

5G-2

知識処理ツール "IKE"

— 仮説推論システム —

上田 芳弘 間藤 隆一 上田 謙一
松下電器産業株式会社 東京研究所

1. はじめに

第2世代のエキスパート・システム構築ツールとして知識処理ツール(IKE)の開発を行っている[1][2]。IKEは推論サブシステムとして以下を有する。

- 1) ルール・ベース・システム
前向き推論システム、後ろ向き推論システム
- 2) フレーム・ルール・ベース・システム
前向き推論システム、後ろ向き推論システム
- 3) 仮説推論システム

さらに、これらの各推論サブシステムでの知識表現を一階述語論理形式で統一している。また、多重世界機構[3]を採用した世界管理システムがルールやファクト、フレームを一括管理している。

ここでは、優先度を付加した3) 仮説推論システムについて報告する。

2. 基本構成

IKEの仮説推論システムはATMS[4][5]を基にしているが、ATMSは確定的な知識しか扱えないので不確定な知識を扱うように拡張を行った。同様な試みには董、中川によるEATMS[6]が報告されている。ただし、EATMSは不確定な知識として確信度を用いた知識表現を仮定している。これに対して我々は、矛盾を表すnogoodに優先度を付加することによって、優先度毎の矛盾のない解を推論できるように拡張を行った。これにより確信度の計算等の処理の軽減ができ、優先度による柔軟な推論が実現できる。

ATMSは、知識ベースの無矛盾性を維持するシステムで、知識はnodeとして表現され、その内部表現は簡単には次のような構造を持つ。

< datum, label, justification >

datumは、知識の実体の内部表現である。labelは、そのnodeが信じられる環境の集合である。例えば、labelは

{ {A} {B C} }

のような集合であり、この場合は仮説Aが信じられている環境、仮説B, Cが信じられている環境でこのnodeは支持される(信じられる)。justificationはそのnodeが導かれた理由付けであり、ルールの条件部にマッチしたnodeのリストの集合である。例えば、

{ (A) (b c) }

のような集合である。justificationは推論システムが計算し、labelはATMSが計算する。(このアルゴリズムがlabel updateアルゴリズム)。IKEの仮説推論システムでは、あるcontext(いくつかの仮説を信じたときの世界)にお

いて、そこで支持されるnodeとルールによって推論システムが新しい結論を導く。そして、推論システムはこの結論nodeに導出の条件となったnodeリストをjustificationとして記録する。ここで、ATMSは結論nodeについてlabelの計算を行うが、このアルゴリズムは優先度の付加によって後述のような拡張が必要である。

優先度は、矛盾であるnode(nogood)だけに付加できるものとし、他のnodeには付加しない。これは、ATMSではnogoodを中心に処理が進むが、nogoodが確定的であるので、ここに優先度を付加すれば、システム全体として不確定的な処理が可能となるからである。また、優先度の取扱は次のような規則に従う。

- 1) 優先度は0以上の整数とする。
- 2) 優先度が0以外のnogoodはその優先度の時のみ有効とし、他の優先度の時は無効とする。
- 3) 優先度が0であるnogoodはすべての優先度において有効とする。

つまり、優先度が0のnogoodは大域的に、優先度が0以外のnogoodは局所的に矛盾とされる。

3. 機能

IKEの仮説推論システム用のルールや諸機能の一部を説明する。

ルール

- 1) 一般的なルール
他のnodeから結論を導出するためのルールであり、前向き推論システム用のルールと同一形式である。
- 2) 矛盾を定義するルール
矛盾の条件を定義するルールであり、システムはこのルールを基に矛盾を検索する。THEN部に"contradiction..."と記述し、さらに優先度を与える。
- 3) 新しいcontextを生成するルール
新しいcontextを生成するときに、ユーザがcontextに加えたい仮説を定義するルールであり、このルールがなくてもATMSは自動的に矛盾のないcontextを生成する。THEN部に"new-context..."と記述する。
- 4) contextの枝刈りをするルール
無駄なcontextの生成を防ぐために予め他の

可能性を調べる必要性がないことが分かっているときにその条件を指定するルールである。ATMSはこの条件を満たす context と並列関係にある context を枝刈りする。THEN 部に "cut-path..." と記述する。

- 5) ゴールを設定するルール
 ゴールとなる context の条件を指定するルールであり、ATMSはこの条件を満足する context より下位の context は生成しない。THEN 部に "goal-context..." と記述する。

その他の機能

- 1) 仮説の定義
 関数 def-assumption によって仮説を定義する。なお、premise は前向き推論のファクトと同様に定義する。
- 2) 仮説の選択
 関数 choose で仮説の集合を指定する。ATMS は context を生成するときこの仮説集合の要素のうち、少なくとも1つは context に含めるようにする。
- 3) 初期 context の定義
 関数 init-context で仮説の集合を指定し、初期 context を定義する。

