

TR-459

プロダクションルールの機能的側面

滝 寛和

March, 1989

©1989, ICOT

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

1. はじめに

我々が、プログラムを作成する場合には、プログラム言語の基本的な機能を組み合わせて、目的とする複雑な機能を実現している。この場合、プログラマーは、プログラム言語の持つ基本機能を自分の使い易いより高いレベルの機能で記述しなおしてそれを利用している場合が多い。それは、サブルーチンやプログラムの記述スタイルとして表されている。同じプログラムの基本要素でも、プログラマーの見方(解釈)によって、別の機能となる。たとえば、「値が1」の意味するところは、色々である。たとえば、存在を表す(値が1で存在する、他の値では存在しないを示すなど)、個数を示す、や属するグループ名が「1」である、やIDナンバー(固有名称)を示すなど種々の意味表現となる。

知識ベースの構築においても、同様のことがおこりうる。同じ知識ベースであっても仕様環境(解釈条件)が変わるとその意味するところは、大きく変わる。しかし、この場合、論理的、つまり演繹的な意味は変わらないことに注目しなければならない。これは、より高次の処理(解釈など)によって影響を受けるものである。たとえば、現在の非単独推論や常識推論で行われている知識の無矛盾な拡大解釈に見られるような演繹を越える推論機構によって知識の機能や表現する内容が変わることはその例である。

知識の解釈の多義性は、意味を考慮した知識ベースのメンテナンスを困難なものとしている。つまり、知識ベースのメンテナンスには、論理レベル、確率レベル(確信度のメンテナンス)以外に、解釈レベル、機能レベルでのメンテナンスが必要である。

知識獲得における知識のリファイン問題を考えると、その知識の使用目的(解釈のされかた)によって、知識の不足、矛盾や冗長が検出される場合が多い。たとえば、MORE[Kahn 85]は診断のドメインモデルを使って、同じ徴候から複数の仮説が導かれる場合に、各仮説を同定(区別)する新たな徴候の獲得を行う。これは、診断の目的からみて、仮説を明確に区別できる知識が必要なのである。診断のための解釈があってこそ、このリファインが可能となる。

計算機の構成エキスパートシステムR1のルールを分析して、よく利用されるプロダクションルールの構造を整理したSEARプロジェクトがある。これは、ある機能を実現するルールをテンプレートとして準備することで、知識獲得を支援することをわらっている。

知識獲得では、解釈が極力1意になるような知識表現を準備できることが望ましい。知識獲得支援システムEPSILON/EM[滝 87][EPSILON/One[棒 88][大崎 88]では、専門家が自分の知識を表現するのに、オペレーション単位で知識を表す専門家モデルを利用している。各オペレーションの意味は、オペレーションの機能表現で規定されている。この機能は、診断型のプロダクションルールの分析から得られたものである。EPSILONでは、知識の解釈問題を知識の機能表現問題として考えている。

専門家が機能レベルで知識を表現でき、かつメンテナンスできる知識表現をプロダクションルールを機能的な見地からみなおしてみることで、探ることになる。そこで、ルールの機能を表

す要素として、次の要素が関連しているのではないかと仮説を立てた。

- (仮説1) ルールの機能は、条件部/アクション部の基本機能の組み合わせで決まる。
- (仮説2) ルールの機能は、WMに書き込まれるデータタイプで決まる。

本論文では、これらの仮説を検証するために、まず、プロダクションルールの基本的機能の分析を行い、その後専門家モデルのオペレーションタイプとの関係について、論じる。

2. プロダクションルールの機能要素

プロダクションルールは、OPS5[Brownston et al. 85]等のエキスパートシステム構築ツールの知識表現として、利用されている。基本的なルールの構造は、「IF-THEN」の形式で表される。また、ルール以外のデータは、WM(ワーキングメモリー)に表されている。今、このルール表現の機能を見るために、ルールは次の表現で表されるものとする。

条件部(IF部):

(要素名 *属性名 値または代入される変数)

(関数 値または参照される変数 代入される変数)

アクション部(THEN部):

(WM操作関数 要素名 *属性名 参照される値)

