

機械設計のための幾何拘束処理システム

安藤 英俊 ・ 鈴木 宏正 ・ 木村 文彦
東京大学工学部

プロダクト・モデリングにおいて製品の形状情報は特に重要であるが、現行の形状モデラを形状の自動設計などの知的CADに応用するには幾つかの問題点がある。それは形状は幾つかの幾何学的な拘束条件から規定されているにもかかわらず、これらの拘束条件や形状情報を統一的に扱う枠組みを持たず、従って元の拘束条件とその結果生じる形状との間の依存関係を捨ててしまっており、結果としての形状だけを保持している。

我々はこれらの拘束条件や結果としての形状情報を統一的に表現するために一階述語論理を導入する。さらに幾何学的拘束関係の評価による形状生成・変更機能の実現のため、幾何学的拘束条件と生成された形状情報との論理的依存関係の管理をATMS (Assumption-based Truth Maintenance System)を用いて行い、これに基づく幾何学的推論機構を提案する。試作システムは仮説生成やバックトラックを行いながら前向き推論によって幾何学的拘束関係から形状を規定して行き、後ろ向き推論によってデータベースに対する検索を行う。幾つかの幾何学的拘束条件を例にこの手法の有効性を示す。

Geometric Reasoning System for Mechanical Part Design

Hidetoshi ANDO, Hiromasa SUZUKI and Fumihiko KIMURA
The Faculty of Engineering, The University of Tokyo
7-3-1, Hongo, Bunkyo-ku, 113 Tokyo, Japan

Current geometric modelers are not sufficient when applied in intelligent CAD systems such as automatic design of mechanical parts. The shape of the product is determined by some geometric constraints, but geometric modelers provide no means to represent and handle these constraints or the dependencies between these constraints and the shape of the product. We introduce first order predicate logic to represent all these constraints and the geometric informations in uniformed way. We also introduce ATMS (Assumption-based Truth Maintenance System) to record and manipulate dependencies between geometric constraints and the shape of the product so that we can create and modify shapes through the manipulation of these constraints. Our system reasons the shapes from geometric constraints by forward reasoning with dependency-directed backtracking and answers queries about geometric informations and about logical dependencies by backward reasoning.

1. はじめに

機械部品の中では形状によってその機能を発揮していると言えるものが多く、プロダクト・モデリングにおいては製品情報のうちでも形状情報は特に重要である。この形状情報を計算機内に構築するのが形状モデラであるが、現行の形状モデラを形状の自動設計の様な知的CADに応用するには問題点が多い。まず次節で現状の形状モデラの問題点の幾つかを取り上げる。3節では形状の生成・変更を幾何学的拘束条件の取扱いの立場から考察し、拘束条件に基づく形状生成のための新しい手法を提案する。4節では本手法に基づく試作システムを紹介し、5節で実行例によってその有効性を示す。

2. 形状モデリングの問題点

以下に現行の形状モデラをCADで応用する際の問題点の幾つかをあげる。[木村87]

①現状の形状構成操作は、基本形状生成操作と、それらを変形し、組み合わせる操作とからなっている。これは計算機の内部処理の都合から決められていることが多く、人が形状を規定していく過程とのギャップが大きい場合が多い。

②工業製品の形状の各部は同様に重要ではない。ある部分は機能的に形状精度が厳密に規定されており、ある部分は単につながっていればよいというようである。なにも規定されない部分には、暗黙に設定されている形状があてはめられる。このように形状精度に差を持たせ、形状処理時に反映できれば有用であるが、現状では困難である。

③設計生産の過程では、すでに定義した形状を修正変更することがしばしば必要になる。設計意図や生産準備の要求に従って整合的に形状を変更していくが、これも現状の形状モデラでは扱いにくい。

④幾何形状はそれ自体が独自に決定されるのではなく、寸法や形状特徴などの幾何学的な拘束条件を満たすものとして構成されることが多い。現在の形状モデラはこのような拘束関係を捨ててしまい、結果の形状だけを持っているので、これを陽に扱うことが望ましい。

