数理モデル化と問題解決 27－12
（1999．11．26）

負制約の等価変換による問題解決の基礎理論

岡田浩一[†] 赤間清[††] 宮本衛市[†††]

論理パラダイムでは否定を正しく計算するのは容易ではない．SLDNFレゾリューションでは，変数を含んだ負リテラルを計算のために選択できないので，否定を含んだ計算はしばしば計算途中で実行不可能になってしまう．本論文では，正当な計算手続きを組織的に発見する方法を用いて，新しい計算手続きを提案する．これは論理パラダイムにおける否定の問題を克服している．すなわち，SLDNFレゾリューションと異なり，本方法では変数が入った負制約を選択して，等価変換ルールにより正しく計算を続けることができる．

A Theoretical Foundation of Problem Solving by Equivalent Transformation of Negative Constraints

KOICHI OKADA,[†] KIYOSHI AKAMA[††] and EIICHI MIYAMOTO[†††]

In logic programming, it is not easy to compute negation correctly. Since non-ground negative literals can not be selected for computation in SLDNF-resolution, computation with negation often halts without obtaining answers. In this paper, we propose a new computation procedure for negation by a method of inventing correct computation procedures systematically. This method overcomes the difficulty of negation in logic programming. In contrast to SLDNF-resolution, the proposed method can select non-ground negative literals and can compute correctly using new equivalent transformation rules.

1. まえがき

我々の長期的な研究目標の1つは，問題を正しく高速に解くための枠組(問題解決の枠組)を確立することである．否定表現を含む問題解決のための計算手続きとしてSLDNFレゾリューションが知られているが，SLDNFレゾリューションによって正しく解ける問題の範囲は狭すぎ，往々にして問題解決の障害となっている[4]．本論文では，SLDNFレゾリューションよりも強力な新しい計算手続きを提案する．

1.1 SLDNFレゾリューションの理論

ボディに否定表現(負リテラル)を許した確定節をプログラム節と呼び，プログラム節の集合を正規プログラムと呼ぶ．正規プログラム P とアトム q が作る問題 (P,q) とは，$comp(P) \models q\theta$ を満たす代入 θ 全体の集合を求める問題である．ただし $comp(P)$ は P の完備化である．この問題を解く手続きとしてSLDNFレゾリューションが提案されている．

正規プログラムのゴール節には，

(1) 正リテラル
(2) 変数を含まない負リテラル
(3) 変数を含む負リテラル

の3種類のアトムが出現する．SLDNFレゾリューションは，これらのリテラルに対する次のような手続きから構成されている．

(1) 正リテラルの計算は，通常どおりのレゾリューション操作で行う．
(2) 変数を含まない負リテラルの計算は「失敗による否定」で行う．つまり負リテラル $\neg A$ の計算をする場合，まず正リテラル A をレゾリューション法で計算し，その結果の真偽を反転する．
(3) 変数を含む負リテラルの計算は行わない．

SLDNFレゾリューションの最大の欠点は，「変数を含む負リテラル」の計算を行なわないことによって，全

† 日本電信電話株式会社 情報流通プラットフォーム研究所
NIPPON TELEGRAPH AND TELEPHONE CORPORATION, NTT Information Sharing Platform Laboratories
†† 北海道大学 情報メディア教育研究総合センター
Center of Information and Multimedia Studies, Hokkaido University
††† 北海道大学大学院 工学研究科 システム情報工学専攻
Division of System and Information Engineering, Hokkaido University

体の計算が中途で停止してしまう恐れがあることである．そのような事態は，ゴールに含まれるすべてのリテラルが「変数を含む負リテラル」である場合に起こる (2.3節)．

1.2 正当な計算手続きの発見方法

本論文の目的は，否定表現を含む 問題解決のための 新しい計算手続きを提案することである．その計算手続きは，次の条件を満たすものとする．

(1) SLDNF レゾリューションの上記の欠点を克服していること．

(2) SLDNF レゾリューションより強力な計算手続きであること．

ただし，計算手続き A が 計算手続き B より 強力 であるとは，A による計算はすべて B でも可能であり，B による計算で A では不可能な計算が存在することである．

この目的を達成するために，我々は，計算手続きの発見方法として **等価変換パラダイム** を用いる．等価変換パラダイムの最も重要な特徴は，問題解決のための正当な手続きを系統的に導く事ができる点にある．それは次のステップで行われる[2)]．

