２０１８年度修士論文

判断推理の多項式モデルによる計算アルゴリズム

早稲田大学　数学応用数理専攻
5117A032-6
高橋　拓也
指導教員：楫元

2019 年 1 月 29 日

1 Introduction

近年における計算代数学の発展は著しく、グレブナー基底や終結式理論を用いることで高度な数式処理が計算機により実装可能である。一方で、[5] において多値論理は多項式モデルによって表現可能なことが知られていて、「与えられた仮定から結論が導かれるか否か」という推論の問は多項式のイデアル従属判定問題に帰着できることがわかっている。

初等幾何については幾何の条件を多項式に変換し推論を自動証明する具体的な例が多々知られているが、多値論理についてはある推論が正しいかどうかを自動証明する具体的な例があまり知られていない。

そこで我々は具体的な問題の例として、入社試験や公務員試験でよく出題される「判断推理」という分野に着目し、計算機を使った解答を試みた。結果、「順序関係」、「真偽判定」、「命題論理」、「試合（総当たり）」、「位置関係」においては多項式モデルを用いて問題を自動で解答できることがわかった。

「真偽判定」や「命題論理」の分野においては二値論理に限定すると SAT 問題とみなし多項式表現するといった手法を紹介している文献はいくつかある。しかし三値以上の値を取る問題に関して多項式表現して解くといった文献はなかなか見られない。例えば「位置関係」という分野では二値論理の問題として解くことも可能であるが、実は三値以上の論理に変換して問題を解くことで変数の量を減らし計算を高速で行うことが可能である。更に例えば「真偽判定」において、二値の範疇であれば手計算で計算可能な問題でも、三値以上に拡張すると手計算では現実的ではないレベルの時間を要するようなものもある。こういった観点から三値以上の値を取る問題を計算機で解く手法はかなり有益であると考えられる。

chapter2 では多項式計算の用語を解説する。chapter3 ではとある制限下における有限体上の多項式環の性質を述べる。chapter4 では論理式と多項式の関係について記述し、chapter5 にて具体的な問題計算例とそのアルゴリズムを載せる。

2 Preliminaries

まず、多項式計算の理論に必要な定義・定理を挙げる。

定義 2.1 (グレブナー基底)**.** k を体とする。多項式環 $k[x_1, ..., x_n]$ と単項式順序 $>$ に対

して、多項式の集合 $G = \{g_1, ..., g_s\}$ がイデアル $J \subset k[x_1, ..., x_n]$ の $>$ に関するグレブナー基底であるとは、

$$\langle LT_>(J) \rangle \ = \ \langle LT_>(g_1), ..., LT_>(g_s) \rangle$$

が成立することをいう

定義 2.2 (消去イデアル)**.** $J = \langle f_1, ..., f_s \rangle \ \subset k[x_1, ..., x_n]$ に対して、J の l 次消去イデアル J_l とは、

$$J_l = J \cap k[x_{l+1}, ..., x_n]$$

で定義された $k[x_{l+1}, ..., x_n]$ のイデアルである。

以上の概念を用いることで、多項式計算において重要な以下の定理を得られる。

定理 2.1 (消去定理)**.** $J = \langle f_1, ..., f_s \rangle \ \subset k[x_1, ..., x_n]$ に対して G を $x_1, ..., x_l > x_{l+1}, ..., x_n$ となる消去順序に関する J のグレブナー基底とする。この時、すべての $0 \leq l \leq n$ に対して、$G_l = G \cap k[x_{l+1}, ..., x_n]$ は J の l 次消去イデアル J_l のグレブナー基底である。

Proof. [1] 参照。 □

次に、不等号付きの多項式を処理するために必要な概念を記す。

定義 2.3 (限量記号除去（Quantifier Elimination）)**.** 存在記号 $\exists$ や全称記号 $\forall$ といった限量記号が付随している一階述語論理式を考える。与えられた論理式に対して、限量記号を外した等価な論理式を導出することを、限量記号除去という。

実数体上の限量記号除去については盛んに研究が行われている。

例 2.1. $\exists x(x^2 + ax + b \leq 0)$ は、解の公式を利用することにより $a^2 - 4b \geq 0$ と同値なことがわかる。

