知識ベース・システムの実時間制御への応用

Application of the Knowledge Base Systems to the Real-time Control

高田昌之[†]　中須賀真一　大須賀節雄[‡]

Masayuki TAKATA　Shin'ichi NAKASUKA　Setsuo OHSUGA

東京大学 大学院 工学系研究科 先端学際工学専攻

The Interdisciplinary Course on Advanced Science and Technology, The University of Tokyo

The knowledge base systems have not been used widely for the real-time control purposes, because of some difficulties such as the hardness on predicting the time period required to solve given problems using backtracking search. In order to ease application of the knowledge base systems to the real-time control purposes, the authors designed another problem solving system, and solved practical scheduling problem with this system. This problem solving system includes some meta-predicates, such as for limiting the time period for the backtracking, and for specifying the way to search the answer. This system is based on the Multi Layered Logic, in order to permit the specifications of predicates as the argument of meta-predicates. And it also equips meta-level control logics for efficient utilizations of meta-knowledges, which enables selective data recovery on the backtracking.

1　はじめに

知識ベース・システムは実時間制御の目的には敬遠されがちであるが、その様々な利点を切実に必要とする問題が現実に存在することに注目し、これらの問題領域に対応できる知識システムを構築できる環境の実現を目的とした。

知識ベース・システムが実時間制御に忌避される主な理由としては、推論の実現が主に探索などによって行なわれ、その処理に必要な時間の予測が困難なことなどが挙げられる。しかしその一方で、惑星間探査プローブなど未知の状況に適応せねばならないシステムや、自律型生産制御システムなど対象とする状態空間が巨大すぎて事前に全状態に対処する方法を準備できないシステムなどが存在する。これらの問題に対処するには、知識をルールの形で蓄え、状況に応じて適切と判断されるルール群を用いた推論を行ないながら、システム全体を制御していく能力が必要とされる。

ここでは、「実時間」とは制御対象の変化速度によって規定される時間内に、与えられた問題の解を得られること、もしくは解が見つけられないという結論を確定できることと定義する。現状では知識ベース・システムの問題解決に要する処理時間の上限を保証することが困難であるため、知識ベース・システムをフィードバック・ループに組み込まずに、状況を解析して人間に示唆を与える自動診断システムに応用するなど、分析型の問題への適用が行なわれてきた。一方、設計・制御・スケジューリングなどの合成型の問題については、知識ベース・システムを用いた問題解決システムはあまり実現されていない。

知識システムを制御などの合成型の問題に応用するためには、各状態で適用可能なオペレータ群から一つを選択した場合、どのような効果と副作用をもたらすかを予測し、その結果に従って適切な決定を下す必要がある。このため合成型問題の解決システムは必然的にシミュレーションを伴い、実時間性が要求される場合には将来予測を実際の現象が生起する時間以内で行なえなければならない。このような探索問題では一般的に組合せ爆発の可能性をはらみ、探索空間を上手に

[†] 現 電気通信大学 総合情報処理センター　Information Processing Center, The University of Electro-Communications at present
連絡先: 〒182 東京都調布市調布ヶ丘 1-5-1　TEL: 0424-83-2161 ext.4744　FAX: 0424-84-1598　E-mail: takata@cc.uec.ac.jp

[‡] 現 早稲田大学 理工学部　School of Sci. and Eng., Waseda University at present

制限できるか否かがシステム性能の鍵を握ることになる。ここに知識を階層化し、適切な場面で適切な知識を利用することで探索空間を狭めていく能力が必要とされる。このような目的には手続き的記述よりも述語的記述の方が適切であると考えられ、知識ベース・システム利用の必要性が発生する。

本研究 [高田 95] では、知識ベース・システムに基づく実時間制御システムを実現し、このシステムの上で実際に問題解決を行なうことによって、どのような知識が (特に知識の用い方に関する知識であるメタ知識が) 有効であるかを示すことを主眼とする。

2 実時間知識処理システムの要求仕様

これらの目的を達成するため、現存する実時間知識処理システムと実時間での問題解決を必要とする問題との分析を行ない、このような問題領域に適用可能な実時間問題解決システムがどのような仕様を持たなければならないかを考察した。その結果を以下に示す。

- 推論 (探索) の過程を充分制御できること
  メタ述語が利用できなければならない。このため処理系が基礎とする論理体系としては一階述語論理では不十分である。ここでは多層論理 (Multi Layered Logic)[Ohsuga 85] に基づく推論を行なった。
- プログラム・レベルで実時間性を記述できること
  メタ述語に時間を引数とする述語を加える必要がある。例えば、ある問題解決に要する時間が指定時間を超過したら、その問題を解決できないと決定できなければならない。
  また、問題解決機構の外部で何らかの例外状態が発生して、その時解決しようとしている問題そのものが不適切になった場合には、その場で解決を中断できなければならない。
- 知識の階層性を活かしたプログラミング
  多くのメタ・レベルの階層におけるプログラミングが、整理された形でわかりやすく行なえなければならない。またシステムがこのような情報を利用して、推論とデータのメタ・レベルに相応しい振る舞いをすることが要求される。
- 探索に用いる知識を切替えたり、編集できる能力
  推論時の環境に応じて、推論に用いる知識の選択や、推論システムが保持する知識の再構成が、よりメタな知識などによって柔軟に行なえなければならない。

