

## フレームの一般化による特徴記述の生成と 知識利用型画像検索システムへの応用

森 敏昭 大川 剛直 馬場口 登 手塚 慶一  
大阪大学工学部

フレーム型知識表現による記述を一般化することにより、特徴記述を生成する手法を提案する。特徴記述とは、ある共通点を持ったインスタンスフレームが存在する場合に、そのクラスを規定する記述を指し、あるインスタンスフレームがそのクラスに属しているかどうかを判別することができる。特徴記述は、フレームの階層性に基づくスロット値の抽象化、値の変数化、スロットの削除などにより、各インスタンスフレームを一般化することにより得られる。また、過度の一般化による無意味な記述の生成を防ぐために、入力されるインスタンスフレームは同一クラスに属しているという制限を加えている。最後に、この手法の知識利用型画像検索システムの知識獲得に対する応用についても述べる。

### A Method for Generating Characteristic Description by Generalizing Frames and Its Application to Knowledge Based Image Retrieval System

Toshiaki Mori, Takenao Ohkawa, Noboru Babaguchi, Yoshikazu Tezuka

Faculty of Engineering, Osaka University  
2-1 Yamadaoka, Suita-shi, 565, Japan

This report presents a learning method to generate characteristic descriptions for frames. A characteristic description, defined as a description of a class that is formed by objects given as instance frames, can be obtained by generalizing each of instance frames. Frames can be generalized by comparing the generality between frames, that is a partial order relation. In addition, the constraint that all of instance frames should belong to the same class is introduced to suppress overgeneralizations. Finally, we show how to apply this method to knowledge acquisition for a knowledge based image retrieval system.

## 1. まえがき

現在、知識情報処理システムの知識ベースの設計には多大な労力と時間を要し、システム開発におけるボトルネックとなっている。このボトルネック解消のため、機械学習による知識獲得の自動化に対する期待は大きい。とりわけ例題に基づく一般的規則の獲得を目指した帰納的学習に関する研究の歴史は古く、これまで多くの研究成果が得られている[Cohen82, Michalski83, 86]。

一般に帰納的学習とは、与えられた複数の入力言明から、それら全てを説明する一般的な規則、法則を帰納的言明として発見する過程であると定義できる。即ち、得られる帰納的言明は、各入力言明に対する一般的な記述となる。従って、各入力言明の一般化により、帰納的言明が生成される。このように、帰納的学習においては、一般化の概念が重要となる。

一般化を行うにあたり、論議領域の背景に存在する抽象-具体関係を積極的に利用することが、有効であると考えられる。これは、より抽象的なものが、より具体的なものに比べて一般的であるといえるからである。これまで提案されている学習法においては、述語論理あるいはその変形が、言明の記述言語としてしばしば用いられている。しかし、述語論理を記述言語とした場合、抽象-具体関係は陽には示されない。そこで、述語あるいは概念間の抽象-具体関係を述語の階層木として外部に定義するといった手法も提案されている[Kodratoff86]。

一方、概念の抽象度に基づいた階層性を有する記述言語としてフレーム型知識表現がある。フレームにはクラス、インスタンスの概念があり、抽象-具体関係が明示的に規定されている。そこで、本稿ではフレームを学習アルゴリズムにおける記述言語として採用し、その一般化手続きを提案する。

さて、これまで様々な帰納的学習法が提案されているが、一部の対象依存のものを除き、汎用的アルゴリズムがそのほとんどである。このため、知識獲得に対する実際の利用を考えた場合、入力される例に対する制限があまり存在しないため、限られた数の例から意味のある知識を獲得するためには、与える例について考慮し、適切なものを選ばなければならない。

そこで、ここでは入力言明に対して、次のような制限を加えることにより、より実用的な学習を目指す。即ち、

一つの入力言明は一つの対象に関する記述であり、入力言明として記述された対象は全て同一クラスに属するものと仮定する。この仮定は、学習アルゴリズムの観点からは、非常に強い制約となるが、入力言明の集合は同一クラスに属したフレームの集合となり、各フレームの使用スロットの間に高い類似性が期待できる。これにより、情報喪失の可能性を伴う条件削除を抑えた一般化が可能になると考えられる。

本稿では、フレームの一般化の基準となるフレームの一般性について定義し、フレームの一般化による特徴記述生成の手続きについて述べる。また、現在開発を進めている知識利用型画像検索システムのための知識獲得に対する本手続きの応用について言及する。

## 2. 事例記述と特徴記述

帰納的学習とは、与えられたいくつかの例を基に、それらの例を全て説明する一般的な法則を導くものである。帰納的学習アルゴリズムを考える上で、まず最初に、例として与えられる入力言明、学習の結果得られる帰納的言明の意義を決定しなければならない。そこで、以下のように各言明の意義を定める。

