

## 概念の形成と概念構造の融合

中島 誠      渡邊 美香      伊藤 哲郎

大分大学 工学部 知能情報システム工学科

870-11 大分市大字旦野原 700 番地

概念は対象の各クラスタに属性を介した記述を定めると得られるが、あらかじめ属性が準備されていなくても入力対象の変化に合わせて概念の形成ができる生成的モデルの構築が望まれている。ここでは、必要な属性を作りながら概念が学習できるモデルをもとに、別な構造にあっても互いに関連あると見なせる概念同士を対応づけることで、複数の構造をひとまとめにして扱う方法を定式化し、入力の変化に応じた生成的な概念形成を試みる。またこの対応づけに従い、それぞれの構造の共通点ならびに相違点を明らかにする方法も定式化する。提案した方法の有効性は色概念構造の構築を通じて実験的に検討する。

## Forming Concepts and Merging Concept Hierarchies

Makoto Nakashima    Mika Watanabe    Tetsuro Ito

Department of Computer Science and Intelligent Systems, Faculty of Engineering, Oita University,

700 dannoharu, Oita-shi, Oita-ken 870-11, Japan

Concepts can be formed by providing object clusters with attribute based descriptions. There have been proposed many concept learning models. To get a model which can form concepts constructively under non-enough given attributes and adaptively to the varying inputs, however, is most attractive for modeling how human beings acquire the concepts from the real world. For modeling adaptive concept formation, we here try to formulate a process of treating several concept hierarchies, obtained constructively under various input environments, as a unique one by corresponding concepts seemingly different but specified by similar attribute-values. Further, we try to clarify the human way of mutual understanding of concepts by finding corresponding parts in different hierarchies. The proposed methods are also examined computationally for the color concept hierarchies.

## 1. はじめに

少数の事実に基づいて多くの場面に通じる一般化を行ったり、一見無秩序な観測データから規則性を見いだしたりできる人間の能力を、計算機を用いて解明する研究分野に概念の帰納学習がある。概念は対象の各クラスタに、その全体をひとまとめで言及できるような属性を介した一般化記述を定めることで形成されるが、対象や属性の扱い方の違いによって種々の概念形成モデルが考えられる [9]。中でも、属性を生成しながら扱う対象の変化に応じた概念の形成法の開発が望まれている [14]。これに関連して、新しい属性を作りだせる生成モデル [11, 13] や対象が入力される都度既存の構造を修正してゆける逐次モデル [2, 3] が提案されてきた。しかし、そこには利用できる属性を前もって決めておかねばならなかったり、得られた概念同士を明確に区別する基準がなかったりする欠点がみられる。

属性を準備せずに概念を形成するには、概念に加えて属性も入力対象の集まりに依存して作り出さなければならない [7, 12]。ここでは、逐次的な処理はできないが、対象の観測データから必要な属性を生成しながら同時に概念が形成できるモデル CNC [7] をもとに、(a) 別々な対象の集まりから作られた複数の概念構造をひとまとめで扱う融合操作を定式化することで、入力の変化に応じた生成的な概念形成を試みる。融合操作は小規模の構造を一つにまとめるためにも利用できる。また、(b) 別な構造にありながら互いに同じと見なせる概念を対比させる方法を定式化し、それぞれの構造の共通点ならびに相違点を明らかにすることも試みる。

以下、2 節ではモデル CNC について簡単に説明する。3 節では CNC で作られた概念構造を相手に、概念同士の対応づけをもとにして複数の構造を融合する方法および構造中の概念を対比させる方法について議論する。この場合、それぞれの概念は別々な対象の集まりから作られているため、どの概念同士を対応づければよいか直接調べることは不可能である。これには共通基盤という考え方を導入し、この上での関連性を属性値の構造、そしてもとの概念構造にと反映させてゆくようにする。4 節では色データを用いて提案した方法の有効性を実験的に検討し、最後の 5 節では応用面について簡単に考える。

## 2. 概念形成

モデル CNC [7] は、対象を観測して得た数値データとしてのパターンを受け取り、それらの間の類似度からクラスタを求める。そして、そのパターンがクラスタの要素となった各対象についての記述 (パターンからの記述と呼ぶ) をもとに、それらを一括して言及する一般化記述 (クラスタからの一般化記述と呼ぶ) を得ることで概念を形成する。この場合、あるクラスタからの一般化記述が別なクラスタ中のパターンを生んだ対象にもあてはまれば、記述はクラスタを識別したとはいえなくなり、クラスタの設定そのものが問題となる [8]。そのため、求められた一般化記述がクラスタを識別できるとされた場合に概念が形成される。全体的には、一般化記述の与えられた各クラスタのサブクラスタに同様の事柄を繰り返し、各概念ならびに一般化記述がそれぞれ節点ならびに枝に対応した木構造での概念階層を作る。

