積を節約したい。

といった場合は、図形オブジェクトが、縦一列 (vertical), 横一列 (horizontal) などに整列しているという制約をそこに付加し、また、順序に関する関係を明示的に示す必要がある場合にはさらに、特定の順序で並んでいるという制約を付加すればよい (図9)。

本研究では、図形オブジェクトの「集まり」によって、抽象オブジェクトの集合の可視化を行う。「集まり」を扱う目的を整理すると次の2つになる。

- 順序なしの関係に他の関係を加えた関係の可視化が行える。
- 順序なしの関係についての図形オブジェクトの配置の自由度を残した可視化が行える。

ここで考えている図形オブジェクトの「集まり」が元の階層集合構造を抽出することができるものであるために満たすべき条件としては、直観的には次のようなものがあげられる。

- (1) 近くにまとまる。
- (2) 図形オブジェクトどうしは重なり合わない。
- (3) 図形オブジェクトの配置に偏りが無い。
- (4) 同じ「集まり」の構成員でないような図形オブジェクトが混在しない。

これらの性質には、5章で正確な定義を与える。

ここで注意すべき点がある。図10(I)の集まりにおいて、図形オブジェクト A と B は互いに近接していないが同じ集まりに含まれる。これは、間にある図形オブジェクトを通じて、「同じ集まりに含まれる」とい

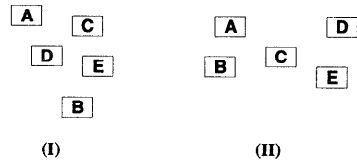


図10 「集まり」の関係の遷移性 (I) と交わる集合 (II)

Fig.10 Transitivity of group relation (I) and joint sets (II).

う関係が推移的に伝わることによる。したがって、抽象オブジェクトの互いに交わる集合を表すためには、図形オブジェクトの「集まり」は適切でない。たとえば、{A, B, C}, {C, D, E} という2つの集合を図形オブジェクトの集まりで表そうとすると、図10(II)のように1つの大きな集まりになり、2つの集合があるようには見えない。

### 2.4 「集まり」とベン図、線接続

集合を図表現する場合、ベン図 (Venn diagram) のように、同じ集合に含まれるオブジェクトを閉曲線で囲んだり、あるいは関係のあるオブジェクトどうしを線で結んだりするという方法がある。このような方法では、図形オブジェクトの配置の近接性によらず、閉曲線で囲んだり、図形オブジェクトを結ぶ線に従って解釈することにより、集合構造を抽出することは可能である。しかし、図形オブジェクトが互いに近くに配置されていない場合、閉曲線や線がこみいった形になり、結び付きを読みとるのは、見る者にとって負担となる。閉曲線や、図形オブジェクトを結ぶ線によって関係を表すような図を作成する際にも、集合に基づいて「集まり」を作れば、より分かりやすい図となる。逆に、閉曲線による囲みや線による接続は、こみいった形にならない場合はある程度距離の離れた図形オブジェクトの結び付きを強調する効果がある。たとえば、図6の process 4 と data D は少し離れてしまっているが、プロセスとデータの関係を水平配置だけでなく、閉曲線による囲みあるいは線による接続をあわせて用いれば、より適切な図となる。また、閉曲線による囲みには、制限はあるものの、交わりのある集合を表せるという利点がある。適切に作られた図形オブジェクトの集まりに対して閉曲線による囲みを適用することで、交わりのある集合の効果的な可視化が可能である。このように、図形オブジェクトの集まりと、閉曲線による囲みや線による接続とは、併用することによって互いの効果を増すものとなる。

### 3. 関連研究

#### 3.1 可視化システム

本研究で用いる抽象データの可視化の基本的な枠組みは、Kamadaら<sup>3)</sup>により提案されたものである。この枠組みに基づき、TRIPと呼ばれる可視化システムが作られている。ネットワーク・ダイアグラムの描画を目的とした、TRIPと類似のモデルがMarks<sup>4)</sup>によって提案されている。制約に基づく可視化を行うユーザーインタフェースシステムとしてはThingLabII<sup>5)</sup>などがある。しかし、これらのシステムで、本研究で扱う階層集合の可視化は行われていない。Rothらによる可視化システムSAGE、Visage<sup>6)</sup>は、多様な情報の可視化を可能にしているが、本論文の可視化モデルが基礎としている制約ベースの方法とは異なるアプローチを採用している。

#### 3.2 グラフ描画

グラフの描画についての研究はTamassiaら<sup>7),8)</sup>による研究をはじめとして、数多く行われている。2.2節末尾の図形オブジェクトの「集まり」の満たすべき性質は、グラフ描画における頂点の配置と類似性があり、Schafferら<sup>9)</sup>はネットワーク図を部分的に縮小、拡大してグラフのノードの階層性を表現する方法を示しているが、階層集合を一般に図形オブジェクトの集まりで表すものではない。Eadesらによる木の描画に関する研究<sup>10)</sup>では、“inclusion”と呼ばれる、箱の包含関係による木の図表現が扱われているが、集合の一般的配置を制約で表すものではない。

また、杉山<sup>11)</sup>は、通常よく見られるようなノード間のリンクによるグラフ描画を網図系、包含関係による表現を領域系とよび、両者を統一的に扱う領域網図複合系の描画について論じているが、集合とグラフに特化しており、「集まり」の制約と他の制約を混在できる枠組みではない。

#### 3.3 制約解消系

制約として表された図の構造から実際の図を作るためには、与えられた制約を満たすような変数の組合せを見つける必要がある。そのような制約解消の手法として、次のようなものがある。

- 代数的解法 (TRIPの制約解消系COOL<sup>3)</sup>, 最適化問題の諸解法<sup>12)</sup>など)
- Local Propagation (ThingLabIIの制約解消系DeltaBlue<sup>13)</sup>など)
- 論理型言語 (CLP( $\mathcal{R}$ )<sup>14)</sup>, Prolog など)

