

## 2G-8

## 知識の逐次的修正用機能の提案

山田 弘 安信 千津子 増位 庄一  
 日立製作所 システム開発研究所

## 1. はじめに

エキスパートシステム構築ツールには、デバッガにおいて知識を逐次的に修正できる機能がある。しかしながら、この機能はまだ十分であるとは言えない。本稿では、まず知識構築過程における知識の修正の意義と機能を明らかにした上で、それに基づき、知識の修正に関して構築ツールが必要な機能を提案する。

## 2. 知識構築モデル

## 2.1 従来の知識構築モデル

[1]では、知識構築の過程を図1から点線部内を取り除いたものでモデル化している。この内「知識構築」の役割は、「専門家モデルの形成」で得られた専門家が問題を解決する方法のモデルである専門家モデルを、ルールのような知識の形に変換することである。

## 2.2 知識修正の意図

知識構築過程における知識修正の意義や機能を探るために、知識のデバッグ時に修正を行う意図を調べてみる。すると、表1のように分類できる。ここで「本来の値に直す」とは、推論実行時に他の誤った知識により、本来なるべき値になっていないものを、なるべき値に直す場合である。

個々の修正の意図が、先にあげた知識構築モデルと適合しているかどうかを考えると、表1の右側のようになる。×印のついた修正が適合しないのは「知識構築」にて行われているにもかかわらず、専門家モデルとはあまり関係せずに、現在ある知識に基づいて細かな修正を行うのみだからである。すなわち、これらの修正は先の知識構築モデルに明確には反映されていないのである。

## 2.3 新しい知識構築モデル

そこで、×が付いた場合を反映する段階を知識構築モデルに入れることにする(図1・点線内部を含む)。この新しく入れた「知識修正」は、対応する修正の意図が示すように「知識正当性の検証」と密接な繋がりがああり、相互にいききできることが必要である。そのため、「知識構築」のように一括して知識を修正するのではなく、一つのルールといったような一知識

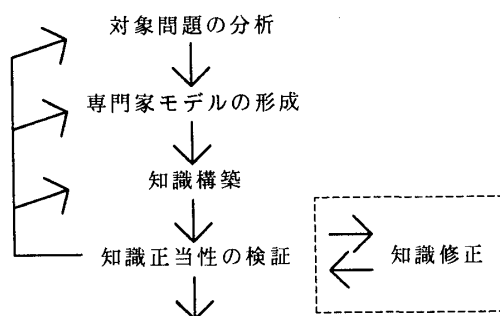


図1 知識構築モデル(点線内が追加部分)

表1 知識修正の意図

|       | 修正の意図                | 構築モデルとの適合 |
|-------|----------------------|-----------|
| 誤りの修正 | 1. ケアレスミス            | ×         |
|       | 2. 本来の値に直す           | ×         |
|       | 3. 専門家モデルを正しく表現していない | ○         |
|       | 4. 専門家モデル以前の誤り       | ○         |
| 実験的修正 | 1. 試行錯誤的によりよい知識を求める  | ×         |

単位ごとに逐次的に修正できることが必要となる。このことを、知識の逐次的修正と呼ぶことにする。

## 3. 知識の逐次的修正に必要な機能

構築ツールではデバッガ中で推論を行うことなどにより、誤った知識を発見する。すなわち「知識正当性の検証」の役割はデバッガが担っている。そのため、デバッガにおいて「知識修正」の役割が果たせるようにすべきである。すなわち、デバッガ内で推論を行うことなどにより修正すべき知識を決め、その場で逐次的に知識を修正し、すぐに推論を行うことなどができる状態になっていなければならない。その際に必要な機能を、前章で述べた逐次的修正の意義と機能に従い、以下の三つの視点より考える。

## 3.1 修正自体に必要な機能

単純な誤りや本来の値にする修正を行う場合、一知識単位での追加や削除、および、ルール内の定値など

の修正が必要となる。このとき、他の知識を参考にできたり、修正単位を容易に選ぶことが必要である。また、実験的修正の場合、知識を試行錯誤的に修正しながらよりよい知識を見つけていくという作業を行うため、一度削除した知識を復活したり、修正前の知識を参考にできる必要がある。以上をまとめると、表2のようになる。構築ツールではこのすべての機能を提供することを提案する。