記述に現れる属性は、値の構造自体が上・下位関係を定めた概念階層とみなせることに注目し、もとの観測データの一部からなる限定化パターン (例えばカラーイメージに対するグレイイメージや色データなど) に概念形成の全プロセスを再帰的に働かせて生成する。ただし、パターンがその限定化パターンを持っていなければ、クラスタに名前をつけることで概念を定める記述が得られるとみなす。概念の階層構造を成長させる各段階では、一つの属性に係わる記述を扱い、その属性の生成には、対象クラスタを得るのに最も大きく寄与した限定化パターンを

選ぶようにしている。ここでは、各レベルの概念ができるだけ同一の属性で識別されているような均質の概念構造が最も理解しやすいこと [10] を考慮して、対象のクラスタを得るのに大きく寄与した限定化パターンのうちでも、概念構造を成長させてきた現時点までと同じ種類のものを選ぶ可能性を高くする。作られた概念の構造や属性値の構造は、結果的に、もとの構造を説明するためにより基本的な概念構造が段階的な背景知識を成しているような形となる。

### 3. 概念構造の扱い

#### 3.1 考え方

CNCを用いて作られた二つの概念階層SとTに対する融合・対比操作を定式化する(複数の構造を扱う場合には二つの構造に対する処理を繰り返す)。これには、一方の構造中の概念を他の構造のもののみとしたとき、影響を受ける極小の部分を見つけ、その中でお互いに関連ある概念同士を対応づければよい。ところがSやTは必ずしも同じとは限らない属性・属性値をもとに作られているため、どの概念同士が関連しているかを直接調べることができない。ここでは共通基盤という考え方を導入してこれに対処する。

異なった対象同士でもその限定化パターンについては同じものを持つ可能性があり、より限定化されたパターンほどその度合いは高い。このことから、最初、これ以上限定化されたパターンをもたないパターンが作る空間を共通基盤とみなす(CNCではこの種のパターンにはクラスタに名前をつけることで概念が形成されたとした)。この空間上で概念同士の関連を見るには、対応するクラスタ中のパターンを類似度にしたがって並べたとき、別なクラスタ中のパターンが別な区分に入るようにできれば、概念間に関連はないとする。全体としては、共通基盤が存在する構造について融合・対比操作を施し、これらの結果を一つ浅い段階での共通基盤とみなしながら、最終的にはもとの概念構造上での処理を行うようにする。

#### 3.2 対応づけ

融合・対比操作を行なうにあたっては、概念的にかけ離れていない概念同士が対応づけられるようになっていくことが望ましい。このことを考慮に入れて対応づけを定式化しよう。

S(あるいはT)中の節点 $s(t)$ の子節点を $\langle s_1, s_2, \dots, s_m \rangle$  ( $\langle t_1, t_2, \dots, t_n \rangle$ )と書く。節点は概念をまた子節点は下位概念を表し、子節点は各構造中で共通基盤として特定の属性に関する値で識別されているとする。いま、 $s_i, 1 \leq i \leq m$ , を定める属性値が $t_j, 1 \leq j \leq n$ , を定める値の上位にあるとき、 $s_i$ は $t_j$ の上位になるという。 $\langle s_1, s_2, \dots, s_m \rangle$ に対して別な $\langle u_1, u_2, \dots, u_o \rangle$ を考えたとき、任意の $s_i$ についてその上位になる $u_k, 1 \leq k \leq o$ , が存在し、かつ任意の $u_k$ についてその下位になる $s_i$ が存在するならば、 $\langle u_1, u_2, \dots, u_o \rangle$ は $\langle s_1, s_2, \dots, s_m \rangle$ を被覆するという。この被覆の考えを使って、二つの $\langle s_1, s_2, \dots, s_m \rangle$ と $\langle t_1, t_2, \dots, t_n \rangle$ 中の $s_i$ と $t_j$ が対応づけられているとは、 $s_i$ と $t_j$ がある $u_k$ に対してその下位になるような被覆 $\langle u_1, u_2, \dots, u_o \rangle$ が存在するときをいう。上での要請から被覆としてはできるだけ他の被覆を被覆しないものを選ぶ。具体的な被覆の求め方として次のようなものを考える。

CC被覆:  $s_1, s_2, \dots, s_m, t_1, t_2, \dots, t_n$ に係わる記述について、共通の子孫を持つ属性値で規定される節点のつながりをひとまとめにし、そこにみられる属性値の極小で共通の先祖についての記述で定められる概念を一つの節点 $u_k$ とする。

MSC 被覆:  $s_1, s_2, \dots, s_m$  それぞれに係わる記述の論理積と  $t_1, t_2, \dots, t_n$  それぞれに係わる記述の論理積について MSC 一般化記述<sup>1</sup>を求めたとき、各部分記述で定められる概念を一つの  $u_k$  とする。