本研究では、SQP (Sequential/Successive Quadratic Programming) とよばれる非線形計画法のアル

ゴリズム<sup>12)</sup>を用いる。

### 4. 抽象データの構造—階層集合モデル

可視化の対象となる抽象データは、抽象オブジェクトの集合と、抽象オブジェクトの間の関係の集合とで表される。本研究では、抽象オブジェクト間の関係のなかで、抽象オブジェクトの「集合」として表されるような関係に注目し、このような関係の可視化の方法を提案する。一般に、抽象オブジェクトを要素として構成される集合は、より高位の集合の構成要素となり、全体的に見ると、階層的な集合構造を構成する。ここでは、このような集合構造を階層集合構造とよび、以下でそのモデルを定義する。

#### 4.1 階層集合

階層集合は、集合、集合の集合、... から階層的に構成される集合である。集合とその要素の間の包含関係を線で結ぶことによって表すと図11のようになる。抽象オブジェクトの集合を  $A$  とする。  $A$  から次のようにして構成される集合を階層集合とよぶ。  $A$  を明示するときは、「 $A$  上の階層集合」とよぶ。

$A$  の部分集合は階層集合である。階層集合を要素とする集合も階層集合である。以上により再帰的に構成された集合だけを階層集合とする。

階層集合  $h$  の要素、要素の要素、... を順にたどっていくと、最終的には1つ1つの抽象オブジェクトにたどり着く。このようにして、階層集合  $h$  からたどり着くことのできる抽象オブジェクト全体の集合を  $h$  の土台とよび、 $Base(h)$  と表す。階層集合  $h$  に対して、 $h$  が  $A$  の部分集合の場合は  $Base(h) = h$ 、それ以外の場合は、 $Base(h) = \bigcup_{x \in h} Base(x)$  となる。  $A$  上の階層集合  $h$  に対して、 $Base(h) \subseteq A$  である。

#### 4.2 階層集合の深さ

階層集合  $h$  の深さ  $Depth(h)$  を次のように定義する。  $h$  が  $A$  の部分集合の場合は  $Depth(h) = 1$ 、それ以外

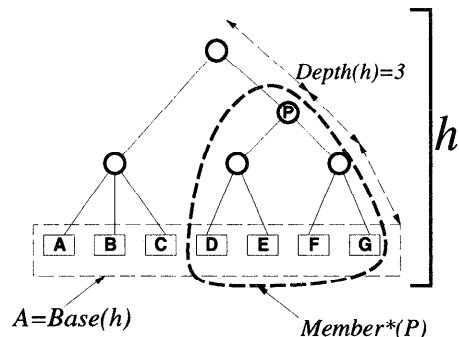


図11 階層集合の構造

Fig. 11 Graph representation of a hierarchical set.

の場合は、 $Depth(h) = Max\{Depth(s) \mid s \in h\} + 1$  とする。

- $Depth(h) = 1$  なる階層集合  $h$  を **0 階層集合** (0-hierarchical-set) とよぶ。
- $Depth(h) = 2$  なる階層集合  $h$  を **単階層集合** (1-hierarchical-set) とよぶ。
- $Depth(h) > 2$  なる階層集合  $h$  を **多階層集合** (n-hierarchical-set) とよぶ。

階層集合の深さの概念は、6.3 節で示す階層集合の可視化モデルにおいて、抽象データとして与えられる階層集合と 5.3 節の方法で構成される図形オブジェクトの同値類による階層集合との対応付けに用いる。

4.3 階層集合の要素の関係の推移的閉包

階層集合  $h$  の要素、要素の要素、... をすべて集めた集合を  $Member^*(h)$  とする。  $Member^*(h)$  は、0 階層集合  $h$  に対して  $Member^*(h) = h$ 、それ以外の  $h$  に対して

$$Member^*(h) = h \cup \bigcup_{h' \in h} Member^*(h')$$

と定義される。

このような  $Member^*(h)$  は 6.3 節で示す可視化のモデルにおける抽象データとして与えられる階層集合と図形オブジェクトから構成される階層集合との対応付けに用いる。

4.4 交わりのない階層集合

$A$  上の階層集合  $h$  に対して「交わりがない」という概念を定義する。交わりのない階層集合と交わりのある階層集合を集合と要素の包含関係を線で表した図の例を図 12 に示す。通常、階層的でない 2 つの集合  $S_1, S_2$  は、 $S_1 \cap S_2 = \emptyset$  であるとき、 $S_1, S_2$  は「交わらない」といわれる。ここでは、階層集合  $h$  のすべての要素となる集合どうしがこれと同様の意味で「交わらない」とき、「 $h$  は交わりのない階層集合である」とする。ただし、集合どうしの交わりの有無については、間接的な包含関係による要素の共有も含めて問題にする。たとえば、

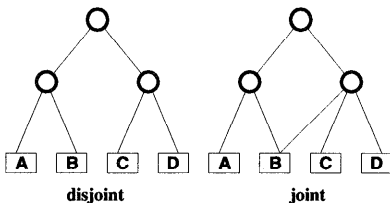


図 12 交わりのない階層集合と交わりのある階層集合

Fig. 12 A disjoint hierarchical set and a joint hierarchical set.

$$\begin{aligned} a_{10} &= \{a_{00}, a_{01}\}, a_{11} = \{a_{02}, a_{03}\}, \\ a_{12} &= \{a_{03}, a_{04}\}, a_{13} = \{a_{05}, a_{06}\}, \\ a_{20} &= \{a_{10}, a_{11}\}, a_{21} = \{a_{12}, a_{13}\}, \\ h &= \{a_{20}, a_{21}\}. \end{aligned}$$

のような階層集合  $h$  を考えると、 $h$  の要素  $a_{20}, a_{21}$  については、 $a_{20} \cap a_{21} = \emptyset$  だが、間接的な包含関係を考慮すると、 $a_{20}, a_{21}$  はオブジェクト  $a_{03}$  を共有している。ここでは、このような階層集合  $h$  も交わりのある階層集合と考える。そこで、交わりのない階層集合は、次のように定義する。

階層集合  $h$  で、 $h$  の任意の相異なる要素  $s, t$  で  $s \notin Member^*(t), t \notin Member^*(s)$  となるものに対して、 $Base(s) \cap Base(t) = \emptyset$  となる  $h$  を交わりのない階層集合とよぶ。また、0 階層集合  $h$  は交わりがない。