計算機上で $a^2 - 4b \geq 0$ を導出するためには、自動的に数式処理するアルゴリズムが存在しなければならない。この場合、まず $x^2 + ax + b$ に対してスツルム・ハビッチ列というものを定義し、それを用いて代数細胞分割を行う。（この方法は、柱状代数分解法（Cylindrical Algebraic Decomposition）と呼ばれる）各々の細胞は、与えられた多項式の取る値が正であるエリア・負であるエリア・0 であるエリアいずれかに対応している。

その中から条件を満たすエリアに対応する不等式を調べることで、限量記号を除去することが可能となる。詳しくは [6] を参照されたい。上記の限量記号除去は基本的に実数体上での計算を仮定している。実は有限体上の限量記号除去についても研究されている [3]。有限体においては大小関係が定まっていないため、$\leq$ や $>$ といった不等号は考えない。したがって実数体上の計算理論とは若干異なった形になっている。以下で有限体上の限量記号除去に必要な理論を述べる。

3 Computational Algebra in finite field

定理 3.1 (フェルマーの小定理)**.** $\mathbb{F}_p$ を標数 p（p : 素数）の有限体とする。この時、任意の元 $a \in \mathbb{F}_p$ に対して、$a^p - a = 0$ が成立する。更に、$V(x^p - x) = \mathbb{F}_p$ が成立する。

有限体は一般的に代数閉体でないため、零点定理が成立しない。しかし、フェルマーの小定理から一定の条件下では零点定理が成立することがわかる。

定理 3.2 (有限体上の零点定理)**.** $J \subset \mathbb{F}_p[x_1, ..., x_n]$ に対して、

$$I(V(J)) = J + \langle x_1^p - x_1, ..., x_n^p - x_n \rangle$$

が成立する。

Proof. 以下の 3 点の事実が証明の本質である。詳細については [4, Theorem2.2] 参照。

- $J + \langle x_1^p - x_1, ..., x_n^p - x_n \rangle$ は根基イデアル
- $\mathbb{F}_p^a$ を $\mathbb{F}_p$ の代数閉包とすると、$I(V^a(J)) = \sqrt{J}$（ただし、$V^a(J) = \{\boldsymbol{a} \in (\mathbb{F}_p^a)^n : \forall f \in J, f(\boldsymbol{a}) = 0\}$）
- $V^a(\langle x_1^p - x_1, ..., x_n^p - x_n \rangle) = \mathbb{F}_p^n$

□

消去定理と零点定理から、有限体上の方程式において「存在記号を消去すること」と「変数を消去すること」が同値であるという以下の定理が示される。

定理 3.3 (限量記号除去と消去イデアルの関係)**.** $J \subset \mathbb{F}_p[x_1, ..., x_n, y_1, ..., y_m]$ を $\langle x_1^p - x_1, ..., x_n^p - x_n, y_1^p - y_1, ..., y_m^p - y_m \rangle$ を含むイデアルとする。
この時、$\pi_n : \mathbb{F}_p^{n+m} \to \mathbb{F}_p^m$ を射影とすると、以下の等式が成立する。

$$\pi_n(V(J)) = V(J_n)$$

Proof. J が $\langle x_1^p - x_1, ..., x_n^p - x_n, y_1^p - y_1, ..., y_m^p - y_m \rangle$ を含んでいることが証明の本質である。詳しくは [4, Theorem3.1] 参照。 □

これより、$J \neq \langle 1 \rangle \Rightarrow V(J) \neq \phi$ が直ちに導かれる。
以上により有限体上の数式処理において、$\langle x_1^p - x_1, ..., x_n^p - x_n \rangle$ を含むものに関する限りでは比較的単純な議論が可能であることがわかった。さらに定理 3.3 を活用することにより、有限体上の限量記号除去アルゴリズムを構築することができる。([4] 参照)

$\langle x_1^p - x_1, ..., x_n^p - x_n \rangle$ を含むイデアルを考察する例としては、多値論理が挙げられる。以下で多値論理の代数的表現について知られていることを述べる。詳しくは [5] を参照されたい。

