事例に基づく概念の学習*

前田 茂†
(財) 新世代コンピュータ技術開発機構‡

1 はじめに

事例に基づく推論 (Case-Based Reasoning: CBR) は，問題解決と学習の両面からとらえることができる。過去の事例を修正し新しい問題に対する解答を与える面では問題解決である。一方，修正を施された過去の事例が，次回からの利用のために解析され，記憶構造の適切な位置に記憶される過程は，学習であると考えられる [1]。

本稿では，抽象度によって階層化された事例記憶を前提とし，事例を修正した結果を利用して領域知識および事例記憶の構造をより適切なものに修正・変更する手法を提案する。これにより，事例の蓄積が進むにつれ，検索効率が改善され，また，より適切な事例を検索できることになる。

2 事例記憶の構造

まずここで前提としている事例記憶の構造を述べる。事例はそのインデックスとして，抽象度により階層化された概念を持つものとする（図1参照）[2]。事例記憶はこの事例のインデックスとしての階層化された概念の集まりとして構成される。これは，検索の効率化を計るためである。各概念は，一階述語論理で表される関係の論理積で表される。つまり，概念をある意味情報の集まりとして表す。

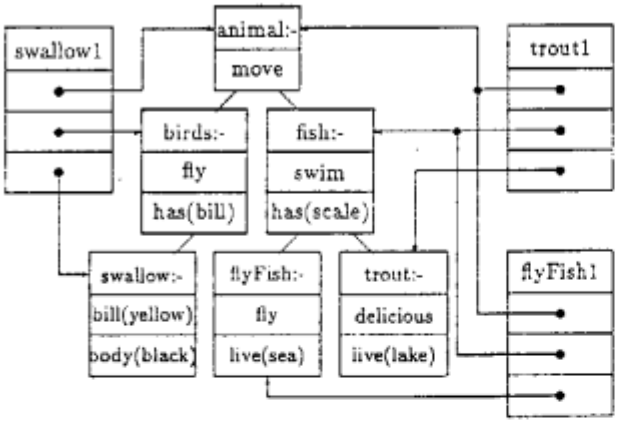

図 1: 階層化された概念による事例のインデックスの表現

3 事例に基づく推論の推論過程

ここでは，事例に基づく推論として以下のものを考える。

まず，与えられる問題は関係の論理積で表される。次に問題に最も類似している事例を検索する。そのとき，関係間の照合にはユニフィケーションを用い，それに失敗した場合は，領域知識（例えば概念木など）を用いて照合を試みる。概念木を用いる場合は，両者の関係を抽象化して照合を試みる。そして，抽象度の高い概念に照合した関係を重視した重み付き最良照合により類似事例を決定する。これは，例えば，以下のような階層構造のもとで

?-match([swim,color(body,[black,white])],X).% 河豚の検索

のような質問をした場合に，抽象度の低いレベルに存在する特殊な要素により誤った検索 (penguin を得る) が行われる事を防ぐためである。

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

修正は与えられた問題の関係と部分的に照合した過去の事例中の概念に対する変換として行われる。例えば，与えられた問題が CPU の設計であり，その一部で 8 ビットの加算器の設計をする場合に 4 ビットの加算器の事例しかないとき，加算器の知識を用いて 4 ビットの加算器を 8 ビットの加算器に変換する。また，過去の事例に存在しない関係に対して明示的に変換規則が存在する場合はそれを用いる。そして，修正された事例は，現在の問題を解決するのに成功するか，失敗するかが評価される。

最後に，修正された事例は，次回からの利用のために，成功した場合は新しい事例として記憶され，失敗の場合は，失敗の原因が解析され，再び失敗を繰り返さないように修復される。本稿ではこの処理を学習としてとらえ，以下にその手法を提案する。

4 概念の再構築による学習

ここでは，ある与えられた問題に対して，類似の過去の事例の修正の結果を利用して，現在の領域知識および概念構造を変更することを考える。修正の結果は修正された事例の各概念に対して次の二通りが存在する。

Cs 修正が成功した概念

*Concept learning by using past experiences
†Shigeru MAEDA
‡Institute for New Generation Computer Technology