6.3 節で示す可視化モデルで、抽象データとして与えられる階層集合は、図形オブジェクトの階層集合と対応付けられる。5.3 節で示すように、この図形オブジェクトの階層集合は交わりのない階層集合である。可視化写像は、抽象データにおける階層集合と図形オブジェクトの階層集合との間の同型写像なので、可視化の対象となる抽象データは交わりのない階層集合を考える。なお、7 章では、交わりのない階層集合だけでなく、交わりのある階層集合に本モデルを適用した例もあわせて示す。

4.5 バランスした階層集合

階層集合が「バランスしている」という概念を定義する。バランスした階層集合とバランスしていない階層集合を図 13 に示す。この図では包含関係を線で表している。0 階層集合  $h$  はバランスしている。それ以外の場合は、 $h$  の任意の要素  $s$  に対して、 $Depth(h) = Depth(s) + 1$  のとき、バランスしているという。

6.3 節で示す可視化モデルで抽象データとして与えられる階層集合と対応付けられる図形オブジェクトの

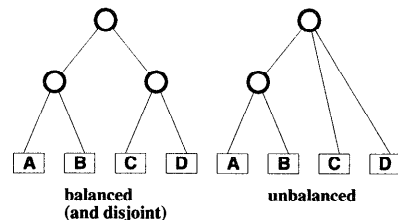


図 13 バランスした階層集合とバランスしていない階層集合  
Fig. 13 A balanced hierarchical set and an unbalanced hierarchical set.



階層集合は、5.3 節で示すように、バランスした階層集合である。したがって、可視化の対象となる抽象データは、バランスした階層集合とする。一般に、バランスしていない階層集合は、ただ1つの要素から構成される集合を適宜挿入して、バランスした階層集合にすることができる。

### 5. 図の構造

階層集合構造を持つ抽象データが与えられたとき、その構造を表すような図を生成することによって、抽象データの可視化を行う。ある一定の方法によって定義される図の構造が、抽象データの構造と同型であるならば、その図によって情報を正しく伝達することができる。ここで使われる図の構造が心理的に認識されやすければ、伝達は効果的なものとなる。

本研究では、図形オブジェクトの近接による集まりを再帰的に構成することによってできる階層集合構造(図14)を図の構造として導入する。たとえば、図4のような図から図形オブジェクトの近接による集まりを再帰的に構成して図5のような構造ができる。この構造が抽象データの階層集合構造と同型であることが図4によって情報が正しく伝達されることの根拠となる。可視化の過程では、「抽象データと同型な構造が取り出せるような図になること」という制約を解くことによって図を生成する。

#### 5.1 遠隔度

図の構造を定めるための基礎となる概念として、遠隔度を定義する。これは、2つの図形オブジェクトの間の距離の概念を一般化したものである。2つの図形オブジェクト  $x, y$  の間に定義される遠隔度  $d(x, y)$  は次の条件を満たす尺度とする。

- 非負値： $d(x, y) \geq 0$
- 対称： $d(x, y) = d(y, x)$
- $d(x, x) = 0$

一般に、距離と呼ばれる尺度は、これらの条件に加え、

- 三角不等式： $d(x, y) + d(y, z) \geq d(x, z)$
- を満たす。本研究で用いる遠隔度においては、三角不等式の成立は必ずしも必要ではない。そのため、たとえば色空間における「集まり」などに遠隔度を適用することも可能である。

以下では、各図形オブジェクトについて定義されている属性  $a_1, a_2, \dots, a_l$  に対し、2つの図形オブジェクト  $p, q$  の属性の関数

$$d(p_{a_1}, p_{a_2}, \dots, p_{a_l}, q_{a_1}, q_{a_2}, \dots, q_{a_l}) \geq 0$$

で遠隔度の規則を満たすものを使う。

たとえば、図形オブジェクトの境界の間の最短距離は、遠隔度の一種である。図形オブジェクト  $A, B$  が円形の図形オブジェクトである場合、 $A, B$  間の最短距離は  $A_{cx}, A_{cy}, B_{cx}, B_{cy}$  をそれぞれ  $A, B$  の中心座標とし、 $A_r, B_r$  をそれぞれ  $A, B$  の半径とすると、

$$d(A, B) = \max\{0, ((A_{cx} - B_{cx})^2 + (A_{cy} - B_{cy})^2)^{\frac{1}{2}} - A_r - B_r\}$$

として定義され、距離の公理を満たす。すなわち、遠隔度の3条件を満たす。

#### 5.2 遠隔度に基づく同値関係

次に、図形オブジェクト間の遠隔度に基づく同値関係  $\sim$  を定める。

ある値  $\Delta$  と遠隔度  $d$  から決まる標準的同値関係  $\sim$  を

$$d(x, y) \leq \Delta \Rightarrow x \sim y$$

で決まる関係  $\sim$  の推移的閉包とする。

#### 5.3 階層遠隔度と同値関係

$i, j \in G_n$  に対して、

$$d_n(i, j) = \text{Min}\{d_{n-1}(i', j') \mid i' \in i, j' \in j\}$$

により  $G_n$  の遠隔度  $d_n$  が  $d$  から再帰的に定義できる。ただし、 $d_n(i, i) = 0$  とする。この遠隔度を階層遠隔度とよぶ。階層遠隔度が遠隔度の条件を満たすことは容易に示される。

すぐ分かるように

$$d_n(i, j) = \text{Min}\{d(i', j') \mid i' \in \text{Base}(i), j' \in \text{Base}(j)\}$$

である。

遠隔度  $d$  と  $\Delta_1$  から標準的同値関係  $\sim_1$  が決まる。 $G_1 = G/\sim_1$  上の階層遠隔度  $d_1$  が  $d$  から決まる。 $d_1$  と  $\Delta_2$  から標準的同値関係  $\sim_2$  が決まる。 $G_2 = G_1/\sim_2$  上の階層遠隔度  $d_2$  が  $d_1$  から決まる。このようにし

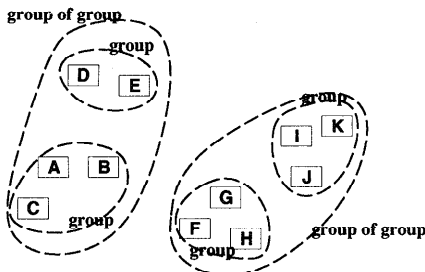


図14 集まりの構成

Fig. 14 Groups of graphical objects.

