知 能 と 複 雑 系 113－2
（1998. 7. 23）

区間変数の領域における 等式制約の解消

繁田 良則* 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
{akama,miya}@complex.eng.hokudai.ac.jp shige@sdel.toshiba.co.jp

変数に 区間を付けた対象を 区間変数と呼ぶ．区間変数は 区間制約の伝播を行えるので，知識処理の高速化に役立つ場合がある．区間変数を用いた計算の基礎理論を構築するために，我々は，いろいろなデータ構造を利用する計算を統合的に扱うことのできる等価変換パラダイムを採用する．等価変換パラダイムでは 多数の等価変換ルールで 宣言的プログラムを変形することで 計算を行う．本論文の目的は，宣言的プログラムに含まれる等式制約を解消することによって，宣言的プログラムを等価的に変換するための新しい理論を開発することである．

区間変数 宣言的意味 等式制約 等価変換

Constraint Solving for Equality Constraints on an Interval Variable Domain

Yoshinori Shigeta Kiyoshi Akama Eiichi Miyamoto
Division of System and Information Engineering, Hokkaido University
North 13 West 8 Kita-ku Sapporo. Tel. 011-706-6814
{akama,miya}@complex.eng.hokudai.ac.jp shige@sdel.toshiba.co.jp

An interval variable is a pair of a variable and an interval. Interval variables often contribute to efficient computing in knowledge processing due to the propagation of interval constraints. To develop a foundation of computation using interval variables, we adopt the equivalent transformation (ET) paradigm, which provides a unified theoretical framework to disscuss computation using many data structures. In the ET paradigm, many equivalent transformation rules are used for computation. The purpose of this paper is to develop a new theory for equivalent transformation of declarative programs that include equality constraints.

interval variable, declarative semantics, equality constraint, equivalent transformation

1　まえがき

1.1　区間変数

区間変数とは，$B:[2,5]$ のように，タグ (B) と 区間 ($[2,5]$) の２項組であり，その区間の範囲の実数 B という直観的意味を持つ．区間変数は，文献 [6, 10] の例が示すように，知識処理の高速化に 著しく貢献する場合がある．高速化が達成される主要な原因は，区間変数によって 大まかな制約が伝播されることにより，最終的に不要となる多くの選択肢が 効果的に枝刈りされることである．

本論文では，「等価変換パラダイム (文献 [1])」のもとで，区間変数を用いた計算の基礎的な理論を与える．

1.2　区間変数を含む項

プログラムの基本要素として，３種類の項を導入する．それは，４のような定数，Y のような純変数[1]，それに $B:[2,5]$ のような区間変数である．純変数と区間変数をあわせて 変数と呼ぶ．数学的には，$B:[2,5]$ のような区間変数は，B のようなタグ[2] と $[2,5]$ のような 区間の２項組として扱われる．これらの基本要素と f や g のような関数[3] を用いて 再帰的に $f(4, B:[2,5], g(Y))$ のような複合項をつくる．このようにして，本論文で対象とする 「区間変数を含む項」全体の集合が得られる．本論文では，簡単のために，区間変数を含む項を 単に項と呼ぶことがある．

区間変数を含む項 全体の集合を $\mathcal{A}_T$ で表す．$\mathcal{A}_T$ の元のうち 変数を含まないものを基礎項と呼ぶ．基礎項全体の集合を $\mathcal{G}_T$ で表す．

1.3　区間変数を含む項に対する特殊化

論理プログラミングでは，項は 代入によって変化 (特殊化) させられる [8]．代入の最も基本的な単位は 束縛 (変数と項のペア) であり，たとえば，$f(X, g(Y))$ は 束縛 $X/h(Z,1)$ によって $f(h(Z,1), g(Y))$ に変化する．束縛の繰り返しによって 任意の代入効果を得ることができる．

しかし 代入の概念だけでは，多様なデータ構造を扱うには 不十分である．代入だけでなく「代入以外の変化」をも 導入したほうが自然なデータ構造が たくさん存在する．区間変数を含む項の世界もその１つである．そこで，本論文では，代入の拡張概念である 「特殊化」の概念 (文献 [2, 3]) を用いる．特殊化は，項に対して任意の変更を可能にする．特殊化の概念を使えば，もはや 代入という特別のタイプの変更だけしか利用できないという (不自然な) 理論的制約に縛られることがない．

