TM-0308 他

情報処理学会 第35回全国大会 論文集
(5G基礎ソフトウェアシステム)

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03) 456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

TM-0308	推論エンジンKORE/IEにおける非単調推論機構の実現	新谷虎松(富士通)
TM-0312	PSIにおける推論エンジンKORE/IEの試作	新谷虎松，二神浩道(富士通)
TM-0313	落書帳からの証明ー計算機による論証支援のための証明方法論の一考察	南　俊朗，沢村　一(富士通)(
TM-0314	制御集合にもとづく線型言語の帰納的推論	高田裕志(富士通)
TM-0315	Inductive Inference of Logic Programs based on Algebraic Semanti	Y-Sakakibara (Fujitsu)
TM-0320	論証支援システムのための論理エディタ	土屋恭子，佐藤かおる，小野越夫，沢村　一，南　俊朗(富士通)
TM-0321	核言語KL1分散実行方式ーソフトウェアシミュレータによる評価ー	大原有理，鳥居　悟，小野越夫，神田陽治(富士通)
TM-0328	GHCプログラムの検証についてー同期による決定的動作の検証ー	村上昌己
TM-0342	制約付き論理型プログラミング言語の構想と実例	相場　亮，坂井　公
TM-0343	項書換システム「Metis」における帰納的証明機能	大須賀昭彦，坂井　公
TM-0344	記号表現と結合表現の接点	岡　夏樹
TM-0345	Architecture Abstraction in GHC	田中二郎
TM-0350	汎用計算機上のGHCコンパイラ	森田正雄，吉光　宏(三菱総研)，太細　孝(三菱)，上田和紀
TM-0380	通信システム用仕様設計における追加仕様の検証方式	上田佳寛，長谷川晴朗，田中　亘，柴田健次(沖電気)
TM-0383	並列論理型プログラムにおける新プラグマ方式とその応用	神田陽治(富士通)
TM-0385	レイヤードストリームを用いた並列構文解析	奥村　晃，松本裕治

6P−2　推論エンジンKORE/IEにおける非単調推論機構の実現

新谷虎松
富士通㈱国際情報社会科学研究所

1. はじめに

KORE/IEは、論理型言語上に構築された高速な推論システムであり、柔軟なルール指向的プログラミングを提供する[新谷 87]。KORE/IEの推論は、OPS5等に代表される前向き推論型プロダクションシステムと同様に、ﾜｰｷﾝｸﾞﾒﾓﾘｰ（WM）を意識したRecognize-Actｻｲｸﾙを実行することにより実現される。一般に、この種の推論は単調に行われ、WMは、単に、ルールを干渉させるための大域的なデータベースとして用いられる。つまり、WMにおける事実の一貫性は特に問題とされない（むしろ、無視されている）。しかしながら、不確実な知識に基づく推論を効果的に実現するためには非単調な推論メカニズムが必要とされる。このような非単調な推論メカニズムの実現は、専門的知識（もしくは、経験的知識）の有効性の範囲を広げ、ルール指向的プログラミングを用いたアプリケーション・プログラムの構築を容易にする。本論文では、特に、KORE/IEにおける非単調な推論メカニズムの実現方式の概要について論じる。具体的には、関係ネットワーク・サブシステムKORE/EDENを用いて知識間の依存関係を表現する。これにより、非単調推論を実現する際に必要となるdependency-directed backtracking（依存関係に基づくﾊﾞｯｸﾄﾗｯｷﾝｸﾞ）相当の機能を効率的に実現する。

2. KORE/IEにおける非単調推論

不確実な知識には、①問題解決（例えば、診断など）において知識を取捨選択する際に必要とされるものと、②問題解決（例えば、スケージュリングなど）において常識的な判断情報として必要とされるものがある。①に関する知識には、普通、確信度と呼ばれる不確実性を表す数値が付加され、知識間の整合性は保証されないのが一般的である。一方、②に関する知識は、当面の問題解決を達成するために必要とされるものであり、知識間の一貫性は保持される必要がある。このためには、過去に行われた問題解決の一部分を修正をする必要がある。KORE/IEにおける非単調推論は②に相当する不確実な知識を扱うために設計され、デフォルト推論[Reiter 80]に基づく推論を実現する。デフォルト推論は、反証の欠落（もしくは、暗黙の了解）に基づいた推論であり、②のタイプの不確実な知識（デフォルト知識）を扱うための枠組を提供する。