3 実時間知識処理システムの基本設計

このような要求仕様を踏まえ、実時間推論制御処理系 RRCS のプロトタイプを開発した。

このシステムは多層論理 MLL に基づく述語論理システムであり、集合の扱いや属性の継承が容易に行なわれるようにした。また、この他に

- 探索の方法を指定する豊富なメタ述語を備え、
- 一定の時間内での問題の解決、
- 問題を解くために用いるルール群の指定、
- 推論のメタ・レベルの特定による backtrack 時のデータ回復の制御、
- 保有する知識の編集

などが行なえるようにした。

3.1 推論のメタ・レベルの特定による backtrack 時のデータ回復の制御

メタ・レベルの管理機能を推論システムに組み込み、 backtrack 時のデータの回復の制御を自動的に行なわせる機能を組み込んだ。

集合は構造を持った定数とも、その要素が変化していく容器のようなものとも考えられるが、このような捉え方の差は集合をどのようなメタ・レベルから見ているかの違いによって発生する。メタ・レベルに対応した視点から以下のような考察ができる。

- メタ・レベルの高い操作は、よりメタ・レベルの低い操作を制御するために存在すると考える。
- 集合の中身を操作するのは、その集合を使って推論を行なうための一種の環境設定 (下準備) と考える。ここから集合の操作はその参照よりもメタ・レベルの高いものと考える。
- 全ての集合に対してそのメタ・レベル sl を、推論にメタ・レベル el を考える。ここではメタ・レベルの高い方から $0, 1, \cdots, n$ と番号を振る。

ここで以下のような規則を設定すると、メタ・レベル間でのデータの操作や backtrack 時の回復の実行を適切に制御することができる。

$$
\begin{cases}
sl > el: & \text{追加・削除可, backtrack 時の回復不可} \\
\quad \cdots & \text{環境がどんなものか判っている場合} \\
sl = el: & \text{追加・削除可, backtrack 時の回復可} \\
\quad \cdots & \text{環境自体も試行錯誤しながら推論をする} \\
sl < el: & \text{追加・削除不可} \\
\quad \cdots & \text{上位から与えられた環境は操作できない}
\end{cases}
$$

これにより、例えばモデルの操作を行なう場合にも、推論のメタ・レベルをモデルを構成する構造のメタ・レベルより高くしたり同レベルにしたりすることによって、同一のモデル操作用の述語群を用いてbacktrack 時に回復されない変更や回復の対象となる変更を動的に実現することができる。

3.2 問題を解くために用いるルール群の指定

問題解決時に用いるルール群を package と呼ぶ集合で表現し、これをメタ述語で指定できるようにした。これにより、問題解決時に利用すべきルールを直接指定できるようにするとともに、述語表現されるルール群をグループ化することによって、プログラミングのモジュール化なども可能にすることができる。

メタ述語に指定する場合には、使用する package をルールの探索順位に従ってリストにして与える。これにより、問題解決の状況に specific な知識から一般的な知識までを順に指定することができる。

3.3 探索の方法を指定するメタ述語

RRCS では推論の制御のために、述語を引数にとるような述語が必須となる。このため、本質的に Multi Layered Logic が要求される。RRCS では、引数に述語を与えることのできる処理系として、可能な限りバリエーションに富んだメタ述語を準備した。

一般にメタ述語は解くべき述語と解を受け取る引数とを持つ。解を受け取る引数に変数が指定されると、メタ述語は全解探索の結果をリストにしてこの変数に返す。引数に与えられた述語に含まれる変数には何も unify されない。メタ述語全体としては、一つでも解が求まれば成功、見つからなければ失敗する。

解を受け取る引数として、値 nil が与えられると単一解探索を行ない、メタ述語は最初に見つかった解を唯一の解とみなす。このため、メタ述語に backtrack で戻って来ても、このメタ述語は新たな解を生成しない。探索が成功すると求まった解はメタ述語の引数に与えられたゴールに含まれる変数に unify される。また、探索が成功した場合であってもゴールとして与えられた述語の副作用は一切残らない。

さらに、解を受け取る引数として整数値が与えられた時には、指定された個数の解を求める。この場合にはメタ述語は最初に見つかった指定数個の解を発見するともう他には解はないものとみなす。この場合にも単一解探索と同様に、ゴールの評価に成功した場合でも一切の副作用を残さず、メタ述語に backtrack で戻って来てもこのメタ述語は新たな解を生成しない。このメタ述語で解が発見された場合には、その解はゴールの変数毎に見つかった順にリストにされ、ゴールの変数そのものに unify される。

3.3.1 package とメタ階層の管理

```
solve( pkgs, goal(X), Ans )
```

指定した package list の clause を使用して与えられた述語を評価する。探索の仕方を指定する引数が存在する場合、その指定に従って探索を行ない、発見した解をリストにして所定の変数に unify する。