4 Multi-valued Logic on Polynomial Model

二値論理に対して命題論理式が与えられたとき、その真理表を考えることができる。真理表を考えることは、命題論理式 X を論理値 $v(X) \in \mathbb{F}_p$ へ移す付値 v を考えることに他ならない。(この付値 v はまずそれぞれのリテラル (原始的な論理式又はその否定のこと) $X_1, ..., X_n$ を定義域として $v(X_1), ..., v(X_n)$ と与えられる。この定義域を拡張することで論理演算子込みの命題論理式全体に対する付値が定義される。拡張の具体的な構成方法は [5] を参照されたい。) 多値論理に関しても同様の付値が定義される。

例 4.1. ウカシェヴィッチの三値論理において、$X \wedge Y, X \vee Y$ の真理表は以下で表される。(*0:true,1:false,2:unknown* とする。)

X	Y	$X \wedge Y$	$X \vee Y$
0	0	0	0
0	1	1	0
0	2	2	0
1	0	1	0
1	1	1	1
1	2	1	2
2	0	2	0
2	1	1	2
2	2	2	2

これらはそれぞれ命題論理式と $\mathbb{F}_3$ における論理値の付値対応を表にしたものと考えられる。

実は命題論理式の集合から論理値の集合 $\mathbb{F}_p$ への付値写像を $\mathbb{F}_p[X_1, ..., X_n]$ 上の多項式の代入写像として表現することができる。ここで、命題論理式の集合を厳密に定義する。ただし、写像 $\mathbb{F}_p^k \to \mathbb{F}_p (1 \leq k \leq n)$ の個数は p^{p^k} であるため、多値論理における論理演算子は同値なものを除くと有限個しかないことに注意する。

定義 4.1. $\{X_1, ..., X_n\} \in P$ かつ、$\Phi_1, ..., \Phi_k \in P \Rightarrow \phi(\Phi_1, ..., \Phi_k) \in P$（ただし、$\phi$ は多値論理における論理演算子、$\Phi_1, ..., \Phi_k$ は原始的とは限らない命題論理式）を満たすものとして、論理演算子込みの命題論理式の集合 P を帰納的に定義する。（同値なものを除いた論理演算子は有限個であるため、P は有限集合だということがわかる。）

例 4.2. 二値論理に対して命題論理式 $X \wedge Y$ は、$\mathbb{F}_2[X, Y]$ 上の多項式 $XY + X + Y$ と表現することができる。多項式の代入写像 $\mathbb{F}_2[X, Y] \to \mathbb{F}_2$ を考えることにより、付値：$P \to \mathbb{F}_2$ の取る値と対応していることがわかる。

以上は一例であるが、命題論理式に対応する多項式が一般に存在することを保証する必要がある。

定理 4.1. P を上記で定義された集合、$A = \mathbb{F}_p[X_1, ..., X_n] / \langle X_1^p - X_1, ..., X_n^p - X_n \rangle$ とする。また、P と A を定義域とする付値をそれぞれ v, v^* とする。このとき、$v^* \circ T = v$ となる写像 $T : P \to A$ が存在する。

Proof. 証明の概略を記す。詳細は [5, THEOREM2.1] を参照されたい。

論理演算子を ϕ とし、この演算によって導かれる論理式を $\phi(X_1, ..., X_n)$ とする。それぞれのリテラルに値を代入したとき、この演算子によって導かれた論理式は論理値を返す。この値を $\bar{\phi}(i_1, ..., i_n)$ $(i_1, ..., i_n \in F_p)$ とする。

$\phi(X_1, ..., X_n)$ に対応する多項式を $T_\phi(X_1, ..., X_n)$ とする。（すなわち、$T_\phi(i_1, ..., i_n) = \bar{\phi}(i_1, ..., i_n)$ となる。）実は $T_\phi(X_1, ..., X_n)$ は以下で表される。（ラグランジュの補間多項式）

$$T_\phi(X_1, ..., X_n) = \sum_{(i_1, ..., i_n) \in \mathbb{F}_p^n} \bar{\phi}(i_1, ..., i_n) \prod_{m=1}^{n} \prod_{j \in \mathbb{F}_p / \{i_m\}} \frac{X_m - j}{i_m - j}$$

この T_ϕ を用いて、以下により写像 T を帰納的に定義する。

$$T(X_i) = X_i \ (i = 1, ..., n)$$
$$T(\phi(\Phi_1, ..., \Phi_n)) = T_\phi(T(\Phi_1), ..., T(\Phi_n))\ (\Phi_1, ..., \Phi_n \text{ は命題論理式})$$

