TR-0787

プール制約評価アルゴリズム：漸増型変数除去法

毛受　哲

July, 1992

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~5
Telex ICOT J32964

Institute for New Generation Computer Technology

ブール制約評価アルゴリズム：漸増型変数除去法

毛受 哲
(財) 新世代コンピュータ技術開発機構
〒108 東京都港区三田 1-4-28

概要

いくつかのブール代数式が与えられたとき，変数の値や変数間の関係を求めるアルゴリズムはいろいろ知られている．しかし，ブール制約評価アルゴリズムとして用いる場合に要請される性質，すなわち正規形があり，インクリメンタルな実行ができ，媒介変数を使わないといった性質を持つ方法はあまり知られていない．本論文では上の性質を持つ新たなアルゴリズムを提案する．このアルゴリズムは Boole の変数除去に基づいている．

1 はじめに

近年，制約論理プログラミングが盛んに研究されているが，実数や複素数領域ばかりでなくブール代数領域に対しても，制約処理の方式がいろいろ提案されている．例えば CHIP[4] では Boole の変数除去に基づいたブーリアン・ユニフィケーション [2, 6] が，Prolog III[3] では SL 融合 (SL-resolution) が，CAL[8] ではブフバーガ・アルゴリズムを修正したブーリアン・ブフバーガ・アルゴリズム [9, 10] が使われている．また，ブール式の標準形やブール式の変換に関する研究は古くから行われている [1, 7] .

ブール等式を解くということは，与えられたすべてのブール等式を満たすような各変数の値の組 (解) を求めることで，通常はそのような解を 1 つ，またはすべて求めることを意味する．さらに全解を求める場合は，すべての具体的な解を列挙する場合と，すべての解を表す式の集合を求める場合とに分けられる．本論文では後者を対象にする．

全解を表す式の集合を求める場合，与えられたブール等式の集合自身もその 1 つであるので，求める集合には何らかの基準が必要である．最低限要請したいのは，変数の値がある定数しか取り得ない場合にその値が求まることである．さらに，同値なブール等式の集合に対して同じ等式の集合が得られることが望ましい．ここでは，そのような集合を正規形と呼ぶことにする．ブーリアン・ブフバーガ・アルゴリズムはブーリアン・グレブナ・ベースという正規形を求めることができる．

本論文では，ブール等式を解く新しいアルゴリズム「漸増型変数除去法」を提案する．このアルゴリズムは Boole の変数除去に基づいているが，ブーリアン・ユニフィケーションと異なり媒介変数を用いないため得られる解が見易い．また，インクリメンタルにブール等式を処理しても正規形を得ることができるといった性質を持っている．これらの性質は，制約を解くアルゴリズムとして使う場合に望ましいものと思われる．また，必要だが興味のない変数が与えられたブール等式に含まれているとき，そのような変数は早々に除去することができるので，興味のある変数のみからなる正規形を高速に抽出することができるといった利点も持っている．

次節では，まずブール制約を定義し，Boole の変数除去を紹介する．第 3 節では漸増型変数除去法を説明し，停止性と健全性を示した後，求まる式の集合が正規形の性質を満たすことを証明する．第 4 節では 2 つの問題に適用した例を紹介し，第 5 節でまとめを行う．本論文で必要とするブール代数に関する知識は基礎的なものだけだが，ブール代数に関しては [5] などが，ブール等式を解くことに関しては [1, 7] が詳しい．

2 ブール制約

ブール代数 $(\mathbf{B}, \vee, \wedge, \neg, 0, 1)$ があるとする．表現のし易さのため，演算子として $\leftarrow$，$\rightarrow$，$\leftrightarrow$，$+$ (対称差，symmetric difference) も許すことにする．このとき，$\mathbf{B}$ の要素である定数と $\mathbf{B}$ の要素に対する変数，および演算子とから作られる式をブール式と呼ぶ．この 2 つのブール式を等号 $=$ で結んだ式をブール等式と呼び，ブール等式の集まりをブール制約と呼ぶ．

等号 $=$ は両辺のブール式が同値であることを表し，ブール式の形も等しいことを表すのに $\equiv$ を使うことにする．また，変数を表すのに英小文字，ブール式を表すのに英大文字を使うことにする．変数には全順序 $\prec$ が与えられているものとし，変数 x より小さい変数を $\prec_x$- 変数と呼ぶことにする．また，興味のある変数とない変数を区別することがあるが，興味のない変数の順序を興味のある変数より大きくすれば同じ効果を得ることができるので，ここでは特に区別しないことにする．ただし，インプリメントするとき区別できるようにすることは有用である．興味のない変数に対する式を出力しないようにすれば，出力が見易くなるからである．

