制約に基づくオブジェクト指向知識表現システム FREEDOMにおける制約充足方式

横山 孝典　　佐塚 秀人

（財）新世代コンピュータ技術開発機構　　NTTデータ通信（株）

1．はじめに

我々は制約に基づくオブジェクト指向知識表現システムFREEDOMを開発している[1]。FREEDOMはオブジェクト指向を基本にその属性値や構成要素などに関する制約記述を可能とし，制約充足に基づく推論機能を有するシステムで，設計型問題における対象モデル表現を目的に開発を始めた。

そしてその第一段階として逐次推論マシンPSI上にESPを用いてプロトタイプFREEDOM/S0を作成した。本報告では制約論理プログラミングとオブジェクト間の制約伝播によるFREEDOM/S0の制約充足方式について述べる。

2．FREEDOMにおける知識表現

FREEDOMの知識表現はオブジェクト指向と制約指向の融合を狙ったものである。クラスは特定の種類の対象に関する制約をモジュール化して表現したものと考える（クラスはオブジェクトではなく，以後オブジェクトと言った場合インスタンスを意味する）。

クラス階層の表現では抽象・具体関係を表現するものを"is_a"関係，抽象・具体関係ではないが性質を継承するものを"includes"関係と呼んで区別する。"is_a"関係における下位クラスは，上位クラスに新たな制約を付加して具体化したオブジェクトの部分集合を表わすものと考える。そしてインスタンスの属するクラスを"is_a"リンクに沿って動的に変更可能とする。この機能は設計型問題における詳細化（具体化）処理の効率化に役立つ[1]。

また，オブジェクトの全体・部分関係のうち，特にインスタンスの成立に必要不可欠な構成要素を"consists_of"関係で定義する。これは構造上の制約条件となる。この時構成要素のクラス名を指定でき，これを構成要素の型に関する制約条件として扱う。この場合，宣言したクラスかその下位クラスであれば制約を満たすものとする。

クラス定義にはオブジェクトが満足すべき属性値に関する制約を宣言的に記述できる。制約式の表現形式は数式または述語である。ただし制約式中で参照できるのは当該オブジェクトとその構成要素の属性に限る。

3．制約充足に基づく推論

FREEDOMはオブジェクトに属性値を設定したり制約を付加した時，クラスで定義された制約と動的に付加された制約を常に満足するようにオブジェクトの状態を保つ。従って設計型問題では，設計知識や要求仕様を制約条件として表現することにより，手続き的な処理を記述することなく，要求仕様を満足する解を生成できる。

FREEDOMの制約充足機能の特徴は，属性値の決定のみでなく，制約を満足するクラスの探索機能をも提供していることである。これにより抽象的なクラスに属するオブジェクトを制約を満足するより具体的なクラスに変更することができる。これを制約に基づく分類法的推論（Constraint-based taxonomic reasoning）と呼ぶことにする。

制約に基づく分類法的推論の非常に簡単な例として，増幅器の種類を決定する例題を図1に示す。オブジェクト「増幅器A」は最初クラス「増幅器」のインスタンスとして生成したが，それを「FET増幅器」か「トランジスタ増幅器」に具体化したい。増幅器Aは増幅率Aと負荷抵抗RLという属性を持ち，増幅率に関する制約条件が付加されている。RLの値が未定の状態ではFET増幅器とトランジスタ増幅器のいずれも可能性があるが，RLの値を図のように決定した段階で，FET増幅器は制約条件を満足せず，そのクラスはトランジスタ増幅器に決定される。

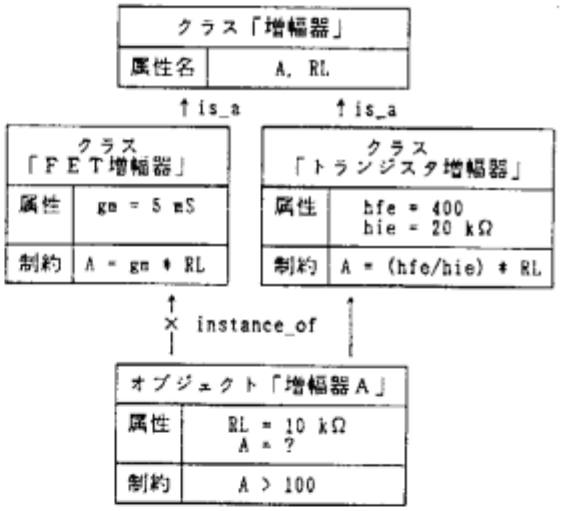

図1 制約に基づくクラスの探索

ところでオブジェクトのクラスの型の制約充足については，クラス変更時に包含クラスや構成要素のクラスの型をチェックし，制約を満たしていなければ満たすクラスに変更する処理を行う。

4．制約論理プログラミングによる制約充足

FREEDOM/S0では属性に関する方程式，不等式，記号的な制約を扱い，宣言的表現に優れた制約論理プログラミングによる解法を基本とする。

Constraint Handling of Constraint-Based Object-Oriented Knowledge Representation System FREEDOM

Takanori YOKOYAMA,　Hideto SAZUKA

ICOT,　NTT Data Communications Systems Corp.

