

拡張項のための重ね合わせ符号を用いた検索方式

1 L - 9

田中 勉, 森田 幸伯

沖電気工業(株) 総合システム研究所

1. はじめに

近年演繹データベース、知識ベース等のシステムでは、Prolog等で用いられている単純な項だけでなく、 ψ 項、素性構造、拡張項など、引数をその位置ではなく、属性名などを用いて、順不同、不定数個記述できるように拡張された項が、広く用いられる。

これら拡張項を要素とする集合から拡張項に対するある種のパターンマッチ検索を行う処理は重要である。しかし、これら拡張項に対しては、検索の高速化を行う技法は知られていない。構造体の検索方式としては、SCW (superimposed code word) 等を利用した様々な方式が知られている[1]~[6]。しかしながら、これらの方式では、対象とするものは、Prologという項であり、引数の数及び位置が固定されており、拡張項などに適用できない。

本稿では、拡張項に対する重ね合わせ符号の構成法を提案し、SCWの手法を適用可能とすることで、高速な検索を可能とする検索方式を述べる。

2. 拡張項

まず、本稿で扱う拡張項について説明する。拡張項は、以下のように定義される。

〈拡張項〉 ::= 〈関数子〉 [" [" 〈属性〉リスト "] "]
 | 〈変数〉 [" [" 〈属性〉リスト "] "]
 | 〈変数〉

〈属性〉 ::= 〈属性名〉 " = " 〈属性値〉

〈属性名〉 ::= 〈拡張項〉

〈属性値〉 ::= 〈拡張項〉 | " [" 〈属性値〉リスト "] "

但し、〈拡張項〉において、〈属性〉リストがない場合(0引数の場合)、定数を意味する。

本拡張項では、以下の特徴がある。

- ・ 順不同、不定数個の引数の記述が可能である。
- ・ 拡張項をネストした構造が扱える。
- ・ 関数子、属性名、属性値として変数が用いることができる。

本稿では、変数付きの拡張項を対象に、パターンマッチするものを求める検索を考える。問い合わせも拡張項であり、問い合わせ中の変数の位置(後で述べる)は拡張項の関数子、属性名、属性値の場合がある。

問い合わせ

(1) X/[age=24,parent={mary}]

検索対象

(2) john[age=24, hobby={music,sport},parent={mary,paul},
works=cs[dname="CS",manager=phil]]

(3) phil[parent={sally,bob}]

(4) sally[age=30,hobby={reading},parent={bob}]

図1 拡張項の例

図1には、検索対象である拡張項および問い合わせの例を示す。図1(2)の属性名worksの属性値は、csを関数子とする拡張項であり、拡張項をネストした構造になっている。図1(1)問い合わせの関数子の変数となっており、この場合図1(2)が解となる。

3. 拡張項の検索方式

本稿で扱う索引検索では、比較する拡張項の変数に対して、適当な拡張項を代入した時、パターンマッチする拡張項を、検索対象から求める。このパターンマッチという意味は、問い合わせの項の中の属性をすべて含み、その属性と関数子が一致するという意味であり、パターンマッチする拡張項は、問い合わせ中の属性すべてを含めば、それ以外の属性を含んでもよい。よって、問い合わせがf[l=a,m=X], 検索対象がf[l=a,m=b], f[l=a,m=b,n=c]の場合、両方とも解となる。

検索処理は、あらかじめ検索対象となる集合および問い合わせに対して索引を作成し、双方の索引同士の論理演算により検索候補を抽出し、その検索候補を用いて解を求める。以下では、索引の構成および検索処理手順について述べる。

3.1 索引の構成

索引は、関数子、属性名、属性値のビットフィールドを定めて、属性名と属性値のビットフィールドの並びを関数子のビットフィールドで覆うように配置し、それぞれのビットフィールドへハッシュを行うことで、構成する。但し、属性名(属性値)が拡張項でかつそのビットフィールドが十分な大きさである場合は、その属性名(属性値)のビットフィールドをその属性名(属性値)の関数子、属性名と属性値のビットフィールドにより構成するものとする。但し、属性名(属性値)が拡張項であってもそのビットフィールドとして十分な大きさをとれない場合は、そのビットフィールドは、その属性名(属性値)の関数子のみで構成する。

