

命題レベル高速推論法を利用するための述語 論理版仮説推論知識のリフォーメーション*

1 N-9

山本 哲† Helmut Prendinger† 石塚 満†

‡ 東京大学工学系研究科

1 はじめに

人工知能の分野における諸問題を記述する為には述語論理による表現が有効であるが、その推論速度は実用上有効な高速化が未だ達成されていない。一方、命題論理版についてはGSATやNBP [1]といった高速推論方法が提案されている。そこで、述語版仮説推論システムを対象として、あらかじめ一階述語論理の知識ベースを変形しておくことにより、効率的な命題論理への変換を可能にする方法が先行研究 [2] で提案されており、本研究はこれを発展させたものである。

2 述語版仮説推論システムの構成

本研究で用いる述語版仮説推論システムの概要は次のようになる。一階述語論理の知識を入力とし、この知識の変形を行なった後に各変数を展開して命題論理へと変換する。この結果を命題論理版仮説推論システムであるNBPによって解く。

ここでは扱う知識を一階述語のホーン節とする。節 C のヘッドを $hd(C)$ 、ボディを $bd(C)$ と表す。 $\mathcal{V}(hd(C))$ 、 $\mathcal{V}(bd(C))$ をそれぞれヘッド、ボディ中に現れる変数の集合とする。また、関数は扱わないものとする。

3 述語知識のリフォーメーション

知識ベース中の不要変数（節中のボディに現れてヘッドに現れない変数）を取り除くことによって、効

率的な命題論理への変換ができるようする。

例 次のような節について考える。

$$C_0 : q(X, Y) \leftarrow p1(X, Z1) \wedge p2(Z1, Z2) \wedge p3(Z2, Y)$$

この節を次のような二つの節に変形する。

$$C_1 : q(X, Y) \leftarrow newp(X, Z2) \wedge p3(Z2, Y)$$

$$C_2 : newp(X, Z2) \leftarrow p1(X, Z1) \wedge p2(Z1, Z2)$$

この例において、各変数がとり得る定数の数をそれぞれ5個であるとすると、 C_0 においては命題論理に展開した際に得られる節の数は $5^4 = 625$ となるが、変形後の C_1, C_2 については $5^3 + 5^3 = 250$ となり、生成される節の数を減らすことができる。

このようにして全ての不要変数を除去するアルゴリズムが [3] において提案されているが、このアルゴリズムは正常な終了が保証されていない。ここではいくつかの不要変数を取り除き、正常に終了する手続きを提案する。

3.1 節の形による分類と手続き

節の形によって次のような分類を行い、それぞれの場合について不要変数除去手続きを行なう。

ブロック 節のボディ中の二つのアトム B_1, B_2 において、 $\mathcal{V}(B_1) \cap \mathcal{V}(B_2) \neq \emptyset$ であるとき、 $R(B_1, B_2)$ と記す。この関係 R によって結ばれるアトムの集合をブロックと呼ぶ。

ブロック中に2つ以上の不要変数がある場合、新しい述語を生成することによって不要変数を除去する。

例えば

$$q(X, Y) \leftarrow p1(X, Z1, Z2) \wedge p2(Z2) \wedge p3(Z2, Y)$$

* First Order Predicate Knowledge Base Reformation for Propositional Hypothetical Reasoning

† Tetsu Yamamoto, Helmut Prendinger, Mitsuru Ishizuka

‡ University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan

e-mail: {tetsu, helmut, ishizuka}@miv.t.u-tokyo.ac.jp

のボディは一つのブロックから構成される。この節に対しては次のような新しい述語が定義される。

$$\text{newp}(X, Z2) \leftarrow p1(X, Z1, Z2) \wedge p2(Z1)$$

そして元の節は $q(X, Y) \leftarrow \text{newp}(X, Z2) \wedge p3(Z2, Y)$ となる。

chain ブロック B_l が、(i) 全ての近接するアトム A_i, A_j に対して $R(A_i, A_j)$ で、(ii) ブロック中の最初と最後のアトムが少なくとも一つは $hd(C)$ 中の変数を持ち、(iii) 他のアトムが $hd(C)$ 中の変数を持たず、(iv) $R(A_i, A_j)$ 中の変数がそれぞれ異なる時、このブロック B_l を chain と呼び、 $\langle A_1, \dots, A_k \rangle (k > 2)$ で表す。