- (1) 一つの入力言明は、ある一つの対象に関する記述である。これを事例記述と呼ぶ。
- (2) 学習の結果得られる帰納的言明は、各事例記述により定められた一つのクラスに依存した記述となる。即ち、入力言明に対応する対象によって形成されるクラスを規定する記述となる。逆に、これはある対象がそのクラスに属しているかないかを判別する記述と考えることもできる。この記述をそのクラスの特徴記述と呼ぶ。

事例記述の例を図1(a)、その特徴記述を(b)に示す。

```
(HOUSE1                                (HOUSE2
(AKO (VALUE (HOUSE)))                   (AKO (VALUE (HOUSE)))
(COLOR (VALUE (WHITE)))                 (COLOR (VALUE (BROWN)))
(STORY (VALUE (2))))                   (STORY (VALUE (2))))

                                           (a)

(x
(AKO (VALUE (HOUSE)))
(COLOR (VALUE (WHITE BROWN)))
(STORY (VALUE (2))))

                                           (b)
```

図1 事例記述と特徴記述の例

さらに、事例記述に対して、次のような制約を設ける。即ち、事例記述には対象のクラスが明示され、アルゴリズムに対して入力される各対象は全て同一のクラスに属するものとする。なお、ここでいうクラスは、前の(2)における学習の結果形成されたクラスとは異なり、予め論議領域内で定義されたものである。

### 3. フレームの一般性

#### 3.1 フレームの準等価性

二つのフレームが存在し、それらのフレーム名及びスロットが全て一致しているとき、それらは等価であるとみなすことができる。ところが、各スロットが形式的には一致していないにもかかわらず、内容的には同一の情報を表現したフレームとして取り扱うことが可能な場合もある。ここではそのようなフレームの準等価性を定義する。

まず、記述内容に大きな変化を与えない操作として次の三つの操作を考える。

##### (1) 暗黙値の実体化

暗黙値の実体化とは、ファセットがVALUEであるスロットが存在しないとき、暗黙値が存在するならばファセットをDEFAULTからVALUEに書き換えて、その値をスロットの値として用いることをいう。図2にその例を示す。

```
(MAN1 (AKO (VALUE (MAN)))
      (AGE (DEFAULT (20))))
      ↓
(MAN1 (AKO (VALUE (MAN)))
      (AGE (VALUE (20))))
```

図2 暗黙値の実体化の例

##### (2) 継承値の実体化

フレームは概念の抽象度に基づいて階層化されており、下位フレームにスロットが存在しない場合にも、上位のフレームにそのスロットが存在すれば、値を継承して下位フレームの値として扱うことができる。このような値を継承値という。

継承値の実体化とは、図3のように上位フレームの持つスロットを、その下位に属するインスタンスフレームにも記述することをいう。

```
(BOOK1 (AKO (VALUE (BOOK))))
(BOOK (MADE_OF (VALUE (PAPER))))
      ↓
(BOOK1 (AKO (VALUE (BOOK)))
      (MADE_OF (VALUE (PAPER))))
```

図3 継承値の実体化の例

##### (3) 書換え規則の適用

ある物体が机の上に置いてあるとき、その物体は机の上方にあるということもできる。また、ある人が木の右側にいるとき、その人は木の隣にいるということもできる。このように、一つの関係を様々な言い方に直すことができる。このような言い換えは、フレームにおいてはスロットを異なるスロットに書き換えることに相当する。

書換え規則の適用とは、図4のようにあらかじめ定められた規則に従ってスロットを書換え、生成されたスロットを新たにフレームに追加することをいう。

図5にその書換え規則を適用した例を示す。

```
ON → ON ∧ ABOVE
RIGHT_OF → RIGHT_OF ∧ NEXT
```

図4 書換え規則の例

```
(CUP1 (AKO (VALUE (CUP)))
      (ON (VALUE (TABLE1))))
      ↓
(CUP1 (AKO (VALUE (CUP)))
      (ON (VALUE (TABLE1)))
      (ABOVE (VALUE (TABLE1))))
```

図5 書換え規則の適用例

以上の(1)~(3)の各操作を用い、次のようにフレームの準等価性を定義する。

##### [定義1] (フレームの準等価性)

フレームFに対して暗黙値の実体化、継承値の実体化、書換え規則の適用の各操作を有限回実行した結果をF'とした時、F'をFに準等価なフレームであるという。

#### 3.2 一般性の定義

スロット値及びフレーム名に変数の使用を許したフレームFならびにF内の変数に対する値の束縛関係を表した代入σを用いて作られるフレームを〈F, σ〉と表現する。ここで、[Plotkin70]等の論理式における一般性に従うと、フレームの一般性は次のように定義される。

[定義2] (形式的-一般性)

フレーム F1, F2 に対して,

$$F2 = \langle F1, \sigma \rangle \quad \dots (1)$$

を満足する代入  $\sigma$  が存在するとき, F1 は F2 より形式的に一般的であるという.