CC ならびに MSC 両被覆は、共通基盤が木の形を取っていれば、任意の  $\langle s_1, s_2, \dots, s_m \rangle$  と  $\langle t_1, t_2, \dots, t_n \rangle$  に対して一意に決まることがいえる。CC 被覆は属性値が共通の子孫を有するような複数の節点同士を対応づけるが、被覆中の各節点には識別的な記述が与えられる。一方、MSC 被覆は属性値が近い位置に共通の先祖を有するような節点を一対一に対応づけるが、必ずしも被覆の各節点には識別的な記述が与えられるとは限らない。ただし MSC 被覆の節点に識別的な記述が与えられていれば、MSC 被覆は CC 被覆に被覆されることがいえるため、融合・対比操作では、まず MSC 被覆での対応づけを利用する。

二つの構造が与えられたとき、そこでの同じと考えてよい部分を見いだす方法として、パターン認識のための緩和法やグラフ照合法、知識処理のための干渉照合法 [6]、類推の基本となる類比操作 [4] などが提案されてきた。しかしながら、それらは節点につけられた名前の同一性あるいは節点間のつながり具合を基本にした方法であり、ここでのように節点の概念的な扱いはなされていない。概念はその背景知識をもとにして形成されていることを考えると、共通部分構造を見つける場合にも、節点の背景知識上での関連性を捉えることが必要である。

### 3.3 融合

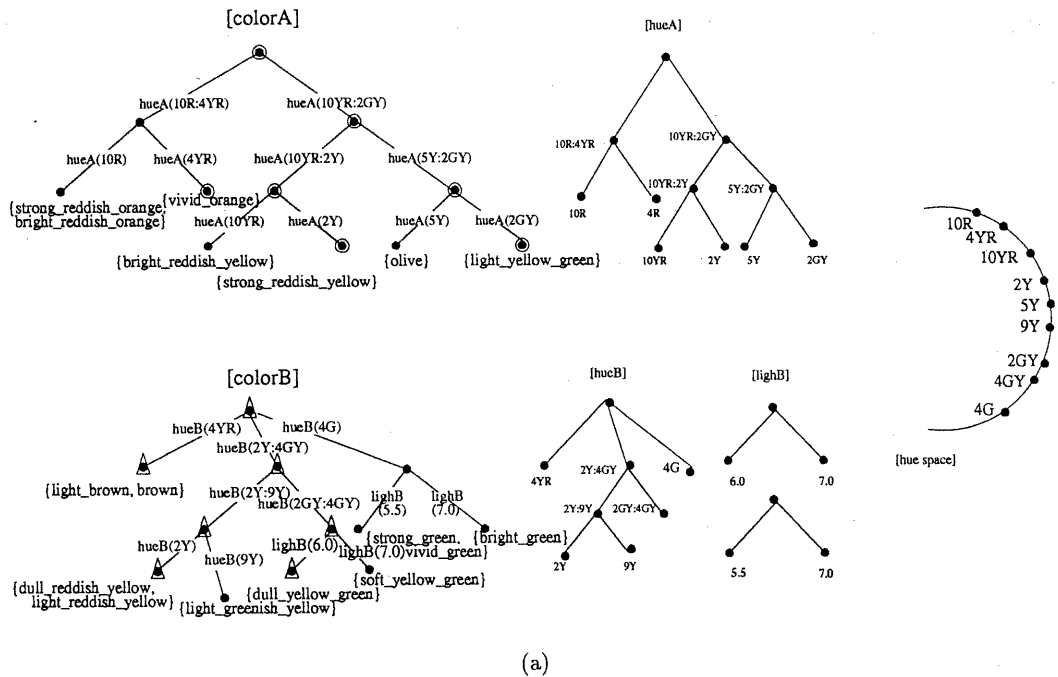
ある  $s_i$  が  $t_j$  と対応づけられかつ  $s_i$  が  $t_j$  の上位になったとする。このときもし、 $s_i$  の子節点が  $s_i$  と同じ属性で規定されていれば、 $s_i$  の子節点と  $t_j$  から作った被覆の方がもとの  $s_i$  と  $t_j$  から作った被覆に被覆される。この  $t_j$  のような節点が複数存在しても同じことがいえる。以下、融合操作を定式化するが、この事実を拡大解釈し、ある  $s_i$  (あるいは  $t_j$ ) が  $t_j, \dots, t_l(s_i, \dots, s_k), 1 \leq l \leq n, 1 \leq k \leq m$ , と対応づけられていれば、 $s_i(t_j)$  の子節点と  $t_j, \dots, t_l(s_i, \dots, s_k)$  から被覆を作ることを試みるようにする。以下、融合操作で作られる構造を融合構造と呼ぶ。

