知識メディアステーション
(4) 推論機構

7G-6

山岡 孝行　熊谷 秀光　和氣 朝臣
三菱電機(株)中央研究所

1 はじめに

知識メディアステーションでは、知識表現と問題解決を記述する知識プログラミング言語として、Ｐｒｏｌｏｇに制約指向を取り入れた制約指向論理型言語[1],[2],[3]を使用する。この言語は、属性リストによるフレーム型知識表現、ドメイン制約，線形方程式，線形不等式，等価，非等価などの制約条件、ＥＳＰのクラスに相当するモジュール階層、Ｐｒｏｌｏｇの後ろ向き推論機構に、前向き推論機構、データ駆動推論機構などを加えた柔軟な推論機構、といった特徴を持つ。これらの特徴と知識プロセッサの機能を活用することで、高度な知識表現と問題特性の宣言的な記述による問題解決が可能となる。本稿ではこの言語の概要を紹介する。

2 フレーム表現

フレーム表現は知識の有効なクラスタリングを可能にする。知識メディアステーションの知識プログラミング言語では、ＣＩＬ[4]と同様に属性リストを用いてフレーム表現を行う。

属性リストは、属性名とそれに対応する属性値の対の集合である。これにより、記述対象を多角的に表現できる。

例. Mutou={@,name!武藤,age!28,sex!male}は名前が武藤で年齢が２８才である男性についての属性リストによる記述である。

属性リストは固定的なものではなく、動的に属性を追加することが可能である。また、属性値として変数を与えることも可能である。

例. 武藤さんが裕子という女性と結婚したときはMutou!wife=Yuukoとすることで奥さんに関する属性記述を追加できる。ただし、Yuuko={@,name!裕子,age!26,sex!female}である。更に、新居が未定の時はMutou!adress=Xとしておいて、新居が決まったときＸを具体化すれば良い。

２つの属性リストＸ，Ｙを等価にするにはＸとＹをユニファイすれば良い。属性リストのユニフィケーションは次のように行われる。ＸとＹの属性記述をまとめ（属性名，属性値対の和集合を取り）それを両者の新たな属性記述とする。このとき、ＸとＹが属性名が同じ属性を持つときは、両者の属性値のユニフィケーションを行う。属性値のユニフィケーションが失敗したときは、属性リストのユニフィケーションが失敗する。

例. 武藤さんについての別の記述Employee1={@,name!武藤,task!engineer}があったとき、Employee1=Mutouとすることで両者を等価にできる。その結果、武藤さんの属性記述は{@,name!武藤,age!28,sex!male,task!engineer,wife!Yuuko,adress!X}となる。

3 制約条件

制約指向プログラミングの基本の一つは、その実行が通常の実行順序とは切り離されて管理される制約条件(Constraints)の導入である。制約条件の実行タイミングはそれが表す条件の判定が可能になったときである。データが不十分なため判定ができないときは、その処理はデータが揃うまで延期させられる（サスペンドする）。

知識メディアステーションの知識プログラミング言語で制約条件として処理されるのは、非等価，不等式，否定，方程式，有限領域制約である。

① 非等価（＝／＝）
　Ｘ＝／＝Ｙは、ＸとＹが等しくないという制約を与える。Ｘ，Ｙに未定義変数が含まれていて「等しくない」ことが判定できない時にサスペンドする。

② 不等式（＜，＞，＝＜，＞＝）
　数および変数と演算子＋，－，＊，／，ｍｏｄ，ｄｉｖで構成される数式Ｘ，Ｙを不等号で結び大小関係の制約条件を与える。Ｘ，Ｙに未定義変数が含まれるときサスペンドする。変数を一つだけ含む不等式は、その変数に関する区間制約として取り扱われる。

③ 否定（ｎｏｔ）
　ｎｏｔ（Ｇｓ）は、ゴール列Ｇｓの計算が失敗したときに成功し、Ｇｓの計算が成功したときに失敗する。Ｇｓに変数が含まれるときサスペンドする。

④ 方程式（＝）
　数式Ａ，Ｂを等号（＝）で結び方程式制約を与える。Ｘ＝Ｙ（Ｘ，Ｙは数または変数）の形に簡約化できない時にサスペンドする。

⑤ 有限領域（ｄｏｍａｉｎ）
　有限領域制約は次の形で与える。
　　ｄｏｍａｉｎ（Ｘ，＜領域リスト＞）
＜領域リスト＞はアトム，数，ストリングからなるリストである。有限領域制約は、Ｘの値を＜領域リスト＞に含まれる要素のいずれかに限定する。

また、起動条件を付加することにより、任意のユーザ定義述語を制約条件とすることができる。起動条件は、述語の各引数について、グラウンドである，ファンクタが決まっている，未定義でよい、のいずれかを指定する。起動条件を満たさない述語呼び出しはサスペンドする。

4 制約条件に関する前向き推論

実行中にたまった複数のサスペンドしている制約条件については前向き推論が行われる。その結果として、解や矛盾やより強い制約条件が導き出される。このような制約条件は、active constraintsと呼ばれる[3]。active constraintsは、有限領域制約関連，区間制約関連，方程式関連に大別される。

① 有限領域制約関連

Knowledge Media Station (4) Inference Engine
Takayuki YAMAOKA, Hidemitsu KUMAGAI, Tomoomi WAKE
Central Research Lab. MITSUBISHI Electric Corp.