3. 幾何学的推論

上記の問題点は、もともと形状は様々な幾何学的拘

束条件から規定されているのに、拘束条件やその拘束条件と結果として生成された形状との依存関係が扱えないこと、形状モデラの用いている形状表現が常に対象形状の完全性を要求しているなどの点に起因するところが大きいと考える。従ってこれらの依存関係を管理、利用することによってこれらの問題点がある程度解決されると思われる。

そこで本研究ではこれらの拘束条件や形状情報の統一的な表現とその上での拘束条件の評価による形状生成・変更機能の実現のために、拘束条件および形状情報の一階述語論理による表現を用いる。また、幾何学的拘束条件と生成された形状情報との依存関係の管理をするためにA T M S (Assumption-based Truth Maintenance System) [de Kleer86a]に基づく幾何学的推論機構(Geometric Reasoner)を提案する。

そこでまず本システムの中核であり、データの依存関係を管理するA T M Sについて、その概略を説明する。

A T M Sにおける基本的なデータ構造は“ノード”で、問題解決システム用のデータ、問題解決システムによって与えられた“justification” (正当化) のリスト、及び正当化からA T M Sによって計算された“label”からなる。

ノード: name:<datum, label, justifications>

justification (正当化): あるノードが他のノードからいかに導かれるかを書き記したもので、条件(antecedent)、帰結(consequent)、および問題解決システムのデータ(informant)よりなる。一つの正当化が推論の1ステップを表しているとも言える。

assumption (仮説): 仮定をするという決断を表現し、何が仮定されるのかを問わない。A T M Sでは“仮定をするという決断と、仮定されたものを明確に区別するべきである”と主張する。以下の例では誤解のないかぎり大文字は仮説を表す。

environment (環境): 仮説の集合。コンテキストは環境によって特徴付けられる。

label: それぞれのノードに付加されたenvironmentの集合で、正当化からA T M Sによって計算される。これはそのノードが最終的にどの仮説に依存しているかを表す。labelが空であるノードはその正当性を裏

付けるものが何もないということを意味し、すべてのコンテキストで偽となる。labelは推論の過程で新しい正当化が加えられたり、矛盾が発見されたりして書き換えられたりするが、ATMSによって常に無矛盾、完全、健全で最小化されている。

(labelが完全であるとは、そのノードを導出するようなすべての無矛盾な環境は、このlabelのある環境を包含することをいう。健全(sound)であるとは、このlabelの各環境からこのノードが導出できることをいう。また最小であるとは、このlabelに属するものの2つの環境の間にも包含関係がないことをいう。)

NoGood : 矛盾を導く仮説の連言。例えば、 $NoGood\{A, B\}$ は仮説Aと仮説Bは相矛盾するということを表す。矛盾を導く環境(仮説の集合)は"**NoGood 集合**"という大域的なデータベースに保持され、新しく生成された環境は常にこのデータベース上で無矛盾性の検査をされる。このデータベースは常に最小化されている。これは、矛盾を導く環境を包含する環境はやはり矛盾を導くという事実に基づいている。例えば、 $NoGood\{A\}$ ならば環境 $\{A, B\}$ も矛盾を導く($NoGood\{A, B\}$)。

ATMSに基づく推論では、仮説のむやみな生成によって引き起こされるバックトラックの回数を最小限に抑える為に、出来るだけ初期の段階で矛盾を検出して以後の推論の範囲を限定することが重要である。従って、効率の問題として推論規則は矛盾を検出するものを最も高い優先順位で実行し、逆に仮説を生成するような推論規則は最も低い優先順位で実行される。

4. 試作システム

上述のATMSをデータベースに持ち、この上で一階述語論理表現された幾何学的拘束条件から前向き、後ろ向きに推論を行うシステムを試作した。本システムはLispマシンSymbolics3640上でCommon-Lispを用いて構築されている。本報告書で用いている論理表現は実際は全てLispのS式の形をしている。

図1に本システムのデータベースおよび推論部の構成を示す。論理データベースにはユーザの入力による拘束関係や推論によって得られた新しい事実が登録され、ATMSによって依存関係と論理的整合性が管理されている。