#### [融合操作]

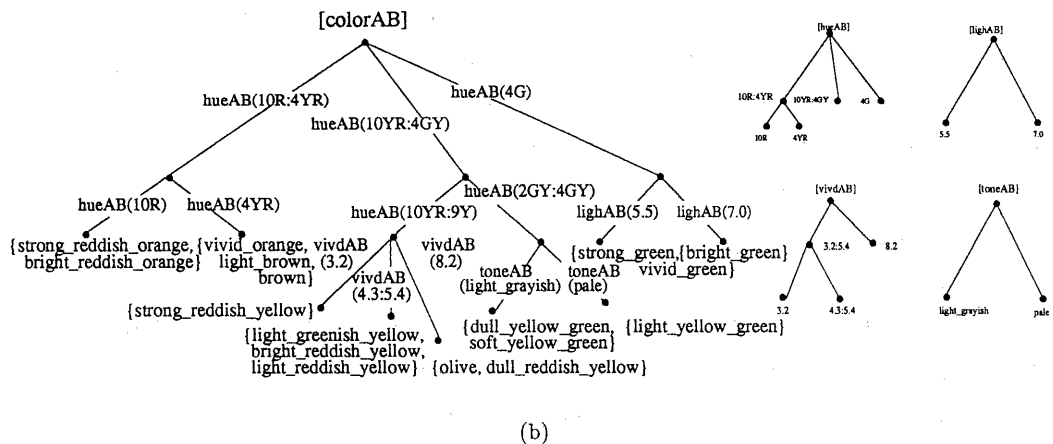
- (step1) 構造 S と T 中の根節点の子節点から葉節点に向かって step2 を実行する。
- (step2) 注目している節点を規定する共通基盤があるならば step3 へ。共通基盤がなく属性が同じであるならば、融合操作を再帰的に用いて共通基盤を求め step3 へ。属性が違っていれば step4 へ。
- (step3) MSC 被覆あるいは CC 被覆が求まるならば step3a, step3b を行なう。
  - (step3a) MSC 被覆の節点の記述が互いに識別的であれば以下を行う。
    - (step3a.1) 仮節点と対応づけられた節点については、それを根とする部分木をそのまま融合構造に組み入れる。
    - (step3a.2) 各対応づけられた  $s_i$  と  $t_j$  について、 $s_i(t_j)$  が  $t_j(s_i)$  の上位になるなら、 $s_i(t_j)$  を融合構造の節点とみなし、 $s_i(t_j)$  の子節点と  $t_j(s_i)$  について step2 から繰り返す。そうでなければ被覆中の  $u_k$  を融合構造の節点とし、 $s_i$  ならびに  $t_j$  の子節点を取り出し step2 から繰り返す。
  - (step3b) CC 被覆の節点の記述が互いに識別的であれば以下を行う。
    - (step3b.1) いずれとも対応づけられていない節点があるならば、その節点を根とする部分木はそのまま融合構造に組み入れる。
    - (step3b.2) 各対応づけられた  $s_i(t_j)$  と  $t_j, \dots, t_l(s_i, \dots, s_k)$  について  $s_i(t_j)$  が  $t_j, \dots, t_l(s_i, \dots, s_k)$  すべての上位になるなら、 $s_i(t_j)$  を融合構造の節点とみなし、 $s_i(t_j)$  の子節点と  $t_j, \dots, t_l(s_i, \dots, s_k)$  について step2 から繰り返す。そうでなければ被覆中の  $u_k$  を融合構造の節点とし、 $t_j, \dots, t_l(s_i, \dots, s_k)$  中  $s_i(t_j)$  の下位になる節点とそうでない節点の子節点をあわせたものと  $s_i(t_j)$  の子節点について step2 から繰り返す。
- (step4) 共通基盤を求めている段階でなければ、注目している節点を根とする部分木中の概念にあてはまる対象について CNC を動かす。

<sup>1</sup>一対一に対応づけられた各  $s_i, t_j$  に係わる記述を含意する一般化記述の論理積を連言的一般化記述というが、そのうちの最も特定化されたものを MSC(maximally-specific conjunctive)一般化記述 [8] という。MSC 一般化記述を求める際、 $m \neq n$  ならば少ない節点を持つ組に不足分だけ仮節点を追加して考える。仮節点は注目している属性に関し、他のどの値に対しても共通の先祖が根 (root) となる値で規定されるとする。

Step2で注目している節点と同じ属性で規定されていない場合、MSC被覆あるいはCC被覆が求まらない場合あるいはstep3b.2で複数の $s_i, \dots, s_k$ が $t_j, \dots, t_l$ のいくつかの上位になる場合にはstep4の必要性が生じる。しかし、ひとたび対応づけが成され融合操作が繰り返されてゆくと、扱うべき部分木のサイズは指数関数的に小さくなるため、CNCの実行に要する手間は極度に小さくなる。融合操作は結果的に、知的分業のもとで獲得された知識を一つの大きな知識として協同で利用できるようにする方法論を提供している。また、CNCを動かさないうで作られる構造を見てみると、これはSとTの両者の大枠をとらえる索引としての働きを持つものとなっている。例を



(a)



(b)

図1 融合例

用いて説明しよう。図1(a)に二つの異なる色の集合をもとにCNCを用いて作った概念構造 colorA と colorB ならびに概念を規定する属性値の構造を示す。各色は色相、明度、彩度が定めるマンセル色空間上のデータとして

扱った [7]. 属性・属性値は、各色についての色相, 明度, 彩度あるいは明度 - 彩度をクラスタ化し, 各クラスタに名前をつけることで求めた. これらから得られた属性を, それぞれ, hue, lightness, vividness ならびに tone と呼ぶことにする.