ここでは定数が 0 と 1 のみの 2 値ブール代数に限定して議論を進めるが，式の表現や演算を拡張して一般ブール代数を扱えるようにすることは容易である．

任意のブール式は，次のような変形操作と分配法則によりブール環 $\langle \mathbf{B}, +, \times, 0, 1 \rangle$ 上の式に変形することができるので，以後すべてのブール式はブール環上の式の形をしているものとする．

$$\begin{aligned} X \wedge Y &\Rightarrow X \times Y \\ X \vee Y &\Rightarrow (X \times Y) + X + Y \\ \neg X &\Rightarrow X + 1 \\ X \leftrightarrow Y &\Rightarrow X + Y + 1 \\ X \rightarrow Y &\Rightarrow (X \times Y) + X + 1 \end{aligned}$$

また，ブール環の性質 $X \times X = X$ かつ $X + X = 0$ は重要である．特に後者は等式 $X = Y$ と $X + Y = 0$ とが同値であることを意味するので，入力したブール等式は右辺が 0 で左辺はブール環上の式であるような等式に変形することができる．そこで等式に関してもこのようなもののみを扱うことにする．本論文でも，見易さのため慣例により “$\times$” を省略する．

変数を含むブール式は，順序に関して最大の変数 x に対し交換法則，結合法則，分配法則を使って $Ax + B$ の形に直すことができる．ここで A，B は x を含まず，演算子 $\times$，$+$ のみからなるブール式である．$Ax + B$ の A，B に対しても変数があれば同様に書き換えることができるので，このような書き換えを繰り返せばブール式の表現は一意になる．そこで，この書き換えを可能な限り繰り返す操作を norm で表し，$\mathrm{norm}(A)$ と書いた場合，ブール式 A に上の操作を適用し得られた式を意味することにする．そして，この操作によって得られる形をブール式の標準形と呼ぶことにし，標準形をしていることを特に表す場合は $Ax \oplus B$ と書くことにする．また，このとき A を x の係数と呼ぶことにする．標準形は次のように再帰的に定義することもできる．

定義 2.1　標準式は次のように構成されるものと定義する．

1. 定数 0 と 1 は標準式である．
2. $A\ (\neq 0)$ と B が $\prec_x$- 変数または定数のみから成る標準多項式であるとき，$Ax + B$ もまた標準多項式である．

例 2.2　変数 a，b，c が $c \prec b \prec a$ のとき，

$$\mathrm{norm}((a \wedge b) + ((b \vee c) - c)) \equiv ba \oplus (c \oplus 1)b$$

である．厳密に言えば $\mathrm{norm}(b)$ は $1b \oplus 0$ であるが，このように 0 の項や係数 1 は省略することにする．

次に，大小関係 $\leq$ をブール式の間にも通常のように定義する．

定義 2.3　任意のブール式 A，B に対し，

$$AB = A \Longleftrightarrow A \leq B$$

である．

この大小関係 $\leq$ は，明らかに半順序関係である．

等式 $Ax \oplus B = 0$ があるとき，$Bx = (Ax)x = Ax = B$ より $B \leq x$ である．また，$(A+B+1)x = Ax + Bx + x = x$ より $x \leq A+B+1$ であるので，$B \leq A+B+1$ すなわち $(A+1)B = 0$ でなければならない．このことを最初に示したのは Boole である．

性質 **2.4**　**(Boole)** $F(x) = 0$ のとき，x の値に関係なく $F(1) \times F(0) = 0$ である．

つまり，$F(x) = Ax \oplus B = 0$ とおけば $F(1) \times F(0) = (A+B)B = AB + B = (A+1)B = 0$ を意味する．$(A+1)B = 0$ は変数 x を含まないので，変数の数は少なくとも 1 つ少なくなっている．さらに $(A+1)B = 0$ に同様の操作を続けると，少しずつ変数の少ない等式を得ることができる．

逆に $(A+1)B = 0$ のとき x の値を $B \leq x \leq A+B+1$ のように取れば $Ax \oplus B = 0$ の解になる．このことをブーリアン・ユニフィケーションでは任意の値を取る媒介変数 u を使って $x = (A+1)u + B$ のように表現し，変数 x と右辺とのユニフィケーションを行う．これによって他の式の x に右辺を代入する効果を得ることができる．

