前向き推論による論理プログラミング言語Monologの一計算方式*

7 B − 1

細井 健太郎† 坂元 紫穂子‡ 中村 克彦¶
東京電機大学大学院理工学研究科§ 東京電機大学理工学部||

1 まえがき

われわれは，超融合法(hyper resolution)に基づく前向き推論による論理プログラミングの計算方式の研究を進めてきた．Prologに代表される多くの論理プログラムはSLD融合または後向き計算に基づくゴール駆動型計算モデルを採用しているが，前向き推論に基づくアプローチは一般的な論理プログラミングとしてはあまり発展していない．しかし，Prologでは，多くの言語のもつ配列や連想記憶の使用などのデータ構造がないため，大量のデータの集合の処理を効率よく行ないにくい．また，ゴール駆動型の並列計算モデルは，共通の変数の代入による同期に多く依存しているため論理的な意味を多く失っているなどの問題点がある．

われわれは，前向き推論に基づく論理プログラミング言語Monologと処理システムを作成して，広範囲の分野への応用をはかっている．今回は，効率の高い計算のために，部分的マッチングの結果を２進探索木に格納する方法について述べる．

2 Monolog言語

Monologのプログラムはホーン節の規則の集合であり，入出力データは単位節である．規則は次の形式をもつ．

$$A_1, \cdots, A_n \rightarrow B_1, \cdots, B_m.$$

*computation of Logic Programs Based on Forward Reasonig
†Kentaro HOSOI
‡Shihoko SAKAMOTO
§Tokyo Denki Univ. Graduate School of Science and Engineering
¶Katsuhiko NAKAMURA
||Tokyo Denki Univ. The Faculty of Science and Engineering

ここで，各A_iとB_jはアトムである．この規則は，左辺が代入σによって単位節により充足された場合に単位節$B_1\sigma, \cdots, B_m\sigma$を生成する．右辺に$\sharp put B_i$の形式のアトムが含まれていとき，単位節$B_i\sigma$が外部へ出力される．

2.1 Monologプログラムの例

図1 に示した有限オートマトンから正則式を求めるMonologプログラムを図2に示す．左辺の$\sharp(r(X_1, \cdots, X_n))$の形式のアトムは変数に対する条件を表わす．プログラム中の$p(X, Y, N, C)$は，状態Xから $(C - 1)$個以下の状態しか経由せずに状態Yへ状態推移させる正則式がNであることを表わしている．$q(X, Z, [N1, N2], Y)$中の$[N1, N2]$は$p(X, Y, N1, Y)$，$p(Y, Z, N2, Y)$の正則式$N1$，$N2$を連接した正則式であり，$r(X, Z, [N1, [N3, *], N2], Y)$中の正則式は，正則式$N1$と正則式$N2$の間に$p(Y, Y, N3, Y)$で表わされる正則式$N3$を加えたものである．

入力データとして状態Xから状態Yへ直接状態推移させる入力記号の集合Nを表わす単位節$p(X, Y, N, 1)$を与える．結果はリスト$[0, +, [1, [0, *], 1]]$となる．

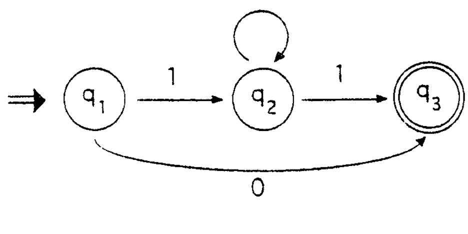

図 1: 有限オートマトン

$p(X,Y,N1,Y),\ p(Y,Z,N2,Y),\sharp(N1 = \phi; N2 = \phi)$
$\quad\to r(X,Z,\phi,Y).$
$p(X,Y,\sharp(N1 \neq \phi),Y),p(Y,Z,\sharp(N2 \neq \phi),Y)$
$\quad\to q(X,Z,[N1, N2],Y).$
$p(X,X,\sharp(N \neq \phi),X),\ q(Y,Z,[N1, N2],X)$
$\quad\to r(X,Z,[N1,[N,*],N2],X).$
$p(X,X,\phi,X),\ q(Y,Z,[N1, N2],X)$
$\quad\to r(X,Z,[N1, N2],X).$
$p(X,Y,\sharp(N1 \neq \phi),C),r(X,Y,\sharp(N2 \neq \phi),Y)$
$\quad\to p(X,Y,[N1, N2],C + 1).$
$p(X,Y,\phi,C),\ r(X,Y,\sharp(N2 \neq \phi),Y)$
$\quad\to p(X,Y,N2,C + 1).$
$p(X,Y,N1,C),\ r(X,Y,\phi,Y)$
$\quad\to p(X,Y,N1,C + 1).$
$p(1,3,N,4) \quad\to \sharp put(\mathrm{N}).$

図 2: Monolog プログラム

3 規則の条件部の処理

左辺に2つ以上のアトムをもつ規則は,

$$A_1, A_2 \to B_1, \cdots, B_m$$

の形式の規則に分解できる. Monolog プログラムの計算では, 単位節 P_i の集合に対して, 左辺のアトムのすべてを満足する規則を求める操作が基本となる. これを効率よく行うため, 次のような方式を用いる.

1. ある代入 σ のもとで $A_1\sigma = P_i$ (または $A_2\sigma = P_i$) なる単位節 P_i が存在するならば, 規則 $A_1, A_2 \to B_1, \cdots, B_m$ から組 $(A_2\sigma,(B_1,\cdots,B_m)\sigma)$ (または $(A_1\sigma,(B_1,\cdots,B_m)\sigma)$) を生成し, $A_2\sigma$ ($A_1\sigma$) の中に含まれる定数の組をキーとして2進探索木に格納する. 規則は固定しているので実際には, 代入 σ だけを作成し, 格納すればよい.

2. 新しい単位節 P_{n+1} が生成されたとき, P_{n+1} に含まれる定数の組をキーとして2進探索木を探索し, 組 $(P_{n+1}, (B_1,\cdots,B_m)\sigma)$ が得られたならば, $(B_1,\cdots,B_m)\sigma$ を新しい単位節として生成する.

4 定理証明への応用

一般の1階述語論理の定理証明のためのモデル生成法は, 与えられた節集合から充足するアトムの集合を前向き計算によって求める. Monolog の前向き計算法式は, 出力されるアトムの集合をモデルとすることでモデル生成型定理証明に応用できる. 定理証明では, 規則として非ホーン節も取り扱う. 非ホーン節を表わすために, Monolog の規則の形式を

$$A_1, \cdots, A_n \to B_1; \cdots; B_m$$

を含むように拡張する. 拡張された規則は, 条件部が単位節の集合により充足されたならば, $B_1\sigma$ を生成する. その後, モデル棄却となったときに, バックトラックにより, 単位節 $B_1\sigma$ を削除し順次別のアトム $B_i\sigma$ を生成する.

長谷川, 藤田らのMGTPでは, 本システムと同様に条件部の一部の照合結果をスタックに記憶している. 本システムで用いた2進探索木によって, 効率の高い探索が可能となっている.

この方式は逐次処理だけでなく, 並列処理にも適用可能である.

5 むすび

本論文では, Monolog の計算方式として, 部分的マッチングの結果を2進探索木に格納する方法について述べた. この方式に基づいて, Prolog により Monolog インタプリタを作成し, さらに定理証明問題に応用した. 今後の課題としてハッシュリストを用いた記憶方式により, さらに効率の良いシステムの作成を予定している.

参考文献

[1] 中村 克彦:“単位消去法：論理プログラムの上向き計算法” *LPC'90*, ICOT.1990.