本論文では，まず 区間変数を含む項を変更する基本的な操作として「基本特殊化」を導入し，基本特殊化の逐次適用によって 特殊化を導入する．これは 束縛の逐次適用によって 得られる代入全体の集合を考えるのと類似しているが，基本特殊化として，束縛という１種類だけでなく，以下の４種類のものを考える．

純変数は，変数代入によって特殊化される．例えば，Z は $(Z,2)$ という基本特殊化によって２へ 特殊化される．区間変数は，タグを変更することによって特殊化される．例えば，$B:[3,8]$ は (B,C) によって $C:[3,8]$ に特殊化される．区間変数は，区間をその部分集合（部分区間）に変更することによって特殊化される．$B:[3,8]$ は $(B,[4,5])$ という基本特殊化によって $B:[4,5]$ に特殊化される．区間変数は 区間に属する定数に特殊化される．$B:[3,7]$ は $(B,5)$ によって ５に特殊化される．

４種類の基本特殊化のうち，後の２種類の基本特殊化は，場合によっては 適用不可能になることに注意したい．たとえば，$(B,[4,5])$ という基本特殊化は $B:[6,8]$ には 適用できない．これは $[4,5]$ が $[6,8]$ の部分区間ではないからである．同様に，$(B,5)$ という基本特殊化は $B:[6,8]$ には 適用できない．これは５が $[6,8]$ の要素ではないからである．

等価変換パラダイムにおける 特殊化の概念は適用不可能性を 部分写像 の概念によって 扱っており，区間変数の場合も それを利用することができる．たとえば，$(B,5)$ という基本特殊化の適用を １つの部分写像で表現し，その部分写像は，$B:[4,6]$ を５に移すが，$B:[6,8]$ では 未定義であるとする．

特殊化は，基本特殊化の列として定義される．たとえば，$[(Z,2),(B,C),(C,[4,5]),(C,5)]$ は 特殊化の例である．空列 [] も 特殊化の１つである．特殊化全体の集合を $\mathcal{S}$ で表す．

特殊化の適用は，特殊化の中の基本特殊化の繰返し適用によって定義される．本論文では，特殊化を θ や σ などのギリシャ文字で表記し，特殊化の作用を後置記法で表す．これは 代入の場合の記法の自然な拡張である．たとえば，項 t が $f(Z, B:[3,8])$ であり，特殊化 θ が $[(Z,2),(B,C),(C,[4,5]),(C,5)]$ であるとき，

$$
\begin{aligned}
t\theta &= f(Z, B:[3,8])[(Z,2),(B,C),(C,[4,5]),(C,5)] \\
&= f(2, B:[3,8])[(B,C),(C,[4,5]),(C,5)] \\
&= f(2, C:[3,8])[(C,[4,5]),(C,5)] \\
&= f(2, C:[4,5])[(C,5)] \\
&= f(2,5)[] \\
&= f(2,5)
\end{aligned}
$$

[1] 純変数とは，いわゆる変数のことであるが，区間変数と区別するためにこう呼ぶ．
[2] 区間変数で使われる変数を特にタグと呼ぶ．
[3] これは 数学上の写像の意味の関数ではなく，単なる記号である．

となる．ここに見られるように，空列 [] による特殊化は 項を変化させない．基本特殊化が 適用不可能な場合が あるので，特殊化についても 適用不可能な場合が 存在する．特殊化が適用可能なのは，それに属する基本特殊化が 順次 すべて適用可能なときである．

1.4 区間変数を含むアトムと その特殊化

述語と (区間変数を含む) 項の並びからなる $p(t_1, t_2, \cdots, t_n)$ の形[4] の表現を 区間変数を含むアトムと呼ぶ．以後 本論文では，しばしば，単にアトムとも呼ぶ．区間変数を含むアトム 全体の集合を $\mathcal{A}$ で表す．$\mathcal{A}$ の元のうち 変数を含まないものを基礎アトムと呼ぶ．基礎アトム全体の集合を $\mathcal{G}$ で表す．

(区間変数を含む) 項 に対する特殊化は 自然に (区間変数を含む) アトムに拡張できる．特殊化の アトムへの作用も 後置記法で表す．変数を含まない (引数が基礎項だけからなる) アトムを 基礎アトムと呼ぶ．