例 **2.5**　ブール式 $xyz + yz + xy + y = 1$ $(x \prec y \prec z)$ を考える．標準形は

$$((x \oplus 1)y)z \oplus ((x \oplus 1)y \oplus 1) = 0$$

である．z を除去すると

$$((x \oplus 1)y + 1)((x \oplus 1)y \oplus 1) = (x \oplus 1)y \oplus 1 = 0$$

となり，さらに変数 y を除去すると

$$((x \oplus 1) + 1) \times 1 = x = 0$$

となるので $x = 0$ である．x に 0 を代入すると $y = 1$ が得られ，さらに y に 1 を代入すると $z = 0$ が得られる．従って与えられたブール式は $x = 0$，$y = 1$，$z = 0$ と同値である．

3　漸増型変数除去法

インクリメンタルに実行することを想定しているので，通常は制約となるブール等式が入力されたときには既に処理した制約等式の集合が存在する．そこで，このような集合を解集合と呼び，新たに入力された等式を入力式と呼ぶ．アルゴリズムを示す前に，まず基本となる考えを示す．

基本的には Boole の変数除去の拡張を用いる．前節で示したように $Ax \oplus B = 0$ は $B \leq x \leq A+B+1$ と同値であり，その式はそのままで x に関する制約を表す式と考えることができる．そこで媒介変数を導入して式を変形することはせず，そのままの形で持つことを考える．ここでは，解集合におけるこのような x に対する式を $\mathrm{constr}(x)$ と記し，すべての $\prec_x$- 変数に対する式の集合を $\mathrm{constr}(\prec_x)$ と記すことにする．

制約式をそのままの形で持つ場合には 2 つの問題があり，それはブーリアン・ユニフィケーションにおける代入と同じ効果をいかに得るかに該当する．

1 つ目は $B \leq x \leq A+B+1$ を意味する $\mathrm{constr}(x)$ を他の式の中の x にどう反映させるかである．ブーリアン・ユニフィケーションの場合は制約式の代入によって x の制約は自然に反映される．ただし代入しただけでは式が複雑なままなので，ブーリアン・ユニフィケーションの場合でも式の変形をしなければならないのは同じである．この問題に対しては，$Ax \oplus B = 0$ といった形の式を使って x の係数を書き換えるリダクションを導入することにより解決することができる．また，これにより制約の反映のみならず正規形を得ることができるようにもなる．このリダクションについては，アルゴリズムの記述の後に述べることにする．

2 つ目は入力式 $Cx \oplus D = 0$ をどう解集合と融合するかである．ブーリアン・ユニフィケーションの場合は，解集合に $x = (A+1)u + B$ があるなら代入によって入力式は $C((A+1)u+B) + D = 0$ のようになっている．そこでその式に対し普通に Boole の変数除去を行い，媒介変数 u を媒介変数表現して $x = (A+1)u + B$ に代入すれば融合が自然に行われる．これに対し我々のアルゴリズムでは同じ変数に対する制約を融合する操作が必要である．解集合に $Ax \oplus B = 0$ があり $Cx \oplus D = 0$ が入力式のとき，後で証明するように両式から得られる x に対する制約は $\mathrm{norm}(AC + A + C)x \oplus \mathrm{norm}(BD + B + D) = 0$ であるので，その式を $Ax \oplus B = 0$ の代わりに $\mathrm{constr}(x)$ とする．これだけでは融合の前後が同値にならないので，同値にするため $ACD + BC + CD + D = 0$ を新しい入力式とみなして融合操作を繰り返す．

リダクション抜きのアルゴリズムは次のようになる．

ステップ 1 入力式を標準形に直し，$Cx \oplus D = 0$ とする．

ステップ 2 解集合に $\mathrm{constr}(x)$ があれば，その式を $Ax \oplus B = 0$ とする．
なければ $A = 0$，$B = 0$ とする ($0x \oplus 0 = 0$ は常に成り立つ)．

ステップ 3 $\mathrm{norm}(AC + A + C)x \oplus \mathrm{norm}(BD + B + D) = 0$ を解集合における新しい $\mathrm{constr}(x)$ とする．このとき $\mathrm{norm}(AC + A + C) \equiv 1$ であれば，x に $\mathrm{norm}(BD + B + D)$ を代入し，書き替わった解集合の式を標準形に直す．

ステップ 4 $\mathrm{norm}(ACD + BC + CD + D) \equiv 0$ なら充足可能で終了．
$\mathrm{norm}(ACD + BC + CD + D) = 1$ なら充足不能で終了．
それ以外なら $ACD + BC + CD + D = 0$ を新しい入力式として，ステップ 1 へ．