プロダクションシステムの処理は、認知実行サイクルによって制御されている。まず、条件部の成立するルールが選択されて、その中の1ルールが、競合解決戦略によって選ばれる。選ばれたルールのアクション部が実行されて、WMの内容が書き換えられる。新しい内容のWMで条件の成立するルールが次に評価される。動作としては、単純なサイクルであり、WMに対する作用は、追加/削除/変更のみであるが、機能的にみると、選択したものにフラグを付けて識別を行ったり、条件に合う2つの要素を組み合わせで新しい要素を作ったりしている。本章では、まず、プロダクションルールの細かい機能について分析し、その組み合わせとデータの意味付けから機能を探ることとする。

2.1 条件部の機能

条件部の機能として、第1に挙げられるのが、同値判定機能である。これは、次の様な表現となっている。

例1: 同値判定機能(同形式パターンの存在確認)

(animal *habitat sea): 海に棲む動物の存在

(animal *habitat seaside *has wings): 海辺に棲み羽を持つ動物の存在

次に挙げられるのが、参照検索機能である。これは、次の表現で表される。

例2: 参照検索機能(属性値の参照)

(animal *habitat <X>): 動物の棲息地を参照して、Xに代入する

(animal *habitat <X> *has fins): 鰭(ひれ)を持つ動物の棲息地を探す。

以上が、基本的な機能であるが、その他に、参照した値を持つ変数に対する判定と演算がある。

例3: 比較判定 (a) 同値判定

(animal 'legs {<X> = 2}): 動物が2本足であるか調べる。
(animal 'habitat land 'legs {<Y> = 4}): 陸に棲む動物が4本足であるかを調べる。

これは、例1の様に、(animal 'legs 2) や (animal 'habitat land 'legs 4) と記述してもよい。

例1との大きな違いは、

(animal 'habitat land 'legs <X>)
(vehicle 'wheel <X>): 陸に棲む動物の持つ足の数と乗り物の車輪の数が同じかを調べる。

この例の様に、2つの要素の属性の等価性を調べることができる。

(b) 参照検索

上記の例は、「見方により、「2本足の動物を探す。」
「陸に棲む4本足の動物を探す。」
「陸に棲む動物の持つ足の数と同じ数の車輪を持つ乗り物を探す。」とみることが出来る。

(c) 比較判定

(animal 'legs {<X> >= 2}): 動物が2本以上の足を持つか調べる。
(animal 'habitat land 'legs {> 1 < 10}): 陸に棲む動物が1本より多く、10本までの足を持つか調べる。
(animal 'habitat land 'legs <X>)
(vehicle 'wheel {<Y> > <X>}): 陸に棲む動物の足の数より多くの車輪を乗り物が持つかを調べる。

(d) 演算

この演算は、数値演算を意味する。一般にプロダクションシステムでの数値演算は、compute関数または、LISP/Prologなどの再帰呼び出しで行われる。

(animal 'legs <X>)
(compute <X> / 2 <Y>): 動物の足の対の数を求める。演算は、アクション部でも利用される。

2.2 アクション部の機能

アクション部で扱われる値は、固定値/要素の値/演算で求めた値がある。ここで注目すべきことは、条件部で求められた値(検索や演算で求められた値)がアクション部へ伝播することである。

アクション部の基本的な機能は、WMへの値の追加/削除と変更である。そこでの値は、「条件部からの伝播したもの」と「固定値(アトムまたは、数値)」である。WMに対する操作は、要素レベル/属性レベル/値レベルについて、「値の設定/変更」「属性の追加/削除」「要素の追加/削除」が行われる。

2.3 値の意味分類

プロダクションルールの機能の意味は、単に、上記に説明した機能で規定出来ない。次の例は、構文や操作は同じであるが、機能的な意味付けは異なる。

例4: 意味付けと機能の関係

(a) 陸に棲む6本足の生物は、昆虫である。

IF ((creature 'habitat land 'legs 6) <insect>)
THEN (modify <insect> 'genus insect)

(b) 材質が鉄で、倉庫にある荷車を選ぶ。

IF ((gear 'place in-stock 'material steel) <goods>)
THEN (modify <goods> 'use selected)

例4では、(a)(b)は、同じ構文をしている。しかし、(a)は言い換え(変換)であり、(b)は選択を行っている。これは、厳密には、条件部

やアクション部の属性と値の関係で、そのルールの機能が決まることを意味する。もう少しラフにみると、アクション部の属性とその値が意味することで、機能が決まると言える。この例では、属性 'useの取りえる値は、selectedかnon-selectedのいずれか(negation as failure: その属性がないときは、否定である。この考えを考慮にいと、属性 'useがないときは、non-selectedと考える。)である。つまり、存在の有無を示すことになり、「選択」機能を持つことになる。また、属性 'genusの取りえる値は、非常に多く存在する(哺乳類、鳥類など)。つまり、言い換え(変換)の他に、この取りえる値のどれかに「分類」されたと見こともできる。物の定義などで、我々がよく、XXの性質をもつ物をYYと呼ぶ(名付ける)と言う表現を使うがこれは、言い換えでもあり、分類でもある。