1.5 区間変数を用いた確定節

区間変数を含むアトムを用いて，確定節を作ることができる．たとえば，

$p(X:[2,5]) \leftarrow q(X:[3,8], Y:[2,5])$

は，確定節の例である．通常の理論と同じように，$\leftarrow$ の左を ヘッド，右をボディと呼ぶ．確定節のボディに さらに制約も書きたい場合がある．たとえば，

$p(Z) \leftarrow Z = X:[2,5], q(X:[3,8], Y:[2,5])$

は，等式制約 $Z = X:[2,5]$ をボディの先頭に持つ．

制約を形式的に扱うために，本論文では，制約対象 a と制約領域 G の組 (a, G) で 制約を表す．この制約 (a, G) の直観的意味は，制約対象 a が 制約領域 G の元だけにしか具体化されないことである．たとえば，上記の等式制約 $Z = X:[2,5]$ を正式に表現するには，制約対象として

$a = equal(Z, X:[2,5])$

を用い，制約領域として

$G = \{equal(g,g) \mid g \in \mathcal{G}_T\}$

を使えばよい．

制約 (a, G) は，a が変数を含まないとき，基礎制約と呼ばれる．$a \in G$ を満たす基礎制約 (a, G) を真な制約と呼ぶ．$a \notin G$ を満たす基礎制約 (a, G) を偽な制約と呼ぶ．

特殊化 θ が制約 (a, G) に適用可能なのは，θ が a に適用可能なときであり，その適用結果 $(a, G)\theta$ は制約 $(a\theta, G)$ である．このとき (a, G) はθ によって $(a\theta, G)$ に特殊化されるという．

[4]ただし，通常の Prolog の記法にならって，$n = 0$ のときは 括弧は省略し 述語だけを書く．

ヘッドがアトムであり，ボディがアトムや制約からなる節を制約つき確定節と呼ぶ．本論文では 制約つき確定節をしばしば 単に確定節，あるいは，節と呼ぶことがある．基礎アトムまたは 基礎制約だけからなる制約つき確定節を 基礎節と呼ぶ．

制約つき確定節

$C = (H \leftarrow B_1, B_2, \cdots, B_n)$

と 特殊化 θ に対して，もし θ が $H, B_1, B_2, \cdots, B_n$ のすべてに適用可能であるならば，θ は C に適用可能で，適用結果 $C\theta$ は

$C\theta = (H\theta \leftarrow B_1\theta, B_2\theta, \cdots, B_n\theta)$

であると定義する．$C\theta = C'$ のとき，C は θ で C' に特殊化される，または，C' は C の例であるという．基礎節 C' が C の例であるとき，C' は C の基礎例であるという．

1.6 宣言的プログラムとその意味

制約つき確定節を任意個 含む集合を制約つき宣言的プログラムと呼ぶ．制約つき宣言的プログラムを，単に宣言的プログラム，あるいは，プログラムと呼ぶことがある．

次に 宣言的プログラムの「意味」を次の 3 ステップで定義する．

1. ステップ 1 宣言的プログラム P から 基礎節の集合 P' を作る．
2. ステップ 2 基礎節の集合 P' から 写像 $T_{P'}$ を作る．写像 $T_{P'}$ は 基礎アトムの任意の集合を 基礎アトムの集合に変換する．
3. ステップ 3 写像 $T_{P'}$ から 基礎アトムの集合 を決定する．

最後に決定された基礎アトムの集合が 宣言的プログラム P の 意味であり，$\mathcal{M}(P)$ と書かれる．

各ステップの内容を定義する．

1. ステップ 1 宣言的プログラム P に含まれる確定節に特殊化を作用させることによって 基礎節を得る．ただし，特殊化は確定節の構成要素であるすべてのアトムと制約に適用可能なものに限る．また，確定節に出現する制約は すべて 真なる制約に特殊化されなければならない．そうして得られる基礎節から 制約 (すべて真) を取り除いて，基礎アトムだけからなる確定節を得る．この操作によって P から得られる確定節全体の集合を P' とする．

2. ステップ 2 基礎アトムの任意の集合 I が与えられたとする．P' に含まれる すべての基礎節のうち，ボディに出現する基礎アトムが すべて I に属するものを考える．そのような基礎節のヘッドをすべて集めると，1つの集合が定まる．基礎アトムの任意の集合 I に対して，こうして決まる集合を対応させる写像が $T_{P'}$ である．