ステップ 2 で $Ax \oplus B = 0$ がない場合，$A = 0$，$B = 0$ とおく．このとき，ステップ 3 の操作は $Cx \oplus D = 0$ を $\mathrm{constr}(x)$ として解集合に入れることを意味し，ステップ 4 で作られる式は $CD + D = 0$ になるので，このアルゴリズムは Boole の変数除去をインクリメンタルに行うような拡張になっていることがわかる．

アルゴリズムではループごとに入力式に含まれる変数が減るため，停止することは明らかである．また，アルゴリズムの正当性は次のように証明できる．

補題 3.1　(1) の 2 式と (2) の 2 式は同値である．

(1)　$Ax + B = 0$，　$Cx + D = 0$
(2)　$(AC + A + C)x + (BD + B + D) = 0$，　$ACD + BC + CD + D = 0$

証明
((1)⇒(2)) $Ax + B = 0$，$Cx + D = 0$ より

$$\begin{aligned}(AC + A + C)x &= ACx + Ax + Cx \\ &= (Ax)(Cx) + Ax + Cx \\ &= BD + B + D\end{aligned}$$

また，$Cx + D = 0$ より $CD + D = 0$ なので

$$\begin{aligned}ACD + BC + CD + D &= (AC)(Cx) + (Ax)C \\ &= ACx + ACx \\ &= 0\end{aligned}$$

((2)⇒(1)) $ACD + BC + CD + D = 0$ より

$$\begin{aligned}(C + 1)(ACD + BC + CD + D) &= (C + 1)D \\ &= 0\end{aligned}$$

これを $ACD + BC + CD + D = 0$ に代入すると

$$\begin{aligned}ACD + BC + CD + D &= AD + BC \\ &= 0\end{aligned}$$

である．よって

$$\begin{aligned}C((AC + A + C)x + (BD + B + D)) &= Cx + BCD + BC + CD \\ &= Cx + AD + AD + D \\ &= Cx + D \\ &= 0\end{aligned}$$

であり，$BD = BCD = AD$ なので

$$\begin{aligned}(AC+A+C)x+(BD+B+D) &= ACx+Ax+Cx+BD+B+D\\ &= Ax+AD+BD+B\\ &= Ax+B\\ &= 0\end{aligned}$$

□

補題 3.1 は，同値を言うだけなら (2) の $ACD+BC+CD+D=0$ は冗長であり，$AD+BC=0$ で十分である．しかし，次の補題で示すように充足性の条件として $CD+D=0$ が必要なので $ACD+BC+CD+D=0$ としている．

補題 **3.2**　$(A+1)B=0$ のとき $CD+D=0$ ならば

$$((AC+A+C)+1)(BD+B+D)=0$$

である．

証明

$CD+D=0$ は $(C+1)D=0$ なので，

$$\begin{aligned}((AC+A+C)+1)(BD+B+D) &= (A+1)(C+1)(BD+B+D)\\ &= (A+1)(C+1)B\\ &= 0\end{aligned}$$

□

補題 **3.3**　$Px\oplus Q=0$ が解集合にあるとき，$\mathrm{constr}(\prec_x)$ が成り立てば $(P+1)Q=0$ が成り立つ．

証明

初期状態，すなわち解集合が空のときは明らかに成り立つ．

次に $Ax\oplus B=0$ が解集合にあり，$\mathrm{constr}(\prec_x)$ が成り立てば $(A+1)B=0$ が成り立つと仮定する．補題 3.2 から $Cx\oplus D=0$ を融合して得られる式 $\mathrm{norm}(AC+A+C)x\oplus\mathrm{norm}(BD+B+D)=0$ に対しても，$\mathrm{constr}(\prec_x)$ が成り立てば $((AC+A+C)+1)(BD+B+D)=0$ が成り立つことがわかる．

□

補題 3.3により，Boole の変数除去と同様に $\prec_x$- 変数の値が $\mathrm{constr}(\prec_x)$ を満たすとき，x の値を $Q\le x\le P+Q+1$ の任意の値とすることができ，かつそれ以外の値とすることはできないことがわかる．