先程の例の C_0 は chain であり、 $p1(X, Z1)$ と $p3(Z2, Y)$ 中の X, Y を embracing variable と呼ぶ。

chain に対しては、新しい述語

$$\text{newp}(X_1, \dots, X_n) \leftarrow A_1 \wedge \dots \wedge A_{k-1}$$

が生成される。ここで X_1, \dots, X_n は A_1, A_{k-1} の embracing variable である。

孤立ブロック部 節 C が $A \leftarrow B_1 \wedge \dots \wedge B_n$ の形であるとする。ここでアトム B_i を、ある $X \subset \mathcal{V}(B_i)$ に対して $X \cap (\mathcal{V}(\{A, B_1, \dots, B_n\}) \setminus X) = \emptyset$ となるとき、 B_i を孤立ブロック部と呼ぶ。

$$q(X, Y) \leftarrow p1(X, Z) \wedge p2(Z, Y) \wedge p3(Z, Z1)$$

において、 $Z1$ は孤立変数であり、 $p3(Z, Z1)$ は孤立ブロック部である。

この場合には新しい述語

$$\text{newp}(Z) \leftarrow p3(Z, Z1)$$

が生成される。

4 実験結果

次の節を用いて知識変形による効率改善の実験を行なった。

$$\text{path}(X, Y) \leftarrow \text{link1}(X, Z1) \wedge \text{link2}(Z1, Z2) \wedge \text{link3}(Z2, Z3) \wedge \text{link4}(Z3, Y)$$

この節は次のように変形される。

$$\begin{aligned} \text{path}(X, Y) &\leftarrow \text{newp1}(X, Z3) \wedge \text{link4}(Z3, Y) \\ \text{newp1}(X, Z3) &\leftarrow \text{newp2}(X, Z2) \wedge \text{link3}(Z2, Z3) \\ \text{newp2}(X, Z2) &\leftarrow \text{link1}(X, Z1) \wedge \text{link2}(Z1, Z2) \end{aligned}$$

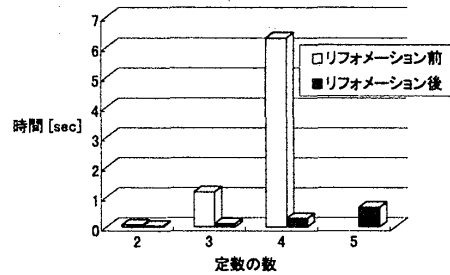


図 1: 知識リファメーションによる推論時間の短縮

ここで、各変数がとり得る定数の数が5であるとすると、変形前の知識によって生成される命題論理節の数は3125個であるが、変形後の知識では375個となり、大幅に縮小することができる。

また、図1に示すように変形後の知識においては推論時間が大幅に短縮されている。なお、この推論時間は述語知識の変形、述語知識から命題知識への変換、NBP法によるコストに基づく準最適解の計算時間を含んでいる。

5 むすび

本研究で提案された述語知識のリファメーションにより、述語論理の知識から生成される命題論理知識の数を大幅に縮小することが可能であり、述語版仮説推論システム全体の高速化につながるものと期待される。

参考文献

- [1] Ohsawa, Y. and Ishizuka, M.: Networked bubble propagation: a polynomial-time hypothetical reasoning method for computing near-optimal solutions, *Artificial Intelligence*, vol. 91, pp. 131-154, 1997.
- [2] 棚橋, 福田, 石塚: 命題論理版高速仮説推論法を利用した述語論理版仮説推論システム, 人工知能学会全国大会, No. 20-04, June 1998.
- [3] Proietti M. and Pettorossi, A.: Unfolding — definition — folding, in this order, for avoiding unnecessary variables in logic programs, *Theoretical Computer Science*, vol.1 42, pp. 89-124, 1995.