(1) **[意味写像の定式化]** 意味写像 $\mathcal{M}$ とは，表現 d に 集合 S を対応させる写像である．意味写像 $\mathcal{M}$ により，表現 d には 意味 $\mathcal{M}(d)$ が与えられる．

(2) **[ルール集合の作成]** 意味写像 $\mathcal{M}$ のもとでの等価変換ルールを幾つか集めてルール集合 R を作る．r が 意味写像 $\mathcal{M}$ のもとで等価変換ルールであるとは，r が d_1 を d_2 に変えるとき，$\mathcal{M}(d_1)=\mathcal{M}(d_2)$ を満たすことである．

この結果 得られるルール集合 R は，(非決定的) 計算手続き と見なせる．R の決定する 計算手続きは，ある写像 f と意味写像 $\mathcal{M}$ と 表現 d_1 を用いて $S=f(\mathcal{M}(d_1))$ で定義される S を求めるために，R の中のルール $r_1,r_2,\cdots,r_n$ で表現 d_1 を

$$d_1 \to d_2 \to \cdots \to d_n$$

のように 繰り返し等価変換して，d_1 から d_n を得て，$S=f(\mathcal{M}(d_n))$ により S を計算する．この計算手続きは d_nを得るための計算方法がいろいろありうるという意味で非決定的である．

この方式の重要な特徴として次の２つを挙げる．

(1) 正しい等価変換ルールを用いるかぎり，得られる手続き R の正当性は 厳密に保証されている

(2) ルール集合に 等価変換ルールを増やすほど，より 強力な 計算手続きが得られ，多様な計算が達成される．

等価変換パラダイムのもとで **[意味写像の定式化]** と **[ルール集合の作成]** とを行えば，得られる計算手続きは正当であることがいえる．またルールを追加することによって どんどん強力にできる．したがってこれは，正当で強力な計算手続きを発見するための方法として使うことができる．

1.3 本論文の方法の概要

SLDNF レゾリューションにおいて レゾリューション操作の部分を unfold 変換で置き換えれば，SLDNF レゾリューションで達成されるのと同じ計算が等価変換によって行えるのは明らかである．したがって unfold ルール を準備☆すれば，少なくとも SLDNF レゾリューションに匹敵する計算が可能であることがわかる．

本論文では，さらに，負のリテラルを選択して実行するための変換ルールを導入する．これにより，次の２点も達成される．

(1) SLDNF レゾリューションの上記の欠点を克服していること．

(2) SLDNF レゾリューションより強力な計算手続きであること．

1.3.1 意味写像の定式化

従来の宣言的記述を拡張して，確定節のボディに「負制約」を入れることを許す．これにより「制約つき宣言的記述」のクラスが得られる．また，任意の制約つき宣言的記述 P に 意味 $\mathcal{M}(P)$ を定義する．

これにより $\mathcal{M}$ は，制約つき宣言的記述を「表現」とする 意味写像と見ることができる．

1.3.2 ルール集合の作成

本論文では，特に,「負制約」の等価変換のための最も基礎的な理論を与える．それによれば，次のような等価変換が可能となる．

(1) **[繰り込み変換]** 負制約を計算するために 新しい宣言的記述をつくり出す．

(2) その宣言的記述を (任意のルールで) 等価変換する．

(3) **[依存解消変換]** その宣言的記述が 等価変換されて 単位節集合になったときに負制約を解消して 単純な制約にする．

これらの (1) と (3) から２つのルールが得られる．それと エクスパンド変換や 基礎制約の変換ルールなどを加えて，ルール集合を構成する．

☆ 本論文では その代わりに エクスパンド変換と制約の変換を用いている．

1.4 SLDNF レゾリューションの克服

制約つき宣言的記述による定式化と繰り込み変換や依存解消変換を含むルール集合は，論理パラダイムにおける否定の問題を克服している．すなわち，SLDNF レゾリューションと異なり，本方法では変数が入った負制約をも選択することができる．変数が入った負制約が選択されると，繰り込み変換や依存解消変換などにより正しい計算が行われる．その処理は 常に正当であることが保証されている．

2. 等価変換による解法例

2.1 非回文問題の 表現の例

"たけやぶやけた" のように 前から読んでも，後ろから読んでも同じ文字列を回文 (palindrome) と呼ぶ．本章では，「回文でない aZb の形のすべての文字列を求める」問題を等価変換で解くことを考える．後述するように，この問題を 論理プログラミングで解こうとすると問題が生じる (2.3節) ので，本論文の方法の利点を示すことができる．