4. 1 幾何学的拘束関係と対象形状の表現:

形状自体の表現とともに、それらを規定する幾何学

的拘束条件を形状表現と融合された形で表現するために、すべての拘束条件とその結果として生成される形状を一階述語論理により表現する。試作システムで現在取り扱っている形状は2次元図形で、線分と円弧から構成されるものである。図2にこれらの形状情報および幾何学的拘束関係の述語論理表現の例を挙げる。

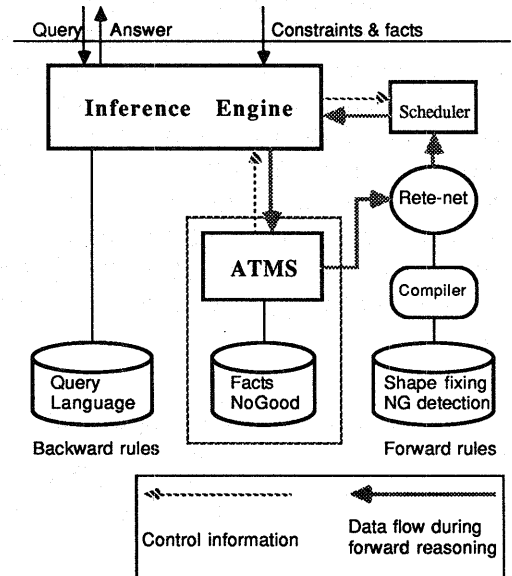


図1 システム構成図

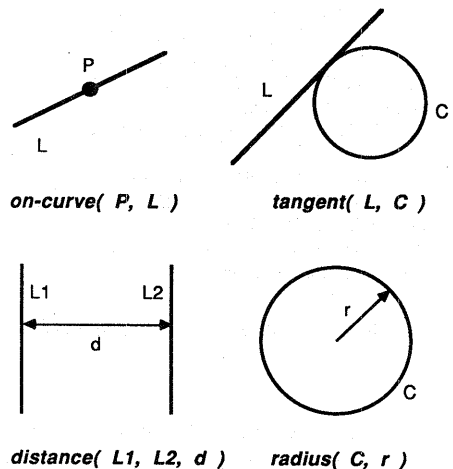
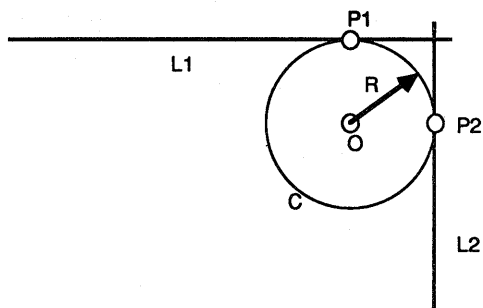


図2 論理表現された幾何学的拘束関係の例

論理表現を用いることによって拘束条件が規定されていない形状などは、未決定のままに残しておく。本システムではPrologなどとは異なり、事実節と推論規則と明確に分けており、事実節の中では自由変数の使用は許さない。

4.2 幾何学的拘束条件に基づく形状決定：

述語論理によって表現された幾何学的拘束関係と与えられた拘束関係から出発して、IF-THEN型の前向き推論規則の適用によって拘束の伝播(Constraint Propagation)を行い、対象形状の属性を決定して行く。推論規則中の論理式には変数の使用を許し、これらの変数はマッチングに成功した事実節中の定数の値に置き換えられる。図3はこのような形状決定のための推論規則の例である。推論の過程では幾何計算などのために外部の関数や手続きを呼出して評価する事が可能である。



```

tangent( L1, C ), tangent( L2, C ),
not-equal( L1, L2 ),
on-curve( P1, L1 ), on-curve( P1, C ),
on-curve( P2, L2 ), on-curve( P2, C ),
line-parameter( L1, A1, B1, C1 ),
line-parameter( L2, A2, B2, C2 ),
radius( C, R ), center-of( O, C ),
side-of( O, L1, S1 ), side-of( O, L2, S2 )
=>
bind(( Ox, Oy, P1x, P1y, P2x, P2y ),
      calc-circle7( A1, B1, C1, A2, B2, C2, R, S1, S2 ))
coordinate( O, Ox, Oy ),
coordinate( P1, P1x, P1y ),
coordinate( P2, P2x, P2y ).

