

論理プログラムに対するモード情報の双方向抽出

7Q-5

周 能法 高木 利久 牛島 和夫
(九州大学)

1. まえがき

モード情報は論理プログラムの実行を高速化するための重要な情報である。本論文ではより詳細なモード情報を抽出するために双方向抽出アルゴリズムを提案する。従来の抽出アルゴリズム¹⁾と比べると、このアルゴリズムには以下の二つの特徴がある。1) このアルゴリズムはトップレベル述語のモードだけではなく、プログラム中で使用している組み込み述語のモードも利用する。2) トップレベル述語のモードをトップダウンに、組み込み述語のモードをボトムアップに伝播させることによってより詳細なモード情報を抽出できる。

2. モードの自動抽出

与えられた項 t の具象化状況 I を次に定義する。

もし、 t が基礎項であれば、 I を c (closed)とする。もし、 t が変数であれば、 I を f (free)とする。その他の場合には、 I を d ("don't-know")とする。

ゴールのパターンとはゴールの引数の具象化状況を表すもので、 $[I_1, \dots, I_n]$ というリストで表記する。ここで I_i はゴールの第 i 番目の引数の具象化状況である。ゴールの呼び出しパターンとは評価が始まる直前のゴールの各引数の具象化状況を表すもので、ゴールの返しパターンとは評価が終了直後のゴールの各引数の具象化状況を表すものである。

p を引数の数が n である述語とする。与えられたプログラムに対して p のモードを $[M_1, \dots, M_n]$ で表記する。 M_i を p の第 i 番目の引数のモードと言う。 p の呼び出しモード(返しモード)は次のように与える。もし、述語 p がプログラムの実行中にゴールとして現れる場合に、そのゴールの呼び出しパターン(返しパターン)の第 i 番目の要素が常に c であれば、 M_i を c とする。もし常に f であれば、 M_i を f とする。この他の場合には、 M_i を d とする。

モードの自動抽出問題とは与えられたプログラム及びトップレベル述語の呼び出しモードを入力とし、プログラムに現れるすべての述語の呼び出しモードと返しモードとを求める問題である。

述語 p のある引数の呼び出し(返し)モードを M とする。以下の二つの条件を満足すれば、 M を安全な呼び出し(返し)モードと言う。1) 実行中にゴールとして現れる述語 p に対して、もし M が c であれば、評価が始まる直前(直

後)に対応する引数は常に基礎項である。2) もし M が f であれば、対応する引数は常に変数である。

抽出したモードを使ってプログラムの高速化などを図ろうとするためには、そのモードは安全でなければならない。また、抽出したモード情報ができるだけ多くのモード c と f とを含むようにする必要がある。

3. モードの双方向抽出アルゴリズム

P をcall, assert, retractのメタ述語が使われていない静的なPrologプログラムとする。 M_{top} を $\langle p, CM \rangle$ の形をした要素の集合とする。ここで p はトップレベル述語の名前で、 CM はプログラマが与えた p の呼び出しモードである。 M_{bottom} を $\langle p, CM, RM \rangle$ の形をした要素の集合とする。ここで p はプログラム中で使用している述語の名前で、 CM は p の呼び出しモードの初期値で、 RM は p の返しモードの初期値である。例えば、 p が算術比較の組み込み述語である場合に、 CM と RM とは $[c, c]$ である。isという述語が使われていれば、 M_{bottom} には以下のものが含まれる。

```
<iscc, [c, c], [c, c]>,
<isce, [c, e], [c, c]>,
<isec, [e, c], [c, c]>,
<isee, [e, e], [c, c]>.
```

e は引数のモードについて何も情報が得られていないことを意味する。プログラム中で使用している述語 is を引数により $iscc$, $isce$, $isec$, あるいは $isee$ に書き換える。例えば、" $X1$ is $X-1$ "というサブゴールに対して、評価直前に変数 X が束縛されなければならないため、この is を $isec$ に書き換える。

モードの双方向抽出アルゴリズムは P と M_{top} と M_{bottom} とを入力とし、 P に現れるすべての述語の呼び出しモードと返しモードとを抽出するアルゴリズムである。このアルゴリズムは図1に示すように三つの過程からなる。

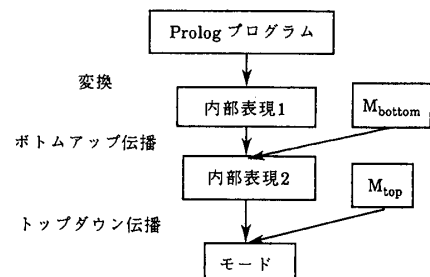


図1 モードの双方向抽出アルゴリズム

3. 1 内部表現

$q_0(T_0) :- q_1(T_1), \dots, q_m(T_m)$ を一つのProlog節とする。ここで q_i は述語名で、 T_i は引数の列である。 $q_0(G)$ を $q_0(T_0)$ と単一化可能な一つのゴール、 CP_0 をこのゴールの呼

び出しパターン, RP_0 をこのゴールの返しパターンとする。 $CP_i (i > 0)$ を $q_i(T_i)$ の呼び出しパターン, $RP_i (i > 0)$ を $q_i(T_i)$ の返しパターンとする。Prologの左から右へという計算規則により, パターン間に以下の依存関係がある。