本節の最初に述べたように，リダクション抜きでは各変数に対する制約が他の式に反映されない．その簡単な例を次に示す．

例 **3.4**　$a\prec b\prec c$ で解集合が $\{ab=0\}$ のところへ $a\wedge b\wedge c=c$ を入力したとする．このとき，入力式の標準形は

$$(ab+1)c=0$$

となるが解集合に $\mathrm{constr}(c)$ がないため $(ab+1)c=0$ が $\mathrm{constr}(c)$ となって終了する．$ab=0$ が $(ab+1)c=0$ に反映されないので，$c=0$ を導くことができない．

正規形を導くリダクションはいくつか考えられるが，代表的なのは次のリダクションである．リダクションは少なくとも最後にすべての変数に対して実行すれば十分であるが，計算の無駄を省くために途中で行うことも考えられる．

リダクション　$Ax\oplus B=0$ が解集合にあり，$Cx\oplus D$ $(C\neq 1)$ が他の式の中にあるとき，条件を満たせば $Cx\oplus D$ を次のように書き換える．このリダクションは，x の係数を 1 にできるときは 1 にし，その他の場合はできるだけ係数が小さくなるように書き換えることを意味する．

$(\mathrm{norm}(AC+A+C) \equiv 1)$ のとき
$Cx \oplus D \rightarrow x \oplus \mathrm{norm}(BC+B+D)$
$(\mathrm{norm}(AC+A+C) \not\equiv 1,\ \mathrm{norm}(AC) \not\equiv 0)$ のとき
$Cx \oplus D \rightarrow \mathrm{norm}(AC+C)x \oplus \mathrm{norm}(BC+D)$

次に，このリダクションが停止して同値な正規形を求めることを証明する．証明中では議論を簡単にするため，$\mathrm{constr}(x)$ がない場合は $0 \times x \oplus 0 = 0$ があると考えることにする．同様に変数 x に対する項が式に含まれない場合も，$0 \times x$ が含まれていると考えることにする．

補題 3.5　上のリダクションは停止性，健全性を持つ．

証明
$Ax \oplus B = 0$ が解集合にあり，Cx をリダクションするとする．
$\mathrm{norm}(AC+A+C) = 1$ のとき

$$\begin{aligned} Cx+D &= (AC+A+C)x+(C+1)Ax+D \\ &= x+(C+1)B+D \\ &= x+BC+B+D \end{aligned}$$

である．また $\mathrm{norm}(AC) \not\equiv 0$ のとき

$$\begin{aligned} Cx+D &= (AC+C)x+ACx+D \\ &= (AC+C)x+BC+D \end{aligned}$$

であるので，リダクションする前の式と後の式は同値である．
ある式の項 Cx が $Ax \oplus B = 0$ によってリダクションされると，x の係数は 1 か $AC+C$ になる．$(AC+C)A = 0$ なので，新たに x の項が付け加わらない限り x の項が再びリダクションされるのは，係数を $AC+C$ から 1 にできることがわかったときのみである．新たに項が付け加わるのは x より大きい変数に対するリダクションが行われたときのみなので，x より大きい変数の項に対しリダクションできなくなれば高々 2 回しかリダクションされないので停止する．
□

補題 3.6　2 つの同値なブール制約の集合にアルゴリズムを適用して得られる解を S_1，S_2 とする．このとき S_1，S_2 の $\mathrm{constr}(x)$ がそれぞれ $Ax \oplus B = 0$，$Cx \oplus D = 0$ であるなら，$A = C$ かつ $B = D$ である．

証明
$Ax \oplus B = 0$ は $B \le x \le A+B+1$ を示し $Cx \oplus D = 0$ は $D \le x \le C+D+1$ を意味するので，$B = D$，$A+B+1 = C+D+1$ でなければならない．よって $B = D$，$A = C$ である．
□

補題 3.7　2 つの同値なブール制約の集合にアルゴリズムを適用し，最後にできる限りのリダクションを行って得られる解を S_1，S_2 とすると，$S_1 \equiv S_2$ である．

証明
最小の変数 x に対する S_1，S_2 の $\mathrm{constr}(x)$ をそれぞれ $Ax \oplus B = 0$，$Cx \oplus D = 0$ とすると，補題 3.6 より $A = C$，$B = D$ であるが A，B，C，D は 0 か 1 なので，$A \equiv C$ かつ $B \equiv D$ である．
$\mathrm{constr}(\prec_x)$ が S_1 と S_2 で等しいとき，S_1，S_2 の $\mathrm{constr}(x)$ をそれぞれ $Ax \oplus B = 0$，$Cx \oplus D = 0$ とおいて $A \equiv C$ を導く．$B \equiv D$ の証明は同様である．
$A \not\equiv C$ と仮定する．補題 3.6 より $A = C$ なので，$\mathrm{norm}(A+C) \equiv Py \oplus Q = 0$ $(P \not\equiv 0)$ となる $\prec_x$- 変数 y が存在する．$Py \oplus Q = 0$ は y に対する制約であるので，S_1，S_2 に $Sy \oplus T = 0$ があって $Q \le T \le y \le S+T+1 \le P+Q+1$ でなければならない．これは $P \le S$，つまり $PS = P$ を意味する．また，変数 y が残っているので $S \not\equiv 1$ すなわち $S \neq 1$ である．
P を作る A，C のそれぞれの y の係数を A'，C' とおく．従って $\mathrm{norm}(A'+C') = P$ である．