図 1 からわかるように colorA, colorB 中の概念の関連性は属性を用いて, また属性値の関連性は共通基盤としての色相 (あるいは明度, 彩度, 明度 - 彩度空間上) でとらえられることになる. 構造 colorA と colorB において, 根節点の子節点を規定する属性はどちらも hue であることから, 色相空間を共通基盤として, hueA と hueB の融合構造を求める. これら属性値の構造の根節点の子節点は, それぞれ  $\langle 10R:4YR, 10YR:2GY \rangle$  と  $\langle 4YR, 2Y:4GY, 4G \rangle$  である. 前者に仮節点を組み入れ, MSC 被覆を求めると,  $10R:4YR$  と  $4YR, 10YR:2GY$  と  $2Y:4GY$ , 仮節点と  $4G$  が対応づけられ,  $\langle 10R:4YR, 10YR:4GY, 4G \rangle$  となる. これらは色相空間上で互いに別な区間を作ることから step3a の条件が成立する. Step3a.1 では hueB の節点  $4G$  を根とする部分木を求めるべき融合構造 hueAB の部分木として組み入れる. 次に, step3a.2 では  $10R:4YR$  が色相空間上で  $4YR$  の上位になることから  $10R:4YR$  を hueAB の節点とみなし,  $10R:4YR$  の子節点  $\langle 10R, 4YR \rangle$  と  $4YR$  について step2 から融合操作を繰り返せば,  $10R$  と  $4YR$  が hueAB の新たな節点となる.

ところで,  $10YR:2GY$  と  $2Y:4GY$  の対応づけについては, 色相空間上でどちらが上位になるともいえないので, MSC 被覆  $10YR:4GY$  を hueAB の節点とし,  $10YR:2GY$  と  $2Y:4GY$  の子節点を取り出して step2 から繰り返す. それぞれについての子節点  $\langle 10YR:2Y, 5Y:2GY \rangle$  と  $\langle 2Y:9Y, 2GY:4GY \rangle$  に対する MSC 被覆を求めても被覆の要素は色相空間上で互に関連があることから, step3a の条件が満たされない. CC 被覆は,  $10YR:2Y$  に  $2Y:9Y$  が,  $5Y:2GY$  に  $2Y:9Y$  と  $2GY:4GY$  が対応づけられ,  $\langle 10YR:9Y, 2Y:4GY \rangle$  となる. しかしこれらの要素も互に関連があることから step4 に進むが, 融合操作は共通基盤を求めている段階のため操作を終了する. 結果的に共通基盤 hueAB (図 1(b)) が求まり, colorA と colorB に対する融合に戻る.

属性 hueAB を用いると colorA および colorB の注目している節点はそれぞれ hueAB ( $10R:4YR$ ) と hueAB ( $10YR:4GY$ ) ならびに hueAB ( $4YR$ ), hueAB ( $10YR:4GY$ ), hueAB ( $4G$ ) で定められているとみなせる. この記述をもとに hueA, hueB に対する融合と同様の操作を行なうと, 図 1(b) のように hueAB ( $10YR:4YR$ ) で定められる節点を根とする部分木が求まる. Step3a.2 においては hueAB ( $10YR:4GY$ ) で規定される節点の子節点同士による対応づけは必要な共通基盤が得られなかったため, step4 によって colorA, colorB の hueAB ( $10YR:4GY$ ) で定められる節点を根とする部分木中の概念にあてはまる色の集まりについて CNC を動かす. 最終的に得られたそれら対象の集合から概念が形成され属性とその値の構造が求まる. これら一連の操作で得られた融合構造が図 1(b) の colorAB である.