て同値関係  $\sim_1, \dots, \sim_n$  が定めることができる。このとき、 $G, G_1, G_2, \dots, G_n$  はそれぞれ交わりのないバランスした階層集合となる。

## 6. 階層集合の可視化モデル

ここでは、抽象データとして、

- 抽象オブジェクトの集合  $A$
  - $A$  上の階層集合  $h$
  - 抽象オブジェクトに関するその他の関係  $\ast AC$
- が与えられているとき、これを可視化するモデルを示す。

### 6.1 可視化の忠実性

抽象データを可視化する場合、生成される図によって、抽象データの中の情報が伝達できる必要がある。すなわち、図から、ある一定の手順に従って抽出される構造が、抽象データの構造と同型である必要がある。

ここで扱う階層集合の可視化においては、図形オブジェクトの遠隔度  $d$  が定義されたとき、 $d$  上の階層遠隔度に基づいて 5.3 節の方法で図から構成される階層集合  $G_n$  が、抽象オブジェクトに関する階層集合  $h$  と同型であるとき、この図によって、階層集合構造を正しく伝達することができる。このとき、 $G_n$  を  $h$  を忠実に可視化した図とよぶ。図上の図形オブジェクトの集合を  $G$  とすると、この同型写像  $f$  は、

$$f: A (= \text{Base}(h)) \rightarrow G$$

すなわち、各抽象オブジェクトを図形オブジェクトに対応付ける写像である。

図形オブジェクトの配置から、ある遠隔度  $d$  に基づく階層遠隔度により構成された階層集合は、交わりがなく、バランスした階層集合である。逆に、交わりがなく、バランスした階層集合を忠実に可視化することは、遠隔度が一定の条件を満たせば可能である<sup>☆☆</sup>。

### 6.2 図形オブジェクトの可視性

図形オブジェクトが最もよく集まっている状態は、その図形オブジェクトがすべて同じ場所に置かれている状態である。しかし、このとき、図形オブジェクトどうしが重なり合っ見えなくなるという問題が生じる。そこで、「集める」という制約を課す場合は、可視化における暗黙の仮定である各図形オブジェクトが

「見える」ことを特に条件として課すことが必要となる。この、図形オブジェクトどうしが十分離れている条件

$$\forall i, j \in G, d(i, j) > m$$

を可視性条件とよぶ。可視性条件は、可視化の際の制約条件として、その他の制約とともに与える。

### 6.3 階層集合の可視化モデル

階層集合の可視化は次のように進める。

抽象オブジェクトに対し、それぞれ対応する図形オブジェクトを生成する。図の遠隔度  $d$  および同値類を構成する際の閾値  $\Delta_1, \Delta_2, \dots$  を決める。

図形オブジェクトの集合を  $G$  とし、抽象オブジェクトと図形オブジェクトとの対応付けを

$$f: A \rightarrow G$$

とする。  $f$  は  $A$  上の階層集合  $h$  から  $G$  上の階層集合への写像に自然に拡張される。これを  $\bar{f}$  とする<sup>☆☆</sup>。

$A$  中に現れる集合に対応する図形オブジェクトの階層集合は、 $\text{Member}^*(\bar{f}(A))$  の要素  $s$  である。これが「集まって」配置されるための制約条件を  $\text{group}(s)$  と表す。これは  $s$  が 5.3 節で定義した  $\sim_{\text{Depth}(s)-1}$  の同値類になるという条件である。

抽象オブジェクトの、階層集合  $h$  の構造に含まれないさまざまな関係  $AC$  については、図形オブジェクトの配置順序に関する制約や、縦一列や横一列の整理といった、図形オブジェクトに関する制約  $GC$  に変換する規則を別途与えておくものとする。

以上を用いて抽象データ  $(A, h, AC)$  の可視化を

- $ao \in A (= \text{Base}(h)) \mapsto go (= f(ao))$
- $as \in h - \text{Base}(h) \mapsto \text{group}(f(as))$
- $ac \in AC \mapsto GC$

という可視化写像によって、図形オブジェクトおよび、図形オブジェクトに関する制約に変換する。以上の制約に、6.2 節の

- 図形オブジェクトの可視性条件

を加えた制約を解くことによって、階層集合の可視化を実現する。

### 6.4 最適化モデルへのマッピング

同値類構成の条件  $\text{group}$  は、同値関係が推移的閉包により定義されているため、 $g \sim_k g'$  に対し、

$$d_k(g, g') \leq \Delta_k, d_k(g_1, g') \leq \Delta_k, \dots, d_k(g_n, g') \leq \Delta_k$$

を与える  $g_1, \dots, g_n$  が存在することである。したがっ

\* 縦一列、横一列、などの制約に翻訳される関係、また、順序関係など。

☆☆ 図形オブジェクトの集合  $G, G'$  に対して、 $G$  中の任意の  $x, y$  の遠隔度  $d(x, y)$  と  $G'$  中の任意の  $x', y'$  の遠隔度  $d(x', y')$  を不変に保ったまま  $d_1(G, G')$  として  $[\Delta_i, \Delta_{i+1})$  の値を 1 つ以上 (任意の  $i$  に対して) とらせることができること、などの条件。図形オブジェクト間の通常の距離を遠隔度として使えばこの条件は満たされる。

☆☆☆  $o \in A$  に対し、 $\bar{f}(o) = f(o)$ 。  $o \in \text{Member}^*(h) - A$  に対し、 $\bar{f}(o) = \{f(o') \mid o' \in o\}$ 。