3. ステップ 3 空集合 $\emptyset$ に $T_{P'}$ を適用して $T_{P'}(\emptyset)$ を作る．次に それに $T_{P'}$ を適用して $T_{P'}(T_{P'}(\emptyset))$ を作る．さらにそれに $T_{P'}$ を適用して $T_{P'}(T_{P'}(T_{P'}(\emptyset)))$ を作る．これを無限に繰り返すことによって，得られるすべての集合
$$[T_{P'}]^n(\emptyset) \qquad (n = 1, 2, 3, \cdots)$$
の和集合として 基礎アトムの集合を決定する．

ここでは P の意味を定義するのに 3ステップで説明したが，ステップ1とステップ2を統合した定義を採用する場合も多い．本論文の 2.2 節では そのように 2ステップの定義を与えている．その場合 写像 $T_{P'}$ のかわりに「P から 決まる写像」という意味で T_P が 用いられる．

1.7 等価変換に基づく問題の表現と解法

等価変換パラダイムにおける 問題解決の流れは次のようなものである．

1. ある未知の集合を求める問題を想定し，
2. 未知の集合を記述する問題文を ある宣言的記述で表現し，
3. その宣言的記述を，等価的に変換して 簡単な宣言的記述を得，
4. その簡単な宣言的記述から，未知の集合を見出す．

これを背景として，等価変換パラダイムにおける計算は，「宣言的記述を，その意味を保存して，等価的に変換すること」と捉えることができる．

より具体的には，宣言的記述として 宣言的プログラムを用い，1.6 節で定義された意味を保存して等価的に変換する．

1.8 論理プログラミングとの違い

宣言的プログラムの等価変換に基づく理論体系は，推論に基づく論理プログラミングと基本的な点で異なる．

論理プログラミングでは 問題を 論理式で表現し，推論で計算する．この枠組では，モデルの概念が重要である．論理式の背後には モデルが存在し，推論は モデルを基礎として正当化される．

しかし等価変換パラダイムでは，このような考えは採用しない．1.7 節で述べたように，宣言的プログラムはある集合を表現するためのものと考える．その集合が モデルであるべき必然性はない．基本的には 何らかの集合を記述できれば よい．等価変換パラダイムにおける計算は，推論ではなく，等価変換である．推論の概念を使わない以上，モデルという概念を扱う必要はない．

推論ではなく 等価変換を考えるのは，計算の概念を拡大するためである．論理プログラミングの推論 (SLD レゾリューション) は，等価変換による計算の特殊な場合と見ることができる [5]．このように計算の概念を拡大することにより，高速な計算が達成しやすくなる．実際，制約充足問題 [11] や自然言語理解システム [6] などで，著しい高速化が 達成されている．

論理プログラムのプログラム変換 [9] は，宣言的意味を保存する点で等価変換パラダイムの計算と類似している．しかし，論理プログラムのプログラム変換が，実際の問題解決のための計算 (＝実計算) ではなく，プログラムを高速化するための計算であるのに対して，等価変換パラダイムでの等価変換は 実計算である．両者は 異なる目標を持つので，データ構造を項に限定した場合ですら，いろいろな違いが出て来る．たとえば，プログラム変換では フォールド変換が基本的であるが，等価変換ではそのようなモジュラー性のないルールは使わない．

1.9 等価変換ルールのモジュラー性

等価変換パラダイムでは，多数のルールを適切に用いて効率的な等価変換を行うことを目指す．これは 論理パラダイムが アンフォールド変換というただ1つの等価変換ルールを使う (と見なせる [5]) のと対照的である．

等価変換ルールには モジュラー性がある．すなわち，等価変換ルールの正当性は，他の等価変換ルールとは無関係に判定でき，正当な等価変換ルールを使う限り，いろいろな等価変換ルールが実行時に どのような順序で適用されようと，その結果の正当性が保証できることである．

これは 理論開発においても 実際のシステム構築においても 極めて重要である．それは いろいろな等価変換ルールを個別に正当化し，蓄積して行けるからである．

1.10 等式制約の変換

区間領域で必要な等価変換ルールは，平坦化変換，エクスパンド変換，等式制約の変換，包含制約の変換，否定を扱う変換，一階論理制約の変換などを行うルールで

ある[5]．このうち最も基本的なのは，平坦化変換，エクスパンド変換，等式制約の変換のためのルールである．

確定節

$p(t_1, t_2, \cdots, t_n) \leftarrow B$

を，この節に出現していない 純変数 $X_1, X_2, \cdots, X_n$ を用いて，

$$p(X_1, X_2, \cdots, X_n) \\ \leftarrow X_1 = t_1, X_2 = t_2, \cdots, X_n = t_n, B$$