<p>《フレーム F1》 (a (AKO (VALUE (DOG))) (COLOR (VALUE (b))) (SIZE (VALUE (c))))</p>	<p>《フレーム F2》 (DOG1 (AKO (VALUE (DOG))) (COLOR (VALUE (BROWN))) (SIZE (VALUE (MEDIUM)))</p>
---	--

(但し, a, b, c は変数)

図6 フレームの形式的-一般性の例

しかし, フレームは階層構造を持ち, 階層の上位-下位間に抽象-具体関係が成り立つが, これを形式的-一般性により説明することはできない. 例えば, 図6は定義1の一般性に従い F1 は F2 より一般的といえるが, 図7(a)に示すフレーム F1, F2 間の抽象-具体関係を定めることはできない. そこで, 階層構造を考慮したフレームの一般性について定義する.

フレーム F に準等価なフレームから, 任意のスロットを削除して得られるフレームを  $F^*$  とする. また,  $value(\sigma, x)$  を代入  $\sigma$  における変数  $x$  の値とすると, 代入  $\sigma 1, \sigma 2$  に出現する全ての変数  $x$  について,

- (1)  $value(\sigma 1, x)$  が  $value(\sigma 2, x)$  と等値
- (2)  $value(\sigma 1, x)$  が  $value(\sigma 2, x)$  の上位クラス
- (3)  $value(\sigma 1, x)$  が  $value(\sigma 2, x)$  を包含
- (4)  $value(\sigma 1, x)$  が変数

のいずれかが成立するとき,  $\sigma 1 \succ \sigma 2$  と記す. これらを用いてフレームの一般性を次のように定義する.

[定義3] (本質的-一般性)

フレーム F1, F2 に対して,

$$\left. \begin{array}{l} F1 = \langle F, \sigma 1 \rangle \\ F2^* = \langle F, \sigma 2 \rangle \\ \sigma 1 \succ \sigma 2 \end{array} \right\} \dots (2)$$

を満足するフレーム F ならびに代入  $\sigma 1, \sigma 2$  が存在するとき, F1 は F2 より本質的に一般的であるという.

例えば, 図7(a)に示すフレーム F1, F2 に対しては(2)式を満たすフレーム F, 代入  $\sigma 1, \sigma 2$  が同図(b)のように存在するため, F1 は F2 よりも本質的に一般的であるといえる.

《フレーム F1》

(a (AKO (VALUE (ANIMAL)))  
(COLOR (VALUE (b)))  
(SIZE (VALUE (SMALL))))

《フレーム F2》

(BIRD1 (AKO (VALUE (BIRD)))  
(COLOR (VALUE (WHITE)))  
(SIZE (VALUE (SMALL)))  
(SHAPE (VALUE (GLOVE))))

(但し, a, b は変数)

(a)

《フレーム F》

(w (AKO (VALUE (x)))  
(COLOR (VALUE (y)))  
(SIZE (VALUE (z))))

$$\sigma 1 = \{w \leftarrow a, \quad \sigma 2 = \{w \leftarrow BIRD1, \\ x \leftarrow ANIMAL, \quad x \leftarrow BIRD, \\ y \leftarrow b, \quad y \leftarrow WHITE, \\ z \leftarrow SMALL\} \quad z \leftarrow SMALL\}$$

(b)

図7 フレームの本質的-一般性の例

#### 4. 特徴記述生成アルゴリズム

3. 2で定義したフレームの本質的-一般性に基づく特徴記述生成アルゴリズムを示す. 特徴記述の生成はスロット間の対応をとりながら行われ, 全ての事例記述に出現しないスロットは削除される. しかし, 条件削除を安易に適用することは過度の一般化につながるため, スロットに書換え規則を適用することによりスロットの削除を最小限に抑える. また, 一般化を行う際にフレームの階層性を利用する.

##### 4. 1 フレームの単一関係と多重関係

フレームはそれぞれが独立したものではなく, 各々は相互に何らかの関係の有していると考えられる. 即ち, 他のフレーム名をスロット値とすることができ, そのフレームとはスロットにより表される関係を持つ. スロットの値が一つであれば対応は一对一に定まるが, 複数の場合は一对多となりフレーム間の関係は複雑になるため, 特徴記述を生成する際に一对一の場合とは, その対処の方法が異なると考えられる.