### 3.4 対比

次に, 異なった構造 S と T にありながら互いに同じとみなしてよい (対比しているという) 節点同士を見つけることを考えよう. 融合操作でも一対一の対応づけで融合構造の節点を作った節点同士が対比していると考えられるが, 複数の  $s_i, \dots, s_k$  が  $t_j, \dots, t_l$  のいくつかの上位になる場合には step4 が実行され, 対応づけが成されなくなってしまう. 対比操作を行なう際には, step4 の代わりに, この多対多の関係を step3a.2 が実行できるよう一対一の対応づけに分解することを考える. ただし, step2 で注目している節点を規定する属性が違っていても step4 が実行されるが, ここでは構造 S (あるいは T) に注目して S (T) の属性に合うよう T (S) での属性を解釈し直す

ようにする。SあるいはTのいずれかの構造に注目することは、自身の構造を重視しながら他の構造中で同じとみなしてよい概念を見つけてゆくことに相当する。

#### [対比操作]

(step1,step2,step3) 上と同じ(ただし、融合構造は作らない。Step2で属性が違ってればT中の属性をS中の属性に合わせて解釈し直す。またstep3a.1で仮節点と対応づけられた節点ならびにstep3b.1でいずれとも対応づけられていない節点については無視する)。

(step4) 複数の  $s_i, \dots, s_k$  が複数の  $t_j, \dots, t_l$  と関係あるとすると、MSC 被覆を求め直し、そこでの可能な対応づけの各々について、step3a.2と同じ操作を繰り返す。

Step4では多対多の関係がMSC被覆を求め直すことで複数の可能な一対一の対応づけに分解される。対比操作によって作られた構造を用いることでSとT両構造の類似点と相違点を明確にすることができる。

例として図1のcolorAおよびcolorBを対比させてみよう。(対比の状況を図1(a)中、それぞれ○と△を付して示す)。融合操作でのstep1からstep3で作られた部分構造については、上で試みたのと同様にして対比が求められるため、colorA, colorB中それぞれhueA(10YR:2GY)とhueB(2Y:4GY)で定められる節点を根とした部分構造についての対比操作をみってみる。これら根節点の子節点を規定する属性について、その値の構造上での<10YR:2Y, 5Y:2GY>と<2Y:9Y, 2GY:4GY>の対応づけを求めるには、step4でMSC被覆を使う。すると10YR:2Yが2Y:9Y, 5Y:2GYが2GY:4GYに対応づけられる。Step3a.2では10YR:2Yと2Y:9Yそれぞれの子節点<10YR, 2Y>, <2Y, 9Y>を取り出しstep2から繰り返せば、そのCC被覆<10YR, 2Y, 9Y>より、2Yと2Yが対応づけられる。

5Y:2GYと2GY:4GYの対応づけについては2GY:4GYの子節点を取り出せないで操作を終了し、colorAとcolorBの対比操作に戻る。colorAとcolorB中のhueA(10YR:2Y)とhueB(2Y:9Y), hueA(2Y)とhueB(2Y)およびhueA(5Y:2GY)とhueB(2GY:4GY)で定められた節点同士がそれぞれ対比される。次にhueA(5Y:2GY)とhueB(2GY:4GY)で定められた節点の子節点に対して対比操作を行なう。これらの子節点はそれぞれ、hueA(5Y)とhueA(2GY)とならびにlightnessB(6.0)とlightnessB(7.0)で定められており、属性が違っていることから、colorB中の属性lightnessをcolorA中の属性hueに合わせて解釈し直すと、lightnessB(6.0), lightnessB(7.0)はそれぞれhueB(2GY), hueB(4GY)となる。ここでlightnessBの上側の構造を2GY, 4GYを節点とするhueの構造とみなし、対比操作を行なうと、<5Y, 2GY>と<2GY, 4GY>のCC被覆<5Y, 2GY, 4GY>を得る対応づけから、2GYと2GYが対応づけられ、このことより、colorA, B中のhueA(2GY)とlightnessB(6.0)(解釈し直すとhueB(2GY))で定められた節点の対比がなされる。

#### 4. 実験的考察

ここでの方法の有効性を調べるため計算機実験を行った。扱う対象としては、表1に示すような自然界にみられる色を取り上げ、上の例で示したような方法で共通基盤を求めながら融合ならびに対比操作を行った。融合操作に係わる実験では、30色と20色からなる色の集合を二つ(それぞれCol30, Col20と書く)考え、Col30についての構造は既に存在するとして、これにCol20から得た構造を融合することを試みた。ただし、これら集まりがどの程度同じ要素を含むかによって融合操作の効率が違ってくる。それゆえ最初の実験として、これらの色を色相空間上に置いたとき、Col30のうちの右端に位置するいくつかの色がCol20に現れるようCol30とCol20を設定して、そこでの同じ要素の割合の違いによる操作効率の比較を行った。Col20の色のうち75%, 50%あるいは25%がCol30の色と共通とした集まりをそれぞれCol20(75), Col20(50), Col20(25)とする。効率については、