(a) 存在

属性が取りえる値が、2種類である場合で、且つ、一方が他方の否定になる場合を言う。この値をWMに書き込まれることは、選択/選別の操作を意味する。ただし、(大きい、小さい)の対は、「順位」を暗示する。また、値が1種類のときは、その属性が存在しないときにはその値の否定を意味するものとして、この中に含める。

(b) 分類

属性が取りえる値が、3つ以上で有限個である場合を言う。ただし、取りえる値が、1から100までの整数のような場合で、他のルールが、その値に数値演算を施すような場合には、「その他」とする。(属性値が要素名や属性名を示す場合には、階層化分類とみることができる。)

(c) 順序

属性が取りえる値に全順序がある場合を言う。

(d) 上記以外の属性値(個数、サイズ、メッセージを示す文字列など)

属性が取りえる値が無限に存在する場合や数値(存在と分類に使用されない数値)を言う。

3. プロダクションルールの機能分類

まず、条件部の機能を判定(比較)、参照と数値演算の3つとする。値は、条件部からの伝播したものとアクション部での固定値とに分ける。また、参照/判定に関係する要素が、1つの場合と2つの場合に分ける。まず、この分類をした後、値にタイプ付けを行う。プロダクションシステムのWMでは、同じ要素名が複数存在できるが、以下では、要素名と属性名の対は、1つだけしか許さないことにする。要素は、フレーム認知論表現の1フレームと考える(ただし、継承/デモンはないものとする)。

3.1 機能の組み合わせ

(type1-1) ある要素の内容を判定して、その要素の属性に固定値を書き込む。(属性は、参照したものと操作を受けるものが同じであっても異なってもよい)

(type1-2) ある要素の内容に演算を行い、その値をその要素の属性に書き込む。

(type1-3) ある要素の内容を判定して、別の要素の属性に固定値を書き込む。

(type1-4) ある要素の内容に演算を行い、その値を別の要素の属性に書き込む。

(type2-1) ある2つの要素の内容を比較して、その一方に固定値を書き込む。

(type2-2) ある2つの要素の内容を比較して、第3の要素に固定値を書き込む。

(type3-1) ある2つの要素の内容を比較して、その一方に他方の値を書き込む。

(type3-2) ある2つの要素の内容を比較して、第3の要素にいずれか

の値を書き込む。

(type3-3) ある2つの要素の内容を演算して、その一方に演算結果を書き込む。

(type3-4) ある2つの要素の内容を演算して、第3の要素に演算結果を書き込む。

(ただし、判定/比較は、参照した後、判定/比較を行っているときのみ。演算は、参照した後、演算を行っているときのみ。単なる参照は、何も行わない演算であらう。さらに、(type3-1)を2度適用すれば、値の交換ができる。)実際のルールでは、上記の場合より複雑なものが存在するが、この基本形の条件部に、その他の要素の判定が加わったものとみることが出来る。アクション部の追加/削除/修正についてみると、削除は否定値を書き込むものであり、追加は、否定値を修正するものとする。

3.2 書き込まれる値と機能の組み合わせ

さらに、値は、存在/分類/順序/その他に分ける。ここには、操作を受ける要素が条件部から伝播したものかどうかの情報について組み合わせをみることにする。

(1) 存在

(type1-1) ある条件を満たす要素を選択する。(2群に分類する)

(type1-2) 稀有。(存在値を計算で求めるのは、論理演算である)

(type1-3) 並立/排他する要素の選択を行う。

(type1-4) 稀有。(存在値を計算で求めるのは、論理演算である)

(type2-1) より条件に合う要素を選択する。

(type2-2) ある要素間の関係による要素選択を行う。

(type3-1) 共に成り立つか、共に成り立たないかを求める。(内容は、存在値であるから2要素同時の選択)

(type3-2) 3要素同時選択。2要素並立、1要素否定の選択。

(type3-3) 並立性や排他性による選択。(存在値を計算で求めるのは、論理演算である)