(1) A', C' の一方が 0 のとき
対称性により $A' \equiv 0$ とする．このとき $P = C'$ より $SC' = C'$ である．$C' \equiv 1$ とすると $S = 1$ となって矛盾するので $C' \neq 1$ であり，$SC' = 0$ より $C' = 0$ なので $P = P'z \oplus Q' = 0$ とおくことができる．

(2) A', C' の一方が 1 のとき
対称性により $A' \equiv 1$ とする．C' は $C' \neq 1$ なので $SC' = 0$ であり，$PS = P$ より $(1 + C')S = 1 + C'$ となって $C' = 1 + S$ が得られる．$C' \equiv \mathrm{norm}(1 + S)$ だと $\mathrm{norm}(SC' + S + C') \equiv 1$ となり矛盾するので，$C' \neq \mathrm{norm}(1 + S)$ でなければならない．よって $\mathrm{norm}(C' + 1 + S) \equiv P'z \oplus Q' = 0$ とおくことができる．

(3) A', C' がともに 1 以外のとき
$SA' = 0$, $SC' = 0$ なので $PS = 0$ であり，よって $P = 0$ である．すると $P = P'z \oplus Q' = 0$ とおくことができる．

いずれの場合も変数の少ない式 $P'z \oplus Q' = 0$ に帰着できるが，$Py \oplus Q = 0$ と同様の議論を続けると変数の数は有限なのでいずれ矛盾する．よって $A = C$ でなければならない．
□

定理 3.8　リダクション付きの漸増型変数除去法は必ず停止し，入力された制約集合が充足可能であればそれと同値な解集合が求まり，それは正規形である．

証明
一連の補題より明らか．
□

4 例題

最後に，論理回路の問題と Lewis Carroll のパズルを例題として取り上げる．

4.1 論理回路の問題

and- 素子，or- 素子，not- 素子で作られた 5 入力 3 出力のある回路をそのままブール式で表現すると，次のようになるとする．

$$
\begin{array}{llll}
\mathtt{I1} = \mathtt{X1} \wedge \mathtt{X2}, & \mathtt{I2} = \mathtt{X1} \vee \mathtt{X2}, & \mathtt{I3} = \mathtt{X3} \wedge \mathtt{X4}, & \mathtt{I4} = \mathtt{X3} \vee \mathtt{X4}, \\
\mathtt{I5} = \neg \mathtt{I1}, & \mathtt{I6} = \neg \mathtt{I2}, & \mathtt{I7} = \neg \mathtt{I3}, & \mathtt{I8} = \neg \mathtt{I4}, \\
\mathtt{I9} = \mathtt{I1} \vee \mathtt{I3}, & \mathtt{I10} = \mathtt{I1} \wedge \mathtt{I3}, & \mathtt{I11} = \mathtt{I6} \vee \mathtt{I8}, & \mathtt{I12} = \mathtt{I6} \wedge \mathtt{I8}, \\
\mathtt{I13} = \mathtt{I5} \wedge \mathtt{I2}, & \mathtt{I14} = \mathtt{I7} \wedge \mathtt{I4}, & \mathtt{I15} = \neg \mathtt{I14}, & \mathtt{I16} = \mathtt{I14} \vee \neg \mathtt{I13}, \\
\mathtt{I17} = \mathtt{I13} \vee \mathtt{I15}, & \mathtt{I18} = \mathtt{I13} \vee \mathtt{I14}, & \mathtt{I19} = \neg \mathtt{I13} \vee \mathtt{I15}, & \\
\mathtt{I20} = \mathtt{I9} \wedge \mathtt{I4} \wedge \mathtt{I2} \wedge \mathtt{X5}, & & \mathtt{I21} = \mathtt{I11} \wedge \mathtt{I7} \wedge \mathtt{I5} \wedge \neg \mathtt{X5}, & \\
\mathtt{I22} = \mathtt{X5} \wedge \mathtt{I16} \wedge \mathtt{I17}, & & \mathtt{I23} = \neg \mathtt{X5} \wedge \mathtt{I18} \wedge \mathtt{I19}, & \\
\mathtt{Y1} = \mathtt{I20} \vee \mathtt{I10}, & \neg \mathtt{Y2} = \mathtt{Y1} \vee \mathtt{I21} \vee \mathtt{I12}, & \mathtt{Y3} = \mathtt{I22} \vee \mathtt{I23} &
\end{array}
$$