具体的には、デフォルト推論を実現する手法として、TMS(Truth Maintenance System)[Doyle 79]で用いられた知識間の矛盾解消法を導入する。つまり、矛盾の発生時には、そこに到った推論連鎖の記録に基づいてバックトラッキング(dependency-directed backtracking)し、矛盾発生の原因となるデフォルト知識を見いだし、その取消しを行うものである。

3. KORE/IEにおけるＴＭＳの実現

KORE/IEにおいて、TMSは命題間の整合性を保ちながら非単調推論を行うための管理機構として用いられ、デフォルト推論を実現するための基本的機能となる。TMSを具体的に実現するためには、①推論連鎖の記録、②①に基づくバックトラッキング、③矛盾の原因となったデフォルト知識の発見・変更等の手段を実現する必要がある。

次に、KORE/IEにおける具体的な実現方式を述べる。

3.1. 推論連鎖の記録

①の推論連鎖の記録は、推論により新たに得られたデータに対して、その推論ルールのトリガーとして用いられたデータをjustificationとして付加することにより達成する。具体的には、ルールのRHSにおいて生成されるWM要素に対して、LHSにおけるLHSパターンにマッチしたWM要素を関係付けることである。KORE/IEでは、この関係付けにはWM要素のタイムタグをキーとして行われ、これら

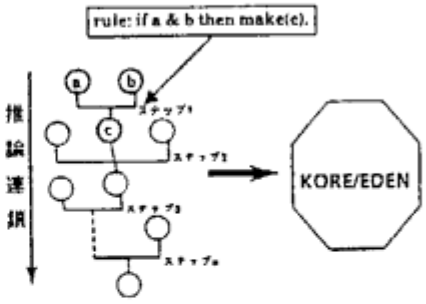

図1. KORE/EDENを用いた推論連鎖の記録

関係情報はKORE/EDEN［新谷 86］を用いて管理される（図1参照）。KORE/EDENは、知識間の関係情報をビット情報として効率的に保存し、これら関係情報により構成

An Approach to Non-monotonic Inference Mechanism on KORE/IE
Toramatsu SHINTANI
IIAS-SIS, FUJITSU LIMITED

されるネットワークを操作するための基本的な機能を提供する。

3.2. TMSバックトラッキング

②のバックトラッキングは、KORE/IEにおける推論を制御する推論スッテパーを用いて実現する。ステッパーにはrun及びbackがある。runコマンドは、推論ステップ数を指示することにより、Recognize-Actサイクルを前向きに進ませ、これにより推論が実行される。逆に、backコマンドは、推論ステップ数を指定し、このサイクルを戻すことにより推論の取消しを行う。この推論の取消しに際しては、各Recognize-Actサイクル毎に記録されたWMへのWM要素の出入り、及び、競合集合の情報が、用いられる。例えば、backのための情報として、サイクル毎に次のようなアサーションが保存される。

back_data (*Step*, *Flag*, *Instantiation*, *WM_list*)

ここで、*Step*はRecognize-Actサイクルにおけるステップを表す整数であり、推論ステップに相当する。*Instantiation*は各ステップで実行されたインスタンシエーションを表す。*WM_list*はWMの出入りを表すリストであり、具体的には、追加されたWM要素及び、削除されたWM要素がそれぞれ、in_wm(<WM要素>)、out_wm(<WM要素>)で表現される。*Flag*は、先に挙げた③の矛盾となったデフォルト知識の発見・変更を実現する際に用いられる重要な情報であり、タイプとして、*real*, *unreal*, *contradiction*がある。*real*は通常のRecognize-Actサイクルにより獲得された情報、*unreal*は矛盾解消手続きが実行されているときのRecognize-Actサイクルにより獲得された情報、そして*contradiction*は矛盾解消手続きの際に逆戻りの対象となった情報であることを示す。