(type3-4) 3要素間の並立性や排他性による選択。(存在値を計算で求めるのは、論理演算である)

(2) 分類

(type1-1) ある条件を満たす要素に分類指標を与える。

(type1-2) 稀有。分類指標を計算で求める。

(type1-3) 2要素を同じ分類とする。新しい分類を別の要素からもらう。

(type1-4) 稀有。分類指標を計算で求め、他の要素の分類を決める。

(type2-1) 2要素間の関係からその要素の分類を決める。

(type2-2) ある要素間の関係で要素の分類を決める。

(type3-1) 2要素を同じ分類とする。

(type3-2) 3要素同じ分類。ある要素を2要素のいずれかと同じ分類にする。

(type3-3) 稀有。

(type3-4) 稀有。

(3) 順序

(type1-1) ある条件を満たす要素に順序性のある分類値を与える。

(type1-2) 要素の持つ値を交換して、他の要素と順序比較出来るようにする。

(type1-3) ある要素に関係ある要素の順位(順序性のある分類値)を決める。

(type1-4) 稀有。

(type2-1) 稀有。

(type2-2) 稀有。

(type3-1) 2要素を同じ順位(順序性のある分類値)にする。ある要素の新しい順位を別の要素からもらう。

(type3-2) 2要素の順序関係が、ある要素の順位をいずれかの要素と

同じになるように決める。

(type3-3) 他の要素の値を基準にして、ある要素を順序比較出来るようにする。

(type3-4) 稀有。他の要素の値を基準にして、ある要素を順序比較出来るようにする。

(4) その他

(type1-1) 要素の持つ複数の内容を概略表現に変換する。

(type1-2) 要素の持つ値を別の基準で換算し直す。

(type1-3) 稀有。不明。

(type1-4) 関連する要素の値を補正して、別の要素に伝播/継承する。

(type2-1) 他の要素との関連からある要素の値を概略表現に変換する。

(type2-2) 稀有。

(type3-1) ある要素の値を別の要素に伝播/継承する。

2要素の値を適切な方に合わせる。

(type3-2) ある要素の値を別の要素に伝播/継承する。

2要素の値の内適切な値を第3要素に伝播する。

(type3-3) 他の要素の値を基準にして、ある要素の値を換算/変換する。

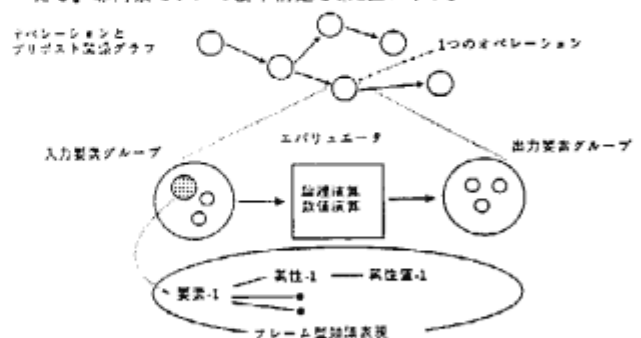
(type3-4) 他の要素の値を基準にして、ある要素の値を換算/変換する。

この組み合わせ結果からみて、ルールの機能的な意味付けは、WM操作の際のデータのタイプであることが分かる。存在/分類データの場合は、そのデータがルールの機能を表すのに大きな影響をもつ。順序データは、ルールの基本機能と合わせることで(たとえば、(type3-1)を2回適用し、値を入れ換えることで)要素の順序付け機能を実現できる。その他のデータは、値の変換/換算や伝播/継承機能に関係する。

4 専門家モデルとプロダクションルールの機能

4.1 専門家モデルの概要

専門家モデルは、オペレーションの集まりとして、専門家的问题解決知識を表わす知識表現である。各オペレーションには、処理される要素が入力され(これを入力要素グループと呼ぶ)、処理結果(出力要素グループ)は、別のオペレーションに渡される。このオペレーションを要素が流れるデータフローが問題解決の過程になる。EPSILON/Oneは、まず、このオペレーションとその関係(プリポスト関係:コントロールフロー)を獲得する。その後、専門家がオペレーションにタイプを指定すると、EPSILON/Oneは、そのオペレーションを実現するのに必要な情報をそのタイプに依存する形で、獲得する。このオペレーションのタイプは、診断型のエキスパートシステムの知識ベースの分析から得られた。それらは、「選択、分類、組み合わせ、順位付け、変換、入力と出力」である。また、この機能は、ナレッジエンジニアのルール記述ノウハウとも習える。専門家モデルの基本構造を第1図に示す。



