人 工 知 能 75－9
（1991. 3. 7）

事例の修正結果に基づく事例ベースの洗練化

前田 茂

（財）新世代コンピュータ技術開発機構

事例ベース推論 (Case-Based Reasoning: CBR) の性能向上のためには，類似事例の検索の効率化は，特に蓄積される事例数の増加にともない重要な問題となる。本論文では，抽象度によって階層化された事例の記憶構造を前提とし，事例の照合結果および修正結果を用いて段階的に領域知識，事例の記憶構造を修正することにより，事例ベース推論の性能向上をはかる手法を提案する。事例の記憶構造の修正は，4種類のオペレーターを定義し，与えられた問題と類似事例との照合結果および事例の修正の成功，失敗の結果の組み合わせの各場合によって適切なオペレーターの適用により行う。これにより，事例の蓄積が進むにしたがい，より適切で効率の良い検索を行うことができる。

Refinement of Case Knowledge Base by using result of modification of cases

Shigeru MAEDA

Institute for New Generation Computer Technology

Mita Kokusai Building 21F, 1-4-28 Mita, Minato-ku, Tokyo 108

It is an important issue to increase efficiency of retrieval of similar case(s) on Case-Based Reasoning (CBR), especially when the number of cases stored in Case-Base becomes large. In this paper, supposing hierarchy of elements of cases according to abstract level, I propose the method of how to improve performance of CBR, by which refine domain knowledge and case hierarchy increasingly by using result of modification of case(s). To achive this, I define four operators to refine knowledge and use them selectively to states by combination of information of matching and that of modification. It enables the system to retrieve more suitable cases effectively.

1 はじめに

事例ベース推論 (Case-Based Reasoning: CBR) [Kolodner 87] の推論過程は，主に，（1）類似事例の検索，（2）事例の修正，（3）修正事例の評価，（4）修正事例の格納からなる。事例ベース推論の性能向上のためには，類似事例の検索の効率化，事例修正の効率化などの問題を解決する必要がある。中でも，類似事例の検索の効率化は，蓄積される事例数の増加にともない重要な問題となる。従来，その解決法としては，事例の indexing 手法 [Barletta et.al. 88]，並列検索 [新田ほか 90] などが知られている。

本稿では，事例の照合結果，事例の修正結果の情報を用いて，事例の検索の効率化に必要な情報を追加，修正を行うことでこの問題を解決する手法について述べる。

まず，事例ベースの事例は，自動的に抽象度によって階層化されているものと仮定する。一度，ある問題に対して推論が行われると，そのときの事例の照合結果，修正結果の情報により，事例の検索知識および事例ベースの構造を適切なものに修正し，次回からの事例の検索の効率を改善する。これにより，事例の蓄積が進むに従い，段階的により適切でかつ効率の良い検索を行うことが出来るようになる。

また，労働判例を事例として用いて法的推論を行うシステム [新田ほか 90] に適用する場合について考察した。

2 事例ベースの構造

まず，前提とする事例ベースの構造 [前田 89] について述べる。事例の検索の効率化のためには事例ベースの構造化の問題が重要になる。なぜなら，構造のない平板な事例ベースでは，全ての事例の全ての要素を検索する必要があり，非常に効率が悪いからである。したがって，事例ベースに何らかの構造を与え，その構造にしたがって検索候補を段階的に絞っていく必要がある。

本稿で扱う事例ベースでは，事例は，入力時には一階述語形式で表された属性要素の集合として表され，各属性要素の抽象度は考慮しなくてよい。その後，各々の属性要素は，事例ベース中に存在する概念要素と照らしあわされ，照合された概念要素の出現頻度によって抽象度を決定され，図1のような事例ベース中の概念要素の階層構造として記憶される。ここで，概念要素とは，複数の属性要素の集合であり，抽象度により自動的に単一の概念の構成要素として判断されたものである。最後に，入力された事例は，各概念要素に対するインデックスの集合として表される。

図では，SWALLOW1, TROUT1, FLYFISH1 は，それぞれ概念要素に対するインデックスの集合として表された事例であり，それ以外の部分が事例ベースの概念要素の階層構造を表している。例えば，SWALLOW1 は，ANIMAL → BIRD → SWALLOW という概念構造へのインデックスで表される。各概念要素の抽象度は事例からのインデックスの数から判定できる。各概念要素は，図では仮にその一番上の欄に書いてあるものを名前とし，それ以下に書いてあるものを各属性要素として表している。ANIMAL, BIRD, などは概念要素の仮の名前である。たとえば，SWALLOW という概念要素は {bill(yellow), body(black)} という属性要素の集合として表される。

この手法は，E-MOP [Kolodner 84] 等に比べて，属性要素を重複して持つことがないため，記憶効率が良く，また，抽象度により階層化されているので，階層をたどる検索の効率も良い。