3.3. 矛盾解消手続き

デフォルト推論において、矛盾は、反証の欠落に基づいて用いられたデフォルト知識が原因となり発生する。そこで、KORE/IEでは、矛盾を解消するために、これら矛盾の原因となるデフォルト知識を見いだし、その反証を宣言することにより達成する。デフォルト知識はその他の知識と区別するためのフラグが立てられる。具体的には、次のステップにより達成される。

ステップ1： 矛盾をルートとする推論連鎖木において（推論連鎖木は普通の木の逆の形になるから（図1参照））、反証の欠落に基づいて用いられた全てのデフォルト知識をリストアップする。これは、KORE/EDENを用いて効率的に得られる。具体的には、推論連鎖木の葉に相当する位置にあるデフォルト知識を得る。これにより得られたリストLには、これらデフォルト知識が推論連鎖木の深さ順にソートされる。これにより、矛盾に近い知識が先にチェックされる（近いもは信頼性がないとする）。

ステップ2： もし、リストLが空リストなら、矛盾解消手続きの失敗となり、さもなければ、ステップ3へ行く。

ステップ3： リストLの先頭のデータL_{head}（つまり、$L=[L_{head}|L_{tail}]$）の反証を宣言する。具体的には、L_{head}が見つかった推論ステップまでbackコマンドを用いて推論サイクルを戻し（途中、back_dataデータの*Flag*を*contradiction*に変更する）、L_{head}の否定を宣言する。

ステップ4： ステップ3でさらに矛盾が発生しなければ、L_{head}に依存した知識を消去し、手続きを終了させる。具体的には、先のback_dataデータの*Flag*の位置に*contradiction*があるものを消去し、さらに*unreal*があるものを*real*に変更する。さらに矛盾が発生したなら、ステップ3の宣言を無効にして（具体的には、back_dataデータの*Flag*の位置に*unreal*があるものを消去する）$L=L_{tail}$としてステップ2へ行く。

ＴＭＳは、矛盾解消のために矛盾発生の原因となる仮定に依存したノードの全てのstatus（つまり、inもしくはout）をボトムアップ的に変更することにより矛盾解消を図る。これに対して、本手続きの特長は、トップダウン的にこれら仮定ノードに相当するデフォルト知識の反証をテストして行くことにより矛盾解消を図ることである。ステップ4における*Flag*の変更はPrologの制御によりシーケンシャルに行われ、ＴＭＳにおけるdependency-directed backtrackingに相当する特別な制御はいらない。これにより、ＴＭＳに比べ、バックトラキングによる負担を軽減し、効率的に矛盾からの脱却を図ることが可能になる。

3.4. 矛盾解消手続きの呼びだし

矛盾解消手続きは、ルールのRHSに*contradiction*というキーワードを記述することにより達成する。例えば、KORE/IEでは、次のように用いられる。

cry: if cry(name=X,about=Toy) then *contradiction*.

そして、もしこのルールが適用されると、矛盾解消手続きが呼び出され、cry(name=X,about=Toy)を導き出す原因となったデフォルト知識をチェックし矛盾の解消を図る。

4. おわりに

KORE/IEでは、非単調推論を実現する枠組としてデフォルト推論を導入し、これを実現する技法としてＴＭＳ的矛盾解消手段を用いた。本アプローチの特長は、OPS5に代表される前向き推論型プロダクションシステムで非単調推論を実現したことであり、またＴＭＳに比べ効率良い矛盾解消手段を構築したことにある。尚、本研究は第五世代コンピュータプロジェクトの　一環として行われたものである。

参考文献

[Doyle 79] J.Doyle : A Truth Maintenance System, AI Vol.12, pp.231-272,(1979).

[Reiter 80] R.Reiter : A Logic for Default Reasoning, AI Vol.13,pp.81-132,(1980).

[新谷 87] 新谷：推論エンジンKORE/IE 矛盾メカニズムに基づく高速な推論エンジン, Proc. LPC'87, pp.233-242,(1987)

[新谷 86] 新谷, 他：問題解決支援環境KORE/(その2)知識記憶利用機構 KORE/EDENとその応用-, 情報32回大会,5L-9,PP.1145-1146,(1986).