と変形するのが平坦化変換である．平坦化した節を用いて,「アンフォールド変換」(ヘッドにマッチするアトムをボディの形に展開) するのが エクスパンド変換である．その結果，等式制約が 節のボディに得られることになる．通常のアンフォールド変換が行う変換は，平坦化変換，エクスパンド変換，等式制約の変換に分けて行うことができる．

等価変換パラダイムでは，アンフォールド変換単独ではなく，小規模な変換に 分けて問題解決を行う方法を採用している．これにより，理論が単純化して証明が簡単化できるばかりでなく，必要な変換だけを組み合わせて使う方法や，特定の変換を遅延することによって 計算の高速化を図ることが可能になる．また，ルール生成 [7] などのためにメタルールを作る場合，小規模な変換に分けた方が 正当なメタルールをより多く得ることができる．また，包含制約の変換，否定を扱う変換，一階論理制約に関する変換などを追加することによって，等価変換という 唯一の原理のもとで 均質で統合的な計算の枠組を構築できる．

1.11 本論文の内容

本論文の目的は，区間変数を含む項に関する等式制約を解消するための等価変換を基礎付ける理論を与えることである．等式制約を解消するための等価変換ルールは，区間変数を含む項を用いた計算の最も基礎的な部品の1つである．

本論文の構成を示す．2章では，等価変換を正当化するための最も基礎的な理論を復習する．3章では，区間変数の領域での基礎的な命題を与える．4章では，区間変数の領域における等式制約を正しく等価変換するための理論を与える．

紙面の都合上，証明はすべて割愛する．

2 等価変換の正当性のための基礎

本章では，すべてのデータ構造領域に共通に使われる定義 や 命題を復習する．

[5]他のデータ領域におけるこれらの変換に関する記述が [3], [2], [4], [12] などにある．

2.1 記号

区間変数の領域のみならず，多くのデータ構造の領域は，アトム全体の集合 $\mathcal{A}$，基礎アトム全体の集合 $\mathcal{G}$，特殊化全体の集合 $\mathcal{S}$，特殊化の作用を定義する写像 μ の4項組で規定できる．その4項組を特殊化システム[6] と呼ぶ．本章では，特殊化システム $\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ を 任意に1つ仮定して 議論する．

a を $\mathcal{A}$ の元とするとき，a の基礎例全体の集合を $rep(a)$ で表す．制約つき確定節 C のヘッドを $head(C)$，ボディ中のアトムの集合を $obj(C)$，ボディ中の制約の集合を $const(C)$，また，$obj(C)$ と $const(C)$ の和集合を $body(C)$ で表す．$Tconst(\Gamma)$ はΓ 上の真な制約全体の集合である．Γ 上の基礎節全体の集合を $Gclause(\Gamma)$ で表す．

2.2 制約つき宣言的プログラムの宣言的意味

制約つき宣言的プログラムの宣言的意味を定義する．

まずΓ 上のプログラム P に対して，写像 $T_P : 2^{\mathcal{G}} \to 2^{\mathcal{G}}$ を定義する．

定義 1 【写像 T_P】

$\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ とする．写像 $T_P : 2^{\mathcal{G}} \to 2^{\mathcal{G}}$を，任意の $I \subset \mathcal{G}$ に対して

$$T_P(I) \stackrel{\text{def}}{=} \{head(C\theta) \mid C \in P, \\ \theta \in \mathcal{S}, \\ C\theta \in Gclause(\Gamma), \\ const(C\theta) \subset Tconst(\Gamma), \\ obj(C\theta) \subset I \ \}$$

で定義する．□

上の定義中の条件 $const(C\theta) \subset Tconst(\Gamma)$ は，P の節 C から得られる基礎例 $C\theta$ のうち，$C\theta$ のボディ中の制約がすべて真であるような基礎例だけが宣言的意味を求める際に考慮されることを要請している．

次に写像 T_P を使って，プログラム P の宣言的意味を定義する．

定義 2 【プログラムの宣言的意味】

P を Γ 上のプログラムとする．プログラム P の宣言的意味 $\mathcal{M}(P)$ を，

$$\mathcal{M}(P) \stackrel{\text{def}}{=} \bigcup_{n=1}^{\infty} [T_P]^n(\emptyset)$$