3 推論過程

次に，ここで用いる事例ベース推論の推論過程について述べる（図3参照）。

まず最初に，問題は属性要素の集合で与えられる。それにしたがい，事例ベースから与えられた問題に類似の事例を検索する。属性要素の照合には，unification による部分照合，概念木などの領域照合知識による類似照合を用いる。

まず，unification により，与えられた問題の属性要素と事例ベース中の概念要素に含まれる属性要素との照合を行う。照合が取れた場合は，unification によって照合が取れたという情報を出力する。照合が取れない場合は，両者をそれぞれ概念木に沿って抽象化して照合を試みる。共通の上位概念を持つ場合は，両者の照合は成功したものとし，概念木による照合を行っ

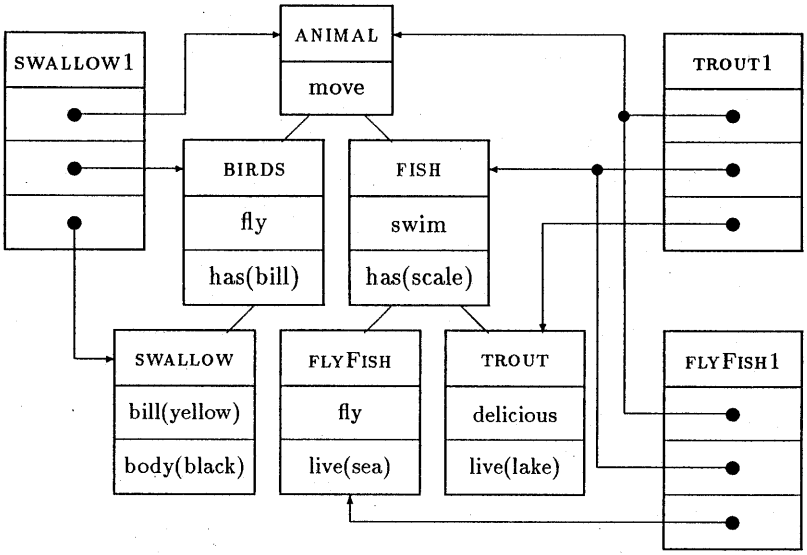

図 1: 階層化された事例ベースの構造

```
                    animal:-move.
                   /              \
bird:-fly,has(wing).                fish:-swim,has(scale).
          |                                  |
penguin:-swim,walk,color(body,[black,white]).      salmon:-color(body,[silver]).
```

図 2: 事例ベースの簡単な例

たという情報を用いた概念木と共に出力する。それでも照合が取れない場合は，照合は失敗したものとし，照合失敗の情報を出力する。

各属性要素間の照合優先順位は，事例ベース中の概念要素の抽象度に従う。すなわち，事例ベース中の抽象度の高い概念要素と照合した属性要素を優先する最良照合を用いる。これは，抽象度の高い一般的な概念要素から該当事例を絞り込むためである。

例として，図 2 のような事例ベースの階層構造を考える。":-" の左側は仮の概念要素の名前，右側は属性要素の集合である。このとき，以下のような質問をした場合を考える。

```
?-match([swim, color(body, [black,
  white])], X). % 河豚の類似事例の検索
```

問題はこのように， swim, color(body, [black, white]) という個々の属性要素の集合として与える。この意味は，水中を泳いでいて，体の色が黒と白のもの（ここでは河豚を意味している）の類似事例を検索することを表している。

まず，概念要素の上位の部分から照合を始める。animal :- move. は，問題には存在しない。したがって，次に， bird :- fly, has(wing). と fish :- swim, has(scale). を検索し，概念要素 fish の属性要素 swim と照合が取れる。 fish の一部と照合が取れたので，その下位概念要素である salmon を検索し，照合を取るが照合には失敗し，結果として fish を類似事例として得る。照合情報を，{[問題の属性要素, 照合した概念要素, 照合した属性要素, 照合状態] ... } で表すとすると， {[swim, fish, swim, unification]} となる。

このように，抽象度の高い概念要素を優先することで，抽象度の低いレベルにある特殊な属性要素によって照合 (penguin) を防ぎ，目的とする fish を得ることができる。したがって，照合には抽象度の高い概念要素を優先する必要がある。

次に，類似の事例が検索されると，それを与えられた問題に適合するように修正する。修正は部分照合した属性要素と概念要素間の概念変換として行われる。今，労働判例を考えると，たとえば，問題中の勤務形態が交代制勤務であるが，類似事例中の勤務形態が夜間勤務だとする。そして，類似事例には労働条件が厳しいとなっているが，問題中には労働条件の記述はない。しかし，夜間勤務より交代制勤務の方が労働条件は緩いという知識があったとする。夜間勤務を交代制勤務に適用できるように変換すると，労働条件が厳しいからやや厳しいに変換される。