システムが起動された時に選択されている default の package list は { defaultPackage } であり、どのような package 集合を選択してもこの package に定義された述語は最低優先順位で探索の対象となる。またシステム定義述語は最高の優先順位を持ち、どのような package にもオーバーライドされない。

```
shift-level( pkgs, level-num, goal(X), Ans )
```

推論のメタ・レベルを変更して、与えられた述語を評価する。レベル番号には、与えた述語を解く時に用いる推論のメタ・レベル番号を、正の値で指定する。

3.4 一定の時間内での問題解決の指定

```
solve-within( pkgs, due-time, goal(X), Ans )
```

一定の時間を示してその時間内に与えられた問題を解く。制限時間はミリ秒で表現し、その内部に制限時間を設けたメタ述語がある場合には、最も早く期限が来るものが有効となる。

探索中に、制限時間を使い果たした場合は、その時点までに求められた解がすべてであるとみなして、指定された方法で解を返す。一つでも解が求まればメタ述語自体は成功、そうでなければ失敗となる。

3.5 保有する知識の編集

知識ベース内のすべてのルールを、処理系が扱う一般のデータと同じ構造で実現することによって、ルール自体を知識によって操作編集できるような能力をも与え、将来の知識の自動獲得に至る可能性を実現できるようにした。

RRCS 処理系では個々の clause, それらを構成する term, clause からなる procedure, それらを集めた package などはすべて集合またはリストとして実現さ

れる。このため、知識ベースに蓄えられる全てのルールはデータとして取り扱うことができ、集合やリストを操作する述語群によって自由に編集できる。

編集された term, clause, procedure, package などの構造は与えられたメタ・レベルに従って、内容の編集や backtrack 時の回復動作の可否を制御される。このため、これらの構造のメタ・レベルが推論のメタ・レベルと等しければ、探索によって必要なルール集合を生成することも可能である。また、そのようなルール集合を受け取ったよりメタ・レベルの低い推論過程からは、その内容の変更は行なうことができず、また backtrack 時の回復動作に備えた操作も行なわない。

すなわち、 assert や retract がメタ階層を変えることで、単なるリストの変更として扱えるようになり、上位のメタ・レベルに属する処理が、下位の処理で利用される知識を自由に編集し、環境に適応した制御を可能にすることができる。

4 実時間知識処理システムの実問題への応用

さらにこのシステムの上でアプリケーションとして OTV ネットワーク [Nakasuka 92] のスケジューリング・システムを試作した言語処理系上に開発した。

OTV (Orbital Transfer Vehicle) とは、地球周回軌道に存在する燃料ステーションから燃料補給を受けて、人工衛星などを所定の軌道へ低コストで輸送するロケットのことである。複数の燃料ステーションと複数の OTV がある場合、その運用形態はネットワーク様になる。衛星が低高度軌道に打ち上げられてから目標軌道へ達するまで、どのステーションを経由しどの OTV で輸送するかを決定するのが OTV ネットワークのスケジューリング問題である。この問題は計画問題であるが、 OTV の故障や、天候の影響などによるペイロードの低高度軌道投入時刻の変更などのため、柔軟な対応ができるように、実時間でスケジューリングしながら運用できなければならない。

この問題の解決において、探索空間を狭めるために、その時の状況に対応したメタ知識を用いた枝刈りと各種の探索の仕方を指定したメタ述語を用い、探索空間を制御する事によって解の質をあまり低下させる事無く探索時間を減らす事が可能である事、その場合には状況の特徴を抽出してそれに対応したメタ知識を利用することが探索空間を減らすために有効である事が実験できた。

また、同一のルールを用いた推論のメタ・レベルを適切に制御する事によって、世界モデルの変更を決定的にも非決定的にも行なえるようにし、推論時のシミュレーションを容易に行なえる事も確認できた。

5 おわりに

本研究では探索時間などを制御するメタ述語の利用により、実時間での問題解決を行なう推論処理系を試作し、その有効性を確認できた。今後の課題としては、さらに大規模な問題を効率良く解決できるようにするための実時間推論制御処理系の改良、効率良く正確に状況の認識を行なうメタ知識の同定、知識の自動獲得につながる状況の特徴抽出と抽象化、戦略決定時に見られる問題にあまり依存しないメタ・メタ知識の同定とシステム組み込み、などが挙げられる。今後とも引続き、これらに関する継続的な研究を行なう予定である。

参考文献

[Nakasuka 92] Nakasuka, S. and Tanabe, T.: Built-in Simulator and Intelligent Manager Architecture for Autonomous Scheduling of OTV Network Operations, in *Proc. of IFAC Workshop on Spacecraft Automation and Onboard Autonomous Mission Control* (1992).

[Ohsuga 85] Ohsuga, S. and Yamauchi, H.: Multi-Layer Logic – A Predicate Logic Including Data Structure as Knowledge Representation Language, *New Generation Computing*, Vol. 3, No. 4, pp. 403–439 (1985).

[高田 95] 高田昌之: 知識ベース・システムの実時間制御への応用に関する研究, 東京大学大学院工学系研究科 先端学際工学専攻 学位請求論文 (1995).