- 1) $CP_i (i > 0)$ は CP_0 及び $RP_j (j=1, \dots, i-1)$ に依存する。
- 2) RP_0 は CP_0 及び $RP_j (j=1, \dots, m)$ に依存する。

これらの依存関係を直接に表すためにプログラムの内部表現を定める。 q_i の引数の数を n_i とすると, " $q_0(A_0, B_0):-q_1(A_1, B_1), \dots, q_m(A_m, B_m)$ "と表す。ここで A_i と B_i とは長さが n_i であるリストである。リストの要素は具象化状況 c , 変数, または変数のリストである。図2は一つのプログラム及びそのプログラムの内部表現を示す。

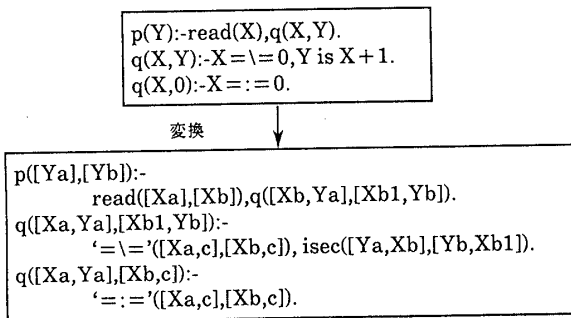


図2 プログラムとプログラムの内部表現

3. 2 ボトムアップ伝播

ボトムアップ伝播の場合に, 述語 q_i の呼び出しモード CM を A_i にある変数を通じて, 返しモード RM を B_i にある変数を通じて節内に伝播する。この伝播関係を図3に示す。図4は, ボトムアップ伝播後, 図2に示す内部表現から変換されたものを示す。

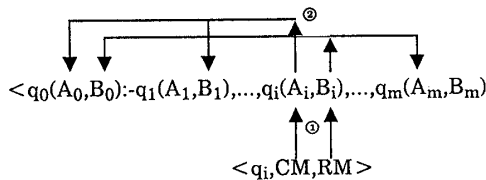


図3 ボトムアップ伝播

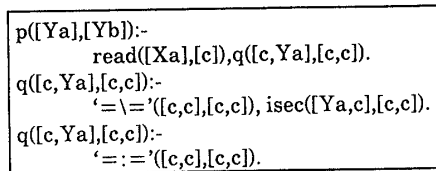


図4 ボトムアップ伝播後の内部表現

3. 3 トップダウン伝播

トップダウン伝播の場合に, 述語 q_0 のゴールの呼び出しパターン CP を A_0 にある変数を通じて節内に伝播する。また, 本体のゴールの返しパターン $RP_i (i=1, \dots, m)$ を B_i にある変数を通じて B_0 に伝播することによってゴールの返しパターン RP を決める。この伝播関係を図4に示す。

3. 4 性質

双方向抽出アルゴリズムは必ず停止する。また, 与えられたPrologプログラムに対してトップレベル述語と組

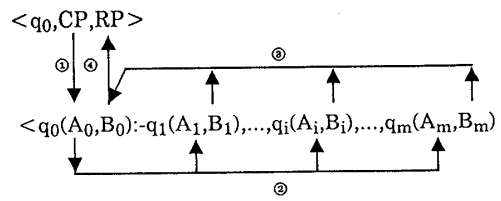


図5 トップダウン伝播

み込み述語のモードが安全であるならば, 双方向抽出アルゴリズムが抽出したモードは安全である。

3. 5 実現

モード抽出を高速化するために我々は以下の技法を考案し, 実現に採用した。

ボトムアップ伝播過程において, M_{bottom} を伝播するだけでなく, トップダウン伝播に役立つないゴールを内部表現の節の本体から削除する。

3. 6 評価

双方向抽出アルゴリズム(BI)は従来の抽出アルゴリズムよりもっと詳細なモード情報を抽出できる。表1は文献²⁾の最後に掲載される6本の実用的なプログラムに対する評価結果を示したものである。SI1はボトムアップ伝播過程を省略し, しかも組み込み述語のモードを利用しない単方向抽出アルゴリズムである。SI2はボトムアップ伝播過程は省略するけれども, 組み込み述語のモードは利用する単方向抽出アルゴリズムである。スラッシュ(/)の左側の数字は呼び出しモードの中で c または f として抽出されたモードの数を示す。右側の数字は返しモードの中で c または f として抽出されたモードの数を示す。

3.5節で述べた実現技法を用いれば, 抽出精度だけではなく, 抽出効率の面においても平均的にはBIはSI2に劣らない。

表1 評価結果

	Program	Number of Clauses	SI1	SI2	BI
1	game (mastermind)	23	25/24	28/34	30/34
2	game(nim)	36	15/19	37/49	39/50
3	game(kalah)	73	23/17	84/107	88/107
4	credit evaluation expert system	54	51/44	57/58	57/58
5	equation solver	123	50/50	77/93	80/94
6	compiler	72	21/17	46/55	47/55

参考文献

- 1) S.K. Debray and D.S. Warren : Automatic Mode Inference for Logic Programs, *J. Logic Programming*, 5, 3, pp.207-229(1988).
- 2) L. Sterling and E. Shapiro : The Art of Prolog. The MIT press(1986).