修正が終了すると得られた解は問題解決に成功するか，失敗するかを評価される。本稿の着眼点はこの結果を用いて事例ベース推論の段階的な効率向上をはかることである。

最後に，問題解決に成功した修正事例は，照合結果を用いて事例ベースの概念構造の中の適切な位置に格納される。

4 事例ベースの洗練化

与えられた問題に対し，過去の類似の事例は修正される。この結果を用いて現在の領域知識，事例ベースの階層構造を洗練する手法を説明する。領域知識としては以下のものがある。

概念抽象化知識 一般的な概念要素の階層関係

概念間変換知識 異なる概念要素間の変換知識

概念抽象化知識は一般に概念木とも呼ばれ，概念要素間の階層関係を木の形で表したものである。たとえば，船長と機長は共に乗りものの責任者であり，人間であるという知識は， isa(船長, 乗りものの責任者)., isa(機長, 乗りものの責任者)., isa(乗りものの責任者, 人間)., isa(乗りものの責任者, 人間). と表される。

概念間変換知識は，一般的に概念と概念の変換ルールとして表される。たとえば，先の労働条件の例は，以下のように表される。

IF trans(労働形態, { 夜間勤務, 交代制 })
THEN trans(労働条件, { 厳しい, やや厳しい }).

修正される概念要素中の属性要素は，与えられた問題の属性要素との照合結果によって次の三通りに分類される。

$\mathbf{P_1}$ unification により完全に照合した属性要素

照合結果の出力: unification

$\mathbf{P_2}$ 概念抽象化知識を用いて照合した属性要素

照合結果の出力: [tree, 用いた概念木]

$\mathbf{P_3}$ 照合しなかった属性要素

照合結果の出力: noMatch

P_1 は，完全に照合したため修正は行われないので，修正される概念要素は次の三通りに分類される。

$\mathbf{C_1}$ P_2 を含む概念要素

$\mathbf{C_2}$ P_3 を含む概念要素

$\mathbf{C_3}$ P_2 と P_3 を含む概念要素

また，個々の概念要素は修正した結果によって次の二通りに分類される。

Cs 概念要素の変換が成功した場合

Cf 概念要素の変換が失敗した場合

さらに事例ベース中の階層構造を変更するオペレーターとして次の四つを用意する (図 4)。

Op_1: 統合 異なる概念要素を同一の概念要素とみなす

Op_2: 融合 複数の概念要素を一つの概念要素に統合する

Op_3: 分離 ある概念要素に含まれる属性要素集合を別の概念要素として分離する

Op_3: 削除 ある概念要素を必要のないものとして削除する

以下では上記の概念要素と修正結果の組み合わせの各場合について，領域知識と事例ベースの概念階層の洗練化の手法を説明する。組み合わせは，[修正される概念要素・修正結果] で表す。

[C_1· Cs]

step1: 概念抽象化知識の洗練化 抽象化された概念要素間の照合結果を獲得する。たとえば，クモ膜下出血と高血圧性脳出血が共通の上位概念要素脳出血で照合がとれ，修正が成功した場合，両者を直接照合する知識を加える。これにより，次回からは概念木をたどる必要がなく，照合の効率化がはかれる。

step2: 事例ベースの階層構造の洗練化 照合知識に二つの概念要素が等価であるという知識を加えたので，$\mathbf{Op_1}$ を行い，二つの概念要素を一つに統合する。したがって，次回からはこの概念要素一つ検索するだけですむ。

さらにこの概念要素の直接の上位／下位概念要素を融合できるかを判断し，上位／下位概念要素が共に概念変換に成功していれば $\mathbf{Op_2}$ を行い両者を一つに統合する。

[C_1· Cf]

step1: 概念抽象化知識の洗練化 過剰一般化による概念間の照合の失敗を獲得する。たとえば，船長と機長が共通の上位概念乗車物の責任者で照合がとれたが，修正が失敗した場合，この一般化を禁止する知識を加える。これにより，次回からは過剰な一般化による照合によって生じる修正の失敗を防ぐ。

step2: 事例ベースの階層構造の洗練化 類似事例の概念要素は単一の概念として照合を行い，修正を行った結果失敗したので，概念を分離する必要がある。したがって，概念要素中の照合に成功した属性要素集合を上位概念要素とし，概念抽象化知識によって過剰一般化による照合を行った属性要素集合はその下位概念要素として $\mathbf{Op_3}$ によって分離する。

[C_2· Cs]

事例ベースの階層構造の洗練化 概念要素中の照合に失敗した属性要素集合を $\mathbf{Op_3}$ によって下位概念に分離し，さらに，$\mathbf{Op_4}$ により削除する。これにより，次回からの無駄な照合を防ぎ，照合の効率化をはかる。

