

PSI 上のエキスパートシステム
開発支援ツール (3)

5K-3

— 推論処理部 —

永井 保夫、澤本 潤、久保野 秀雄、岩下 安男

(財) 新世代コンピュータ技術開発機構

進藤 静一

(株) 三菱電機

1. はじめに

本論文では、PSI 上のエキスパートシステム開発支援ツールPROTON[1]の推論処理(ルールベース)部について述べる。推論処理部の特徴は知識源(Knowledge Source:KS)とKSの制御を行なうメタ知識という概念を導入することで、推論処理のモジュール化と推論方式の柔軟な切り換え機能を備え、かつKEに対しては容易な知識ベースの構築可能な環境を提供することを目的としている。以下では、メタ知識と知識源について説明する。

2. メタ知識

メタ知識とは知識を利用するための知識、つまり推論を制御する知識でありルールベース・システム、フレームベース・システム、両者を統合したシステムなど様々なアーキテクチャを有したシステムに取り入れられている。多数のルールベース・システムやフレームベース・システムではメタ知識がメタルールやメタフレームといったメタ用の階層的な表現形式となっており、メタ知識と処理対象に関する知識の分離のなされる場合が少なくない。本システムでも同様に適用対象となる問題をサブ問題に分割して解決する(複数の知識源に分割して問題解決を行なう)場合、それぞれの処理(知識源)を制御する知識をメタ知識として分離し、システム全体の見通しを良好なものとしている。

表現形式

メタ知識は、メタルール、メタルールの制御戦略(first hit strategy 又は、CRS)、メタルールの適用-実行サイクルの停止条件などから構成されており、メタルールの形式は現状ではプロダクションルールの前向き推論用ルールと同様にしている。メタルールの付加情報にはルールの適用についての指定及びルールの適用優先度がある。メタルールの条件部はWorking Memory(WM)要素パターン、ユーザ定義の述語(メソッド)又はESP(KLO)述語などのand結合となっており、実行部はWM要素取り扱ひ用述語、ユーザ定義の述語(メソッド)又はESP(KLO)述語などのand結合と知識源名(但し、KSが後向き推論を行なう時は後向き推論に対して与えられるゴールパターンも付加される)から構成される。

知識源(KS)の制御

知識源(KS)の制御はメタ知識とメタ知識用の推論機構を用いて行なうものである。メタ知識をメタルールという記述形式にしたのは、メタルール表現により様々なKSの起動-実行メカニズムが実現可能だからである。この場合、ユーザはルールによる推論機構で不可欠となるWMの設計をKSの制御用とドメイン用に使い分けて設計することが好ましい。

KSの制御(つまり、メタ知識の推論)のメカニズムを要約すると以下のようになる。

1. メタ知識(ルール)から適用可能なものを1つ選択

2. 選択されたメタ知識(ルール)の実行部に記述されているKSを実行

3. KSの実行結果がsuccessの場合(FC用またはMIXED用は常にその結果がsuccess、BC用の結果はsuccess/fail)、メタ知識(ルール)の実行部に記述されたアクション(通常、WM上の制御用情報に副作用を発生させる)を実行させる。結果がfailの場合には何もしない。

4. 1へ戻る

さらに、ユーザがこのようなメタ知識を拡張して使用すればトラックボードモデルの実現も可能である。(但し、その場合WM機能も拡張して用いる必要がある)。

上記の項目1と2はそれぞれ先程述べたメタルールの評価(選択)-実行(recognize & act)サイクル中の評価(選択)と実行に相当する。これには、アジェンダ(conflict set)を生成しないでメタルールの記述順序だけに依存した形でルールの条件部が満足されたものを、即、選択する場合と、すべてのルールの条件部を評価してアジェンダ(conflict set)を生成し、何らかの評価に基づいてその中から1つを選択する場合の2つが与えられる。本システムでの後者のケースにあたるCRS(conflict resolution strategy)は、OPS5のLEX戦略とほぼ同様でありここでの説明は省略する。

最後に、KSの制御メカニズムについての概略を図により説明する。メタ知識中のmeta_rule1が選択されて、それを実行すると知識源1が起動され結果として知識源中のルールが実行される。知識源がsuccessして終了した場合にはmeta_rule1の右辺のstmntkが実行され、WM中のKS制御用情報がupdate(delete)される。これがKS制御の基本サイクルとなっている。ユーザはオプションとして、メタルールの選択-実行サイクル毎に停止条件をチェックできるような記述が可能である。