4. アルゴリズム

データ構造は優先度の付加にともなって、次のように拡張した。label, nogood データベースは優先度と環境のペアからなる集合とする。例えば、label は以下のような集合となる。

$$\{(0 \quad \{A \ B\}) \ (1 \quad \{B \ C\})\}$$

ATMS では label の冗長性の検出、矛盾の検出時には環境の包含関係を調べる必要がある。上のような拡張によって新しく優先度と環境のペアについて包含関係の定義を次のように与える。

定義: $(p_i \ \alpha_i)$ が $(p_j \ \alpha_j)$ の superset, iff
 $(p_i = p_j \ \vee \ p_j = 0) \wedge \alpha_i \supseteq \alpha_j$

label update アルゴリズムは次のように拡張する。

- 1) 親 node の label をマージする。

$$\bigcup_k \{x \mid x = \bigcup_i x_i \text{ where } x_i \in j_{ik}\}$$

j_{ik} : 親 node の k 番目の justification の i 番目の label

$\bigcup_i (p_i \ \alpha_i)$: すべての i に対して、 $p_i = p_j \ \vee \ p_j = 0$ のときのみ

$$\{(\max_i p_i \ \cup \ \alpha_i)\}$$

を結果とする。

- 2) この新しい label から冗長な環境と矛盾である環境を削除する。ただし、包含関係については上の定義を用いる。
- 3) ここまでで計算した label が今までの label と同じであれば、その node についての計算は終了する。
- 4) もし、結論 node が矛盾であれば、計算した label のすべての環境をの nogood データベ-

スに加え、優先度が0で矛盾である環境はすべての node の label から削除する。

もし、結論 node が矛盾でなければ、子 node (この結論 node を justification に持つ node) の label についても再帰的に 1) からこのアルゴリズムを適用する。

元の label update アルゴリズムとの変更点は 1) と 4) である。

また、choose 機能を追加することによって label の consistency と completeness を保証するための規則 (hyperresolution) も次のように拡張する。

consistency を保証するための規則

$$\frac{\text{choose}\{A1 \ A2 \ \dots\} \\ \text{nogood}(p_i \ \alpha_i) \text{ where } A_i \in \alpha_i \wedge A_{j \neq i} \notin \alpha_i \\ \wedge (p_i = p_{j \neq i} \vee p_i = 0) \text{ for all } i}{\text{nogood}(\max_i p_i \ \cup \ [\alpha_i - \{A_i\}])}$$

completeness を保証するための規則

$$\frac{\text{choose}\{A1 \ A2 \ \dots\} \\ \langle \beta \ \Lambda \ \delta \rangle \text{ where } \Lambda = \{(p_1 \ \lambda_1) \ (p_2 \ \lambda_2) \ \dots\} \\ \text{[nogood}(p_i \ \{A_i\} \cup \ \alpha_i) \vee (p_i \ \{A_i\} \cup \ \alpha_i) \in \Lambda] \\ \wedge A_{j \neq i} \notin \alpha_i \wedge (p_i = p_{j \neq i} \vee p_i = 0) \text{ for all } i}{\langle \beta \ [\cup_i \{(\max p_i \ \alpha_i)\}] \cup \ \Lambda^* \ \delta \rangle}$$

Λ^* : Λ から $\cup_i \{(\max p_i \ \alpha_i)\}$ のすべての superset である要素を除いた集合

これらの規則はそれぞれ label update アルゴリズムのステップ 4) と 1) に追加される。

5. おわりに

知識処理ツール (IKE) の仮説推論システムとして優先度を付加した ATMS の実現方法と本システムの機能について述べた。優先度による不確定的処理によって、より広範囲なアプリケーションに ATMS が適用できるようになると考えている。今後は、ユーザ・インタフェースの充実と、実際のアプリケーションで評価を行う予定である。

【参考文献】

- [1] 上田 他: 知識処理ツール IKE - システム概要 - 情報処理学会第36全国大会(1988)
- [2] 間藤 他: 知識処理ツール IKE - 知識表現融合方式 - 情報処理学会第36全国大会(1988)
- [3] 中島秀之: 知識表現と Prolog/KR 産業図書(1985)
- [4] de Kleer, J: An Assumption-based TMS Artificial Intelligence 28 (1986)
- [5] de Kleer, J: Extending the ATMS Artificial Intelligence 28 (1986)
- [6] 董、中川: 不確実な知識における ATMS 知識工学と人工知能52-1 (1987)