Cf 修正が失敗した概念

概念構造は先に述べた通りであるが，領域知識は主に次の二つに分類できる。

概念間変換知識 異なって表された二つの概念を同一のものとして扱うようにする知識。

概念抽象化知識 ある概念の上位／下位概念に関する知識。いわゆる概念木に相当する。

次に，与えられた問題に現れる概念を，類似事例との照合結果によって以下の4つに分類する。

C1 完全に照合した概念

C2 領域知識を用いて照合した概念

C3 一部しか照合しなかった概念

C4 完全に照合しなかった概念

C1 の場合は，概念構造および知識の修正は何も行われない。

C2 の場合は修正の成功／失敗によってさらに次の二つの場合が考えられる。

Cs の場合，知識の修正は概念抽象化知識に対してだけ行われる。知識によって抽象化された概念間の照合結果を獲得する。これにより，次回からは類似の関係に対しては直接この知識を利用すればよく照合の効率が改善される。

Cf の場合，まず，どちらの知識に誤りが存在したかを解析する。概念間変換知識が誤りだと判断された場合は，その知識の修正を行う。一方，概念抽象化知識の場合は，修正の誤りは知識自身の誤りよりも過剰一般化による照合に基因するものと考える。したがって，知識に過剰一般化による照合を制限する制約を加える。これにより，次回からの失敗を回避する。

C3, C4 の場合は，概念構造の再構築を行う。その時行う操作は次の四つである（図2参照）。

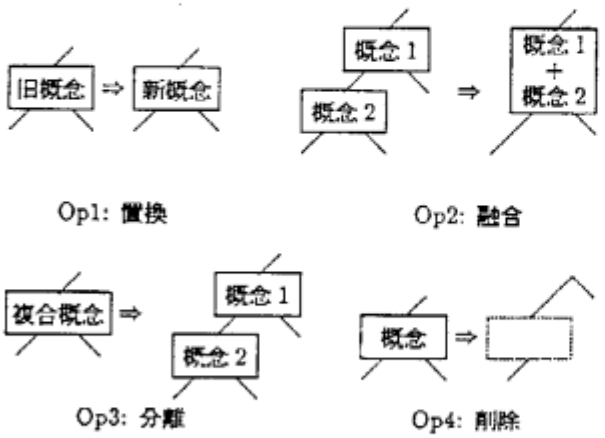

図 2: 概念構造変更操作

Op1: 置換 ある概念に含まれる一部の関係を他の概念の関係に置き換える。

Op2: 融合 複数の概念を一つの概念に統合する。

Op3: 分離 ある概念に含まれる関係を別の概念として分離する。

Op4: 削除 ある概念を必要のないものとして削除する。

Op1 は Cs の場合に行う。与えられた問題と類似事例に含まれる概念中の関係の一部が異なる場合でも，その修正結果に成功をもたらす時，それらの概念は等価なものとして置き換えられる。次回からの照合には置き換えられた新しい概念が直接使用され，検索の効率が改善される。

Op2 は Cs の場合に行う。類似事例中に連接する概念（直接の上位／下位概念）が存在し，どちらも与えられた問題に存在し，かつ修正が共に成功した場合，これらを一つのより一般的な概念として統合する。これにより過剰な概念の細分化を防ぎ，照合回数を減らす事ができる。

Op3 は Cf の場合に行う。類似事例に含まれるある概念の修正が原因で失敗が発生した場合，与えられた問題の概念と共通に存在する関係を上位概念として，類似事例の概念のみに存在する関係を下位概念として分離する。これは，類似事例のみに存在する関係を特殊なものと考え分離することで，次回からの失敗を回避するものである

Op4 は Cs の場合に行う。類似事例に含まれるある概念で，与えられた問題に含まれる概念と照合が取れその修正が成功したもののうち，与えられた問題の概念に存在しない関係がある時，その関係は修正に必要ないものとして分離された後削除される。これにより余分な照合の手間が省ける。

以上の操作を行う事により，与えられた問題を修正した結果を事例の記憶構造に反映させる事ができ，段階的に検索効率を改善する事ができる。

5 まとめ

以上，事例の修正結果を用いて領域知識の修正および事例記憶の構造の変更を行い，事例に基づく推論の問題解決能力の向上および照合の効率改善を行う手法を提案した。今後は，各概念間の依存関係による修正の影響についても考えていきたい。

参考文献

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

[2] 前田 茂，事例に基づく推論のための事例ベース構築方法，情報処理学会第39回全国大会予稿集，pp.311-312, 1989.