

知識ベース管理システム Kappa — 非正規形モデルと項の処理 —

3M-6

宮地泰造^{*} 横田一正^{**} 椿野宣行^{***} 丹羽恵理子^{***}

* 三菱電機㈱, ** ICOT, *** 三菱電機東部コンピュータシステム ㈱

1. はじめに

知識ベース管理システム Kappa は、大量の知識の入力物であり、蓄積可能な限りの多量の知識を蓄積することが1つの主要な目的である。Kappa は、項 (term) を基本データ型として蓄積・利用可能にしている。利用者は、一般のルールや定理証明のための定理などを蓄積することができる。項の基本操作には、単一化 (unification) とパターンマッチとによる検索機能が追加されている。

今回は、項を知識として知識ベースに蓄積する手法として、Nested relation を利用する方法を提案する。

2. 知識ベースにおける項の表現

2.1 単一化とパターンマッチのための知識の整理

知識を蓄積するためには、知識を利用できる形式に整理する必要がある。項をストリングなどに変換して単に蓄積することは容易であるが、単一化やパターンマッチによる利用には不十分である。そこで、知識の整理が必要となる。知識の整理の対象となるものは、項の中のつぎの要素である。

- ・ ファンクタ
- ・ 引数中の変数
- ・ 引数中のアトム
- ・ 引数中のファンクタ

Kappa では、これらの対象をファンクタの位置 (深さ) の情報とともに整理する。

【例1】項とその構成

```

      ファンクタ      引数
      └──┬──┘      └──┬──┘
      a (b (C, d), E)
      1  2-1 2-2 2-3 3    ... 位置情報
      f  f  v  a  v    ... 対象の種類
                        (f: functor, v: variable, a: atom)
  
```

項の構成情報は、テーブルの形式で表現すると図1のようになる。

この情報を、項本体とともに、知識ベースに蓄積・検索できるようにしておけば、単一化やパターンマッチに必要な条件を生成し検索することにより、それらの処理の目標とする項を得ることができる。

対象	位置情報	対象の種類
a	1	functor
b	2-1	functor
C	2-2	variable
d	2-3	atom
E	3	variable

図1. 項の構成情報

2.2 非正規形モデルによる項の蓄積

本稿の非正規形モデルは、Nested relation のことである。非正規形モデルとは、簡単に言えば、1レコード中の属性の値が複数個存在してもよいテーブルである。

【例2】世帯情報の非正規形モデル

世帯主	小児名	趣味	趣味の道具
綾野光一	貴史	サッカー	シューズ
			プロテクタ
			ボール
	美穂	バイオリン	楽譜たて バイオリン
		ピアノ	...
		水泳	...

非正規形モデルでは、複数個の属性値を1レコードに格納することができるので、項本体と項の構造に関する複数個の情報を素直に効率よく1レコードに格納できる。例えば、ファンクタの情報を重要視すると例3のようになる。

パターンマッチや単一化に成功する項を得るためには、この項の情報に対して、項の要素を検索条件に変換して、単に非正規形モデルをアクセスすればよい。

Knowledge Base Management System Kappa

Term Support on Nested Relational Model

Taizo MIYACHI* Kazumasa YOKOTA** Noriyuki TSUBAKINO*** Eriko NIVA***

* Mitsubishi Electric Corp. ** ICOT

*** Mitsubishi Electric Computer Systems

【例3】非正規形モデルによる項の構造情報の蓄積例

項の本体	ファンクタ	引数
a(b(C,d),E)	1 \$ a	2 _ 2 \$ _ 2 _ 2
	2 _ 1 \$ b	2 _ 3 \$ b
		3 \$ _ 3

3. 単一化とパタンマッチ

3.1 単一化

単一化は、PROLOGの単一化と同一である。

【例4】 $a(x, y)$
 $\swarrow \updownarrow \searrow$ 成功
 $a(x, Y)$

3.2 パタンマッチ

“パタンマッチ”とは、項の各引数がよりgeneralな項を見付けてくることである。“general”とは、各引数においてつぎの条件が成立することである。ここでは“ $A > B$ ”と記述して、“AはBよりもgeneralである”を意味することとする。

- ・変数 $>$ 項
- ・変数 $>$ アトム
- ・変数 vs. 変数については、変数間の等値性を考慮する必要がある。等値性が存在しないほうが、よりgeneralである。

【例5】変数：Xで記述された2引数は等値である。
 $f(X, Y, Z) > f(X, Y, X)$

【例6】 $a(X, Y) > a(x, y)$

【例7】 $a(X, y) \not> a(x, Y)$

【例8】 $f(X, Y, Z, Y) \not> f(X, Y, X, Y)$

3.3 単一化の検索条件

単一化の検索は、項のファンクタや引数の各構成要素が同一の位置の要素と単一化可能である項を得ることである。よって、ファンクタ、アトム、変数、項に対して、それぞれ、同一の位置に存在する単一化可能な要素を検索するための検索条件を作ればよい。単一化可能な要素はつぎに示すとおりである。

項 $\rightarrow$ 変数（但し、等値条件に矛盾しない）
 ファンクタ $\rightarrow$ ファンクタ
 アトム $\rightarrow$ アトム、変数
 変数 $\rightarrow$ 何でもよい（但し、等値条件に矛盾しない）

3.4 パタンマッチの検索条件

パタンマッチの検索は、項のファンクタや引数の各構成要素が同一の位置の要素とパタンマッチ可能である項を得ることである。よって、ファンクタ、アトム、変数、項に対して、それぞれ、同一の位置に存在するパタンマッチ可能なよりgeneralな要素を検索するための検索条件を作ればよい。よりgeneralな要素はつぎに示すとおりである。

項 $\rightarrow$ 変数（但し、等値条件に矛盾しないか、よりgeneralである）
 ファンクタ $\rightarrow$ ファンクタ
 アトム $\rightarrow$ アトム、変数
 変数 $\rightarrow$ 変数（但し、等値条件に矛盾しないか、よりgeneralである）

3.5 等値性の処理

項の各構成要素に対して、単一化可能もしくはパタンマッチング可能な要素を同一の位置に持つ項の検索条件の作成の方法は、3.3 節、3.4 節に示した。

しかし、単一化やパタンマッチの演算が実際に起きると同一変数には、同一の値や構造体が固定されるので、同一変数間における値や構造体の伝搬を考慮した演算可能性の検査が必要になる。すなわち、単なる検索により得られた項の中には、このような値や構造体の伝搬に矛盾する項が存在する可能性があり、これを除去する処理を検索された項に対して行なう必要がある。これが、3.3 節、3.4 節の但し書きの処理に対応する。

【例9】単一化の場合

		検査が必要
$a(X, Y, Z)$	$a(X, Y, X)$	
$\updownarrow \updownarrow \updownarrow$ 成功	$\updownarrow \updownarrow \updownarrow$ 失敗	
$a(o, p, q)$	$a(o, p, q)$	

もし上記の検査を検索条件の中に埋め込もうとすると、知識ベース中の項における変数間の関係を、複雑な構造体に対しても検索条件の形式に変換する処理が必要となるが知識ベース中の項の構造情報を見なければならぬので、非効率的で意味がない。

4. ま と め

知識ベースに項を蓄積する方法について、基本的手法を報告した。項を蓄積するテーブルの形式は他にも多く考えられるが、項に対して項の構造情報が複数値存在することには変わらない。このことから、項を蓄積する知識ベースが非正規形モデルであることが有効であると言える。

【参考文献】

- [1] 溝口他, “知識ベース管理システムKappa-データベースから知識ベースへ” 情報大全35回, Sep, 1987.