で定義する．□

[6]本論文では，紙面の都合上，その一般的な定義の記述を割愛する．詳しくは 文献 [2, 3] 参照．

2.3 基礎的命題

命題 1 P, P_1, P_2 を制約つき宣言的プログラムとする．任意の $x \subset \mathcal{G}$ に対して $T_{P_1}(x) = T_{P_2}(x)$ ならば $\mathcal{M}(P \cup P_1) = \mathcal{M}(P \cup P_2)$ が成り立つ． □

2.4 恒偽制約による節の消去

$rep(A) \cap G = \emptyset$ を満たす制約 (A, G) を 恒偽制約と呼ぶ．

命題 2 P を制約つき宣言的プログラム，C を 制約つき確定節，(A, G) を C 中の制約であるとする．もし $rep(A) \cap G = \emptyset$ ならば，

$$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P)$$

が成り立つ． □

2.5 節の置き換え

節 C に対して，$\mathcal{M}(C)$ を

$$\begin{aligned}\mathcal{M}(C) = \{\ & (head(C\theta), obj(C\theta)) \\ & \mid \theta \in \mathcal{S}, \\ & \quad const(C\theta) \subset Tconst(\Gamma), \\ & \quad C\theta \in Gclause(\Gamma)\ \}\end{aligned}$$

で定義する．

命題 3 P を 制約つき宣言的プログラム，C と C' を 制約つき確定節とする．もし $\mathcal{M}(C) = \mathcal{M}(C')$ ならば，

$$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\})$$

が成り立つ． □

2.6 節の特殊化

命題 4 P を制約つき宣言的プログラム，C を 制約つき確定節，(A, G) を C 中の制約であるとする．もし 特殊化 θ が，

「$C\sigma \in Gclause(\Gamma)$ かつ $A\sigma \in G$ を満たす任意の $\sigma \in \mathcal{S}$ に対して，$C\sigma = C\theta\rho$ を満たす $\rho \in \mathcal{S}$ が存在する」

を満たすならば，

$$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C\theta\})$$

が成り立つ． □

3 区間変数領域の基礎的命題

3.1 記号

定数集合 K は 実数の空でない集合とシンボルの集合の和集合とする．関数集合 F は K と互いに素な集合，変数集合 V は $K \cup F$ と互いに素な 可算無限集合，タグ集合 W は $K \cup F \cup V$ と互いに素な 可算無限集合とする．述語集合 R は 任意の集合とする．F の各要素（関数）にはアリティと呼ばれる非負整数が対応していると仮定する．K, F, V, W, R の組を アルファベット と呼ぶ．このアルファベットから，作られるアトム (区間変数を含む項と述語からなる) のなす特殊化システムを $\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ で表す．前章までと異なり，本章以後は，$\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{S}, \mu \rangle$ は 区間変数領域の特殊化システムに限定される．

制約つき確定節 C に含まれる純変数 全体の集合を $pvar(C)$，区間変数 全体の集合を $ivar(C)$ と書く．

3.2 正則な確定節

$u : [2, 6]$ と $u : [5, 7]$ のように，2つの区間変数が，同一のタグを持ち，しかも異なる区間を持つとき，それらの区間変数は互いに「食い違っている」という．節が正則であるとは，互いに食い違っている区間変数を持たないことである．プログラムは，正則な節だけからなるとき，正則であるといわれる．任意のプログラム に対して，それと等価な 正則プログラムが存在すること[7]が容易に証明できる．

3.3 基本的な命題

命題 5 制約つき確定節 C が正則ならば，$C\rho \in Gclause(\Gamma)$ を満たす特殊化 $\rho \in \mathcal{S}$ が存在する． □

命題 6 C_1 ，C_2 が制約つき確定節で，

$$head(C_1) = head(C_2)$$
$$body(C_1) \subset body(C_2)$$

とする．もし C_2 が正則ならば，$C_1\sigma \in Gclause(\Gamma)$ を満たす任意の特殊化 σ に対して，$C_1\sigma = C_1\Sigma$, $C_2\Sigma \in Gclause(\Gamma)$ を満たす特殊化 Σ が存在する． □