```

図3 形状決定のための推論規則の例

全ての前向き推論規則の前提部はルール・コンパイラによってコンパイルされ、Rete-net[Forgy82]とよばれるネットワークに展開される。Reteアルゴリズムは、OPSタイプのプロダクション・システムで採用さ

れており、推論規則の前提部とデータベース中の事実節の高速な照合を行うものである。データベースに加えられた新しい事実節は、このネットワーク中をトークン(前提部に照合する事実節の組)となって流れる。ネットワークの各節には照合の為のテスト関数あり、1つあるいは2つのトークンを入力引数として評価し、テストに成功した場合は新しい1つのトークンが生成されて次の比較のための節に送られる。推論規則の前提部に完全に照合したトークンだけがネットワークの外に放出される。このアルゴリズムの特徴は、①前提部のパターンの一部が同じルール同士ではその部分をネットワーク上でも共有できること、②中間的な解の候補であるトークンをネットワークの各ノードで保持しているため、データベースに生じた事実の変更分についてのみ新しいトークンを作って古いトークンと比較すればよい。しかしその反面、ネットワーク中の各ノードで大量のトークンを保持しなければならず、これがノードでのテスト関数の呼出し回数とメモリ消費を決定付ける。

本研究ではテスト関数の評価時に単なるパターンの照合を行うだけでなく、照合の成功によって生成された新しいトークンを構成する事実節のlabelから、算出される新しいトークンのためのlabelが空になる(このトークンを構成する事実節の組合せは無矛盾なコンテキストに存在しない)かをNoGoodデータベースの情報を利用して調べる。照合には成功して生成はされたがlabelが空になってしまったトークンは、現在のノードに止められ、下位のネットワークへは送られない。なぜならば、この様なトークンと他のどの様なトークンと組合せてもやはり空なlabelをもつトークンしか生成されないからである。もし後でこの様なトークンを構成する事実節がユーザの入力や推論によって新しい正当化を与えられた場合には、再びトークンのlabelを計算して、もし空でなくなればこの時点で初めて次のノードへと送られる。この様にできるだけ無駄なトークンの生成と照合の回数を押えている。これによって前提部に照合する事実の同じ組合せを生成することを避けられるのでATMSに基づく推論機構にとって必要なConsumer手法による推論が可能になる。ルールの前提部にマッチした事実の組は競合集合となって実行の待ち行列(Agenda)に貯められ、そのうちの一つが推論を制御するスケジューラによって取り出され実行される。スケジューラ自身は推論を制御する上位の手続きから呼び出される。

推論ははじめにユーザによって入力された事実節を出発点とし、推論の過程で必要に応じて仮定(Assumption)を作り出してゆく。推論の結果得られる事実は特定の公理と仮定に依存しているため、これらの事実や仮定の組(コンテキスト)を指定することによって、スケジューラは推論を行う範囲を特定のコンテキストに限定することができる。つまり全ての拘束関係のうちのある特定の拘束の組に着目することによって、解こうとする問題のある部分問題にだけ着目して、その範囲での推論を行うことが可能になる。

4.3 整合性管理:

前向き推論規則には、ある条件は矛盾であるということを書き記述するための矛盾検出規則があり、例えば『全ての“点”は必ずただ一つの座標値を持つ』という拘束条件は次の様に、否定が矛盾を導くという形で記述できる。

```
coordinate(P, X1, Y1), coordinate(P, X2, Y2),
not( and( X1 = X2, Y1 = Y2 ) )
=> NoGood (矛盾)
```

データベースのうちの一部は矛盾をあらわす“NoGood”とよばれる集合で、推論の結果矛盾が導かれた場合にはその原因となる仮定の組を同定して記録し、依存関係に基づくバックトラッキングによって矛盾解消がはかられる。矛盾を含むコンテキストにおいては以後推論を行なわない。