まずこの問題を文字列領域の負制約を用いて定式化する．

$R = \{ans, np, pal, rev, pal', \cdots\}$

とする．$ans(\alpha)$ は「α は 答である」を，また，$np(\alpha)$ は「α は 回文ではない」を意味するものとする．$pal(\alpha)$ は「α は 回文である」を，また，$rev(\alpha, \beta)$ は「α と β は 互いに逆順の文字列である」を意味するものとする．pal' は 繰り込み変換のために pal に対応して導入される述語である．すなわち，

$\phi(pal(\alpha)) = pal'(\alpha)$

である．

この問題を 制約つき宣言的記述の意味を用いて記述するために，宣言的記述 P とそれが参照する宣言的記述 Q を次のように定義する．

$P = \{C_1\}$

$C_1 = (np(Z) \leftarrow [pal(Z) \notin \mathcal{M}(Q)]\ \}$

$Q = \{C_2, C_3, C_4\}$

$C_2 = (pal(X) \leftarrow rev(X, X))$

$C_3 = (rev(\epsilon, \epsilon) \leftarrow)$

$C_4 = (rev(AX, YA) \leftarrow rev(X, Y))$

文字列 aZb が 回文でないような 文字列 Z をすべて求める問題は，宣言的記述 P' を

$P' = P \cup \{ans(Z) \leftarrow np(aZb)\}$

とし，$\mathcal{G}$ の部分集合 G に対して 集合を与える写像を

$\pi(G) = \{x \mid ans(x) \in G\}.$

とするとき，$\pi(\mathcal{M}(P'))$ を求める問題として定式化できる．

2.2 非回文問題の解決例

前章までの命題や定理を用いて 非回文問題を解く例を示す．非回文問題は，2.1節の宣言的記述 P' を用いて，$\pi(\mathcal{M}(P'))$ によって記述されている．

$P' = \{\ ans(Z) \leftarrow np(aZb).$

$\qquad np(Z) \leftarrow [pal(Z) \notin \mathcal{M}(Q)].\ \}$

P' を等価変換して，ans 節を単位節に変形するために，ans 節とそれに関係する参照記述 Q を選んで 変換を続ける．等価変換の結果，$ans(Z) \leftarrow$ という単位節が得られる．したがって 問題の答は 文字列全体の集合である．つまり Z が どんな文字列でも aZb は 回文にはならない．この答が得られる過程を以下に示す．

p1: $ans(X) \leftarrow np(aZb).$

p2: $ans(X) \leftarrow [pal(aZb) \notin \mathcal{M}(Q)].$

p3: $ans(X) \leftarrow [pal'(aZb) \notin \mathcal{M}(Q \cup Q')].$

$Q' = \{pal'(aZb) \leftarrow pal(aZb)\}$ として，
pal'節を変換する．

q11: $pal'(aZb) \leftarrow pal(aZb).$

q12: $pal'(aZb) \leftarrow rev(aZb, aZb).$

これは 次の 2 つの節に展開される．

q13-1: $pal'(aZb) \leftarrow [aZb = \epsilon],$
$[aZb = \epsilon].$

q13-2: $pal'(aZb) \leftarrow [aZb = AX],$
$[aZb = YA],$
$rev(X, Y).$

q13-1 は 消滅．q13-2 は 次の節になる．

q14: $pal'(aZb) \leftarrow [aZb = Ya],$
$rev(Zb, Y).$

q15: pal'節が消滅

p4: $ans(X) \leftarrow [pal'(aZb) \notin \{\ \}].$

p5: $ans(X) \leftarrow.$

上記の等価変換過程で用いた変換の根拠を，以下に簡単にまとめておく．

p1 から p2 へ	エクスパンド変換
p2 から p3 へ	繰り込み変換（定理 A）
p3 から p4 へ	依存解消変換（定理 B）
p4 から p5 へ	制約の変換
q11 から q12 へ	エクスパンド変換
q12 から q13 へ	エクスパンド変換
q13 から q14 へ	制約の変換

q14 から q15 へ　　制約の変換

エクスパンド変換は,「平坦な節を用いたアンフォールド変換」である☆. 上記の「制約の変換」とは, 依存制約でない制約に関する等価変換を指す. p4 から p5 への変換では, 恒真な制約 (空集合の元ではないことは常に成立する) を除去している. q13-1 は 恒偽な制約 (aZb は ϵ になれない) により 消去される. q13-2 は $[(A,a),(X,Zb)]$ により 節が特殊化されて, q14 が得られる. q14 は 恒偽な制約 (aZb の右端は b だから Ya とは一致しえない) により 消去される.

