

一階論理表現の等価変換を用いた問題解決の正当性

吉田 忠行[†] 赤間 清^{††} 宮本 衛市[†]

[†] 北海道大学 大学院 システム情報工学専攻
札幌市北区北 13 条西 8 丁目. Tel. 011-706-6814

^{††} 北海道大学 情報メディア教育研究総合センター
札幌市北区北 11 条西 5 丁目. Tel. 011-706-6814

論理プログラミングの理論では、当初は確定節で問題を表現していた。その後、否定の表現を取り入れ、さらには、否定を含む確定節を基礎として、一階述語論理の表現も確定節のボディで扱うことができるようになった。しかし論理プログラミングは、このような一階述語論理を含む確定節で表現されたプログラムを計算することに関して、正当性と柔軟性に問題を抱えている。本論文では、この問題を解決するために、等価変換パラダイムを採用する。

等価変換の枠組で一階述語論理の表現を扱うために、一階論理制約の概念を導入する。一階論理制約とは、基本的な制約を論理記号によって組み立てた複合的な制約である。まず、一階論理制約の表現を定義し、その表現に対する意味を与える。そして、一階論理制約のための等価変換ルールを提案し、その正当性を証明する。

等価変換 問題解決 一階論理制約 宣言的記述 正当性

The Correctness of Problem Solving using Equivalent Transformation of First-order Logical Expressions

Tadayuki Yoshida[†] Kiyoshi Akama^{††} Eiichi Miyamoto[†]

[†] Division of System and Information Engineering, Hokkaido University
North 13 West 8 Kita-ku Sapporo. Tel. 011-706-6814

^{††} Center of Infomation and Multimedia Studies, Hokkaido University
North 11 West 5 Kita-ku Sapporo. Tel. 011-706-6814

In the theory of logic programming, the class of definite clauses was adopted first. Then the concept of logic programs was extended to include negative literals and arbitrary first-order formulas in the body of definite clauses. Computation of first-order expressions has, however, some difficulties with respect to the correctness and the flexibility. In this paper, the equivalent transformation paradigm (ET paradigm) is adopted to overcome these difficulties.

We introduce first-order logical constraints, which are compound constraints composed of basic constraints constructed using logical connectives and quantifiers. We define the class of FLCs and determine the meaning of them. Next, we propose ET rules for FLCs and prove the correctness of these rules.

equivalent transformation, problem solving, first-order logical constraint,
declarative description, correctness

1 まえがき

1.1 論理パラダイムでの一階述語論理式の表現と計算

論理プログラムでは、問題を論理式によって表現するというのがすべての基本になっている [4]。はじめは、表現を正リテラルの確定節のみに限定することによって、処理の高速度を達成した。そのあと、表現の拡大のため、否定をボディにもつ表現（負リテラル）が導入された。さらに、任意の一階述語論理式をもボディに記述できるように拡張され [5]、表現力が飛躍的に拡大した。

このように、表現方法に論理式を用いることで、一階述語論理式も知識表現に取り入れることが可能である。一方、計算は、論理式での表現を扱うために、SLDNF-レゾリューションに基づく推論を用いる [4]。一階述語論理式を扱うときには、いったん一階述語表現を含む確定節を、アトムを否定のみを含む節に論理的に変換してから推論を行う [5]。論理パラダイムではこのように、取り入れた表現によって計算の方法が決められる。しかし、その結果、計算に困難が生じることがある。その困難は、計算が2段階に分かれていて、一貫性がないことなどに起因している。これについては、2章でより詳しく述べることにする。

1.2 等価変換パラダイムの採用

本論文では、論理パラダイムが抱える問題を解決するために、等価変換パラダイム [1] を採用する。等価変換パラダイムでは、問題を論理式ではなく、集合を記述するための宣言的記述で表現する。表現には、それに対する意味を与える。明確な意味を定義できるものならば表現はどんなものでも採用できる。これにより論理式でのみ表現する論理パラダイムでの方法とは違い、多様な表現方法を取り入れることができる。また計算は、“宣言的記述の意味の保存 (=等価変換)” という単一の原理だけを採用する。表現に対する意味を与えると、それを保存するルールを数学的に定めることができる。等価変換パラダイムでは、計算は、等価変換ルールを適用することにより宣言的記述を単純化することである [1]。