て、 $\sim_k$ を調べるには、このような  $g_1, \dots, g_n$  が存在するか否かを決定しなければならないが、これを厳密に解こうとすると、 $g_1, \dots, g_n$  の可能性に関して組合せの爆発が起こる。そこで、図形オブジェクトの「集まり」具合を測る関数として、その値が十分小さいことが  $group(s)$  すなわち、図形オブジェクトの集合  $s$  が同値関係  $\sim_k$  による同値類となることに対応する関数を考える。

$s \in Member^*(\bar{f}(h)), k = Depth(s), s', s'' \in s$  に対し、 $s', s''$  は  $s' \sim_{k-1} s''$  であり、 $s' \not\sim_{k-2} s''$  でなければならない。

- (A)  $s' \sim_{k-1} s''$  は  $s'$  と  $s''$  のそれぞれの要素間の遠隔度の2乗和の平方根を目的関数として最小化することにより近似する。
- (B)  $s' \not\sim_{k-2} s''$  は、階層遠隔度を使うとすると、

$$\forall i' \in Base(s'), \forall i'' \in Base(s''), \\ d(i', i'') > \Delta_{k-2}$$

という条件となるので、 $d(i', i'') > \Delta_{k-2}$  からのずれを目的関数として最小化することで近似する。

階層集合  $h$  の可視化を、これらの目的関数の和の最適化問題として扱うことにする。

### 7. 実行例

前章までの考え方に基づいて、階層集合の可視化プロトタイプシステムを実装した。実装はUNIX\*上で行い、抽象データ定義、評価関数定義、表示用ユーザインターフェイスはSTk (Scheme Tk)<sup>15)</sup>を用いて7000行ほどのプログラムで実現した。

評価関数は次のものを用いた。表示する抽象データの階層集合を  $h$  とする。

$$e = w_s \sum s_i + w_h \sum h_i + w_v v$$

$w_s, w_h, w_v$  は適当な係数である。以下の実験は  $w_s = w_h = w_v = 1$  として行った。

$s_i$  は、図形オブジェクトの中心位置の分散の項であり、6.4節の(A)で図形オブジェクトをできるだけまとめるために導入した。

$$s_i = \sum_{\substack{o \in Member^*(h) \\ Depth(o)=i}} \sqrt{\sum_{\substack{s', s'' \in \\ f(Base(o))}} d(s', s'')^2}$$

$h_i$  は6.4節の(B)で述べた、階層遠隔度  $d_1, d_2, \dots$

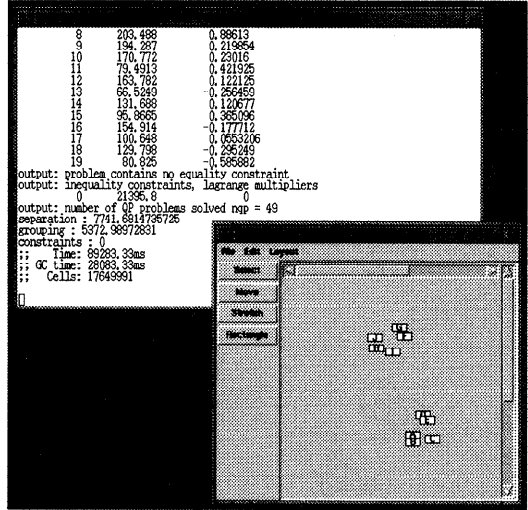


図15 プロトタイプシステムによる実行  
Fig. 15 Prototype system.

を与えたときに決まる図形オブジェクトの標準的同値関係が階層集合と同型になるところからどれだけずれているかのうち、「集まり」すぎを抑える項である。

$$h_i = \sum_{\substack{o \sim_i o' \\ o, o' \in Member^*(f(h))}} P(d(o, o') - \Delta_{i-1})$$

$P(x)$  は条件からのずれを数値的に評価するものであり、プロトタイプシステムでは  $P(x) = \text{if } x > 0 \text{ then } 0 \text{ else } x^2$  を使っている。

$v$  は6.2節の可視性条件からのずれを評価する項である。

$$v = \sum_{o, o' \in Base(\bar{f}(h))} Q(d(o, o') - m)$$

プロトタイプシステムでは  $Q(x) = \text{if } x > 0 \text{ then } 0 \text{ else } x^2$  を使っている。

各パラメータに適当な初期値を乱数により与えた後で、SQP (Sequential/Successive Quadratic Programming)<sup>12)</sup>を使用して、評価関数の最小化を行った。SQPはC言語で実装しており、STkインタプリタからダイナミックにリンクして呼び出している。

$h = \{\{\{a, b, c\}, \{d, e\}\}, \{\{f, g\}, \{h, i, j\}\}\}$  の可視化を実行した例を図15に示す。この例はSun4/20で実行し、実行時間約89秒(うちGarbage Collection時間約28秒)である。

例1 単階層集合を表示する。

$$h = \{\{0, 1\}, \{2, 3\}, \{4, 5\}, \{6, 7\}, \{8, 9\}\} \\ m = 30, \Delta_1 = 50, \Delta_2 = 70$$

\* SunOS 4.1.3, 4.1.4-JLE1.1.4, 5.5.1, FreeBSD-2.1.6で動作を確認した。

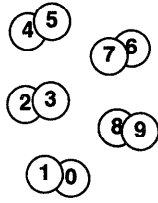


図 16 単階層集合の表示例

Fig. 16 Example of visualization of 1-hierarchical set structure.

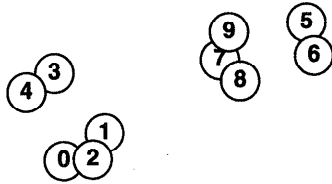


図 17 2階層集合の表示例

Fig. 17 Visualization of 2-hierarchical set structure.

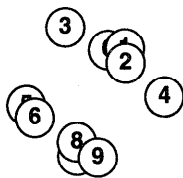


図 18 遠隔度の和の項を図形オブジェクトについてだけにした場合  
Fig. 18 Only  $s_1$  is calculated in the term of  $\sum s_i$ .