2.3 論理パラダイムの困難

非回文問題を 論理パラダイムで解くために, まずこの問題を表現する 文字列領域の論理プログラム P を書く.

$$P = \{C_0, C_1, C_2, C_3, C_4\}$$
$$C_0 = (ans(Z) \leftarrow np(aZb))$$
$$C_1 = (np(Z) \leftarrow \neg pal(Z))$$
$$C_2 = (pal(X) \leftarrow rev(X,X))$$
$$C_3 = (rev(\epsilon,\epsilon) \leftarrow)$$
$$C_4 = (rev(AX,YA) \leftarrow rev(X,Y))$$

これに対して 問い合わせ $\leftarrow ans(Z)$ を与えると,

$$\leftarrow np(aZb)$$
$$\leftarrow \neg pal(aZb)$$

まで変形されるが, SLDNF レゾリューション[4] ではこれ以上計算を進めることができない☆☆. なぜなら SLDNF レゾリューションは 変数の入った負のリテラルを選択することを禁止しているからである. 負のリテラルを選択すれば 計算の正当性が保証できない. 一方, 前節で見たように, 本論文の等価変換に基づく方法はこの困難を克服している.

3. むすび

本論文では, 文字列領域において負制約を正しく計算するための基礎理論を与えた.

等価変換パラダイムは, 集合の宣言的記述を等価変換するという方法論であり, 負制約の扱いは その方法論から直接的に帰結する. すなわち, 負制約の概念は, 集合の宣言的記述の基本演算子である補集合演算を利用して定義され, 負制約の計算時には この定義に従って正確に等価変換する.

この方法論では, 等価変換ルールの正当性は 他のルールとは 独立に判定できる. したがって, 等価変換ルールを蓄積することによって, よりよい計算が可能になる[3]. 本論文の定理から, 2 つの等価変換ルールを作ることができる. 定理 A から, 制約対象の情報を参照記述の中に繰り込むルールが得られる. こうして得られた宣言的記述を等価変換して 単位節の集合が得られた場合, その情報を 呼びだし側の簡単な制約として返すルールが, 定理 B から得られる.

この 2 つのルールは, 負制約の計算に対する最も基礎的なルールである. このルールは, Prolog の「失敗による否定」とは 対照的に, 否定されるアトムが変数を含むか否かには 無関係に 正当である.

否定の効率的な計算のためには, 本論文の理論に加えて, 参照記述とその "呼びだし記述" の間の密接な情報交換をする等価変換ルール[7] が しばしば 有用である. そのような ルールを基礎付ける理論は 別の論文で与える.

等価変換による否定の計算 (上記の情報交換のルールを含めて) は すでに等価変換プログラミング言語 ETC[5] で実現されている. ETC を用いて, 自然言語理解システムが実現されており[6], 否定の正当で高速な計算は, システム作成の重要な鍵となっている.

☆ 文献[1] 参照.

☆☆ ここでは 文字列領域の論理プログラムを用いたが, 通常の項領域の論理プログラムでも本質的に同じ困難が起きる.

参考文献

1) 赤間清, 川口 雄一, 宮本衛市: マルチセット領域上の等式制約の等価変換, 人工知能学会誌, Vol.13, No.3, pp.395–403 (1998).
2) 赤間清, 清水伴訓, 宮本衛市: 宣言的プログラムの等価変換による問題解決, 人工知能学会誌, Vol.13, No.6, pp.944–952 (1998)
3) 畑山満美子, 赤間清, 宮本衛市: 等価変換ルールの追加による知識処理システムの改善, 人工知能学会誌, Vol.12, No.6, pp.861–869 (1997)
4) Lloyd,J.W. : Foundations of Logic Programming, 2nd edition, Springer-Verlag, p.212 (1987)
5) 清水 伴訓, 赤間清, 宮本衛市: 等価変換プログラミング言語 ETC, 電子情報通信学会 技術研究報告, SS 96-19, pp.9-16 (1996)
6) 吹田慶子, 赤間清, 宮本衛市: 等価変換に基づく自然言語理解システムの構築, 電子情報通信学会技術研究報告, SS 97-35, pp.23-30 (1997)
7) Yoshida, T., Akama, K. and Miyamoto, E.: Problem Solving by Equivalent Transformation of First Order Logical Constraints, Preprints Work. Gr. for ICS 98-ICS-114-2, pp7–12 (1998).