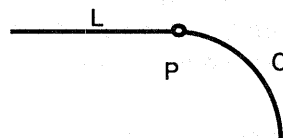
4.4 データベース検索:

本システムにおいては、すべての情報が論理表現されているので、PureなPrologと同様にパターン・マッチングを用いた後ろ向き推論による検索がおこなえる。現在本システムでは後ろ向き推論は純粋にデータの検索のために用いており、カット・オペレータのような制御や、assertやretractの様なデータベースに対する破壊的操作は禁じている。

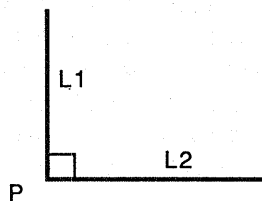
後ろ向き推論による検索もATMSの管理のもとに、ある特定のコンテキストに限定して行なったり、あるいは目標節を満足するようなコンテキストの組を求めたりすることが可能である。また前向き推論の間に、推論された事実とその事実を推論するのに必要となった事実との論理的依存関係が求められATMSによって管理されているので、任意の事実節にたいして、その事実が依存する公理や仮定を同定することができる。

4.5 暗黙の拘束による推論:

ATMSの矛盾回避機能を利用して、形状のうち陽に拘束の与えられていない部分の情報に暗黙値(Default)を仮定できる。例えば機械図面などの図形に広く用いられている暗黙の拘束関係の例を図4に示す。1つは輪郭線は直線分と円弧の接続の場合には多くの場合滑らかであるということであらわし、他方は2直線は直角に交じわることをあらわす。図中でM-tangent(...)とあるのは論理的には“tangent(...)”と矛盾する事実が現在の公理系には存在しない”という意味である。従ってこの推論規則の意味は“もし直線分と円弧が接続しているならば、明らかに誤りでない限りは、この線分と円弧は接している”と仮定してもよい”ということである。しかし本システムでは効率上の問題からさらに“tangent(...)”が現在の公理系で既知でない”という条件も付加して評価することによって不要な仮説の生成を抑えている。



```
line( L ), circle( C ), on-curve( P, L ),
on-curve( P, C ), M-tangent( L, C )
==> tangent( L, C )
```



```
line( L1 ), line( L2 ), on-curve( P, L1 ),
on-curve( P, L2 ), M-orthogonal( L1, L2 )
==> orthogonal( L1, L2 )
```

図4 暗黙の拘束による推論

5. 実行例

ここでは上述の幾何学的拘束関係の論理表現から試作システムを用いて実際に形状を生成・変更してゆく様子を示す。

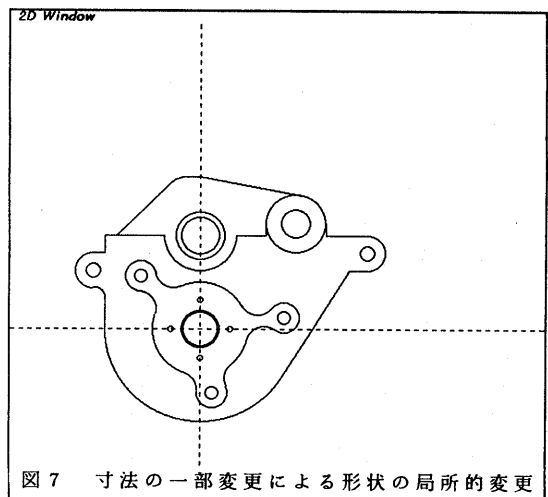
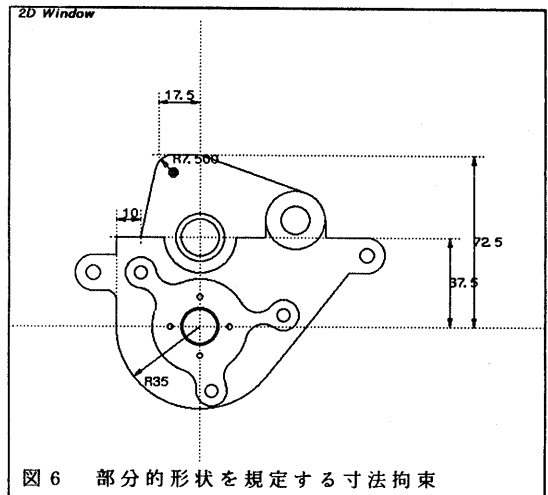
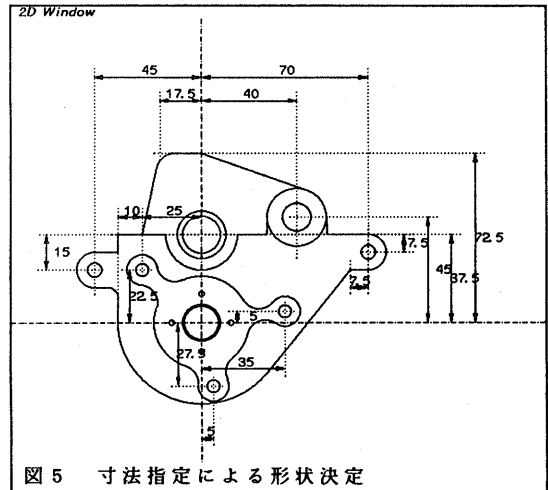
5.1 パラメトリック・デザイン