1.3 本論文の目的

本論文では、一階述語論理に相当する表現力を宣言的記述に与え、その表現を計算するための等価変換ルールの正当性を証明する。これにより、論理パラダイムでは困難であった一階述語論理に対する計算の正当性の保証を、等価変換パラダイムにおいて達成する。

2 一階述語論理を用いた問題解決の困難

2.1 計算が困難になるケース

次のような問題を考える。

$\{1,2,3\}$ の部分集合で、要素が1つのものを求めなさい。

この問題に対して、論理パラダイムでは、まず部分集合の関係を定義する述語 *subset* を、一階述語論理式を用いて次のような論理プログラムで表現する。これをプログラム (1) とする。

```
subset(X,Y)
  ← ∀E (member(E,X) ⇒ member(E,Y)).
```

また、問い合わせとして、

```
← subset([X],[1,2,3]).
```

のようなゴール節を与える。これを (q) とする。

これらの論理プログラムを論理パラダイムでは計算を推論によって行う。論理パラダイムにおいて (1) のような一階述語論理式を用いた論理プログラムを計算するには、次のような2段階に分かれた方法を採用している [5]。

- (1) もとの論理式をそれと論理的に同値な論理積と否定だけを用いた表現に変換する。
- (2) 先に変換された論理式に対して推論をし、推論の際に引き起こされる代入により解を得る。

先のプログラム (1) を論理積と否定のみを含む形に変換すると、次のようなプログラムが得られる。

```
subset(X,Y) ← ¬q(X,Y).
q(X,Y) ← member(Z,X), ¬member(Z,Y).
```

これをプログラム (2) とする。このプログラム (2) は、“ZがXに含まれていて、ZがYに含まれていないなら $q(X,Y)$ が成り立ち、 $q(X,Y)$ でないなら $subset(X,Y)$ が成り立つ” ということを表現している。

ゴール節 (q)

```
← subset([X],[1,2,3]).
```

はプログラム (2) の *subset* 節で展開され、

```
← ¬q([X],[1,2,3]).
```

となる。しかし、このあとSLDNF-レゾリューションは $¬q([X],[1,2,3])$ を選択することができない。なぜなら、SLDNF-レゾリューションには

否定を含むリテラル（負リテラル）が中に変数を含んでいる場合、その負リテラルを選択できない

という制限があるからである。したがって、SLDNF-レゾリューションでは、この問題の解 X を得ることができない¹。

2.2 計算の一貫性と柔軟性

このように、論理パラダイムでは、前半の論理的な変換が成功しても、SLDNF-レゾリューションの段階で計算が不可能になることが起こり得る。この直接的な解決方法は、新しい計算方法を提案し、SLDNF-レゾリューションの問題を克服することである。

しかし、計算が困難になる要因は、SLDNF-レゾリューションに限られたものではない。論理パラダイムにおいては、一階述語論理表現に対する計算が2段階に分離していて完全な一貫性がなく、全体を統一的に扱うことが

¹ 論理パラダイムの言語 Prolog では、このような場合でも無理矢理計算を進めるが正しい解が得られる保証がない。

できない。段階が分離していると、第一段階が終了しなければ、第二段階に移ることができないために、計算に柔軟性を欠くことになる。このことも、計算が停止する可能性を増大させる要因の一つである。

本論文では、等価変換パラダイムを採用することによって、一貫性と柔軟性の問題を解決し、一階述語論理表現の新しい計算の枠組を提案する。

3 本論文の概要

本論文では、前章で述べた論理パラダイムにおける問題を克服するために、等価変換パラダイムを採用する。本章では、等価変換パラダイムにおける一階論理表現に対する枠組の概要を述べる。

3.1 一階論理制約の導入

等価変換パラダイムでは、問題を宣言的記述によって表現する。宣言的記述とは、基本的にはアトムによって構成される確定節の集合である。このアトムと基本的な制約を含む宣言的記述でも、多くの問題を表現することができる。しかし、われわれが扱う問題の中には、複雑な構造の制約を含む問題も数多く存在する。それらの問題をアトムと基本的な制約のみで表現するには、制約の表現能力の乏しさから、煩雑な記述にならざるを得ない。したがって、宣言的記述の表現力の拡大のために、一階述語論理で用いられる表現力を宣言的記述に導入する。

