制約論理型言語処理の一方法

1Q−10

車谷 博之
(株) 日立製作所システム開発研究所

1 はじめに

制約論理プログラミング(CLP Constraint Logic Programming)[1)2)]は、Prologのユニフィケーションを拡張し、等式や不等式の制約を解くことができるようにしたものと理解することができる。CLPはPrologの数値演算処理を高機能化するものとして魅力的である。

しかし、CLPの制約処理は、Prologのユニフィケーションに比べて遥かに重い処理となることが予想される。そこで、Prolog処理で可能な機能はPrologで実行できるように、CLPとPrologプログラム両者を実行し、互いを呼び出すことができるインタフェースを提供する拡張Prolog処理系を実現する。

2 方針

(1)CLP言語仕様

CLP構文は、Prolog言語構文と同一にする。そして、Prologのユニフィケーションを等式、"<"と"<="述語を不等式と解釈する。

(2)制約機能

Prologとの整合性の面から浮動小数点データを対象として、等式、不等式制約を実現する。

(3) CLP-Prolog間インタフェース

PrologからCLPを呼び出す述語clp_call/2、CLPからPrologを呼び出す述語prolog_call/1、CLPプログラムをコンサルトする述語clp_consult/1を提供する。

3 CLP-Prolog間インタフェース

(1) clp_consult(File) : CLPプログラム(File)をコンサルトする。

(2) clp_call(P,ConstList) : (:-mode clp_call(+,-)) CLPプログラムの述語Pを呼び出す。制約処理の結果、求めた解はPのパラメタにユニファイする。ConstList : Pのパラメタである変数の制約リストを返す。

(3) prolog_call(P) : Prologプログラムの述語Pを呼び出す。ここで、CLPプログラムの制約は述語Pの実行には、引き継がれない。Pの実行直後のパタメタがprolog_call(P)実行直前の制約を満たすか否かを判定する。満たす場合true、満たさない場合fail。

4 拡張Prolog処理系

制約処理プログラムはPrologで実現する。これには以下の述語を実現する。

(1) clp_unify(X,Y,Envi,Envo) : 等式 X=Y の制約を実現する。Envi:制約情報入力。Envo:制約情報出力。

(2) clp_le(X,Y,Envi,Envo) : 不等式 X<=Y の制約を実現する。

(3) clp_lt(X,Y,Envi,Envo) : 不等式 X<Y の制約を実現する。

さらに、CLPプログラムはPrologと上記(1)〜(3)の述語に変換するプリプロセッサを実現する。

An Implemetaion for CLP Programs
Hiroyuki KURUMATANI
Systems Development Laboratory,Hitachi, Ltd.

(1) ヘッドユニフィケーションの変換
図1のように、ヘッドユニフィケーションは clp_unify述語に変換する。

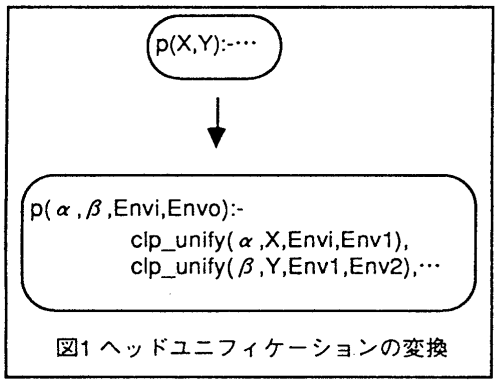

図1 ヘッドユニフィケーションの変換

(2) "=","<","<="述語
"X=Y","X<Y","X<=Y"はそれぞれclp_unify(X,Y,Envi,Envo),clp_lt(X,Y,Envi,Envo),clp_le(X,Y,Envi,Envo)に変換する。

5 制約実現方法
(1) 等式制約:clp_unify(X,Y,Envi,Envo)
ガウスの消去法を漸次に行う。X=Yを次の形式に整理する。
(a) A1*X1+…+An*Xn=0
(b) Y1*…*Yn=Xi
Ai:定数、Xi:変数
非線型項は、(b)のように変数に置き換えて、等式処理を行い、(b)が線型になるまで遅延する。
(a)と次の等式を比較し、共通の変数について解き、変数消去を行う。
変数値を求めた場合、この変数に変数値をユニファイする。さらに、他の変数へこの値を波及させる(後退代入)。
変数消去した変数の情報は、Enviに付加して、Envoとする。
(2) 不等式制約:clp_le(X,Y,Envi,Envo)
X<=Yの各変数について解き、X<=Expとする。Expが含む式の変数消去を次のように行っていく。
Expの変数YをY<=Exp1で置き換える。さらに、Exp1の変数を他の不等式で置き換えていく。このステップを同一変数は一度だけ使用し繰り返す。この結果は、X<=定数、またはX<=f(X)となる。
X<=f(X)となる場合、Xについて解く。
(a) A<=0 (A<=0)の場合、true
(b) A<=0 (A>0)の場合、false
(c) X<=A
(d) A<=X となる。
同様に、X>=Exp についても実行し、各変数について上限値(<=A)と下限値(>=A)を得る。
さらに、各変数の、A1<=X<=A2について、
(a) A1=A2の場合 X=A1
(b) A1<=A2の場合 true
(c) A1 > A2の場合 false となる。
同様にして、clp_lt についても実現する。

6 おわりに
本報告では、CLPプログラムの一実行方法を示した。本方法の特徴は(1)CLP処理系をProlog処理系の拡張機能として実現し、(2)CLPプログラム、Prologプログラム間述語呼出インタフェースを提供した点にある。

参考文献
1) J.Jaffar,S.Michaylov,"Methodology and Implementation of a CLP System",Proc. of ICLP-87,pp.196-218.
2) M.Dincbas,P.V.Hentenryck, et al.,"The Constraint Logic Programming Language CHIP",Proc. of FGCS-88,pp.693-702.