拘束条件からの形状決定の例として“寸法指定”による形状定義・変更をあげる。本システムでは上述の様に寸法指定も含む全ての幾何学的拘束条件が一階述語論理により表現されている。寸法が指定されると形状決定規則と前向き推論によって形状が決定される。

(図5) 人間が過不足のない寸法指定を行う事は難しい。過剰な寸法が指定されて相矛盾するならば幾何学的な整合性が崩れてるので、前述のような矛盾検出規則とATMSの機能を用いて、その原因となる寸法指定を見つけ出してユーザに伝える事が出来る。決定された形状についてはATMSが依存関係を管理しているので、例えば図6の●で示した点を規定する寸法拘束を辿る事が出来る。これらの拘束条件はこの点の座標を表すデータのlabelに直接現れている。また、これらの寸法指定のうちの一部を変更する事によってそれに依存する部分的形状を図7の様に変更する事が出来る。他の拘束条件は満たされたままである。

5.2 公差評価

公差のつかない寸法から形状を決定することは殆ど試行錯誤を要しない直線的な前向き推論で行える。しかし対象となる形状を純粋に幾何学的な存在としてではなく、実際に加工される部品として捉えた場合、一般に公差を伴わない寸法というのはありえない。しかも加工をする側のNCマシニングセンタなどでは公差付きの寸法を直接に取り扱うことはできない。ここでとりあげるのはこのような公差を伴った寸法から、全ての公差を満たすような寸法の具体的な値の組を求めるとい問題である。特に参照寸法などの、寸法指定としては冗長なもの加わり、かつそれらにも全て公差が与えられている場合などは単に公差中心をすべての寸法に具体的な値としてわりあてると、一部の参照寸法の値が公差範囲を満たさなくなることもある。そういった場合には公差範囲を外れる原因となった寸法指定を一部変更することによって公差範囲を外れた参照寸法を公差範囲に収めるようにする。その変更によって今まで公差範囲内に収まっていた参照寸法が公差範囲を外れてしまうこともあるので、その場合にはある程度の試行錯誤が必要になる。



この問題は本質的には連立不等式に帰着される。すなわち以下の式を満たす変数 (X_1, \dots, X_n) の具体的な値を求めることである。 X_i は各寸法の値、 Y_j はそれらに從属してきまる参照寸法の値を表わし、これらの値には不等式によって範囲が指定されている。 dX_i^+ は X_i に対するプラス方向への変動許容値、 dX_i^- はマイナス方向への許容値を表す。

$$\begin{aligned} X_i - dX_i^- &\leq X_i \leq X_i + dX_i^+ \\ Y_j &= f_j(X_1, \dots, X_n) \\ Y_j - dY_j^- &\leq Y_j \leq Y_j + dY_j^+ \end{aligned}$$