方程式は Buchberger アルゴリズムを用いた制約論理プログラミング言語 CAL [2] の制約評価系を利用して解く。これにより連立方程式を解くことができる。不等式については，現在はその不等式を満足するかどうかのチェックのみを行っているが，今後 Sup-Inf 法[3] を導入して解くことを考えている。なお記号的な制約については Prolog 的な述語呼び出しを実行することとし，その定義及び解法はホーン節で記述する。

ところで制約充足のため，既に設定されている属性値の変更が必要となる場合があり，その機能も提供している。この時変更可能な属性を特に指定しない場合，できる限り少数の任意の属性値を変更する。なお，変更したくない属性値については明示的に制約として表現すればよい。

5．制約伝播による制約充足

FREEDOM はオブジェクト単位の並行処理への発展を考えており，できるだけオブジェクトが独立に問題解決を進めることが望ましい。しかし制約論理プログラミングのみでは複数のオブジェクトにわたる制約条件を独立性を保ちながら解くのは難しい。そこで，オブジェクト間の制約伝播による制約充足機能を取り入れることとした。

オブジェクト間の基本的な関係は "consists_of" リンクを用いた全体・部分関係であるが，FREEDOM/S0 では全体を表わすオブジェクト（以下，親と呼ぶ）は構成要素（部分）の属性を参照できるが，構成要素は親の属性を参照することはない。従って属性の相互参照はなく，制約伝播により効率のよい制約充足が可能である。

まず構成要素の共有が存在しない場合の制約充足について述べる。この場合は，属性値が決定した時その値を伝播することにより比較的容易に制約充足可能である。

例えば図2で，2段増幅器は初段増幅器（a1）と次段増幅器（a2）を構成要素としている。ここで図のような制約が存在したとすると（制約式中で a1→A は構成要素a1 の属性 A を意味する），初段増幅器の増幅率 A が決定した時点で図のような属性値の伝播が起こり，次段増幅器の増幅率 A の値が決定される。

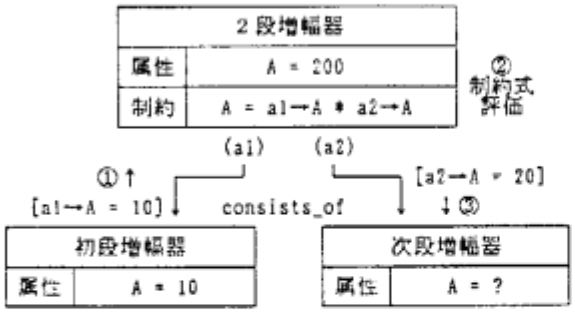

図2 オブジェクト間の属性値の伝播

構成要素の共有がある場合には，制約式における参照関係のループを生じ，属性値の伝播では解けないことがある。このような場合には制約式を伝播することにより解く。

すなわち共有しているオブジェクトの持つ属性のうち親から参照されているものの値を記号化して表現し，親のオブジェクトに伝播する。記号化された属性を受け取ったオブジェクトはその記号を含む制約式をその親に伝播していく。そして上位の共通の親のところで制約論理プログラミングの技法により解く。

例えば図3で，初段増幅器の構成要素 ro と次段増幅器の構成要素 ri が同一のオブジェクト「共有抵抗」である。この場合には共有抵抗の属性 R の値を例えば記号 α で表現して親のオブジェクトである初段増幅器と次段増幅器に伝播し，さらにそれを含む制約式を2段増幅器に伝播することにより，そこで制約式を解き，属性値を決定できる。

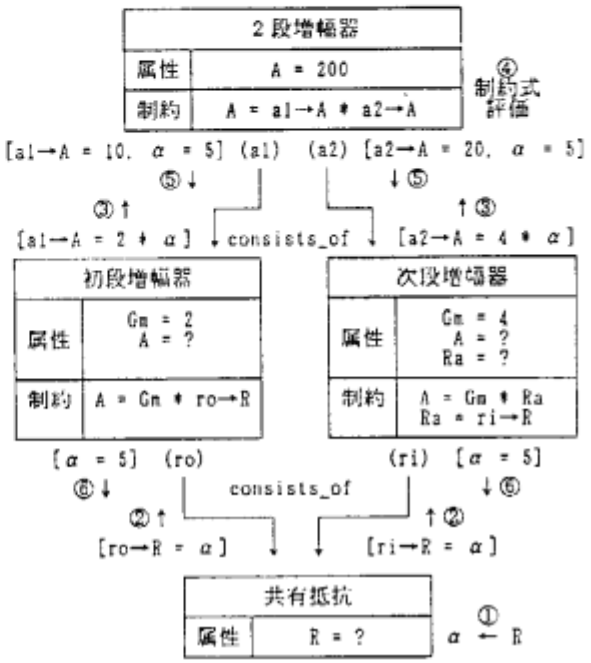

図3 オブジェクト間の制約式の伝播

なお制約充足のため属性値の変更が必要となる場合には，オブジェクトの独立性を重視し，可能な限りオブジェクト単位の局所的な変更とする。また全体・部分関係においてはできる限り部分が全体に影響を与えることは避ける。

6．おわりに

以上 FREEDOM/S0 の制約充足機構について述べた。本方式は制約論理プログラミングを基本に制約伝播を取り入れたもので，宣言的な制約記述と制約充足処理におけるオブジェクトの独立性を重視したものである。

FREEDOM は設計型問題のみでなく，分類型問題やパターン認識のように，制約を満足する解を生成したり与えられたデータに矛盾しない解を探索する形式の問題に対して有力な知識表現システムになり得ると考えている。今後，汎用的で実用的なシステムを目指して機能強化を図っていく。

また，オブジェクト単位の並行処理機能や並列制約充足機能，並列探索機能などを取り込み，並列推論機能を有する知識表現システムへと発展させる予定である。

参考文献

[1] 横山 "設計対象記述のための知識表現システムFREEDOMの提案"，情報処理学会研究会資料，88-AI-60 (1988)
[2] 坂井，相場 "ＣＡＬ：制約論理プログラミングの理論と実例"，電子情報通信学会研究会資料，SS87-22 (1988)
[3] 大木他 "Sup-Inf法に基づいた制約論理プログラミング言語"，ソフトウェア科学会第５回大会，C1-1 (1988)