ここでは, スロット値の数に注目し, フレーム間の関係について次のように分類する.

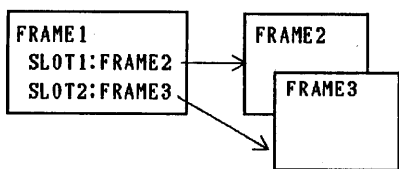


図8 単一関係を持つフレーム

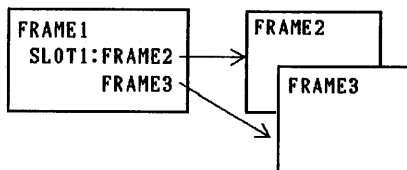


図9 多重関係を持つフレーム

(1) 単一関係を持つフレーム (図8)

他のフレームとの関係を記述する全てのスロットが単一の値のみを持つようなフレームをいう。

(2) 多重関係を持つフレーム (図9)

他のフレームとの関係を記述するスロットの一つ以上が複数の値を持つようなフレームをいう。

4.2 単一関係を持つフレームの一般化

本節では、単一関係を持つフレームのみを対象とし、その一般化アルゴリズムを与える。即ち、単一関係を持つ複数のインスタンスフレームを、入力される事例記述と考え、その一般化により、特徴記述を生成する方法について述べる。

本アルゴリズムにおいて、代入  $\sigma_i$  ( $i=1, \dots, n$ ) に対し、

$$\sigma \succ \sigma_i \quad (i=1, \dots, n) \quad \dots (3)$$

の成立する代入  $\sigma$  を、最も制限された形で求める必要がある。そこで、まず最初に、最小共通代入という概念を導入し、これを次のように定義する。

[定義4] (最小共通代入)

$n$  個の代入  $\sigma_i$  ( $i=1, \dots, n$ ) に出現する全ての変数  $x$  について、以下の規則に従い、新たな束縛を考える。

(1)  $\text{value}(\sigma_i, x)$  がクラスフレームの場合

上位-下位の関係を示したリンクを用いてフレームの階層を上り、全ての  $\text{value}(\sigma_i, x)$  の上位クラスの中で同一のクラスを探す。得られたクラスのうち、最も下位のもので  $x$  を束縛する。

(2)  $\text{value}(\sigma_i, x)$  がインスタンスフレームの場合

$x$  の束縛を解放する。

(3) (1), (2) 以外の場合

$$x \text{ を } \bigvee_{i=1}^n \text{value}(\sigma_i, x) \text{ で束縛する。}$$

このようにして作成された束縛関係により新たに得られた代入  $\sigma$  を代入  $\sigma_i$  の最小共通代入という。

最小共通代入  $\sigma$  は、(3) 式を満足する最も制限された  $\sigma$  といえる。次に、単一関係を持つフレームの一般化アルゴリズムを示す。

[単一関係を持つフレームの一般化アルゴリズム]

ステップ1

入力された各事例記述、即ちフレームの中から、値が一般化変数でないスロットの一つを選択する。ここで、一般化変数とは本アルゴリズムにおける局所的な変数であり、ステップ2において生成されるものである。値が一般化変数でないスロットがなければ、ステップ3へ。

ステップ2

新しい一般化変数を生成し、ステップ1で選択されたスロットの値と置き換える。このとき、生成された変数にはスロット値が代入されるが、その代入は各事例記述ごとに記憶する。

ステップ3

一般化変数を持つスロット、即ちスロットと変数の対が全ての事例記述に出現するかどうかをチェックする。

ステップ4

スロットが出現しない事例記述について、3.1の(1)~(3)の操作によりスロットの生成を試みる。スロットが生成できなかったならば、そのスロット自身に書換え規則を適用し、ステップ4を初めから繰り返す。但し、スロットの書換え規則は同じスロットには一度しか適用しないものとする。スロットの生成も、書換え規則の適用もできないならば、そのスロットを削除する。

#### ステップ5

各事例記述において、一般化変数に置き換えられていないスロット値が存在するならば、再びステップ1へ。存在しないならば、各事例記述のフレーム名を一般化変数に置き換える。

#### ステップ6

この段階で各フレームは形式的に同一構造であり、このフレームをFとする。ここで、各事例記述毎に記憶した代入 $\sigma_i$ の最小共通代入 $\sigma$ を前の定義に従って求め、

$$Fg = \langle F, \sigma \rangle$$

を得る。Fgを返し、本アルゴリズムを終了する。

以上のアルゴリズムにより求められたフレームFgは、事例記述として与えられたフレームFi各々の一般形となっており、事例記述により形成されるクラスを規定する特徴記述といえることができる。

