

3K-4

演繹データベースシステム
EX_DDBにおけるゴール表現の検討

中川 俊夫* 土井 晃一* 嶺野 和夫** 吉田 敦* 田中 英彦*

(*東京大学工学部 **富士通株式会社)

1. はじめに

関係データベースに推論機能を持たせた演繹データベースでは、単一化を伴う関係演算(拡張関係演算)によって、推論処理が実行されるが、ゴールなどの表現の仕方によって、その実行方法は大きく変わってくる。演繹データベースEX_DDB_{csj}(以下、EX_DDBと略す)では、節の定義や変数に対する束縛は、関係の形で表現されているが、ゴールだけは別の表現となっており、逐次、関係に変換されて推論処理が行なわれる。しかし、関係データベースの機能拡張としての演繹データベースとしては、ゴールを含めすべてが関係で表わされている方が自然である。そこで、本稿では、ゴールの関係による表現形式と、それに基づく推論実行方式、そしてその得失について述べる。

2. EX_DDBの推論実行方式

EX_DDBの推論処理は、データベース問合せ言語として用いることを考え、原則として質問に対する全探索を行なう。また、その実行は、拡張関係演算によって行なわれるが、単一化を伴う結合演算(unify join)は、ハッシュとソートによるアルゴリズム_{c2j}を用いて、高速に実行される。

2.1 ゴール表現

EX_DDBでは、ゴールは図1のように、goal構造体と変数の束縛情報を表わしたbinding relationとで表わされ、ゴールスタックに格納される。

?-p1(a1,b1),p2(a1,_2).
?-p1(a2,b2),p2(a2,_2).
?-p1(a3,b3),p2(a3,_2).
?-p3(c1,d1).
?-p3(c2,d2).

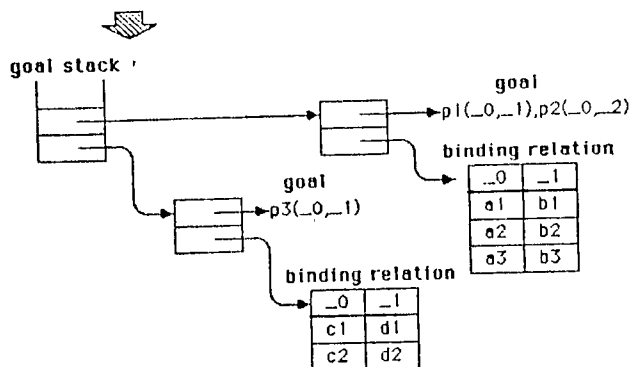


図1. EX_DDBのゴール表現

2.2 reduction方法

EX_DDBの推論処理は、factに対しては変数の束縛情報を生成し、それによって探索空間を絞り込むインタプリタ(即時評価)方式_{c3j}によって、つぎのように実行される。まず、ゴールスタックより一つずつゴール構造体を取り出し、束縛情報をゴール構造体に代入したゴールインスタンスの関係を生成する。その際、ゴールの中から処理するリテラルを一つ選択する。そのリテラルの定義の関係を2次記憶から取り出し、ゴールインスタンスの関係とのunify joinを実行することによって、reductionが行なわれる。生成された新ゴールは、ゴール構造体と束縛情報に変換され、ゴールスタックに格納される。

2.3 ゴール表現に関する問題点

現在のゴール表現では、ゴールスタックより一つずつゴールを取り出して処理するため、拡張関係演算の際にどうしてもタプル数の少ない小さな関係を扱うこととなり、ハッシュとソートを用いたアルゴリズムの高速性が有効に生かされない。また、関係データベースへのゴールスタックといった関係以外のデータ型の導入によるアーキテクチャの複雑化や、ゴール構造体+束縛情報と関係との間の相互交換の手間も問題となっている。

3. ゴールの関係表現による推論実行方式

EX_DDBでは、上記のような問題点があるので、ゴールの関係表現による推論実行方式を現在検討中である。

3.1 ゴールの関係表現形式

ゴールの関係表現には、次の二通りの方法が考えられる。

- (1) 1ゴールを1タプルで表現する。
- (2) 1リテラルを1タプルで表現する。

このゴールの関係を定義の関係とunify joinすることで、関係演算の並列性により、(1)ではOR並列が、(2)ではAND並列が実行できると予想される。ここでは、(1)のゴールの関係表現によるreductionについて述べる。図1と同じゴールを関係表現すると下図2のようになる。

gid	literal	others	ans
g1	p1(a1,b1)	p2(a1,_2)
g2	p1(a2,b2)	p2(a2,_2)
g3	p1(a3,b3)	p2(a3,_2)
g4	p3(c1,d1)	
g5	p3(c2,d2)	

図2. ゴールの関係表現

図2の関係は、1タプルで1ゴールを表わすが、各ゴールの中から処理されるリテラルが戦略によって一つ選択され、選択されたリテラルは属性literalの値として、ゴールの残りのリテラルは属性othersの値として格納される。属性ansには、最初の質問の中に含まれる変数に代入される値が、解として格納される。