0と1, 2と3, 4と5, 6と7, 8と9の「集まり」が、それぞれ判別できる(図16)。

例2 多階層集合を表示する。

$$h = \{\{\{0, 1, 2\}, \{3, 4\}\}, \{\{5, 6\}, \{7, 8, 9\}\}\}$$

$$m = 30, \Delta_1 = 50, \Delta_2 = 70$$

0~4の「集まり」が5~9の「集まり」と分離されて見え、集合の階層的構造が読みとれる(図17)。

例3 遠隔度の和の項  $\sum s_i$  を  $s_1$ , つまり図形オブジェクトの遠隔度の和だけにすると、うまくまとまらないことが多い。たとえば、例2と同じ集合

$$\{\{\{0, 1, 2\}, \{3, 4\}\}, \{\{5, 6\}, \{7, 8, 9\}\}\}$$

を可視化すると図18のようになり、0~4の「集まり」はできるが、3, 4の「集まり」がうまくできない。したがって、遠隔度の和の項の定義が階層集合に基づいていることには意味がある。

領域への拘束

本論文で提案する階層集合の可視化の方式は、評価関数の最小化をベースにしているのので、評価関数の形式で書ければ、他の条件を組み合わせることが可能である。



図 19 領域への拘束の例 1

Fig. 19 Group bound to a region 1.

例として、図形オブジェクトの特定領域への拘束を条件として与えたものを示す。凸でない領域の内部への拘束を、不等式を論理演算子で組み合わせて表すと、選言的論理演算子のために、解に組合せ的な困難がともなってしまう。ここでは、領域への拘束条件を評価関数として表して、全体の評価関数に加えることにより、組合せ的問題を避ける。

例4  $h = \{\{A, B, C\}, \{D, E, F\}\}$  で、 $\{A, B, C\}$  を、かぎ形の領域の内部に置く(図19)。

かぎ形を左上の座標が(200, 200), 右下の座標が(250, 250)の正方形と左上(200, 200), 右下(220, 220)の正方形に分け、左上(200, 200), 右下(250, 250)の内部, 左上(200, 200), 右下(220, 220)の外部を評価関数で表す。左上(200, 200), 右下(250, 250)の内部なら、評価関数は

$$\sum_{o \in G} \begin{pmatrix} \text{if } (o_l > 200) \text{ and } (o_r < 250) \\ \text{and } (o_t > 200) \\ \text{and } (o_b < 250) \\ \text{then } 0 \\ \text{else } \max\{200 - o_l, 200 - o_t, \\ o_r - 250, o_b - 250\} \end{pmatrix}$$

とする。ただし、 $o_l, o_t, o_r, o_b$  はそれぞれ、図形オブジェクト  $o$  の左上および右下の座標値。

領域の左上の欠けた部分を拡張して(200,200), (230, 230)とすると図20のようになる。

例5  $h = \{\{A, B, C\}, \{D, E, F\}\}$  で、 $\{A, B, C\}$  も  $\{D, E, F\}$  も十字形の内部に置く(図21)。十字形の内部という条件は、次の5つの条件を表す評価関数とする。

- $\{A, B, C\}, \{D, E, F\}$  が

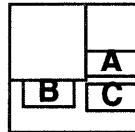


図 20 領域への拘束の例 2  
Fig. 20 Group bound to a region 2.

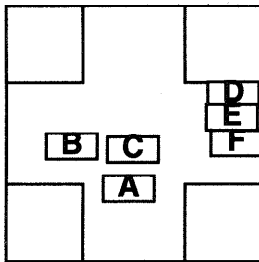


図 21 領域への拘束の例 3  
Fig. 21 Group bound to a region 3.

- 長方形 (100, 100), (200, 200) の内部にある。
- {A, B, C}, {D, E, F} が  
長方形 (100, 100), (130, 130) の外部にある。
- {A, B, C}, {D, E, F} が  
長方形 (170, 100), (200, 130) の外部にある。
- {A, B, C}, {D, E, F} が  
長方形 (100, 170), (130, 200) の外部にある。
- {A, B, C}, {D, E, F} が  
長方形 (170, 170), (200, 200) の外部にある。

例 6 (図形オブジェクトの整列) TRIP<sup>3)</sup> で使われた vertical のような述語も評価関数で近似的に表すことができる。{A, B, C, D, E, F} を vertical(A, B, C, D, E, F) を条件として可視化する (図 22)。vertical は中心位置の x 座標の分散を用いている。なお、この例では、分散の項  $s_i$  として、図形オブジェクト間の最短距離の和を使用している。

交わりのある集合

交わりのある集合の場合、階層集合構造を忠実に可視化することは一般にはできないが、できるだけ条件を満たす目的で本論文のモデルを使うことはできる。

交わりのある集合の場合、階層遠隔度が使えないので、次のようにして図形オブジェクト間の遠隔度の下限を直接定める。抽象オブジェクト  $a, b (a \neq b)$  に対し、 $a \in Member^*(c), b \in Member^*(c)$  となるような  $c$  で、 $i = Depth(c)$  が最小となるものを求め、



図 22 制約 vertical の使用例  
Fig. 22 Grouped and vertically aligned.

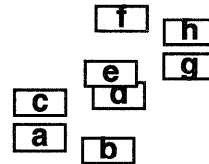


図 23 交わりのある集合の表示例  
Fig. 23 Visualization of joint sets.

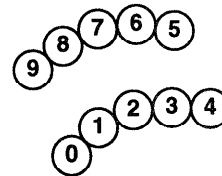


図 24 交わる集合を連ねた例  
Fig. 24 Series of joint sets.