### 3.2 修正後の推論再開時点

デバッガで推論を行うことにより誤った知識を発見し、その時点で推論実行を止めて知識の修正を行った場合、その後どの地点から推論を再開できるようにしておくべきかを考える。

単純な誤りの修正では、推論にまだ利用されていない知識であれば、止めた次から再開すればよい。すでに利用された、もしくは利用すべきだった知識であれば、その知識が利用される、もしくは利用されるべきだった前から推論を再開すればよい。本来の値へ修正する場合は、修正後の知識の状態が正しい現在の状態であるので、推論を止めた次から再開すればよい。実験的修正であれば、推論実行中のある特定の時点から、知識を色々と変化させた場合について推論実行を行い知識の妥当性を調べる。そこで、指定した時点から推論を再開する必要がある。以上をまとめると表3のようになる。構築ツールではこのすべての機能を提供することを提案する。

### 3.3 修正の有効範囲

知識は、推論中にその値などが変化するもの(ワーキングメモリエレメントなど)と変化しないもの(多くの場合ルール)に分けられる。また、変化するものの場合、実際に変化したかどうかで三つの形態に分けることができる。ここで、デバッガを用いて人手により逐次的に知識を追加した場合は、変化がなかった場合と同様に考えることができる。修正の有効範囲として、一連の推論が終了するまでの「この推論中のみ」、デバッガを抜けるまで、および、一度推論を終えてデバッガを抜け出して、再び推論を行っても影響がある「以後ずっと」の三種類がある。これらに応じて修正の有効範囲を分類すると表4の様になる。

この表より2種類の有効範囲が必要となることが分かった。しかしながら、この2種類は機械的に判断することは難しく、かといって利用者にどちらであるかいちいち問い合わせるのは利用者にとって不便である。そこで、推論中に変化しない知識の場合は「以後ずっと」のみであることとの整合性を考えて、有効範囲のデフォルトとして「以後ずっと」とし、それ以外の場合は利用者が指定することを提案する。

表2 知識修正に必要な機能

| 修正の動機          | 必要な機能   |
|----------------|---|
| 単純な誤り<br>本来の値へ | 一知識単位での追加、削除<br>→ 他の知識を参考にできる<br>一知識内の一部の修正<br>→ 修正単位を簡単に選べ、修正できる |
| 実験的            | 一度削除した知識の復活<br>修正前の知識の参照  |

表3 知識修正後の推論再開時点

| 修正の動機 | 再開時点   |
|-------|--|
| 単純な誤り | 推論に未使用で、使用される必要もなかったならば、次から<br>それ以外ならば、使用前または使用されるべき前から(初めからも含む) |
| 本来の値へ | 次から  |
| 実験的   | 指定したある時点から   |

表4 知識修正の有効範囲

| 知識の性質 | 現在の形態             | 修正の動機 | 有効範囲    |
|-------|-------------------|-------|---------|
| 動的に可変 | 変化なし              | 単純な誤り | 以後ずっと   |
|       |                   | 実験的   | この推論中のみ |
|       | 変化あり              | 本来の値へ | この推論中のみ |
|       |                   | 実験的   | 以後ずっと   |
|       | 実行時追加             |       | この推論中のみ |
| 動的に不変 | 「動的に可変」の「変化なし」と同じ |       |         |

表5 修正に関する機能(提案)

|         |                        |
|---------|------------------------|
| 修正単位    | 一知識単位、一知識単位の一部         |
| 修正機能    | 追加、削除、修正、<br>削除した知識の復活 |
| 修正補助機能  | 他の知識、修正前の知識の参照         |
| 推論再開時点  | 初めから、次から、<br>任意の時点から   |
| 修正の有効範囲 | (デフォルトとして)以後ずっと        |

## 4. おわりに

知識構築過程において、新たに設けた知識修正の意義は単純な誤りの修正などにあり、機能は知識の逐次的で部分的な修正にあることを述べた。そして、知識の修正のために構築ツールはデバッガにおいて表5のような機能が必要であると提案した。

## 5. 参考文献

- [1] B.G.Buchanan, et al.: Constructing an Expert System, Building Expert Systems, Addison Wesley, pp.127-168(1983)