3.2 ゴール関係による reduction方法

ゴール関係(goal relation)を用いた reductionは図3のように行なわれる。まず、質問は pregoalという形に変換され、その各ゴールから1リテラルずつ選択され、ゴール関係が生成される。そして、必要な定義を一つの関係definitionにまとめ、ゴール関係とunify joinを行なう。その結果の関係(Unify Join (Goal vs Definition))の属性bodyの値と属性othersの値をつなげて、新しいゴールが pregoalという形で生成される。ここで、pregoalの属性goalの値がnullならば、属性ansの値が解となる。

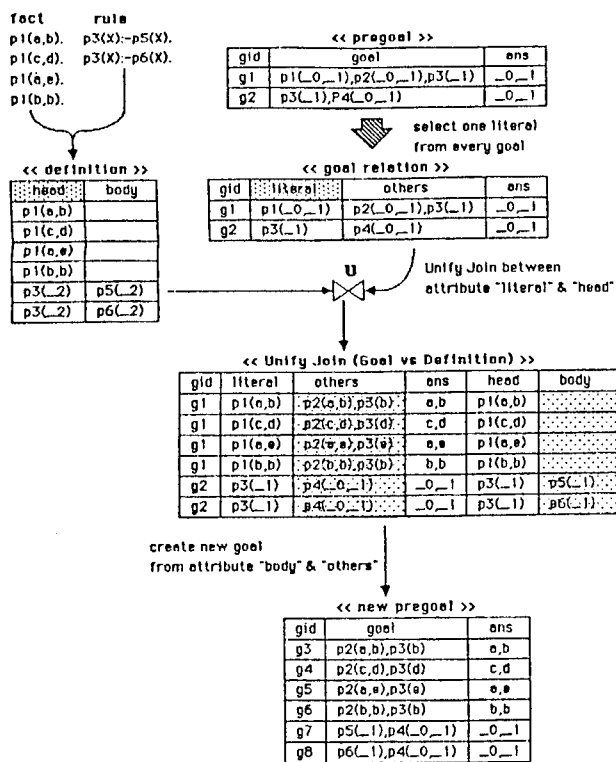


図3. reductionの実行方式

3.3 ゴールの関係化に対する考察

ゴールを関係化したことによって、次のことが言える。

- (1) ゴールを含めたすべての情報が関係の形に統一されたので、アーキテクチャの負担が減る。
- (2) 複数のゴールを格納したゴール関係に対する拡張関係演算によって、自然なOR並列が行なわれる。
- (3) EX_DDBの従来方式より扱う関係が大きくなるので、ハッシュとソートを用いたアルゴリズムによる関係演算の高速性が生かされる。

(4) 複数ゴールが同時に処理されるので、定義の2次記憶からの読み込みや拡張関係演算にかかる手間が減らせる。

このうち、特に(3)の関係が大きくなるのが、ハッシュを用いるための必要条件であり、それによりハッシュとソートを用いたunify joinが効率的に行なわれて、拡張関係演算の手間を減少させることが可能となるのである。

3.4 リテラル選択の戦略

このreduction方式では、ゴール関係を生成する際にリテラル選択を行なうが、ここでゴール中のどのリテラルを選択するかによって、後のreductionの手間が大きく違ってくる。そこで、リテラル選択は、reductionの手間が少なくなるような戦略によって行なわれる。戦略は、質問の前処理_[1]で決定される静的情報と、reduction中のゴールに含まれる残存変数の数などの動的情報から決定される。

また、定義の関係とのunify join時に、同じ定義のリテラルがいくつかのゴールに含まれているならば、そのリテラルを選択し一度のunify joinで済ますようにする。それによって、定義探索による2次記憶へのアクセスが減らせる。ただし、こういった選択は、後のreductionの手間が大きく増加することのない範囲で行なわれる。

3.5 組み込み述語の実行

EX_DDB上では、ROTHORNという述語論理型言語によって、推論処理を記述する。ROTHORNは、四則・比較演算や入出力操作を組み込み述語として持つ。その組み込み述語の実行は、reduction cycleの先頭、pregoalの時点で行なわれる。ゴール内の組み込み述語が実行可能かどうか引数のチェックが行なわれ、可能なら即実行して、その結果によりゴールを書き換えが行なわれる。

4. おわりに

EX_DDBでの推論処理を実行するのに、ゴールの関係表現による方法を示した。ゴールの関係表現化により、関係データベースのより自然な機能拡張としての演繹データベースが構成されるであろう。しかし、現在の方法ではゴールの長さの変動が大きく、データベースに文字列として格納する際のレコード長が問題となる。そこで、3.1(2)の1リテラルを1タプルで表現する方法によるAND並列の実行も現在検討中である。

参考文献

- [1] 吉田、中川、土井、嶺野、田中：「演繹データベースシステムEX_DDBの概要」第34回情報処理学会全国大会 3k-1
- [2] 大森、田中：「単一化可能バクン検索の最適化」第33回情報処理学会全国大会 5L-6
- [3] 吉田、土井、大森、嶺野、田中：「推論機能と関係データベースの融合 - 実行方式の評価 -」第33回情報処理学会全国大会 5L-8