なお、実際には、各 $\sigma_i$ において、 $\text{value}(\sigma_i, x)$ がインスタンスフレーム名である場合があり、その場合それらについても一般化する必要がある。そこで、各インスタンスフレームを新たな事例記述と考え、本アルゴリズムを繰り返し特徴記述を再帰的に生成する。生成された特徴記述は逐次保存し、最終的には、それら全てが与えられた事例記述に対する特徴記述となる。

#### 4.3 多重関係を持つフレームの一般化

4.2では、単一関係を持つフレーム、即ちフレーム間の関係を記述する全てのスロットが、同時に二つ以上のインスタンスフレームをスロット値として持たないフレームの一般化について述べた。このような場合は、アルゴリズムを再帰的に適用することによりフレームを一般化することができた。しかし、人が傘と鞆の二つを持っているなど、一つのスロットの値として複数のインスタンスフレーム名をとる場合には、再帰的にアルゴリズムを適用する際に4.2と同様の手順で行うことができない。そこでこれに対応するために4.2で述べたアルゴリズムをさらに拡張し、複数のインスタンスフレームを一つの事例記述として扱うことが可能なフレームの一般化アルゴリズムについて述べる。

このアルゴリズムでは、まず事例記述を二つ選び、事

例記述間でインスタンスフレームを一つずつ組合せて一般化を行う。ここで複数の一般化された記述が生成されるが、その中には過度の一般化による無意味な記述も含まれるため、そのような記述を削除する。そして、残った記述を新しい事例記述として、同様に一般化を繰り返す。このように、生成された記述を順次削除することにより効率化を図っている。

[多重関係を持つフレームの一般化アルゴリズム]

#### ステップ1～5

4.2におけるアルゴリズムと同様である。

但しステップ4で、(1)～(3)の操作は事例記述の全てのフレームに適用し、どのフレームにも一般化変数をもつスロットが生成できなかった場合にのみスロットに対する書換えを行う。

#### ステップ6

フレーム名及びスロット値が変数で、一つの事例記述に存在する全てのスロットを含むフレームFを作成する。

#### ステップ7

任意に二つの事例記述を選択する。選択された各事例記述の中から一つずつフレームを選び、それらに伴う代入の最小共通代入を求める。但し、値がインスタンスフレーム名の場合はそれをリストにしたものを値とする。また、変数に代入される値が一方にない場合は、その変数の束縛を解放する。最小共通代入が求められなかった場合は、変数を値とする。得られた代入を、“SOLUTION”に追加する。

#### ステップ8

ステップ7で選択した事例記述から別のフレームの組合せを選び、それらから同様に新たな代入 $\sigma_{NEW}$ を求める。得られた代入 $\sigma_{NEW}$ と“SOLUTION”内の代入 $\sigma_i$ との間に、フレームの一般性の定義で用いた代入間関係が成立するかどうかを調べる。

この結果、

・  $\sigma_{NEW} \gg \sigma_i'$ 、 $\sigma_i' \subseteq \sigma_i$ の成立する $\sigma_i'$ が存在する場合

$\sigma_{NEW}$ を削除

・  $\sigma_i \supset \sigma_{NEW}$ ,  $\sigma_{NEW} \subseteq \sigma_{NEW}$ の成立する

$\sigma_{NEW}$ が存在する場合

$\sigma_i$ を削除し,  $\sigma_{NEW}$ を" SOLUTION"に追加

・ それ以外の場合

$\sigma_{NEW}$ を" SOLUTION"に追加

に従い, " SOLUTION"を更新する.

#### ステップ9

ステップ7で選んだ事例記述の全ての組合せについてステップ8を行う. その結果得られた" SOLUTION"内の各代入 $\sigma_i$ とステップ6で作成したフレームFにより作られるフレーム  $\langle F, \sigma_i \rangle$ を新しい事例記述と考え, 各 $\sigma_i$ を  $\langle F, \sigma_i \rangle$ に対する代入として再びステップ7から繰り返す. 事例記述がなくなればステップ10へ.

#### ステップ10

最終的に得られた" SOLUTION"内の各代入 $\sigma_i$ とFから作られる  $\langle F, \sigma_i \rangle$ を結果として返す. 但し, " SOLUTION"内の代入の中で, ある変数がインスタンスフレーム名のリストに束縛されているならば, 各インスタンスフレームを事例記述として本アルゴリズムを繰り返す.

### 5. 知識利用型画像検索システム

現在我々は知識利用型画像検索システムの開発を進めている[大川88, 89a]. 本節では, 提案した特徴記述の生成手続きを検索システムのための知識獲得に応用する方法について述べる[大川89b].