図形オブジェクト  $f(a), f(b)$  の間の遠隔度の下限を  $d_{i-1}$  とする。ただし、 $d_0 = m$  とする。もし、この条件を満たす  $c$  が存在しないときは、 $f(a), f(b)$  の間の遠隔度の下限は  $d_i$  の最大値  $d_{max}$  とする。

例 7 共有部分を持つ 2 つの集合を表示する。

$$h = \{\{a, b, c, d, e\}, \{d, e, f, g, h\}\}$$

$\{a, b, c, d, e\}$  と  $\{d, e, f, g, h\}$  の交わりの  $d, e$  を挟むように配置される (図 23)。

例 8 互いに交わる集合を順次連ねることによって、順序付の集合を表示する (図 24)。

$$h = \{\{\{0, 1\}, \{1, 2\}, \{2, 3\}, \{3, 4\}\}, \{\{5, 6\}, \{6, 7\}, \{7, 8\}, \{8, 9\}\}\}$$

例 9 例 6 の vertical, horizontal と例 8 を組み合わせる (図 25)。

$$h = \{\{\{0, 1\}, \{1, 2\}, \{2, 3\}, \{3, 4\}\}, \{\{5, 6\}, \{6, 7\}, \{7, 8\}, \{8, 9\}\}\}$$

$$horizontal(0, 1, 2, 3, 4)$$

$$vertical(5, 6, 7, 8, 9)$$

例 10 交わりのない 2 つの集合の、両方に交わる第 3 の集合を加えることにより、それぞれの集合の特定の要素を近付けて配置する (図 26)。

$$h = \{\{0, 1, 2, 3, 4\}, \{5, 6, 7, 8, 9\}, \{2, 3, 7, 8\}\}$$

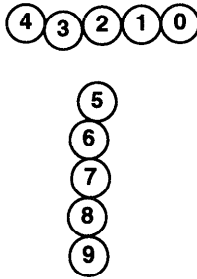


図 25 交わる集合と vertical, horizontal の組合せの例  
Fig. 25 Vertically or horizontally aligned series of joint sets.

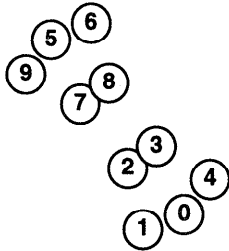


図 26 2つの集合に交わる第3の集合の例  
Fig. 26 The third set over two disjoint sets.

## 8. 議 論

本研究では、可視化システムで通常用いられる図形オブジェクトの整列のような配置指定だけでは適切に可視化しえない構造があることから、図形オブジェクトの集まりによる抽象オブジェクト間の関係（階層集合）の表現を提案した。さらにこれに基づく可視化プロトタイプシステムを作り、いくつかの図を生成した。ここでは、これに基づき、本研究で提案する階層集合の可視化モデルの有効性についての議論を行う。

### 8.1 図形オブジェクトの整列と集まり

特に制約を加えない図形オブジェクトの集まりの生成例（例 1, 2）では、実際に整列していない集まりが作られている。このように、図形オブジェクトの整列していない集まりを作ることにより、「整理された」という印象を与えない図を描くことができ、また、図形オブジェクトの整列から生じる配置の矛盾や意図しない含意を避けることができる。

図形オブジェクトの整列していない集まりは、整理したもの比べて、より大きな描画面積を必要とする傾向がある。ただし、本論文で示した手法では集まりを作るという制約と他の制約との混在が許されることが示されており（例 6）、必要に応じて「一列に並んだ集まり」などを作ることで描画面積を節約することができる。この場合一列に並んだ集まりは図形オブジェ

クトの並び順は特定されていないため、指定した順序どおりの整列で起こりうる配置の矛盾が回避される。むしろ、指定した順序どおりの整列が必要な場合には、集まりを作るのではなくそのような配置を行うようにすればよい。

### 8.2 階層集合のスケール

本論文の可視化モデルに基づき、実際に階層構造を持つような集合も図形オブジェクトの集まりを用いて可視化される（例 2）。こうして構成された図形オブジェクトの集まりから元の階層構造を抽出することは、理論的にはどんなに階層構造が複雑になっても可能だが、実際にそのような図を見て階層構造を認識できるかという問題がある。

本論文では、プロトタイプシステムの機能的制限から、例 2 の範囲を越える階層集合は扱っていないが、3~4 階層程度が限界と推測される。ただし、距離と色などの複数の遠隔度を組み合わせることにより、さらに多くの階層を表現できる可能性がある。また、2.4 節で述べた閉曲線による囲みを併用すれば、より多くの階層がある場合も効果的な可視化が期待される。

### 8.3 可視化対象の制限

本論文で提案する可視化モデルでは可視化の対象となる階層集合は交わりのないバランスした階層集合に限られている。まず、「バランスした」という点に関してであるが、一般に、階層集合のバランスしていない部分（深さの足りない部分）に仮想的な階層を追加することで対応できる。多くの場合、中間階層の追加は、抽象データの構造についての若干の解釈の変更として対応でき、致命的な問題ではないと考えられる。

一方、交わりのある階層集合は、図形オブジェクトの集まりの持つ遷移性から、集まり「だけ」によって表すのは不可能である。しかし、2.4 節で述べたように、閉曲線による囲みを併用することで交わりのある集合に対応することは可能である。たとえば例 7 について、閉曲線による囲みを適用することで、自然に交わりのある集合が表される。

### 8.4 制約系の複雑さ

抽象データの構造がより複雑化した場合、それにとってもなつて図構造データの構造も複雑化し、可視化に際して多くの制約を含む制約系を解かなければならなくなる。このとき、すべての制約を満たすような解が存在しないか、あるいは求めるのが非常に困難になる。

まず、そのような場合にいかにしてより良い解を求めるかという問題がある。本論文では最適化のアルゴリズムを採用しているが、問題が複雑化した場合、無数に存在する局所解への落ち込みのような問題が生じ

る可能性がある。これを避けてできるだけ良い解を得られるようにするため、目的関数のパラメータを適切に決定する必要がある。本研究で示した可視化モデルは必ずしも現在の最適化モデルと不可分のものではないので、他の最適化、制約解決アルゴリズムの適用を含めて検討の余地がある。

また、すべての制約を満たすような解が見つからない場合、最適解に近い解が得られることが期待されるが、それでも、階層集合を図形オブジェクトの集まりだけで表そうとする限り、可視化の忠実性は崩れる。この点も、閉曲線による囲みや、線による接続を併用することによる解決の余地がある。これらの手法を併用した際、図形オブジェクトの集まりは階層集合と厳密に対応しないが、図形オブジェクトの集まりは図の可読性を上げる効果がある。