この回路は，5 つの入力 X1, X2, X3, X4, X5 のうち 1 であるものの個数を 3 桁の 2 進数 Y1 Y2 Y3 として出力するものである．ここではこの 8 つの変数の制約が欲しいので，これらの変数の順位を他の変数より下げることにする．そうすれば他の変数が先に除去されるので，興味のある変数の間の関係を得ることができる．

制約をそのまま解けば一般解が求まるが，回路への入出力を表す変数のいくつかに値を与えて残りの変数の関係を見ることもできる．例えばすべての入力変数を 0 とすると出力変数もすべて 0 となり，出力変数を 1 0 1 とすると十進数の 5 を意味するので入力変数はすべて 1 になる．また，出力変数の値に 0 0 1 という条件を与えると十進数の 1 を意味するので，次のような解が得られる．

$$
\begin{array}{l}
\mathtt{X5} = \mathtt{X4} + (\mathtt{X3} + (\mathtt{X2} + (\mathtt{X1} + 1))) \\
(\mathtt{X3} + (\mathtt{X2} + \mathtt{X1})) \wedge \mathtt{X4} = 0 \\
(\mathtt{X2} + \mathtt{X1}) \wedge \mathtt{X3} = 0 \\
\mathtt{X1} \wedge \mathtt{X2} = 0
\end{array}
$$

これは，次のように解釈することができる．

(1) constr(X1) がないので，X1 の値は自由に決めることができる．

(2) X1 の値が 1 のとき，$X1 \wedge X2 = 0$ より X2 の値は 0 でなければならない．X1 の値が 0 ならば，X2 の値は自由である．

(3) X1，X2 のいずれかが 1 のとき，$(X2 + X1) \wedge X3 = 0$ より X3 の値は 0 でなければならない．X1，X2 いずれも 0 ならば，X3 の値は自由である．

(4) X1，X2，X3 のいずれかが 1 のとき，$(X3 + (X2 + X1)) \wedge X4 = 0$ より X4 の値は 0 でなければならない．X1，X2，X3 いずれも 0 ならば，X4 の値は自由である．

(5) X5 は X1，X2，X3，X4 のいずれかが 1 のとき 0 で，いずれも 0 のときは 1 である．

この解釈からもわかるように，すべての解を表す解集合に対して最小の変数から値を割り当てていくことにより，好みの解を求めることが可能である．

4.2 Lewis Carroll のパズル

Lewis Carroll のパズルは Colmerauer が [3] で取り上げた問題であるが，漸増型変数除去法を使うと高速に解くことができる．この問題は，次のように 16 の命題 a, ..., p に対し 18 の論理式があるとき，「命題 f, n, p の間の関係を求めよ」のように指定した命題間の関係を求める問題である．

$(f \wedge \neg c) \rightarrow g$	$(a \wedge o) \rightarrow b$	$(k \wedge h) \rightarrow n$
$(g \wedge \neg d) \rightarrow \neg f$	$(p \wedge h) \rightarrow j$	$(g \wedge d) \rightarrow h$
$(\neg i \wedge \neg p) \rightarrow \neg n$	$(c \wedge f) \rightarrow h$	$(n \wedge j) \rightarrow (g \wedge l)$
$(k \wedge g) \rightarrow i$	$(p \wedge c \wedge l) \rightarrow e$	$(k \wedge \neg a \wedge \neg b) \rightarrow \neg f$
$(n \wedge \neg c) \rightarrow i$	$(a \wedge m \wedge d) \rightarrow g$	$(g \wedge j) \rightarrow \neg e$
$(n \wedge d) \rightarrow p$	$(\neg o \wedge \neg m) \rightarrow \neg k$	$(i \wedge h) \rightarrow (g \vee j)$

各命題を変数とし，“論理式 = 1” という 18 のブール式に直すことによりブール制約とすることができる．そして指定された命題を表す変数の順序を他の変数より小さくして漸増型変数除去法で正規形を求める．すると興味のない変数から除去されていくので，もし何らかの関係があるなら指定された命題の変数のみからなる式が解集合に得られる．