この問題に限らず与えられた拘束からそれを満たす解を生成する問題を考えた場合、その求め方がよく分かっている場合には手続的に効率よくこれを行うことが出来る。しかし様々な自動設計問題やプログラミングなど、我々がCADの領域で扱おうとする問題に関しては一般に手続的に行かないことが多く、どうしてもある程度試行錯誤的な方法に頼らざるを得ない。しかし単純なバックトラックを用いた生成・検証(Generate&Test)ではとても効率が悪くて手に負えない。一方、対象の定性的挙動を記述したモデルなどがあればこれを用いてかなり効率よく最適に近い解が得られるであろうが、このようなモデルを用意すること自体が大変な作業である。

そこであまり高価なモデルを用いずに、なおかつある程度の効率をもって試行錯誤的に生成・検証を行う事が必要になる。本研究ではATMSの管理する依存関係を利用して、依存関係に基づくバックトラッキング(Dependency Directed Backtracking)を用いた生成検証法によってこの問題を取り扱う。[de Kleer86]

不等式で与えられた拘束条件は、解の候補が生成された後にその解の正当性を評価するために用いることは比較的簡単であるが、解の候補を生成する段階で用いるのは困難であり方程式として扱うか、あるいは与えられた範囲で離散的に近似して扱うしかない。本研究では公差付き寸法は公差範囲で値を離散的に近似して解の候補を生成するために用い、参照寸法は生成された寸法値の組が参法寸法の公差範囲をも満たすかどうかを評価するための拘束条件として用いる。より具体的には、公差付き寸法は公差範囲のある細かさで離散的に分割した寸法値の組に変換され、述語"choose"によって登録される。

choose(A_1, \dots, A_n)は論理的には A_1 から A_n の排他的論理和あらわす。すなわち A_i と A_j は $i=j$ でない限り

矛盾する。なおかつchooseで宣言された事実節の組はATMSによるバックトラックの際に必ずそのうちの1つが解の候補のコンテキストに加えられるように制御される。すなわちchoose(A_1, \dots, A_n)は下式と同値である。

$$\begin{aligned} A_1 \vee \dots \vee A_n \\ A_i \wedge A_j \Rightarrow \text{NoGood} \quad (i \neq j) \end{aligned}$$

①解の候補生成のためのルール

toleranced-distance(L_1, L_2, D, D^+, D^-)

\Rightarrow choose(distance(L_1, L_2, D_1),

.....

distance(L_1, L_2, D_n))

これによってこの例題の場合には、解の候補となるコンテキストには全てのchooseからちょうど1つずつの寸法値が選ばれることになる。

参照寸法は具体的に評価できる時点で自動的に評価され、公差範囲を満たさない場合にはその状態を導く仮定の組合せ(コンテキスト)を矛盾として登録する。この様に、与えられた解の正当性を評価するための拘束条件は矛盾を検出する推論規則を用いて否定的に記述できる。

②解の検証のためのルール

reference-toleranced-distance(L_1, L_2, D, D^+, D^-),

line-parameter($L_1, P_{11}, P_{12}, P_{13}$),

line-parameter($L_2, P_{21}, P_{22}, P_{23}$),

out-of-range($D, D^+, D^-, P_{11}, P_{12}, P_{13}, P_{21}, P_{22}, P_{23}$)

\Rightarrow NoGood

ATMSにおいては効率のため、矛盾を検出するルールは必ず他のルールよりも先に評価される。矛盾が検出されると依存関係に基づくバックトラッキングが起こり、矛盾を導く仮定の組を次の候補に置き換える。新しい候補は、既知の矛盾を導くような仮定の組を決して含まないように生成されるので、特に拘束条件が厳しくて解の候補の多くが失敗するような場合にはNoGoodに記述された情報によって解の探索空間が狭められるので、一般的なPrologにみられるような依存関係を考慮しない逐次型バックトラッキングよりも解の探索効率が良い。

図8は参照寸法も含む公差付き寸法で、これをもとに上述の手法によって公差を満たす具体的な寸法指定を求める。(図9)そして、寸法指定によって規定された形状のデータから、EXAPTのパートプログラムを生成して加工のシミュレーションを行った。(図10)

