論理プログラムに対する制約付き部分計算システムの実現

1G－7

稲垣浩司　世木博久　伊藤英則

名古屋工業大学

1. はじめに

部分計算は、汎用的なプログラムの入力に関する情報が存在する時、その情報に基づき計算可能な部分をあらかじめ計算することにより、特殊化されたプログラムを生成して実行効率を改善する有力なプログラム変換技法である（[1], [2]）が、論理プログラミングの分野でも、1981年にKomorowskiが導入して以来、盛んに研究が行われている（例えば[3], [4]）。

本研究では、一般論理プログラムに対して、有礎モデル意味論（well-founded model semantics）[5]に基づいた部分計算システムの実現について報告する。このシステムは、等号に関する制約[6]を含むような一般論理プログラムを対象としている。また、そのようなプログラムを有礎モデル意味論に基づいて処理する処理系についても述べる。

2. 制約付き部分計算

2.1 有礎モデル意味論に基づく部分計算

Lloydらによって提案された従来の論理プログラムに対する部分計算[7]は、SLDNF反駁に基づいて定式化されており、プログラムの完全化に基づく意味を保存するための条件が調べられていた。それに対して、本報告で用いる有礎モデル意味論のための部分計算[6]では、SLS反駁[5]に基づいた定式化を用いており、ある条件の下でプログラムの有礎モデル意味論を保存することが示されている。

SLDNF反駁と異なりSLS反駁を用いることにより、与えられたゴールのSLS反駁木において無限に展開されるような枝については、それを積極的に失敗枝と解釈する点が大きな違いである。その結果、無限に展開されるような枝に対応する導出節（resultants）は、部分計算プログラムに含まれない。

2.2 制約付き部分計算とその特徴

プログラムをP、問合せのアトムの集合を$\mathcal{A}$、Pを$\mathcal{A}$について部分計算して得られるプログラムをP'とする。このとき、P'に出現している各アトムで$\mathcal{A}$内のあるアトムの述語記号を含むものが、$\mathcal{A}$内のアトムの代入例である場合、P'は$\mathcal{A}$-closedであるという[7]。従来の部分計算では、この$\mathcal{A}$-closedの条件の下で、プログラムの意味が部分計算により保存されることが証明されていた。それに対して、本研究ではこの$\mathcal{A}$-closednessの条件を仮定しない制約付き部分計算[6]について考える。制約付き部分計算は次のように定義される。

[定義]

Pをプログラム、Aをアトム、Tを$P\cup\{\leftarrow A\}$に対するSLS反駁木、$\leftarrow Q_1,\cdots,\leftarrow Q_m$を$T$における全ての失敗しない枝の根でないゴールとする。

R_i $(0\geq i\geq m)$を$\leftarrow A$から$\leftarrow Q_i$までの導出に対応する導出節（つまり、その導出で計算されるAに出現する変数に対する代入をθ_iとすると、$R_i = A\theta_i \leftarrow Q_i$である）とすると、$A$の（通常の意味の）部分計算とは、節の集合$\{R_1,\cdots,R_m\}$である。

$\mathcal{A}$をアトムの有限集合とする。Pを$\mathcal{A}$について制約付き部分計算したプログラムP'とは、Pから以下のようにして得られる：

1. アトム$A\in\mathcal{A}$と頭部が単一化可能な全ての節をPから取り除く。
2. 1で取り除かれた節の替わりに、
$$B\leftarrow\ \neg\theta, L_1,\cdots,L_k$$
なる形の全ての節を付け加える。ただし、節$B\leftarrow\ L_1,\cdots,L_k$は1.で取り除かれた節で、$\theta\equiv\exists(A=B)$、つまり$A$と$B$の最汎単一化代入である。
3. 2.のプログラムに、各アトム$A\in\mathcal{A}$の（通常の意味での）部分計算を加える。 □

制約付き部分計算の例として、二人ゲームでの局面の先手必勝関係を表すwinのプログラムを考える。

[例]

```
win(X) :- move(X,Y),~win(Y).
move(a,b).
move(b,a).
move(b,c).
move(c,d).
```

このプログラムに対して、アトム$A = win(a)$について制約付き部分計算を行うと、次のようなプログラムが得られる。

An Implementation of Constrainted Partial Evaluator for Logic Programs
Koji INAGAKI, Hirohisa SEKI, Hidenori ITOH
Nagoya Institute of Technology

```
win(a) :- ~win(b).
win(X) :- X ≠ a,move(X,Y),~win(Y).
move(a,b).
...            % move の定義節については不変
```