3. 知識源

知識源を導入したメリットはルールを複数の知識源という形式にグループ(モジュール)化して問題解決を行なう事で探索空間の絞り込みによる推論処理の効率化を図ることである。本システムで述べているKSは各々が独立して問題の解決を行ない、1つのKSの実行中に他のKSをよびだすという事は行なわない。又、対象となる問題が複数の問題に分割できない(複数のKSに分割不可能な)場合はKSを1つとみなして処理をおこなう事も可能である。推論方式はKS毎に決定されており、後述するFC、BC、MIXEDの3種類が許されている。

表現形式

KSの推論制御情報はルールの適用-実行サイクルの制御戦略(CRS 又は、first hit strategy)、推論形式(FC、BC、又はMIXED)、及びルールの適用-実行サイクルの停止条件から構成されている。さらに、KSの推論形式がBCの場合には、そのKSに静的にゴールを指定することも可能である。

ルールは処理対象に対する手続的表現であり、関連したものがKSとしてまとめて取り扱われる。ルールの形式にはFC用とBC用の2種類があり、先程述べたKSの推論方式がFCの場合KS中のルール表現はFC用ルールのみで記述され、BCの場合にはBC用ルールのみから記述され、MIXEDの場合KS中のルール記述はFC用ルールとBC用ルールが混在して記述される。ルール記述にはルールの条件部と実行部以外に付加情報として優先度やルールの適用記述の指定が記述可能となっている。

KSが前向き推論を行う、つまりKS中のルールがFCだけから記述されている場合、実行部はユーザ定義の述語、ESP(KLO)述語やWMのupdate処理を行なう組込み述語make、modify、removeという3つの述語などから構成されている。

BC用ルールの構成もFC用ルールとほとんど同じである。但し、注意しなければならないのは、FCと異なってルールの実行部がWM要素のパターンであるか又はpropose 述語(引数としてWM要素のパターンが含まれている)である点である。前者の場合はゴールに対して後向き推論が行なわれてもWMへアサーションされない(副作用がない)。しかし、後者のpropose 述語をルール中で利用した場合はBC

推論の結果としてWMに副作用が起こりWMへのアサーションが行なわれる。ユーザはこのような機能を選択したルールの記述が可能である。また、BCルールの実行中に、ask_user(query)機構を起動してユーザに対するファクトに関する問い合わせを行なうか行なわないかはファクト中のテンプレートに付加手続きとして指定する。

WM要素パターンは対象表現部における構成要素を3つ組形式として表現するか又は構成要素間の関係を述語として表現する。前者はエレメントと言われるシステムの構成要素を属性-属性値で表現しており、1つのエレメントに対して複数の属性-属性値群が記述可能である。対象表現部における構成要素表現ではTE間にhas_a関係が定義可能となっている。TE間での要素(実体)継承は、上記の関係に基づいて行なわれる。このような要素継承関係をルール・シンタックスでは一般にWM要素パターン中の属性名で表現できる。後者の構成要素間の関係は、二項関係や多項関係が記述可能である。このようなエレメントやリレーションはテンプレートとそれによって生成される実体からなっている[2]。

推論形式

KSで実行される推論形式には次の3種類がある。

- forward chaining (FC)
- backward chaining (BC)
- mixed strategy (MIXED)

各処理手順について以下に示す。

前向き推論(FC)はデータ駆動型推論又はパターン主導型推論と呼ばれ、データが入力されることによりそれに適したルールの選択-実行が行なわれ、その結果WM中のWM要素が更新されるといふ処理の繰り返しである。代表的な例としてはOPS5などがある。

後向き推論(BC)は目標駆動型推論と呼ばれ、目標を設定しこれに従って後向き用ルールを起動したりその際にユーザの問い合わせを行なって目標の決定(検証)をする。代表的な例としてはEMYCINがある。

混在型推論(MIXED)は前向き推論で仮説を生成して、各仮説に対して後向き推論を適用して仮説の絞り込みを行なっていくという処理の繰り返しである。本システムの混在型推論はFC主体で動作するようになっており、FCの実行に際して必要な情報を得るためにシステムが自動的(補助的)にBCを起動するものである。