この T は $v^* \circ T = v$ を満たすような、命題論理式から多項式への写像である。 □

例えば三値論理において論理否定はそれぞれ $\neg(0) = 1, \neg(1) = 0, \neg(2) = 2$ となるが、$T_\neg(X) = 2X + 1$ となり、確かに $T_\neg(0) = 1, T_\neg(1) = 0, T_\neg(2) = 2$ となっている。

以上の T を用いることで、論理演繹（推論）の問題が多項式のイデアル従属判定問題に帰着されるという次の定理を与えることができる。

定理 4.2 (演繹判定定理)**.** 論理式 $\Phi_1, ..., \Phi_n, \Psi \in P$ に対して、$\Phi_1, ..., \Phi_n \models \Psi$ と以下は同値である。（ただし、$a \in \{0, 1, ..., p-1\}$ を多値論理における真に対応する値とする。）

$$T(\Psi) - a \in \langle T(\Phi_1) - a, ..., T(\Phi_m) - a, x_1^p - x_1, ..., x_n^p - x_n \rangle$$

Proof. [5, THEOREM4.4] 参照。 □

5 Main Result

本論文では判断推理の問題における「順位関係」、「命題論理」、「位置関係」、「真偽判定」について、計算機代数ソフト singular を用いた自動計算アルゴリズムを提唱する。（有限体上の零点定理や演繹判定定理を活用した例である。）

一般的に判断推理の問題は、以下の段階を踏むことで解くことができる。

- 仮定を述べる上で必要な変数を用意する。（$x_1, ..., x_n$ とする）。
- 変数の取りうる値を考え、必要な値の枠を包括するように体を設定する。（この場合、取りうる値以上の最小の素数を標数とする体 $\mathbb{F}_p$ を考えれば良い。）
- 仮定を多項式表現し、イデアルの元に追加する。（$\langle x_1^p - x_1, ..., x_n^p - x_n \rangle$ を必ず追加する。）このイデアルを仮定イデアルと呼ぶことにする。
- 結論を多項式表現し、仮定イデアルに従属するか判定する。

仮定を満たすような解が存在しない場合、仮定イデアルのグレブナー基底は $\langle 1 \rangle$ となってしまう。このとき、結論に対応する多項式は必ず仮定イデアルに従属する。（仮定が偽ならばいかなる条件命題も真となってしまうことを意味する。）このようなケースは今回考えないことにする。従って、仮定イデアルのグレブナー基底は必ず計算して確認する必

要がある。

また判断推理を考える際、結論が「必ず導かれる」ケースを考える場合と「導かれる時がある」ケースを考える場合とで問題の解決方法が変わってくる。

- 結論が「必ず導かれる」か確かめるためには、結論の否定に対応する多項式が仮定イデアルに従属しないことを示せば良い。
- 結論が「導かれる時がある」か確かめるためには、結論に対応する多項式を仮定イデアルの元に追加したとき、そのイデアルのグレブナー基底が $\langle 1 \rangle$ でなければ良い。（今回の有限体上では $J \neq \langle 1 \rangle \Rightarrow V(J) \neq \phi$ であるため、$\langle 1 \rangle$ でないということは仮定と結論の両方を満たす解が存在することにほかならない。）

5.1 順位関係

次のような問題を自動で解くアルゴリズムを考えよう。

順位関係

P、Q、R、S、T の 5 人で徒競走をした。
5 人の順位について次のことが分かっている。
i）R の順位は、S より上である
ii）T の順位は、R よりも上だが、1 着ではなかった
iii）Q の順位は、P より上である
iv）同着の順位の者はいない

次のア、イ、ウの推論のうち、必ず正しいものはどれか。
ア Q は 1 着である
イ S は 5 着である
ウ 2 着は P または T である

まず、5 人の順位に対応する変数をそれぞれ p, q, r, s, t とする。順位は 1 ～ 5 位の範囲なので、標数は 5 以上の最小の素数を取れば良い。（この場合、標数 5 の体を考えると取りうる値が 0 ～ 4 になる。順位に対応する値を 1 つずらして標数 5 の体上で計算することもできるが、今回は簡単のため標数 7 の体で考える。）