#### 5.1 画像検索システムの概要

人間が画像を見て何らかの情報を得る際には, その画像に描かれている具象的な情報だけではなく, それを基に解釈することにより, より抽象的な意味情報を得ていると考えられる. 知識利用型画像検索システムは, そのような意味情報を検索要求として受け取り, 蓄積された大量の画像の中から要求に合致した画像を検索するものである.

検索の対象となる各画像は, 画像の表象情報, 即ち画像の表層的な特徴や画像に存在する対象物の特徴を表したフレーム形式の画像記述として蓄積されている. 画像記述は次の2種類に分類して記述されている.

#### (1) イメージフレーム

画像の全体的な特徴, 及び画像を構成するオブジェクト名を記述するフレームである. 一枚の画像につき一つのイメージフレームを対応させて設ける.

#### (2) オブジェクトフレーム

画像内に存在する個々のオブジェクトの特徴を記述するフレームである. オブジェクトフレームはさらに, 個々のオブジェクト固有の特徴を記述するインスタンスフレームと, オブジェクトに関する一般的な概念を記述するクラスフレームに分類される.

意味情報に基づく検索は, 検索要求として受け取った意味情報を表象情報に変換することにより行われる. この変換を行う際に, 画像解釈知識と呼ぶルール形式の知識が用いられる. 画像解釈知識は, 具体的な事象をより抽象的なレベルへ変換するが, ここではこのルールを後向きに適用することにより検索要求の抽象レベルの低下を図り, 最終的に画像記述レベルの記述に変換して, これを満足する画像を検索する.

#### 5.2 画像解釈知識の獲得

画像解釈知識はルール形式で表現されており, この知識の質と量がシステムの性能を左右する. しかし, 人間が画像解釈の際に用いる知識をルール形式で抽出するのは容易ではない. さらに, あらゆる知識をシステム構築時に予め蓄積することは事実上不可能であり, 検索利用時に逐次知識を追加する機能が要求される. そこでこのシステムでは, 検索システムの利用者が自分の要求に適合する画像, 即ちある観点から一つ概念を有していると考えられる画像を複数枚選択し, これを基にシステムがルールを生成することにより知識を獲得する. ルールは, 次の(1), (2)の手順に従い, 生成する.

(1) 概念の正の例画像と負の例画像を利用者が指定し, それらに対応する画像記述を基に概念記述を帰納的に生成する.

(2) 生成された概念記述を前件に, またその概念自身を後件に持つルールを生成する.

ここで, 概念の正の例画像とは, その概念を満足する画像のことを言い, 逆に負の例画像とは, その概念を満足しない画像, 即ちその概念の例とならない画像のこと

を意味する。ルール生成の手順を図10に示す。このようにルールの生成は概念記述の生成に帰着される。

複数の画像から概念記述を生成するためには、まず最初に各画像からその概念にとって必要不可欠なオブジェクトを選択する必要がある。それは、画像内に存在する全てのオブジェクトが画像の意味を決定付けているとは限らないからである。さらに、画像の概念はそれらのオブジェクトの特徴に依存していると考えられるので、選択されたオブジェクトの特徴記述を生成する必要があり、提案アルゴリズムで得られる特徴記述を基に概念記述を生成する。

## 6. 生成例

単一関係を持つインスタンスフレームの特徴記述の生成例を図11、多重関係を持つインスタンスフレームの特徴記述の生成例を図12に示す。図11、図12の(a)は入力フレームで、図11の(b)はMAN1とMAN2の特徴記述、図12の(b)はDESK1とTABLE2とDESK3の特徴記述である。また、図12で次の2つのスロット書換え規則を用いた。

ON → ON ∧ ABOVE  
BENEATH → BENEATH ∧ BELOW

これを用いて、提案したアルゴリズムを評価する。フレームの持つ階層性、継承機能の利用、書換え規則の適用を積極的に施すことにより、不用意なスロットの削除を避け、過度の一般化を防ぐことが可能になった。例えば、図11のBOOK1とCUP2のMADE\_OFスロットの値として、BOOK1がBOOKフレームの値を継承することにより値PAPERが得られた。また、図12のTABLE2とDESK3のBENEATHスロットを書き換えることにより、DESK1のBELOWスロットの値の削除を防ぐことができた。また、多重関係を持つフレームの一般化を行う際に一般化を行うフレームの組合せを減らすことが可能となった。例えば、図12の3つのインスタンスフレームのBENEATHスロットの値について一般化を行う場合、単純に組み合わせると8通りの特徴記述を生成しなければならない。本手法では、DESK1とTABLE2の一般化の結果、(DISH1 GLASS2)の組合せによって生成された代入が削除されるため、これが6通りに削減された。これは事例記述の数が増えると、さらに顕著になると考えられる。