MSC 被覆や CC 被覆を求める際に共通基盤を参照した回数, CNC を用いて対象のクラスタに記述を与える際に属性値の構造を参照した回数ならびにこれら参照と CNC の実行の際にクラスタを求めるのに要した時間 (要素数の 2 乗に比例する時間がかかる) で測った。結果を表 2 に示す<sup>2</sup>。この表からは Col20 に Col30 と同じ要素が増えると融合操作の時に CNC を動かせる必要性がでてくるため, 属性値の構造を参照する回数が増加し, 全体として実行時間も増加することがわかる。ただし, 同じ要素が少ない場合の融合は効率よく行なえる。このことは, 既存の構造にその構造と同じ要素を多く含む構造を融合させる場合は, 融合操作の複雑さが増すことを示している。

表 1 色データ

色名	色相	明度	彩度	色名	色相	明度	彩度
strong_pink	1.0R	6.5	9.5	vivid_yellowish_orange	8.0YR	7.0	14
pink1	2.5R	6.5	8.0	light_yellow	5.0Y	9.0	7
pink2	2.5R	7.0	7.0	deep_greenish_yellow	9.0Y	6.0	9.5
dull_red	4.0R	5.0	7.5	dull_yellow_green	2.0GY	6.0	5.5
bright_yellowish_red	7.0R	6.0	13.5	deep_yellow_green	4.0GY	5.5	10
strong_yellowish_pink	7.0R	7.0	10	soft_green	9.0GY	6.5	5
yellowish_pink	10.0R	8.0	8	vivid_yellowish_green	9.0GY	6.5	11.5
brown	4.0YR	4.5	6	bright_yellowish_green	9.0GY	7.5	10
deep_orange	4.0YR	5.0	11	vivid_green	4.0G	5.5	10.5
strong_orange	4.0YR	6.0	12	strong_green	4.0G	5.5	9

表 2 融合操作計算量 1

色集合	参照回数 (回)	実行時間 (sec)
Col20(75)	222	8.9
Col20(50)	180	4.9
Col20(25)	190	4.9

上での事柄をさらに詳しく調べるために, 二番目の実験として Col30 と Col20 が, 同じ要素の代わりに, マンセル色空間上で互いに近い位置にある要素を含んでいる割合によって操作効率がどのようになるかを調べた。この場合の Col30 と Col20 の設定の仕方として, Col20 の作る色相空間上の区間が Col30 のそれに含まれるようにし, 最初の実験と同様に Col20 の要素のうちの 75%, 50% あるいは 25% が Col30 の要素と共通になるようにした (3 種の Col20 の区別は上と同様 Col20(75) 等と書く)。すでにある Col30 の構造にこれらそれぞれから作られた構造を融合した場合の効率を表 3 に示す。比較のため Col30 と Col20(75) (あるいは Col20(50), Col20(25)) 全体に CNC を動かさせた場合についての結果を表 4 に示す。前者は入力の変化に応じて逐次的に概念を形成する場面に, また後者は入力対象を一括して概念を形成する場面に相当する。これらの表からは, 例えば Col20(50) の概念構造を生成しこれを Col30 の概念構造に融合するのに要する共通基盤の参照は, 二つを一括して処理する場合に比べて 60% の回数で済み, また, 実行時間についても 40% の時間で済んでいることがわかる。

表 3 融合操作計算量 2

色集合	参照回数 (回)	実行時間 (sec)
Col20(75)	294	14.3
Col20(50)	309	14.4
Col20(25)	464	26.31

表 4 一括計算量

色集合	参照回数 (回)	実行時間 (sec)
Col20(75)	474	35.04
Col20(50)	494	35.06
Col20(25)	518	35.09

表 3,4 からは, また, ある程度以上の同じ要素がある場合には, 一括的に概念構造を作り直すより効率は良いが,

<sup>2</sup>計測は東芝 Work Station AS4080/30MX 上で行った。



表5 対比操作計算量

色集合	参照回数(回)	step4 操作回数(回)
Col20(75)	45	1
Col20(50)	37	1
Col20(25)	37	2

同じ色の割合が低くなると、大きな部分構造についてCNCを用いた再構築が行われ、融合操作の効率は一括操作の場合と変わらなくなってしまうことも読み取れる。これは互いに似ているが同じではない要素を多く含むような二つの対象の集まりに対しては全体をひとまとめにして概念構造を作る方が効率がよいことを示している。このことから融合操作が良いか一括法が良いか入力に依存するともみれる。しかし、融合操作を用いれば再構築すべき構造の部分だけを見つけてCNCを働かせ、それ以外の部分については対応づけだけで済ませるということが自動的に行なわれるため、最初の実験結果を合わせて考えると全体的には、融合操作を利用する方がすぐれているといえる。

対比操作に係わる実験では、二番目の実験と同じ色の集合を用い、Col30についての構造にCol20から得た構造を対比させることを試みた。対比操作の効率として構造参照回数と対比操作step4での処理回数を表5に示す。これから与えられた色の集まりが違っていても構造参照回数は余り変わらないことがわかる。また対比操作では、多対多であった節点の対応づけをstep4で一対一の対応づけに分解するが、その頻度は対比の曖昧性の割合を表しているとみなせる。その傾向は同じ色の割合が低くなると、頻度が多くなることから、対象の違いの大きいところで形成された概念間で同じとみなしてよいものを見つけようとする、曖昧性の高い対比が行なわれるようになることがわかる。