例えば命題 a，f，i の順序を小さくして解集合を求めてみる．すると a，f，i のみの式が得られないので，この 3 つの命題の間には関係がないことがわかる．次に命題 f，n，p の順序を小さくして解集合を求めてみると，次のような式が得られる．

$$(p + 1) \wedge n \wedge f = 0$$

これは

$$n \wedge f \rightarrow p$$

という関係を意味している．

興味のない変数がある場合，それらに対する式は出力する必要がない．このアルゴリズムは CAL[8] のソルバーとしてインプリメントを行ったが，他のソルバーと同様に興味のない変数を区別し，そのような変数に対する式は出力しないようにしている．

5 まとめ

本論文では，ブール制約式をインクリメンタルに解くためのアルゴリズム「漸増型変数除去法」を提案した．このアルゴリズムの特徴は，ブールの変数除去を拡張してインクリメンタルに式を入力でき，求まる解集合は入力した変数のみで表され，しかも 1 つの正規形を求めることができる点である．また，解集合は全解を表すが，そこから好みの解を簡単に得られることも示した．

その他の利点としては，興味のある変数の間に何か関係が成り立つかどうかを知ることのできる点がある．興味のある変数の順序を相対的に小さくして解集合を求め，それらの変数のみからなる式が得られるときに限り，その式が表す関係が成り立つ．この問題は，漸増型変数除去法によって高速に解くことができる．

欠点としては，多項式表現をしているため解集合が複雑になる場合があることである．ブーリアン・グレブナ・ベースも多項式表現であるが，その場合は各式の次数をできる限り小さくするため，式の数は増えるが相対的に簡単な解が得られる．しかし，漸増型変数除去法では順序の大きい変数に対する式の次数が高くなり易く，従って複雑になり易い．ただし，順序の大きい変数に興味がなければまったく問題はないであろう．

本論文ではドメインを0と1に限定した．しかし，アルゴリズムの基本部分にはこの制限はないので，式の表現や演算を拡張することにより一般ブール代数を扱うことも可能である．実際，ICOT では集合を扱えるようにブーリアン・ブフバーガ・アルゴリズムの拡張を行った [11]．同様の拡張を 漸増型変数除去法に施すことが可能である．

その他には，現在はナイーブなインプリメントしか行っていないが，計算の無駄を省くために内部表現を工夫する等が課題として上げられる．

謝辞

本研究にあたり貴重な示唆や助言を頂いた，筑波大学の坂井 公 先生および ICOT の佐藤洋祐 主任研究員に感謝致します．また ICOT の相場 亮 室長代理には，研究の上でいろいろお世話になりました．

参考文献

[1] Brown, F. M. : *Boolean Reasoning*, Kluwer Academic Publishers, Norwell, Massachusetts, 1990.

[2] Büttner, W. and Simonis, H. : Embedding Boolean Expressions into Logic Programming, *J. Symbolic Computation* 4, 1987, pp.191–205.

[3] Colmerauer, A. : An Introduction to Prolog III, *CACM* 33, 1990, pp.69–90.

[4] Dincbas, M., et al. : The Constraint Logic Programming Language CHIP, *Proc. International Conference on Fifth Generation Computer Systems*, 1988, pp.693–702.

[5] Halmos, P. R. : *Lectures on Boolean Algebras*, D. Van Nostrand Company, 1963.

[6] Martin, U. and Nipkow, T. : Unification in Boolean Rings, Proc. 8th Conf. on Automated Deduction, *LNCS* 230, 1986, pp.506–513.

[7] Rudeanu, S. : *Boolean Functions and Equations*, North-Holland Publ. Co., 1974 ; 森脇幸生訳 : ブール関数と方程式, 工学図書株式会社, 1983.

[8] Sakai, K. and Aiba, A. : CAL: A theoretical background of constraint logic programming and its applications, *J. Symbolic Computation* 8, 1989, pp.589–603.

[9] Sakai, K. and Sato, Y. : Application of Ideal theory to Boolean constraint solving, *Proc. Pacific Rim International Conference on Artificial Intelligence 90*, 1990.

[10] Sakai, K., Sato, Y. and Menju, S. : *Boolean Gröbner Bases*, to appear (1992).

[11] Sato, Y., Sakai, K. and Menju, S. : *Solving Constraints over Sets by Boolean Gröbner Bases*, ICOT-TR 680, 1991.