・複数の有限領域制約に関しては、それらの領域の積集合を取り新たな領域とする。
・有限領域制約と非等価がある時は、領域から非等価にされた値を除く。
・有限領域制約と区間制約がある時、領域のうち区間制約の範囲内にある部分集合を新たな領域とする。

以上の操作で、有限領域が空になるとき失敗する。新たな有限領域が一要素から成るとき、変数をその要素に具体化する。それら以外のときは、新しい領域を持つ有限領域制約に置き換える。

② 区間制約関連

同一変数に関する複数の区間制約については、それらの区間の共通部分を求める。共通部分が空であるとき失敗する。共通部分が一点になるとき、変数をその値に具体化する。それ以外のときは、共通部分を新たな区間制約とする。

③ 方程式関連

方程式は、線形のものは全てまとめられ掃き出し法により解が求められる。非線形のものは、線形になったときに線形方程式系に加えられる。線形方程式系が解を持たない（不能）とき失敗する。

制約条件における前向き推論により、新たな制約条件が導かれたとき、それを導いた制約条件は削除される。

5 プログラム例

図１のようなブリッジ回路の設計を行う問題を考える。回路に関する基本的な記述は次のようになる。

```
E=R1*I1+R2*I2.
I1=I2+I5.
I3+I5=I4.
R1*I1+R5*I5-R3*I3=0.
R2*I2-R4*I4-R5*I5=0.
```

次に、回路の特性についての制約条件を記述する。

```
L=[100,200,300,400,500].
domain(R1,L),domain(R2,L),domain(R3,L),domain(R4,L),domain(R5,L).
I1>=0.2,I1=<0.3,I5>=0.05,I5=<0.1,I3>=I1.
```

以上で回路に関する制約条件の記述は終了する。この状態で両端電圧と抵抗値の具体化を行うと電流値が得られる。

```
E=100.
indomain(R1),indomain(R2),indomain(R3),indomain(R4),indomain(R5).
```

とすると、I1=0.29,I2=0.23,I3=0.47,I4=0.53,I5=0.06 が得られる（indomainは有限領域制約のついた変数について、領域内の値の一つを取って変数を具体化する組み込み述語である）。この結果に対して更に制約を加え、微調整を行うこともできる。

```
I4>=0.54.
```

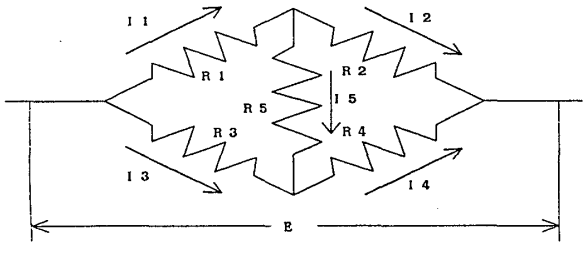

図１．ブリッジ回路

とすると、I1=0.27,I2=0.18,I3=0.45,I4=0.55,I5=0.09 が得られる。

この様な対話型の問題解決は、知識プロセッサの対話的問い合わせ機構[5] の機能により実現される。

6 知識メディアステーションにおける知識表現の特徴

知識メディアステーションにおける知識プログラミングには次のような特徴がある。

＊Ｐｒｏｌｏｇと同様なデータベース機能
＊ＥＳＰのクラス階層に相当するワールド階層による知識のモジュール化機能
＊属性リストによるフレーム表現
＊制約指向による宣言的な知識の記述
＊制約条件についてのデータ駆動推論
＊制約条件による仮説推論

これらのうち最後の３つは制約指向によるものである。論理型言語の特徴の一つに宣言的な知識の記述が可能であることが挙げられる。しかし一般には制御構造を意識して手続き的に書かないと、組み込み述語実行時にエラーが発生したり、正しく動作しても効率の悪いプログラムになったりする。制約指向プログラミングは論理型プログラミングの延長にあり、言語の宣言的記述能力と推論効率とを同時に高めるものである。前節のプログラム例の場合、回路の基本的な記述に関しては、方程式を列挙するだけで良い。これらの方程式の順序については特に意識する必要はない。このように制約指向プログラミングでは、論理型プログラミングに比べて、宣言的な記述能力が高まっている。

また、制約条件として処理される組み込み述語と、起動条件付き述語に関しては、データ駆動推論が行われる。

推論の各時点における解は、そのときサスペンドしている制約条件が全て満たされた時に限って真であるという意味で、条件つきの解といえる [6]。制約条件をサスペンドさせて推論が進めることは、仮説推論を行っていると見なすこともできる。

7 おわりに

知識メディアステーションにおける知識プログラミング言語の概要を紹介した。この言語を用いることで、問題解決の宣言的記述能力の向上と、より柔軟な知識表現が可能となった。

［参考文献］

[1] Lessez,Catherine "Constraint Logic Programming" Byte, Aug., 1987, pp.171-176.
[2] Colmerauer,Alain "Opening the PrologⅢ Universe" Byte, Aug., 1987, pp.177-176.
[3] van Henteryck,P. and Dinkbas,M. "Forward Checking in Logic Programming" Proc. 4th Int. Conf. on Logic Programming, 1987, pp.239-256.
[4] Mukai,Kuniaki "Unification over Complex Indeterminates in Prolog" Proc. the Logic Programming Conf., 1985, pp.271-278.
[5] 熊谷秀光　他“知識メディアステーション（３）対話的問題解決環境”，情報処理学会　第３７回全国大会[6] Vasey,Phil "Qualified Answers and Their Application to Tranformation" Proc. 3rd Int. Conf. on Logic Programming, 1986, pp.425-432.