Managing System of Knowledge Base in Machining

Yasumi NAGASAKA, Makio AKIYAMA, Hideyuki OTAKI and Yoshio ISHIKAWA

It is possible to incorporate a system which manages a knowledge base into an expert system. Because, by using this system, we can rebuild knowledge bases which have no logical contradictions with knowledge of experts and phenomena which we experience daily. However, it is difficult to build beyond systems in the machining field due to the existence of many kinds of knowledge, if we want to achieve rebuilding of the knowledge bases using current representation language. Thus, we developed a truth maintenance system for knowledge bases by using a new predicate-logic representation language. This paper describes how to manage knowledge bases and how to treat the functions in a truth maintenance system using this system when contradictions occur in the practical machining problem.

Key Words: Expert System, Artificial Intelligence, CAM, Cutting, Milling, Knowledge Base, Frame Representation, Predicate-Logic Representation, TMS

1. 緒 言

エキスパートシステムの知識ベースは、専門家の知識や経験に基づいて構築される。しかし、専門家の知識は、装置の稼働状況や経済的な周辺環境などをしながら更新、蓄積していくという過程を踏む、それに応じて知識ベースも更新される必要がある。そのためには、知識ベースを矛盾なく再構築し、知識間の整合性を常に維持するような管理機構が必要となる。このような管理機構として、TMS (Truth Maintenance System)や、ATMS (Assumption-based TMS)が研究開発され、スケジューリングシステムなどの管理分野で具体化されている。

しかし、機械工学分野の知識ベースの管理に適用した例は、著者の知る限りない。機械工学分野の管理家は、一度決定されれば変更のほどとない知識と、逆に時々刻々変化する知識とが混在している種多な知識の中から、最適な知識を選び出し利用している。それゆえ、エキスパートシステムを構築するとなると、必ず複数の知識表現を混ぜざるを得ない。そのため、従来的 TMS や ATMS のように、限定された知識表現でしか利用できないものであっては、これに十分に対処しきれない。

そこで、著者らは知識表現の異なる知識ベースの統合的な管理を目的として研究開発を行ってきた。そして、フレーム表現やルール表現で構築された知識ベース、さらには C 言語による数式などの知識ベースを有機的に結合し、知識表現の異なる知識ベース間の推論を可能にした。しかし、TMSのような知識ベースの管理機構を兼ねているものではなかった。

そこで、本研究では、まず知識ベースの記述形式をフレーム表現の記述形式と同一 S 式 (LISP 言語で処理するデータ: Symbolic expression) と統一し、知識表現の異なる知識ベースの統合的な管理ができるように配慮した。これにより、フレーム表現を基にした知識ベース間の統合性や即席性、さらには知識ベースの管理機構の組み込みやそのアルゴリズム開発に効果的に機能することになった。つまり、知識ベースの管理機構 TMSについては、新たに出力論理表現の Prolog 処理系を開発し、これを適用した。

本報告は、この処理系を用いて機械加工条件を知識ベース化した機械加工管理システムの知識表現、およびこれを管理する TMS の具体例を示した。
2. 開発した述語論理表現（Prolog）処理系

Prolog は、ユニフィケーションとバックトラック機能を特徴とする述語論理表現の言語である。図 1 には、著者の開発した Prolog の記述と一般の Prolog（DEC-10 Prolog あるいは Prolog-KABA）の記述の比較例を示したものである。

一般的の Prolog が「述語名（引数，引数，…）」を基本記述形式とし、記述形式を「（述語名 引数 引数 …）」とした。

この記述形式は、述語名と引数が S 式内で表現されているので、述語名を変数として持つことができる。そのため、メタ知識表現が可能となるので、3・4 節で後述する TMS の「ノード生成」と「矛盾解消」に適用できる。しかも，フレーム表現にある論理機能により，矛盾解消のアルゴリズムをルール表現などの知識表現にも簡単に適用できる。