6. まとめ

以上、知的CADのための幾何拘束処理として幾何学的拘束関係を一階述語論理で表現し、拘束条件から前向き推論による形状生成を行い、拘束条件と形状との論理的依存関係をATMSによって管理する手法を紹介し、部分的な拘束の変更による形状変更や不等式的に与えられた相互に干渉する拘束条件の組を満たす形状を求める問題に適用して、その有効性を示した。

今後の課題は、扱える図形の3次元への拡張、数式処理等の拘束条件解法導入などで、さらに様々な形状生成・変更の問題に適用して行きたい。

参考文献

[de Kleer86a]J.de Kleer:An Assumption-Based Truth Maintenance System, Artificial Intelligence, Vol. 28.

[de Kleer86b]J.de Kleer:Extending the ATMS, Artificial Intelligence, Vol. 28.

[de Kleer86c]J.de Kleer:Problem Solving with the ATMS, Artificial Intelligence, Vol. 28.

[de Kleer86d]J.de Kleer et al.:Back to Backtracking:Controlling the ATMS, AAAI '86.

[木村87]木村文彦:形状モデリングとCAD/CAM、精密工学会誌 Vol. 53 No. 3

[安藤86]安藤英俊他:述語論理に基づく寸法処理システム、第4回設計自動化学講演会論文集

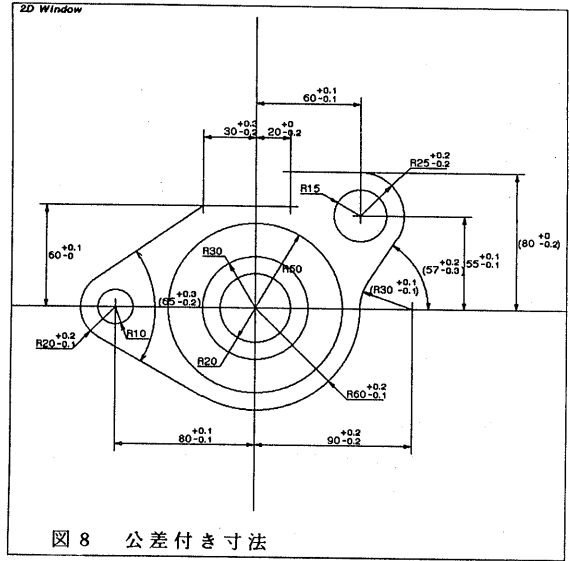


図8 公差付き寸法

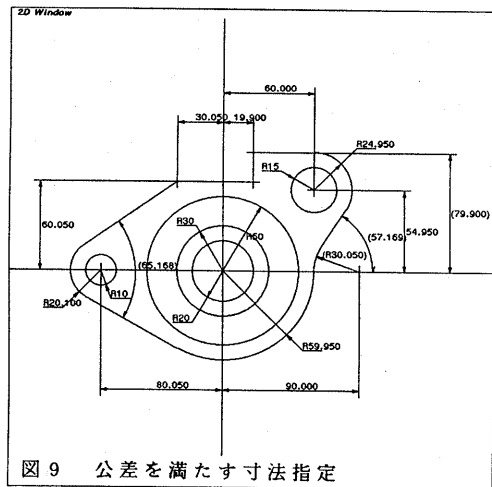


図9 公差を満たす寸法指定

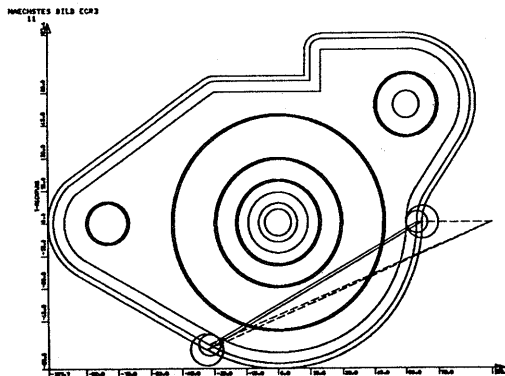


図10 EXAPTによる加工シミュレーション