3.4 恒真制約の除去

$rep(A) \subset G$ を満たす制約 (A, G) を 恒真制約と呼ぶ．

命題 7 P を制約つき宣言的プログラム，

$$C = (H \leftarrow B_1, B_2, \cdots, B_{i-1}, B_i, B_{i+1}, \cdots, B_n)$$

を 制約つき確定節とする．B_i は C 中の制約 (A, G) であるとする．また，C から B_i を取り除いた 制約つき確定節を

$$C' = (H \leftarrow B_1, B_2, \cdots, B_{i-1}, B_{i+1}, \cdots, B_n)$$

とする．もし C が正則で，$rep(A) \subset G$ ならば，

$$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\})$$

が成り立つ． □

[7] 文献 [10] 参照

3.5 特殊化による制約の除去

命題 8 P を制約つき宣言的プログラム，

$C = (H \leftarrow B_1, B_2, \cdots, B_i, \cdots, B_n)$

を 制約つき確定節とする．C 中の制約 B_i を (A, G) とする．また，C から B_i を取り除いた 制約つき確定節を

$C' = (H \leftarrow B_1, B_2, \cdots, B_{i-1}, B_{i+1}, \cdots, B_n)$

とする．もし 特殊化 θ が，

(1) $C\sigma \in Gclause(\Gamma)$ かつ $A\sigma \in G$ を満たす任意の特殊化 $\sigma \in \mathcal{S}$ に対して，$C\sigma = C\theta\rho$ を満たす $\rho \in \mathcal{S}$ が存在する

(2) $C\theta$ は正則である．

(3) $rep(A\theta) \subset G$

を満たすならば，

$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\theta\})$

が成り立つ． □

4 等式制約の等価変換

4.1 区間変数の領域における等式制約

直観的には，2つの項を等号で結んだ式が 等式制約である．厳密には，等号関係を記述する集合 $Equal$ を

$Equal = \{equal(g, g) \mid g \in \mathcal{G}_T\}$

とするとき，任意の $\alpha, \beta \in \mathcal{A}_T$ に対する

$(equal(\alpha, \beta), Equal)$

の形の制約を equal 制約 または等式制約と呼ぶ．簡単のため，$(equal(\alpha, \beta), Equal)$ を $[\alpha = \beta]$ と表記することがある．$[\alpha = \alpha]$ の形の等式制約を恒等制約と呼ぶ．

4.2 命題の前提条件

本章の以降の命題では，次の前提（前提 I と呼ぶ）を共通に用いる．

> [前提 I] P を 制約つき宣言的プログラム，C を 正則な制約つき確定節
>
> $H \leftarrow B_1, \cdots, B_{i-1}, B_i, B_{i+1}, \cdots, B_n$
>
> C 中の B_iを 等式制約 $[\alpha = \beta]$ であるとする． □

等式制約 $[\alpha = \beta]$ の α と β の役割は対称的であり，以下の命題は α と β を交換しても成り立つ．しかし簡単のために，α と β を交換した場合の記述は省略する．

4.3 恒等制約の除去

等式制約 $[\alpha = \beta]$ は，α と β が 同一の項であるとき，恒等制約と呼ばれる．

命題 9 前提 I のもとで，B_i が 恒等制約 $[\alpha = \alpha]$ であるとする．また，C から B_i を取り除いた 制約つき確定節を

$C' = (H \leftarrow B_1, B_2, \cdots, B_{i-1}, B_{i+1}, \cdots, B_n)$

とする．このとき，

$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\})$

が成り立つ． □

4.4 純変数と任意の項の等式制約の処理

命題 10 前提 I のもとで，α を 純変数 X とする．もし X が 項 β に出現しないならば，C から B_i を取り除いた 制約つき確定節

$C' = (H \leftarrow B_1, B_2, \cdots, B_{i-1}, B_{i+1}, \cdots, B_n)$

と，$\theta = [(X, \beta)]$ に対して，

$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\theta\})$

が成り立つ． □

4.5 複合項同士の場合

命題 11 前提 I を仮定する．f を アリティが n の関数，$t_1, t_2, \cdots, t_n$ と $s_1, s_2, \cdots, s_n$ を 項，C 中の B_i を等式制約

$[f(t_1, t_2, \cdots, t_n) = f(s_1, s_2, \cdots, s_n)]$

であるとする．このとき，

$$\begin{aligned} C' = (H \leftarrow\ & B_1, B_2, \cdots, B_{i-1}, \\ & [t_1 = s_1], \cdots, [t_n = s_n], \\ & B_{i+1}, \cdots, B_n) \end{aligned}$$