3. 知識ベースの管理機構を持つ機械加工管理システム

3・1 機械加工管理システムの構成 図 2 は、知識ベースの管理機構を持つ機械加工管理システムの構成を示したものである。

機械加工管理システムは、加工条件推論部、知識ベース（KB：機械加工条件に関する知識を格納）、データベース（DB：推論結果を格納）から構成されている。KB は機械加工の専門家の知識を 2 章の記述形式で構築されたものである。DB は KB の推論結果をフレーム表現で格納している。また、TMS はノード生成システムと矛盾解消システムから構成され，何らの理由で加工条件が不適切な場合に不適切な理由で加工条件が不適切である場合，TMS の括弧の番号に沿って推論過程を追ってみる。

3・2 機械加工管理システムの推論過程 図 3 は，図 2 の機械加工管理システムの推論過程を，その間で以下に見やすい知識表現の例を示したものである。以下に検討の結果を示す。

（1）「フレーズ構成」の必要性。
（2）グローバル処理の有効性（材料，硬さなど）必要性。
（3）入力は空 DB 内の該当する加工条件（工具径，切り込み，回転数など）がユーザに提示される。ここで，DB 内に該当する加工条件が存在しない場合は，ISA 関係により付加手続き（Attached-Procedure-1）が起動し（4）が実行される。また，提示された加工条件が不適切な場合は（5）が実行される。

（4）条件推論の付加手続き（Attached-Procedure-1）の ISA 関係により実行され，加工条件推論部が起動される。すると，加工条件に該当する KB がロードされ，TMS のノード生成を行った後，ノード生成の初期状態（矛盾解消のための推論は実行されない）が結果として示される。この結果は，新たな DB に格納され，同時にユーザに提示される。もし，この提示された条件が不適切な場合は（5）が実行される。

（5）3・4 節で述べる TMS が起動され，矛盾解消

図 1 著者の開発した Prolog の記述の例

図 2 知識ベースの管理機構 TMS を持つ機械加工管理システムの構成

図 3 機械加工管理システムの推論過程
のための推論が実行される。これにより、DB 内容が更新され、同時にユーザに提示される。もし、この加工条件でも不都合な場合は、最終的に満足するまで（5）が繰返され、DB と KB が更新される。このように、KB は論理的表現の TMS により管理され、かつ推論がフレーム表現の付加手続きを用いて起こされる。そのため、専門家の知識を記述性の高いフレーム表現で表示されるので、初心者でも容易に理解できる。また、加工条件推論部は論理的表現で構築されているので、表現言語自体のパルクトラック機能により容易に矛盾解消のための推論機能が実現され、効率的なシステムとなる。

3-3 知識ベースの内部構造 図 4 は、機械加工（フライス加工）の知識ベースの内容を示したものである。これからも明らかのように、機械加工の専門家の知識は論理的表現の「表」と形式により表現されている。

例えば、図 4 中のフライス（Millin-Cutter）の知識は、以下のようなことを意味している。

「(cutter D 80) (diame-r of - cutter 80) (number of - cutter 5) という三つのデータが milling という関係にある」

すなわち、2 場の記述形式と対応させると、記述名は milling、引数は (cutter D 80)(diame-r of - cutter 80)(number of - cutter 5) が該当する。

ここでさらに、機械加工の専門家の表現できる限り近づけるため、数名の専門家の知識表現を分析し、その結果、3-4 節で後述する「もし…」という仮定的な知識（図 4 中の[MJ] と、「ただし…」という制約的な知識（図 4 中の[LJ]）の表現を取り入れている。この[MJ] と[LJ] の表現は、以下のよう記述される。

M (切り込み 3) (切り込み 2) (切り込み 1)
L (切り込み 2) (回転数 200)

この[MJ] の意味は、「もし（切り込み 3）を否定する事実があれば（切り込み 2）を仮定する」という内容を表している。これに対し「L」の意味は、「ただし、（切り込み 2）を（回転数 200）すると、何らの不都合が生じて好ましくない」という内容を表している。

この表現を採用すれば、機械加工特有な「回転数が 270 か 200 とし、切り込みは 3, 2, 1 のいずれかで調整する。ただし、回転数 200 で切り込み 2 で行うと不都合が生じる」といった表現を可能にしている。そのため、専門家の知識に沿って知識表現しやすく、初心者でも知識ベースの内容を容易に理解できる。