最後にCol30とCol20(25)(あるいはCol20(50), Col20(75))の一括処理で作った構造と、それぞれから作られた構造の融合操作で作った構造との対比を行なって、対応づけの段階で一括処理によって作られた構造にあらわれる記述のうち融合操作で作った構造にあらわれない記述の数を調べてみた。Col30とCol20(25)からのものについては、一括処理で作った構造と同じであった。Col30とCol20(50)(あるいはCol20(75))からのものについては、一括処理で作った構造にあらわれる節点のうち約10%が融合操作で作られた構造中にあらわれなかったが、二つの構造間で曖昧な対比を生じさせる原因となるような多対多の節点の対応づけは行なわれなかった。このことを考慮に入れると、後者の場合でも一括処理と融合操作で作った二つの構造間には小さな差異がみられるものの、曖昧な対比が生じるほどの違いは存在しないといえる。

## 5. むすび

段階的な構造の参照ならば被覆にしたがう概念の対応づけにもとづき、それぞれ別な環境で形成された二つの概念構造から一つの融合された構造を求める方法ならびに両構造の概念の対比状況をみる方法について述べた。融合に関する考え方は、例えば時系列データの規則性を見つける場面[1]にも応用できる。これには一定の期間ごとに観察されたデータの概念構造を作りながら、前期間での構造との融合構造のうち現在のデータと関連がない部分を除いたものを結果とする。すると過去のデータの傾向を残しながら現在の規則性を捉えられるようになる。また、心理学的な考察では、我々が共通の問題を解くとき各人が各人なりに理解できる構造を作り上げて行っていると指摘されており[5]、対比に関する考え方は人間の相互理解の活動をモデル化するために利用できると考えられ

る。

ここでの融合操作では、大きな部分構造に対して CNC を働かせなければならない状況が生じた。今後の方向として対比操作によって融合できる部分を見い出しながら全体としてできるだけ CNC を働かせないように済ませる方法を定式化することが考えられる。

#### 参考文献

- [1] Dent, L., Boticario, J., McDemott, J., Mitchell, T., and Zabowski, D. : A personal learning Apprentice, in Proc.of the 10th National Conf. on Artificial Intell., pp.96-103(1992).
- [2] Gennari, J. H., Langley, P., and Fisher, D. : Models of incremental concept formation, Artificial Intell., Vol.40, pp.11-61(1989).
- [3] 半田剣一, 松原仁 : 不均質な領域を対象とした概念形成システム CAFE, 人工知能学会誌, Vol.7, No.6, pp.981-991 (1992).
- [4] 原口誠, 有川節夫 : 類推の定式化とその実現, 人工知能学会誌, Vol.1, No.1, pp.132-139 (1986).
- [5] 波多野諄余夫編 : 認知心理学講座 4, 東京大学出版会 (1982).
- [6] Hayes-Roth, F., and McDemott, J. : An interference matching technique for inducing abstractions, C.ACM, Vol. 21, No. 5, pp.401-411 (1978).
- [7] 伊藤哲郎 : 分類による概念形成, 人工知能学会誌, Vol.6, No.6, pp.881-890 (1991).
- [8] Dietterich, T. G., and Michalski, R. S. : A comparative review of selected methods for Learning from examples, in Michalski, R.S., Carbonell, J.G. and Mitchell, T.M. ed. Machine Learning, Morgan Kaufmann, Los Altos, CA(1983).
- [9] Michalski, R. S., and Kodratoff, Y. : Research in machine learning: Recent progress, classification of methods, and future directions, in Kodratoff, Y., and Michalski, R.S. ed. Machine Learning Vol III, Morgan Kaufmann, San Mateo, CA(1990).
- [10] 中村幸雄 : 情報処理 I, 共立出版 (1968).
- [11] Stepp, R. E., and Michalski, R. S. : Conceptual clustering of structured objects: A goal-oriented approach, Artificial Intell., Vol.28, pp.43-69 (1986).
- [12] 菅沼義昇 : 対象間の差異による学習と知識構造, 人工知能学会誌, Vol.7, No.1, pp.87-104 (1992).
- [13] Thompson, K., and Langley, P. : Concept Formation in Structured Domains, in Fisher Jr, D.H., Paz-zani, M.J., and Langley, P. ed. Concept Formation: Knowledge and Experience in Unsupervised Learning, Morgan Kaufmann, San Mateo, CA (1991).
- [14] Wrobel, S. : Towards a model of grounded concept formation, in Proc.of the 12th Intern.Conf. on Artificial Intell., pp.712-717 (1991).