で与えられる制約つき確定節 C' に対して，

$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\})$

が成り立つ． □

4.6 区間変数同士の等式制約

命題 12 前提 I のもとで，α は 区間変数 (w_x, d_x)，β は 区間変数 (w_y, d_y) で，$w_x \neq w_y$，$d = d_x \cap d_y$ とする．$d \neq \emptyset$ ならば，C から B_i を取り除いた 制約つき確定節

$C' = (H \leftarrow B_1, B_2, \cdots, B_{i-1}, B_{i+1}, \cdots, B_n)$

と，

$\theta = [(w_x, d), (w_y, d), (w_x, w_y)]$

に対して

$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\theta\})$

が成り立つ．また，$d = \emptyset$ ならば，

$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P)$
が成り立つ. □

4.7 区間変数と定数の等式制約

命題 13 前提 I のもとで, α は 区間変数 (w, d), β は定数 k とする. $k \in d$ ならば,
C から B_i を取り除いた 制約つき確定節
$C' = (H \leftarrow B_1, B_2, \cdots, B_{i-1}, B_{i+1}, \cdots, B_n)$
と,
$\theta = [(w, k)]$
に対して
$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P \cup \{C'\theta\})$
が成り立つ. また, $k \notin d$ ならば,
$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P)$
が成り立つ. □

4.8 恒偽制約による節の除去

命題 14 前提 I のもとで C 中の等式制約 $[\alpha = \beta]$ が次のいずれかを満たすとする.

(1) α と β は 相異なる定数
(2) α は純変数で, β は α を含む複合項
(3) α は定数で, β は 複合項
(4) α と β は 相異なる関数を持つ複合項
(5) α は区間変数で, β は 複合項

このとき,
$\mathcal{M}(P \cup \{C\}) = \mathcal{M}(P)$
が成り立つ. □

5 むすび

区間変数領域で必要となる主な等価変換ルールは, 平坦化変換, エクスパンド変換, 等式制約の変換, 包含制約の変換, 否定を扱う変換, 一階論理制約に関する変換などである. 本論文では, 等式制約の変換を正当化するために, 恒等制約, 純変数と項の等式, 複合項同士の等式, 区間変数同士の等式, 区間変数と定数の等式, 恒偽制約の場合に分けて命題を証明した. これは, 平坦化変換, エクスパンド変換, 包含制約の変換, 否定を扱う変換, 一階論理制約に関する変換などと組み合わせて, 区間変数領域での計算の理論を構成することができる.

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, 川口 雄一, 宮本衛市 : 項領域における包含制約の等価変換, 人工知能学会誌, Vol.13, No.2, pp.112–120 (1998).

[3] 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).

[4] 赤間清, 岡田浩一, 宮本衛市 : 文字列領域における負制約の等価変換の基礎, 電子情報通信学会技術研究報告 SS97-81, pp33–40 (1998).

[5] 赤間清, 川口 雄一, 宮本衛市 : 論理的問題の等価変換による解法 (2), SLD 導出の限界, 人工知能学会誌, (1998 掲載予定).

[6] 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).

[7] 小池 英勝, 赤間清, 宮本衛市 : メタプログラムを用いたルールの生成法, 情報処理北海道シンポジウム '98, Info-Hokkaido 98, 38 札幌, pp.78-81 (1998).

[8] Lloyd, J.W.: *Foundations of Logic Programming*, Second edition, Springer-Verlag (1987).

[9] Pettorossi, A. and Proietti, M. : Transformation of Logic Programs: Foundations and Techniques, *The Journal of Logic Programming*, Vol.19/20, pp.261–320 (1994).

[10] 繁田 良則, 赤間清, 宮本衛市 : 区間変数を含む項の空間のための特殊化システムと正則な確定節, 電子情報通信学会技術研究報告 SS98-8, pp17–24 (1998).

[11] 吹田 慶子, 赤間清, 宮本衛市 : 等価変換による制約充足問題の解法, 電子情報通信学会技術研究報告 SS96-18, pp.1-8 (1996).

[12] 吉田 忠行, 赤間清, 宮本衛市 : 一階述語制約の導入による宣言的記述の拡大, 情報処理北海道シンポジウム '98, Info-Hokkaido 98, 42 札幌, pp.92-95 (1998).