上のプログラムで、$X \neq a$ なる形の式が、ここで導入された制約である。 □

制約付き部分計算で導入される制約は、一般的には t_1, t_2 を項とすると $\neg\exists(t_1 = t_2)$ なる形をしている。制約 $\neg\exists(t_1 = t_2)$ は、t_1 と t_2 が単一化可能でない時は「妥当 (valid)」と呼ばれる。また、存在束縛された変数だけに適当な代入を施しても単一化可能でない時は「充足可能 (satisfiable)」という。部分計算において制約を計算する時には、制約が妥当ならばその制約は true と扱われ、制約が充足可能でない時は、その制約は false と扱われる。部分計算の途中で出現する制約については、次のような Clark の等式理論に基づいた書き換え規則を用いて簡略化を行う。以下では $\mathcal{E}$ を等式の列とする。

(1) $\neg(X = X, \mathcal{E}) \Longrightarrow \neg\mathcal{E}$

(2) $\neg(f(X_1, \cdots, X_n) = g(Y_1, \cdots, Y_m), \mathcal{E}) \Longrightarrow true$
但し f と g は相異なる関数記号である。

(3) $\neg(f(X_1, \cdots, X_n) = f(Y_1, \cdots, Y_n), \mathcal{E}) \Longrightarrow$
$\neg(X_1 = Y_1, \cdots, X_n = Y_n, \mathcal{E})$

(4) $\neg(t[X] = X, \mathcal{E}) \Longrightarrow true$
但し $t[X]$ は変数 X を含む項で X とは異なる。

(5) $\neg\exists X(X = Y, \mathcal{E}[X]) \Longrightarrow \neg(\mathcal{E}[Y])$
但し等式の列 $\mathcal{E}[Y]$ は $\mathcal{E}[X]$ に含まれる X を Y に置き換えたものである。

2.3 制約付き部分計算の実現

本報告において実現した部分計算システムは次のような性質を持つ。

1. 部分計算の結果のプログラムが $\mathcal{A}$-closed である場合は、通常の部分計算を行うが、満たされない場合は制約付き部分計算を適用する。これによって $\mathcal{A}$-closedness の仮定を満たさない場合でも部分計算を安全に行うことができる。

2. システムが制約付き部分計算を行う場合、制約の部分は前節の簡略化規則を可能な限り適用する。これによって、冗長な導出節は生成されない。

3. SLS 反駁木が無限の枝を持つようなプログラムとゴールに対しては、部分計算中にループを検出し対処するループ・チェックを行う。本研究における部分計算システムでは、ループ検出条件として、過去に展開したゴールと変数を除いて同じゴールを展開した場合と、過去に展開したゴールと変数を含めて全く同じゴールを展開した場合の 2 つを区別している。前者は、ループ・チェックになったことをユーザに知らせ、計算を続行するか停止するかを選択させる。後者はループ・チェックになった時点でその枝を除去する。

4. 与えられたゴールに対する反駁木を構成する時の様々な探索順序の決定を、ユーザが任意に選択できるようになっている。

3. 制約付き論理プログラムの処理系

我々は、制約付き部分計算の結果として生成される等号に関する制約付き一般論理プログラムを処理するための処理系も実装した。この処理系は、有礎モデル意味論に基づいて制約付き論理プログラムを実行する。処理系の実現方法は、XOLDTNF 反駁 [8] に基づきつつ、等号に関する制約を処理するように拡張されている。

4. おわりに

本研究では、等号に関する制約を含むような一般論理プログラムを対象とした部分計算システムを試作した。また、このシステムによって部分計算されたプログラムを有礎モデル意味論の意味論に基づいて処理する処理系も試作した。今後の課題としては、この制約処理の応用として、否定技法などの負リテラルに対する展開等も可能な部分計算システムの実現等がある。

参考文献

[1] 二村 良彦, プログラムの部分計算法, 電子通信学会誌, 1983.

[2] Sestoft, P. and Zamulin, A. V., Annotated Bibliography on Partial Evaluation and Mixed Computation, *New Generation Computing* 6, 1988.

[3] J.Komorowski, An Introduction to Partial Deduction, *Proc. 3rd Int. Workshop on Meta-programming in Logic*, 1992.

[4] J.Komorowski (editor), Special Issue on Partial Deduction, *J. of Logic Programming*, Vol. 16, No. 1 & 2, 1993.

[5] T.Przymusinski, Every logic program has a natural stratification and an iterated fixed point model. *Proc. of 8th PODS*, 1989.

[6] H.Przymusinska, T.Przymusinski and H.Seki, Soundness and Completeness of Partial Deduction for Well-Founded-Semantics, *LNAI 624*, 1992.

[7] J.W.Lloyd and J.C.Shepherdson, Partial Evaluation in Logic Programming, *J. of Logic Programming*, 1991.

[8] W.Chen and D.S.Warren, A Goal-Oriented Approach to Computing Well Founded Semantics, *Proc. of JICSLP*, 1992.