### 8.5 集合の他の表現法

8.2, 8.3, 8.4 節で指摘したように、可視化の対象となる抽象データの構造が複雑化した場合には図形オブジェクトの集まりだけによる集合の表現では、階層集合の十分に効果的な可視化ができないおそれがある。このような場合には、閉曲線による囲みや線による接続といった手法を図形オブジェクトの集まりと併用することが解決策となりうる。複雑な抽象データを統一的なモデルで可視化するには、これらのような手法を含んだ形でのモデルの構築が必要となる。

### 8.6 分かりやすさの評価

図形オブジェクトを閉曲線で囲んだり、線で結んだりすることは、それだけで関係があることを表したり、関係のない図形オブジェクトを区別する機能がある。しかし、2.4 節で示したように、図形オブジェクトを閉曲線で囲んだり、線で結んだりすることだけで抽象オブジェクト間の関係を表そうとすると、分かりにくい図になる可能性がある。そのため、より分かりやすい図を描くには、これらの手法と図形オブジェクトの集まりとを併用する必要がある。この場合、図形オブジェクトの集まりは、図の可読性のために必要となる。

このような、図の分かりやすさのための図形オブジェクトの集まりの有効性を示すには、なんらかの基準で図の分かりやすさを判定する必要がある。図の分かりやすさについての詳細な議論は今後の研究に委ねるが、直観的には、本研究の結果では「集まり」をうまく表現しているように見える。

## 9. 結 論

本論文では、

- 図形オブジェクトの集まりとはどのようなものか。

- 図形オブジェクトの集まりとはどのように作られるか。

ということについて、階層集合を図形オブジェクトの集まりが忠実に表現するための条件を示し、この条件を評価関数により近似する方法を示した。

評価関数による表現では、評価関数の最小化により、配置を近似的に求めることができ、さまざまな種類の制約条件を組み合わせることが可能である。また、論理型の条件を直接扱う場合におこりがちな組合せ的爆発の問題も避けることができる。

非常に複雑な構造を持つような抽象データについては、図形オブジェクトの集まりによる階層構造の忠実な可視化には困難がともなう可能性がある。しかし、可視化によって生成される図の可読性の点から図形オブジェクトの集まりが、完全な形ではないにせよ必要となる。

また、プロトタイプシステムによる可視化例では、十分に単純な構造を持つ抽象データについては本論文の可視化モデルによって図形オブジェクトの集まりだけでこれを効果的に可視化できることが示された。

複雑な構造を持つ抽象データに対し、本論文のモデルを適用するに際して、閉曲線による囲みや線による接続の併用を含めたアプローチが課題となる。

謝辞 東京大学大学院総合文化研究科広域科学専攻広域システム科学系の皆様へ感謝いたします。なお、この研究の一部は文部省科学研究費補助金（基盤研究 B (2) 08458063）によっている。

## 参 考 文 献

- 1) Anderberg, M.R., 西田英郎 (監訳), 佐藤嗣二ほか (共訳): クラスタ分析とその応用 (Cluster analysis for applications), 内田老鶴圃 (1988).
- 2) 平凡社 (編): 心理学事典, 平凡社 (1981).
- 3) Kamada, T.: *Visualizing Abstract Objects and Relations, A Constraint-Based Approach*, World Scientific (1989).
- 4) Marks, J.: A Syntax and Semantics for Network Diagrams, *Visual Language*, pp.104-110, IEEE (1990).
- 5) Borning, A., Duisberg, R., Freeman-Benson, B., Kramer, A. and Wolf, M.: Constraint Hierarchies, *OOPSLA '87*, pp.48-60, ACM (1987).
- 6) Roth, S., Chuah, M., Kerpedjiev, S., Kolojchick, J. and Lucas, P.: Toward an Information Visualization Workspace: Combining Multiple Means of Expression, *Human-Computer Interaction*, Vol.12, No.1/2, pp.131-185 (1997).
- 7) Tamassia, R., Battista, G. and Batini, C.: Automatic Graph Drawing and Readability of Di-

- agrams, *IEEE Trans. Systems, Man, and Cybernetics*, Vol.18, No.1, pp.61-79 (1988).
- 8) Tamassia, R.: Tamassia's Graph Drawing Page. <http://www.cs.brown.edu/people/rt/gd.html>.
- 9) Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S. and Roseman, M.: Navigating Hierarchically Clustered Networks Through Fisheye and Full-Zoom Methods, *ACM Trans. Computer-Human Interaction*, Vol.3, No.2, pp.162-188 (1996).
- 10) Eades, P., Lin, T. and Lin, X.: Two Tree Drawing Conventions, *International Journal of Computational Geometry and Applications*, Vol.3, No.2, pp.133-153 (1993).
- 11) 杉山公造：図的思考展開支援に関する基礎的研究，技術報告，24，25，富士通株式会社国際情報社会科学研究所 (1988)。
- 12) 茨木俊秀，福島雅夫：FORTRAN77最適化プログラミング，岩波書店 (1991)。
- 13) Freeman-Benson, B., Maloney, J. and Borning, A.: An Incremental Constraint Solver, *Comm. ACM*, Vol.33, No.1, pp.54-63 (1990).
- 14) 溝口文雄，古川康一，Lassez, J-L. (編)：制約論理プログラミング，共立出版 (1989)。
- 15) Gallezio, E.: STk Homepage. <http://kaolin.unice.fr/STk/>

(平成 9 年 8 月 26 日受付)

(平成 11 年 7 月 1 日採録)



酒井 恵光 (正会員)

1967年生。1991年東京大学教養学部基礎科学科第2卒業。1999年同大学院博士課程満期退学。現在，同大学院研究生。ソフトウェア科学会，ACM各会員。



山口 和紀 (正会員)

1956年生。1978年東京大学理学部数学科卒業。1980年同大学院理学系研究科情報科学専攻修了。理学博士。1991年同大学助教授。ACM，IEEE CS各会員。



川合 慧 (正会員)

1944年生。1967年東京大学理学部卒業。1969年同大学理学部助手，1987年同大学教養学部教授。1993年同大学院総合文化研究科教授。理学博士。コンピュータグラフィクス，ユーザインタフェース，プログラム言語。「プログラミングの方法」(岩波書店，1988)。「コンピューティング科学」(東大出版会，1995)。ソフトウェア科学会，電子情報通信学会，ACM各会員。