このようにして得られた特徴記述が、概念記述内に組

み込まれる形で、画像検索システムにおけるルールの前件部を構成する。

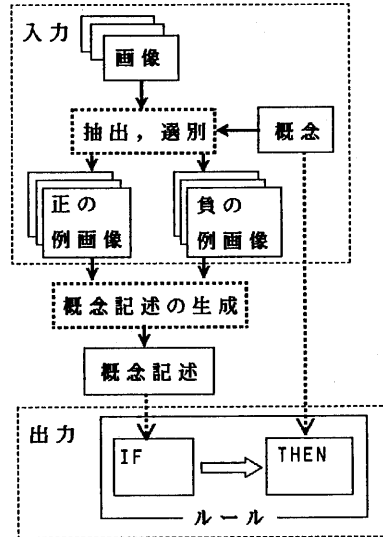


図10 ルールの生成

```
(MAN1
(FRAMETYPE (VALUE (INSTANCE)))
(AKO (VALUE (MAN)))
(HAVE (VALUE (BOOK1))))

(BOOK1
(FRAMETYPE (VALUE (INSTANCE)))
(AKO (VALUE (BOOK))))

(BOOK
(FRAMETYPE (VALUE (CLASS)))
(AKO (VALUE (ARTIFACT)))
(MADE_OF (VALUE (PAPER))))

(MAN2
(FRAMETYPE (VALUE (INSTANCE)))
(AKO (VALUE (MAN)))
(HAVE (VALUE (CUP2))))

(CUP2
(FRAMETYPE (VALUE (INSTANCE)))
(AKO (VALUE (CUP)))
(MADE_OF (VALUE (PAPER))))

(CUP
(FRAMETYPE (VALUE (CLASS)))
(AKO (VALUE (VESSEL))))

(VESSEL
(FRAMETYPE (VALUE (CLASS)))
(AKO (VALUE (ARTIFACT))))

(a)

(x
(AKO (VALUE (MAN)))
(HAVE (VALUE (a))))

(a
(AKO (VALUE (ARTIFACT)))
(MADE_OF (VALUE (PAPER))))

(b)
```

図11 単一関係をもつフレームの特徴記述生成例



<p>(DESK1 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (DESK))) (COLOR (VALUE (GRAY))) (MADE_OF (VALUE (STEEL))) (BENEATH (VALUE (BOOK1)(DISH1))) (BELOW (VALUE (CUP1))))</p>	<p>(BOOK1 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (BOOK))) (ON (VALUE (DESK1))) (COLOR (VALUE (BLUE))) (LEFT_OF (VALUE (CUP1)(DISH1))))</p>	
<p>(CUP1 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (CUP))) (ON (VALUE (DISH1))) (ABOVE (VALUE (DESK1))) (COLOR (VALUE (WHITE))) (MADE_OF (VALUE (POTTERY))) (FILLED_WITH (VALUE (COFFEE))) (RIGHT_OF (VALUE (BOOK1))))</p>	<p>(DISH1 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (DISH))) (ON (VALUE (DESK1))) (BENEATH (VALUE (DISH1))) (COLOR (VALUE (WHITE))) (MADE_OF (VALUE (POTTERY))) (RIGHT_OF (VALUE (BOOK1))))</p>	
<p>(TABLE2 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (TABLE))) (BENEATH (VALUE (GLASS2)(DISH2))) (COLOR (VALUE (BROWN))) (MADE_OF (VALUE (WOOD))))</p>	<p>(GLASS2 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (GLASS))) (MADE_OF (VALUE (GLASS))) (FILLED_WITH (VALUE (WATER))) (LEFT_OF (VALUE (DISH2))) (ON (VALUE (TABLE2))))</p>	<p>(DISH2 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (DISH))) (COLOR (VALUE (IVORY))) (MADE_OF (VALUE (POTTERY))) (RIGHT_OF (VALUE (GLASS2))) (ON (VALUE (TABLE2))))</p>
<p>(DESK3 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (DESK))) (COLOR (VALUE (GRAY))) (MADE_OF (VALUE (STEEL))) (BENEATH (VALUE (CUP3)(CLOCK3))))</p>	<p>(CUP3 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (CUP))) (COLOR (VALUE (WHITE))) (MADE_OF (VALUE (POTTERY))) (FILLED_WITH (VALUE (COFFEE))) (ON (VALUE (DESK3))) (RIGHT_OF (VALUE (CLOCK3))))</p>	<p>(CLOCK3 (FRAMETYPE (VALUE (INSTANCE))) (AKO (VALUE (CLOCK))) (COLOR (VALUE (YELLOW))) (SHAPE (VALUE (BOX_SHAPED))) (ON (VALUE (DESK3))) (LEFT_OF (VALUE (CUP3))))</p>
<p>(TABLE (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (STAND))))</p> <p>(DESK (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (STAND))))</p> <p>(WATER (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (DRINK))))</p> <p>(COFFEE (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (DRINK))))</p>	<p>(CUP (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (VESSEL))))</p> <p>(DISH (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (VESSEL))))</p> <p>(GLASS (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (VESSEL))))</p>	<p>(BOOK (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (ARTIFACT))))</p> <p>(CLOCK (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (ARTIFACT))))</p> <p>(STAND (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (ARTIFACT))))</p> <p>(VESSEL (FRAMETYPE (VALUE (CLASS))) (AKO (VALUE (ARTIFACT))))</p>