ここで仮定を多項式表現しよう。

まず、5 人の順位は 1 ～ 5 位の範囲であることを表現する。例えば P に関して、$(p-1)(p-2)(p-3)(p-4)(p-5)=0$ が成立する。他の変数に関しても同様の多項式を考え、仮定イデアルの元に追加する。(この仮定は、$\langle p^7-p,...,t^7-t\rangle$ を制限したものと考えられる。)

次に「R の順位は、S より上である」を多項式表現しよう。この場合、すべての取りうる値を多項式表現することを考えれば良い。従って、$i_1=(r-1)(r-2)(r-3)(r-4),s-5,i_2=(r-1)(r-2)(r-3),(s-5)(s-4),i_3=(r-1)(r-2),(s-5)(s-4)(s-3),i_4=r-1,(s-5)(s-4)(s-3)(s-2)$ とし、これらの和集合に対応するイデアル $i=i_1\cdot i_2\cdot i_3\cdot i_4$ を仮定イデアルに追加する。他の条件についても同様に多項式表現し仮定イデアルに追加する。

「同着の順位がいない」という条件については、差積が 0 でないことを利用すれば良い。この条件を多項式表現するためには、以降でも頻繁に用いる次の補題のいずれかを利用すれば良い。

補題 5.1. $a\in F_p$ について以下は同値である。

$$a\neq 0 \iff a^{p-1}=1$$

補題 5.2. $a\in F_p$ について以下は同値である。

$$a\neq 0 \iff \exists u\in F_p\ s.t\ au=1$$

計算量の観点からすると、「次数が少ない」場合や「変数が少ない」場合は計算が早く行われることが直観的に明らかである。今回の場合は後者のケースで計算すると時間が短縮されたので、補題 5.2 を活用して条件を多項式化することを試みる。(一般的にどちらのケースを採用すると計算が速くなるかについては十分な研究余地があると思われる。)

改めて「同着の順位がいない」という条件を多項式化すると、以下のような形になる。(ここで多項式環に変数 u を追加する。)
$((p-q)(p-r)(p-s)(p-t)(q-r)(q-s)(q-t)(r-s)(r-t)(s-t))u-1=0$

以上の条件を仮定イデアルに代入し、グレブナー基底を計算すれば良い。計算機代数ソフト singular における本計算のソースコードと結果を以下に記す。

order.txt

```
ring R=7,(u,p,q,r,s,t),lp;
ideal hy=
(p-1)*(p-2)*(p-3)*(p-4)*(p-5),
(q-1)*(q-2)*(q-3)*(q-4)*(q-5),
(r-1)*(r-2)*(r-3)*(r-4)*(r-5),
(s-1)*(s-2)*(s-3)*(s-4)*(s-5),
(t-1)*(t-2)*(t-3)*(t-4)*(t-5);
ideal i1=(r-1)*(r-2)*(r-3)*(r-4),s-5;
ideal i2=(r-1)*(r-2)*(r-3),(s-5)*(s-4);
ideal i3=(r-1)*(r-2),(s-5)*(s-4)*(s-3);
ideal i4=r-1,(s-5)*(s-4)*(s-3)*(s-2);
ideal i=i1*i2*i3*i4;
ideal ii1=(t-2)*(t-3)*(t-4),r-5;
ideal ii2=(t-2)*(t-3),(r-5)*(r-4);
ideal ii3=t-2,(r-5)*(r-4)*(r-3);
ideal ii=ii1*ii2*ii3;
ideal iii1=(q-1)*(q-2)*(q-3)*(q-4),p-5;
ideal iii2=(q-1)*(q-2)*(q-3),(p-5)*(p-4);
ideal iii3=(q-1)*(q-2),(p-5)*(p-4)*(p-3);
ideal iii4=q-1,(p-5)*(p-4)*(p-3)*(p-2);
ideal iii=iii1*iii2*iii3*iii4;
ideal iv=((p-q)*(p-r)*(p-s)*(p-t)*(q-r)*(q-s)*(q-t)*(r-s)*(r-t)*(s-t))*u-1;
ideal HY=hy,i,ii,iii,iv;
ideal j=groebner(HY,"fglm");
j;
```