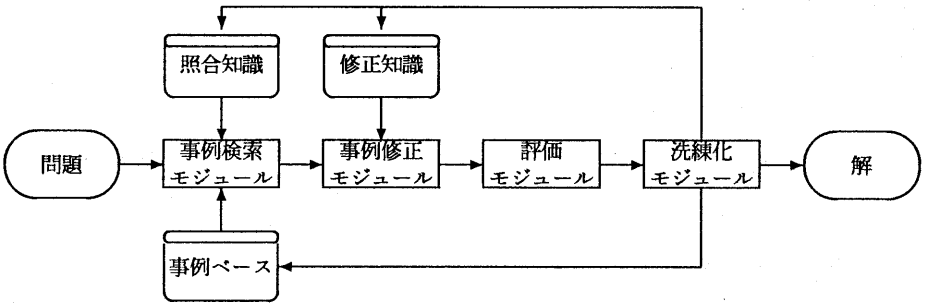

図 3: 事例ベース推論の推論過程

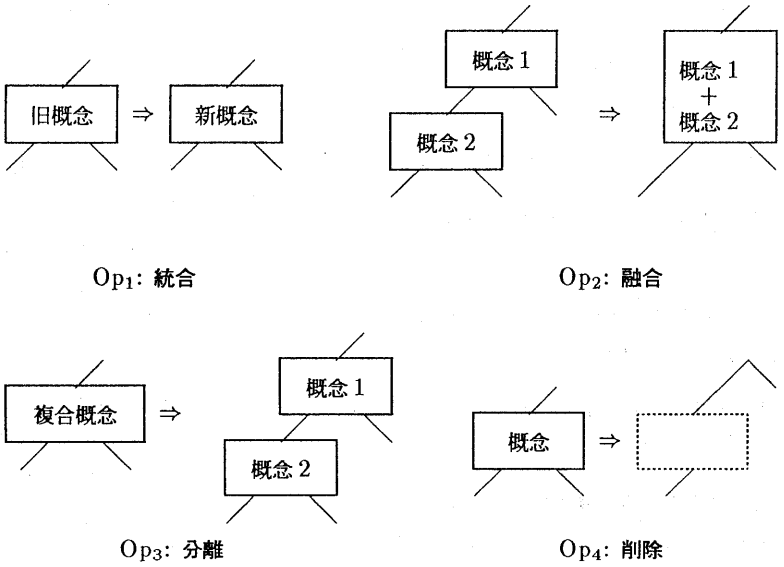

図 4: 概念構造変更オペレーター

[C_2· Cf]

step1: 概念間変換知識の洗練化 概念間変換知識の誤りを修正する。次回からはこの概念変換を行わないようにし，失敗を防ぐ。

step2: 事例ベースの階層構造の洗練化 概念要素中の照合に失敗した属性要素集合を Op_3 によって下位概念に分離する。これにより，次回からの照合の失敗を防ぐ。

[C_3· Cs]

[C_3· Cf]

それぞれ対応する C_1, C_2 に対する操作を上記と同様に行う。

5 まとめ

以上のように類似事例の修正の結果を用いて，領域知識，事例ベースの階層構造の段階的な洗練化を行う手法を提案した。Protos [Bareiss, et.al. 90] では，prototipicality と difference link により事例の構造の再構築を行うが，本手法では，そのようなものを与えずに，事例の修正結果を用いて，実際の推論結果を事例ベースに反映させることを特徴とする。このような手法は，事例ベース推論において効率化をはかるためには必要な技術である。

6 謝辞

本研究の機会を与えた頂いた新田克己 ICOT 第7研究室長に感謝致します。また，ICOT KSA-KAR サブワーキンググループのメンバーの方々との議論が大変参考になりました。

参考文献

[Barletta et.al. 88] Barletta R., Mark W.: Explanation-Based Indexing of Cases, AAAI-88, pp.541-546, 1988.

[新田ほか 90] 事例を用いた法的推論とその並列化: 情処知識工学と人工知能研究会 69-5, 1990.

[Kolodner 87] Kolodner, J.: Extending Problem Solving Capabilities Through Case-Based Inference, Proc. of the 4th Annual International Machine Learning Workshop, 1987.

[前田 89] 前田: 事例に基づく推論のための事例ベース構築方法，第39回情処全国大会 4C-4, 1989.

[Kolodner 84] Kolodner, J.: Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model, LAWRENCE ERLBAUM ASSOCIATES, PUBLISHERS, London, 1984.

[Bareiss, et.al. 90] E. Bareiss, Bruce Porter, Cbaig Wier: PROTOS: An exemplar-based learning apprentice, Machine Learning Vol. III, Morgan Kaufmann, 1990.