まずはじめに、基本的な制約を束縛変数と論理記号で組み立てて作った、複合的な式のクラスを構成する。そして、この式のクラスのうち、束縛変数に対する一定の条件を満たすものを一階論理制約と認める。これにより、アトムと基本制約で表現できる基本的な宣言的記述のクラスを包含する形で、アトムと一階論理制約で表現できる宣言的記述のクラスを構築する。

3.2 参照を含む宣言的記述

等価変換パラダイムでは、宣言的記述が他の宣言的記述を参照することができる [2]。そのために、参照というものを導入している。参照は、アトムと、参照方法を決める写像、それに参照する宣言的記述の3項組によって作られる。例えば、基礎アトムに関する補集合を返す写像を f_c とし、アトム a が宣言的記述 Q から計算される基礎アトムの集合 (Q の意味) に含まれていないことを表現するには、 $\langle a, f_c, Q \rangle$ のように表す。

参照は、それ自身で真偽が決まる制約の一種ともみなすことができる。というのは、基礎アトムによる参照 (g, f, P) は、 P の意味を $M(P)$ とし、 $g \in f(M(P))$ のとき真であり、 $g \notin f(M(P))$ のとき偽である、と定義しているからである。しかし、ここでは制約と参照は別のシンタックスで扱う。

このようにして、ボディに、アトムと、一階論理制約、それに参照制約を含んでいる確定節の集合として宣言的記述のクラスを構成できる。ボディの参照制約は、他の宣言的記述を参照しているので、宣言的記述は全体として、木構造を成していることと見ることが出来る。

3.3 一階論理制約の計算

等価変換パラダイムでは、計算は等価変換ルールによって行われる [1]。したがって、一階論理制約を導入した後は、それに対する計算ルールを必要なだけ用意する必要がある。

本論文では、一階論理制約を計算するための最低限必要なルールの正当性を保証する。最低限というのは、計算することが不可能になるような状況を作り出さないために必要である、という意味である。このため、各論理記号に対する包括的な計算ルールを1つずつ与える。この計算ルールを複数回用いることにより、一階論理制約を用いた宣言的記述は、その一階論理制約がより基本的な制約に分解されながら、意味的に等価な宣言的記述に変換される。

計算においては、参照が重要な役割を果たす。否定以外の論理記号は、それと等価な一階論理制約に変換する“閉じた”変換であるが、否定に関しては、それと意味的に等価な一階論理制約に置き換えることができない。この場合は、宣言的記述の意味に含まれないことを意味する参照を用いて等価変換ルールを作成する。

アトムと、一階論理制約、それに参照を含む宣言的記述は、等価変換によって単純化され、一階論理制約を含まない形に展開される。

3.4 理論の展開

表現に関する部分では、従来の項に相当するデータ構造を基本にする。そして、項における変数と一階論理表現で用いる束縛変数を別々の変数集合として定義し、通常とは少し違う形で変数を扱う。一階論理表現を用いるのは、アトムに対してではなく、制約に対してである。基本的な制約を論理記号により組み立てた式のクラスを定義し、その式の中で束縛変数にはそれぞれに対応する量化記号が必ず1つ存在するという条件を満たすものを一階論理制約として定義する。

計算に関しては、個々の論理記号に対する等価変換ルールを与える。そして、それらのルールの正当性を証明する。

4 両パラダイムの比較

4.1 理論的基盤に関する相違

論理パラダイムでは、論理式により問題を表現し、計算は推論によって行う。理論体系の基礎には、モデルや論理的帰結といった論理構造のための概念が不可欠である。

等価変換パラダイムでは、計算とは等価変換であると考え、集合を記述するための拡大された宣言的記述で問題を表現する。計算は、宣言的記述の意味を保存することのみで規定されていて、理論体系の基礎には、論理構造を必要としない。

4.2 計算の正当性の保証に関する比較

論理パラダイムでは、一階述語論理式を含む論理プログラムと問い合わせに対する計算は、“論理的な同値性”を保つ表現の変換の段階と、SLDNF-レゾリューションに基づく推論の段階という、2つの段階に分かれた構造を採用している。すなわち、全体が1つの原理で統合されていない。この理論構造は複雑であり、全体としての計算の正当性を保証するのは困難がある。

一方、等価変換パラダイムでは、計算を“宣言的記述の意味の保存”という単一の原理を採用している。計算の正当性は、個々のルールが等価変換を引き起こすことを証明することで保証できる。一階論理制約に対する計算に関してもその等価変換ルールの正当性を証明できた。