例として、図2には図3(5)の索引値の作成結果を示す。但し、各ビットフィールドの長さ、集合値のハッ

シュ値は、以下の形式とする。

・対象となる拡張項に対し、その関数子、属性名、属性値のハッシュ値を各々18ビット、8ビット、8ビットとする。

・属性名(属性値)が拡張項の場合、その拡張項の関数子、属性名、属性値のハッシュ値を各々8ビット、3ビット、3ビットとする。

・集合値のハッシュ値は、その要素のハッシュ値のビットOR演算の結果とする。

3. 2 検索処理手順

検索処理の手順は、以下の(1)~(3)のようになる。

(1)索引の生成

あらかじめ検索対象となる集合および問い合わせに対して索引を作成する。各拡張項の索引値については、拡張項の各属性名と属性値の組および関数子に対して、ハッシュ値をとり、それらの論理和を用いて作成する。但し、拡張項の関数子、属性名、属性値に存在する変数のハッシュ値については、その拡張項が検索対象である場合対応するフィールドを全ビット“1”とし、問い合わせの場合は“0”とする。

(2)索引による比較

(1)で生成される索引同士の論理演算結果により、パターンマッチする可能性のある拡張項を判定する。検索対象の索引値S、問い合わせの索引値Qにおいて、 $Q \wedge S = Q$ を満足する項がパターンマッチする可能性のある(候補である)項である。

(3)パターンマッチの検査

(2)で得られる候補項に対して、実際にパターンマッチするかどうかを検査する。パターンマッチする項が問い合わせの解である。

図3には、拡張項(1)~(5)の索引値、及びその検索結果を示す。この場合、候補として、(2)、(5)が挙げられ、その内、(5)がパターンマッチする項である。必要な情報は索引を用いた絞り込みにより欠落することはないので、 $Q \wedge S = Q$ を満たさない項に対するパターンマッチ検査の省略により検索時間の短縮が図れ、高速な検索が可能となる。

問い合わせ

(1) john[age=24, hobby=X, parent={mary}] ----- 索引値の生成 → 索引値Q 1010101111111011001

検索対象

	索引値S	Q ∧ S	
×	(2) sally[age=30, hobby={reading}, parent={bob}]	0100101111101110111 0000101111101010001	← 候補
	(3) bob[age=40, tel[loc=home]=30]	011101111110110101 001000111110010001	
	(4) mary[age=40, parent={sally}, works=X]	101011111111111111 101010111111011001	
○	(5) john[age=24, hobby={music, sport}, parent={mary}, works=cs[dname="CS", manager=phil]]	101010111111011111 101010111111011001	

↑ 比較 (Q ∧ S = Q)

↑ パターン照合可能な拡張項

図3 検索処理

hg(john) = 101010010010010000	
hg(age) = 00000011	hg(24) = 01000001
hg(hobby) = 00001100	hg(music) = 11001010
hg(parent) = 00100001	hg(sport) = 00001010
	hg(mary) = 11000000
hg(works) = 10001000	hg(cs) = 00000110
	h3(dname) = 010
	h3("CS") = 100
	h3(manager) = 001
	h3(phil) = 011

OR

h(john[age=24, hobby={music, sport}, parent={mary}, works=cs[dname="CS", manager=phil]]) = 101010010010010000	
---	--

図2 図3(5)の索引値の作成結果

4. おわりに

拡張項を要素とする集合から拡張項に対するある種のパターンマッチ検索を行う処理として、拡張項に対する重ね合わせ符号構成法を提案し、SCW手法を適用可能とすることで、高速な検索を可能とする検索方式を述べた。今後は、単一化など他の質問形式に対応するように拡張を行う予定である。

【参考文献】

- [1] Berra, P.B., et al, "Computer Architecture for a Surrogate File to a Very Large Data/Knowledge Bases", IEEE COMPUTER, March 1987.
- [2] Ramamohanarao, K., and Shepherd J., "Answering Queries in Deductive Database Systems", Logic Programming: Proceeding of the Fourth International Conference Vol.2, pp.1014-1033, 1987.
- [3] Roberts, C.S., "Partial-Match Retrieval via the Method of Superimposed Codes", in Proceedings of the IEEE, vol.67, NO12, Dec.'79.
- [4] Wise, M.J., and Powers, D.M.W., "Indexing PROLOG Clauses via Superimposed Code Words and Field Encoded Words", in Proceedings of the IEEE Conference on Logic Programming, Atrantinc City, NJ, January 1984, pp. 203-210.
- [5] 有川他, 「重ね合わせ符号を用いた文献検索システムについて」情報処理学会第54回データベース・システム研究会, 1986
- [6] 森田他, 「MPPMを用いた知識ベースマシン(3)」情報処理学会第35回全国大会, 2C-7, 1987.