( a )

<p>(x (AKO (VALUE (STAND))) (COLOR (VALUE (GRAY BROWN))) (MADE_OF (VALUE (STEEL WOOD))) (BENEATH (VALUE (a0)(a1)(a2))) (BELOW (VALUE (b0)(b1))))</p>	<p>(a0 (AKO (VALUE (ARTIFACT))) (ON (VALUE (x))) (LEFT_OF (VALUE (a1)(b1))))</p>	<p>(a1 (AKO (VALUE (VESSEL))) (COLOR (VALUE (WHITE IVORY))) (MADE_OF (VALUE (POTTERY))) (ON (VALUE (x))) (RIGHT_OF (VALUE (a0))))</p>	<p>(a2 (AKO (VALUE (ARTIFACT))) (ON (VALUE (x))) (COLOR (VALUE (BLUE IVORY YELLOW))))</p>
<p>(b0 (AKO (VALUE (VESSEL))) (MADE_OF (VALUE (POTTERY GLASS))) (FILLED_WITH (VALUE (DRINK))) (ON (VALUE (c0))) (ABOVE (VALUE (x))))</p>	<p>(b1 (AKO (VALUE (VESSEL))) (COLOR (VALUE (WHITE IVORY))) (MADE_OF (VALUE (POTTERY))) (ON (VALUE (c0))) (ABOVE (VALUE (x))) (RIGHT_OF (VALUE (a0))))</p>	<p>(c0 (AKO (VALUE (ARTIFACT))) (COLOR (VALUE (WHITE BROWN GRAY))) (MADE_OF (VALUE (POTTERY WOOD STEEL))) (BENEATH (VALUE (b0)(b1))))</p>	

( b )

図 1 2 多重関係をもつフレームの特徴記述生成例

## 7. まとめ

本稿ではフレームの一般性を定義し、それに基づくフレームの一般化による特徴記述生成アルゴリズムを提案した。このアルゴリズムにより、あるクラスに属する複数のインスタンスフレームに対して、そのクラスを規定する特徴記述を生成することができた。

今後の課題として、多重関係を持つフレームの再帰時において複数フレーム間の一般化を行う際に、同一クラスに属するインスタンスフレームの一般化を優先させることによる効率の改善、また、否定等の導入による表現力の拡張があげられる。

## 参考文献

[Cohen82] Cohen, P.R. and Feigenbaum, E.A.:

“The Hand-book of Artificial Intelligence Vol.3”,  
William Kaufmann (1982).

[Michalski83,86] Michalski, R.S., Carbonell, J.G.  
and Mitchell, T.M.(Eds.):

“Machine learning Vol. I, II”, Morgan Kaufmann  
(1983,1986).

[Kodratoff86] Kodratoff, Y. and Ganascia, J.:

“Improving The Generalization Step in Learning”,  
Machine Learning vol11, pp.215-244 (1986).

[Plotkin70] Plotkin, G.D.:

“A Note on Inductive Generalization”,  
Machine Intelligence 5, pp.153-163, Edinburgh Uni-  
versity Press(1970).

[大川88] 大川 剛直,馬場口 登,手塚 慶一:

“知識利用型画像検索システムの構成”,  
情報処理学会知識工学と人工知能研究会報告, A156-9  
(1988).

[大川89a] 大川 剛直,小泉 和彦,馬場口 登,手塚 慶一:

“知識利用型画像検索システムについて”,  
電子情報通信学会1989年春季全国大会論文集, SD-7-2  
(1989).

[大川89b] 大川 剛直,森 敏昭,馬場口 登,手塚 慶一:

“知識利用型画像検索システムのための知識獲得”,  
電子情報通信学会1989年春季全国大会論文集, D-408  
(1989).