3-4 知識ベースの管理機能 TMS は、知識ベースの内容を矛盾なく管理するためのシステムである。具体的には、知識ベースの内容の理由分け（Justification）の集合を新しく得られる理由分けの集合と整合するように、信念（belief）の集合を動的に変化させるシステムといえる。

信念には IN と OUT の二つの状態が持たれる。IN の状態は、信念を信じ得るだけの十分な根拠がある場合で、少なくとも一つの正当な理由づけを持っている。これに対し、OUT の場合は、信念を信じる根拠が全然ないか、正当でないとする理由づけが明確に存在する。TMS はこの IN と OUT の状態を用いて、以下のようにして理由づけの整合性を管理する。

（a）TMS の理由づけ

知識ベースの知識に対して、有効な理由づけを行うためノード (node) を設定する。ノードには、仮定による理由づけ（SL 形理由づけ：Support - List justification）と、条件証明による理由づけ（CP 形理由づけ：Conditional - Proof justification）がある。この二つの理由づけを基にノード間の関係を表現し、知識ベース内の矛盾を解消する。すなわち、

（イ）仮定による理由づけのノード：
（ノード番号 SL IN リスト OUT リスト）のように表現される。ノードの状態が IN であるためには、IN リストのノードがすべて IN で、OUT リストのノードがすべて OUT の場合である。

（ロ）条件証明による理由づけのノード：
（ノード番号 CP 結論部 条件部）のように表現される。ノードの状態が IN であるため
には、例えば、条件部に記述されているノードのすべてがINである場合、結論部がINである場合が該当する。そして、例えば後述する図6のノード1001は、「条件部（125）のノードがすべてINなら、ノード1000がINになる」ということが指示される。このことは、「ノード1000がOUTである」という事実に矛盾する。つまり、この矛盾は（125）が同時にすべてINにならないことによる。そこで、この中のいずれかがOUTになるように探索が実施され、矛盾解消が行われる。

(b) 矛盾解消のための戦略とノード生成

図5は、図4の加工（ milling）に関する知識を基に生成されたノードとその状態を示したものです。つまり、図4中の仮定的な知識「M」と制約的な知識「L」は、2章で述べたメタ知識表現を適用して、図5のようなノードを生成する。そして、この二つの知識「M」と「L」とを組合せることによって、専門家の意図する矛盾解消戦略が実現される。この解消戦略は次の二つの原則に基づいている。

(1) 知識「M」の中で知識要素(図4中の「切り込み」や「回転数」)のノード状態を変更する。
(2) 知識「M」の中では、左側の知識表現からノード状態の変更が効果的に行われる。
(3) 知識「L」内の知識要素と同じノードがINの状態の場合、再度矛盾解消が起動される。

図5のノードは、TMSの矛盾解消過程が上記三つの要領を効果的に行実現できるように生成され、ここで図5に従ってこのノード生成の理由について述べる。

加工に関する知識「M」の知識要素をおのののA0、A1、A2、...、Anとすれば、仮定的な知識「M」は以下のよう示される。

(M,A0,A1,...,An)

知識要素間の関係は、以下のような論理式で示すことができる。なお、Anot-airは知識「M」以降のすべての知識の集合を示している。つまり、Anot-airが仮定された場合、この知識内ではIN状態のノード生成が出来ない。

A0∪A1∪...An∪Anot-air=1
A0∪(A1∪...((An∪Anot-air))...)=1

ここで、A0、A1、...、An、Anot-airは、おののの独立した知識要素としてとらえることができる。以下の関係を考えて考えることができる。

Anot-air=(A0∪(A1∪...((An∪Anot-air))...)
A0∪Anot-air=1

知識「M」のノード生成は、上記論理式の関係で示されるように必ず一つの知識要素が選択される。この際に、この知識要素を選択した正当な理由づけ（SL形理由づけ）を行う。つまり、このSL形理由づけは、上記論理式から「A0がOUT状態であるから、A0はIN状態である」という論理づけを行う。実際に、上式から「A0がOUT状態であるから、A0はIN状態である」という理由づけのノードが生成される。