ルールの適用方式

KS中のルールの適用方式にはfirst hit strategyとCRSの2つがある。従って、3種類の推論形式に対して上記の適用方式が考えられる。しかし、FC及びBCのルールの適用メカニズムは省略して、混在型についてのみ説明する。

混在型はFCの実行に際して、必要な情報を得るためにシステムが自動(補助)的にBCを起動するというメカニズムをとっており、BC起動タイミングと混在型推論自身の制御は現在検討中である。しかしながら、以下には具体的実現の一例として、first hit strategyとCRSについて述べておく。

1. first hit strategy

- 1) FC用ルールだけを評価(適用)-実行する。
- 2) 適用可能なFC用ルールがなくなったら、すべてのFC用ルールの条件部のパターンからゴールパターンを生成し、BC用ルールを使ってこれらのゴールパターンを検証する。
- 3) WMに副作用が起こった場合には、1)に戻り、BC用ルールだけの選択-実行サイクルを繰り返す。そのサイクルで停止条件が満足された場合、混在型推論は終了する。
- 4) BC用ルール実行後、WMに副作用が起こらなかつたり、FC用ルール実行中に停止条件が満足された場合に、混在型推論は終了する。

2. CRS

1) FC用ルールで条件部が発火可能なものをすべてconflict set(agenda)に収容し、その中の1つを選択-実行するというサイクルを繰り返す。

2) 次に、conflict set(agenda)中のFC用ルールが空になったら、すべてのFC用ルールの条件部のパターンをゴールパターンとしてBC用ルールを起動する(WM要素パターンの否定の場合はWM要素パターンをゴールパターンとする)。

3) BC用ルール実行の結果として、WM中に副作用が発生した場合は、1)へ戻り、FC用ルールだけの選択-実行サイクルを繰り返す。そのサイクルにおいて、条件が満足された場合、混在型推論は終了される。

4) BC用ルールの実行結果としてWM中に副作用が発生しなければ、混在型推論は終了される。

4. おわりに

メタ知識を利用して柔軟な知識源の制御を行う推論制御(ルールベース)部について説明した。現在、推論処理部は実装段階に入っており、今後はブラックボードモデルの実現などアプリケーションを通じて種々の問題に適したメタ知識の記述とその適用-実行方式の検討を行っていく予定である。

参考文献

- [1] 澤本等、「PSI上のエキスパートシステム開発支援ツール(1)」、第33回情報全国大会5K-1、1986、10
- [2] 久保野等、「PSI上のエキスパートシステム開発支援ツール(2)」、第33回情報全国大会5K-2、1986、10

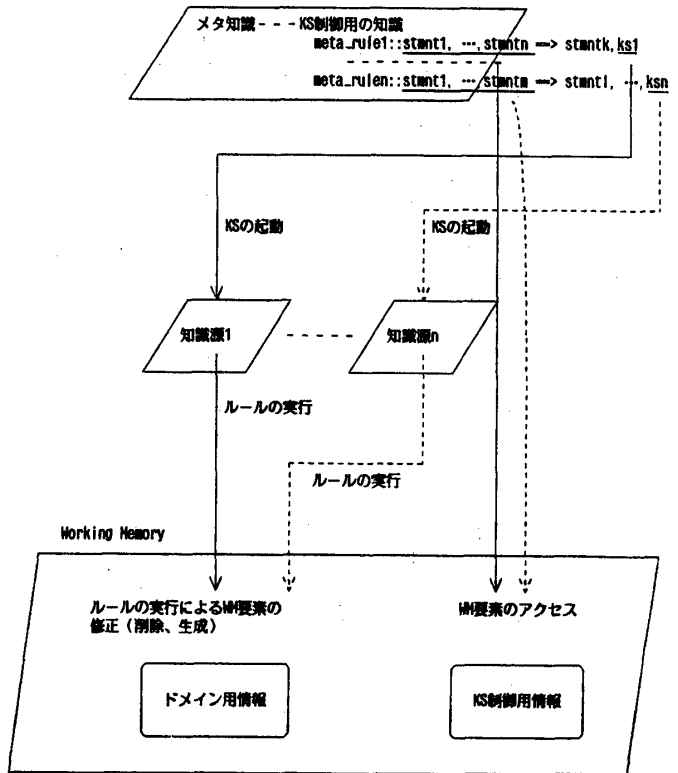


図. KSの制御-起動メカニズムの概略