4.3 計算の柔軟性に関する比較

もともと、論理パラダイムは計算を推論に限定しているので計算の自由度が低いといえる。さらに、SLDNF-レゾリューションにおけるアトムを選択には、“負リテラルの引数には変数を含まない”という制限がある[4]。これは計算を完結できる範囲をかなり狭めている。この問題を解決するためには、変数を含んだ負リテラルを積極的に計算できる方法が必要である。

等価変換パラダイムでは、負リテラルすなわち、否定は一階論理制約として扱い、計算では参照を用いて扱っている。これにより、負リテラルに対する特別の制限を設けることなく、計算が可能になっている。その他の計算に関しても“等価変換”という非常に緩い制限のため計算の自由度を高めている。

4.4 論理パラダイムが持つ困難の解決

一階論理表現を用いた問題解決において、論理パラダイムには困難な点があることを、2章で例をあげて示した。この困難は、等価変換パラダイムでは解決することができる。

つまり、2章で取り上げた例題のような一階論理表現を含む問題は、等価変換パラダイムでは、本稿で正当性を証明した R1 から R5 までのルールを用いることによって、正しく計算できる。具体的には、一階論理制約に関して R1 から R5 のルールを用いると、2章の例題は28回の変換で正しく解くことができる。そのうち、R1 から R5 までのルールは計7回適用されている。

このように困難を克服できるのは、等価変換パラダイムが、“宣言的意味の保存”という緩やかな計算の制限のもと、多様な表現方法を可能にし、広範な計算空間を提供していることに起因している。

5 むすび

論理パラダイムで一階述語論理式を扱う際には、特に計算には、柔軟性や正当性に関して問題点を抱えていることを本論文では指摘した。そして、その論理パラダイムでの困難を解決するために、本稿では等価変換パラダイムを採用した。

この等価変換パラダイムに基づいて、より高度な問題解決の枠組を構築するために、基本的な制約を論理記号で組み立てた一階論理制約を導入することで宣言的記述のクラスを拡大した。一階論理制約含む拡大された宣言的記述の計算に関しては、一階論理制約のための等価変換ルールを導入した。このルールを適用することで、拡大された宣言的記述は、その基本となる宣言的記述に還元される。したがって、表現と計算の両面に関して、一階論理制約という新しい制約の導入が正当に行われたことを述べた。

しかし、さらに等価変換ルールを追加することによって計算の効率を上げることも可能である。例えば、否定記号が先頭にある式は、その形に特化したルールを用いると無駄な節の展開を抑えられる。このようなルールはすでに開発されているが、それらの正当性は、今後の論文で証明することとする。

等価変換パラダイムに基づくプログラミング言語 ETC (Equivalent Transformation Compiler) では、すでに参照に関する実装が行われ[3]、さらに本論文で与えた一階論理制約の等価変換ルールも ETC の中に組み込まれている。したがって、一階論理制約を用いた問題を、ETC の上で実行して解くことが可能である[7]。したがって、これまで以上に多くの問題が簡潔に表現され、計算可能になっている。

参考文献

- [1] 赤間清, 繁田良則, 宮本衛市: 論理プログラムの等価変換による問題解決の枠組, 人工知能学会誌, Vol.12, No.2, pp.90-99 (1997).
- [2] 赤間清, 岡田浩一, 宮本衛市: 否定を表現する新しい枠組-等価変換に基づくアプローチ-, 電子情報通信学会技術研究報告, SS 97-37, pp.1-8 (1997).
- [3] 清水伴訓, 赤間清, 宮本衛市: 等価変換に基づく計算のための多重世界機構, 電子情報通信学会技術研究報告, SS97-32, pp1-8 (1997).
- [4] Lloyd, J.W.: Foundations of Logic Programming, Second Edition, Springer-Verlag (1987).
- [5] Lloyd, J.W. and R.W. Topor: Making Prolog More Expressive, J. Logic Programming 1,3, pp225-240 (1984).
- [6] 吉田忠行, 赤間清, 宮本衛市: 一階論理制約の等価変換を用いた問題解決の枠組, 電子情報通信学会技術研究報告, KBSE98-6, pp17-24 (1998).
- [7] Yoshida, T., Akama, K., Miyamoto, E.: Problem Solving by Equivalent Transformation of First Order Logical Constraints, 情報処理学会研究報告, 98-ICS-114-2, pp7-12 (1998).