これに対し、図4中の知識「L」は、図5中のノード900(not.pairs)を生成する。このノードは、次章で後述する（contradiction）と同様な矛盾解消のためのノードである。つまり、（contradiction）が外部からの矛盾解消のためのノードであるのに対し、（not.pairs）は知識ベース内部からあるノードである。

4. 機械加工におけるTMSの適用例

TMSには、原因が明確な場合と不明確な場合に対する矛盾解消の機能が組み込まれている。図6と7とは、原因が不明な場合のTMSのノードの状態変化を示したものである。図8は、原因が明確な場合の状態変化を示したものである。これで、これらのノードの状態変化による矛盾解消について述べる。

図6と7はノード1000の加工条件で加工を行った結果、「不都合が生じているが原因は不明」という仮定の基に、最適な加工条件を探索して矛盾解消を行う。

(1) 不都合が生じた後に鍛加工条件が不適当であると判断し、その旨を入力した場合は、
(contradiction 1000)

というノードの知識を生成する。これは「ノード1000がノード1000が矛盾を起こしている」ことを示している。

(2) TMSは、ノード1000が状態OUTになる
ようにCP形理由づけによってノードの知識1001（nogood1000）を生成する。そして、パックトラックを起動する。すると、ノード（125）のいずれかが状態OUTになるように、解消戦略に基づいて経路探索がなされる。
（3）前述の解消戦略により、ノード2を状態OUTの候補とし、ノード内のIN/OUTリストを調べ、OUTリストのノード3を状態INへ変更することを試みる。
（4）ノード3は以下に示すノードに変更され、ノード3,2,1000の順に状態が変更される。そして、新たなノード1002が生成される。

図6 第1回めのTMS起動後（制約なしの場合）

図7 第1回めのTMS起動後（制約ありの場合）

図8 第1回めのTMS起動後（原因が明らかになった場合）

図9 第8回めのTMS起動後（制約なしの場合）
3・2節の(5)の状態を示している。図10は、TMS起動直前の状態を示しており、フレーム内部の状態がTMS起動時の状態として示される。図11と図12は、TMS起動時の状態を示しており、「不都合が生じているが原因が不明」と「明らかに回転数に問題あり」という状態を繰り返した実行例を示している。これら図中右部のTMSの起動により、最終的な加工条件が設定状態としてフレーム表現の内部に格納される。なお、TMSの実行では、カッタ、回転数、切込み、テーブル送り速度などの加工条件が提示される。

このように、本システムの知識ベースは、解消戦略によってノードの状態を変化させ、最適な知識ベースの状態になっていることがわかる。

5. 結 言

フライス加工の専門家（平均14年経験）8人を対象に、本システムの評価を行った。専門家の中には、「回転数を徐々に落とすと一度に半分程度にする」、「回転数を落としすぎると」などがあったが、特異の意見（知識）は矛盾消解として最良の消解戦略が与えられているという評価を得た。また、無人化や安全性の点から、本システムがNC加工において有効であるという評価も与えられた。なお、本報告は、知識ベースの管理機構を構築し、その管理機構が矛盾なく機能することを確認することを第一目的とした。そのため、管理機構の動作原理を理解しやすくなるために、矛盾消解の対象を限定して述べてきたが、本来対象の大きさに関係なく同一管理機構で無矛盾の知識ベースを管理することは可能である。ここで、本研究をまとめてみることにするのが以下のようになる。

(1) 知識ベースの管理機構を持つ本システムは、時々刻々変化するような管理機構として、その有用性や構築性に大いに期待される。

(2) 管理機構の矛盾消解において、専門家の消解戦略を知識ベースに組込むことを可能にしている。これにより、専門家の知識を無理なく表現できるので、矛盾消解の専門家より高い評価を得た。

(3) 知識ベースの記述形式の統一により、知識ベース間の組合せや可読性、および管理機能の開発に有効である。特に、本記述形式によりメタ表現を可能にしているため、その実用性が高い。

文 献

(2) Jonan de K., Artificial Intelligence, 28(1986), 197-224.
(3) 大場・松島・長谷・中山, 情報処理学会論文誌, 5:1889-1899, 668-677.
(4) 長坂・ほか2名, 機論, 58:547, C(1991), 1151.
(5) 長坂・ほか2名, 機論, 58:549, C(1991), 1385.