4.2 オペレーションタイプとルールの機能の比較

オペレーションのタイプは、選択、分類、組み合わせ、順位付け、変換、入力と出力であるが、この内、入力と出力タイプは、プロダクションルールに、read/write/fileに関する演算を持つものに対応する。その他の対応は、一応、次の表1ようになる。

専門家モデル	ルール分析
選択オペレーション	存在データの書き込み
分類オペレーション	分類データの書き込み
組み合わせオペレーション	その他データの書き込み + 値の伝播
順位付けオペレーション	順序データの書き込み + 値の交換(注1)
変換オペレーション	その他データの書き込み + 値の換算

表1: 専門家モデルのオペレーションとルール分析結果の対応

注1: (type3-1)の利用による値の交換等で実現

5. データベースの操作とオペレーション機能の関係

専門家モデルは、分割された非常に小さいデータベースを操作/更新して行くモデルである。そこで、データベースの基本操作とオペレーションタイプの機能の関係をみることにする。関係代数でのデータベースの基本操作[Date 86]は、次の8つである。

SELECT(比較判定による参照/選択), PROJECT(属性の参照/選択), PRODUCT(組み合わせ), UNION(和集合), INTERSECT(積集合), DIFFERENCE(差集合), JOIN(要素/属性間の関係づけによる要素の生成), DIVIDE(属性の存在の有無による選択)

これをオペレーションタイプと機能の対応をみると、次の様になる。

専門家モデル	関係代数
選択オペレーション	SELECT, DIVIDE, INTERSECT, DIFFERENCE
分類オペレーション	(SELECT, DIVIDE, INTERSECT, DIFFERENCE)
組み合わせオペレーション	JOIN, PRODUCT
順位付けオペレーション	PROJECT
変換オペレーション	(データの変更)に相当
オペレーション外操作	UNION

表2: 専門家モデルのオペレーションと関係代数のオペレーションの対応

この表から分かることは、データベースの基本操作の機能表現の方がより詳細な機能表現となっていることである。また、変換オペレーションの機能は、データベースの基本操作外にあり、データベースを利用するアプリケーションプログラムでの操作に含まれる。オペレーションに入力される入力要素グループのAND関係が、データベース操作のUNIONに対応している。

6. まとめ

プロダクションルールの機能的な表現に重要な要素としては、WM操作の際のデータの意味であることが分かった。また、その詳細な意味付けには、基本機能「判定」「演算」「条件部からアクション部への要素/値の伝播」であることが分かった。これらの分析から、専門家モデルのオペレーションタイプが導かれる可能性を得た。また、今回の検討では、1ルールに1機能を対応づけ

る様に分析したが、複数のルールで、機能を表す場合やルールのアクション部のデータを別のルールがどの様に処理するかで、そのルールの機能が決まる場合もある。今後の研究として、上記の複数ルールの機能分析と機能レベルでのルールリファインが課題となっている。

謝辞

本研究に、助言頂いたICOT LANRS(学習と高次推論研究)グループメンバーならびに、古川次長に感謝いたします。また、第5世代コンピュータシステムプロジェクトにおいて研究の機会を与えて頂いたICOT第5研究室藤井室長、瀬所長に深く感謝いたします。

参考文献

- [Brownston et al. 85] Brownston, Kant, Farrell, Martin, "Programming Expert System in OPS5", Addison Wesley, 1985
- [Date 86] Date, C.J., An Introduction to Database Systems, vol.1, Addison Wesley, 1986
- [Kahn 85] Kahn, G., Nowlan, S. and McDermott, J.: Strategies for Knowledge Acquisition, IEEE Trans. Pattern Analysis and Machine Intelligence, PAMI7-5, 511/522, 1985
- [大崎 88] 大崎 宏、椿 和弘、滝 寛和: 知識獲得支援システム EPSILON/One(2), 第8回 知識工学シンポジウム資料, 1988年10月
- [滝 87] 滝 寛和、他: 知識獲得における専門家モデル、SICE「ヒューマンモデル」3部会合同研究会、予稿集、1987年7月
- [椿 83] 椿 和弘、大崎 宏、滝 寛和: 知識獲得支援システム EPSILON/One(1), 第8回 知識工学シンポジウム資料, 1988年10月