これにより仮定イデアルのグレブナー基底が計算される。出力は以下になる。

```
> <"order.txt";
j[1]=t2+2t-1
j[2]=st-2s+2t+3
j[3]=s2-2s-1
```

j[4]=rt-2r+3t+1
j[5]=rs+2r-3s+1
j[6]=r2-2
j[7]=q-1
j[8]=p+r+s+t
j[9]=u-2r+2s+2t

ここで結論が仮定から「必ず」推論されるかどうか考えよう。これを確かめるためには、結論の「否定」に対応する多項式が仮定から導かれないことを確かめれば良い。(仮定イデアルのグレブナー基底が $\langle 1 \rangle$ でないことからこの方法が利用できるのである。)

ア、イ、ウそれぞれの否定に対応する多項式は $(q-1)^6-1, (s-5)^6-1, ((p-2)(t-2))^6-1$ である。これらを仮定イデアル j に加えたもののグレブナー基底をそれぞれ j_1, j_2, j_3 とすると以下のように出力される。

> ideal j1=j,(q-1)^6-1;
> groebner(j1);
_[1]=1
> ideal j2=j,(s-5)^6-1;
> groebner(j2);
_[1]=t-2
_[2]=s+3
_[3]=r+3s-1
_[4]=q-1
_[5]=p+r+s+t
_[6]=u-2r+2s+2t
> ideal j3=j,((p-2)*(t-2))^6-1;
> groebner(j3);
_[1]=1

これにより、結論ア、ウの否定に解が存在しないことがわかるため、ア、ウは仮定から必ず推論されることが確認できたことになる。
(実は仮定イデアルを計算した地点で Q が一着にしかなりえないことが、j[7] より容易に判断できる。)

今回は結論に対応する式が 1 式だったため多項式の否定として表現できたが、一般に結論に対応するものは 2 元以上のイデアルである。イデアル $I = \langle f_1, ..., f_s \rangle$ の否定に対応するイデアルは、零点集合の補集合を考慮すると $\overline{I} = (f_1^{p-1} - 1)(f_2^{p-1} - 1)...(f_s^{p-1} - 1)$ と表現することができる。（ド・モルガンの定理を多項式表現している。）イデアルの否定の元は与えられたイデアルの生成元に依存して定まる（つまり well-defined でない）が、今回は零点集合についてのみ考えているため深く考慮する必要はない。

また、論理積・論理和に対応するイデアルも考えなければならない。I と J に対応する論理式の論理積に対応するイデアルは、$\langle I, J \rangle$ と表せる。また、I と J に対応する論理式の論理和に対応するイデアルは、$I \cdot J$ と表せる。

以上に注意すると、一般的に有限人の参加者がいる順位関係の問題については、「X は Y より速い」、「同着の順位はいない」、「X は B 着である」という条件とその論理積・論理和・否定のみを取り入れたものに関して以下のアルゴリズムによって自動計算される。（論理和・論理積のケースでは関数を再帰的に呼び出している。）

Function Orderconditionmodel$(\Phi, a, p, x_1, ..., x_n)$
INPUT Φ: pure logical condition, a :maximal number of order,
p :minimum prime which is equal or larger than a, $x_1, ..., x_n$:participants
OUTPUT I: ideal model of Φ
$R \leftarrow \mathbb{F}_p[x_1, ..., x_n]$;
$I \leftarrow \phi$;
IF Φ= "x_i is faster than x_j";
$I \leftarrow \langle 1 \rangle$;
FOR $k = 1$ to $a - 1$
$I \leftarrow I \cdot \langle (x_i - 1)(x_i - 2)...(x_i - k), (x_j - a)(x_j - (a - 1))...(x_j - (k + 1)) \rangle$;
IF Φ= "x_i's order is s";
$I \leftarrow x_i - s$;
IF Φ= "each order is different";
$R \leftarrow R[u]$
$q \leftarrow 1$;
FOR $k = 1$ to $n - 1$
FOR $l = 2$ to n

IF $k \geq l$

BREAK;

$q \leftarrow q(x_k - x_l)$;

$I \leftarrow qu - 1$;

IF Φ contains negation - (N)

$\Phi \leftarrow \neg\Phi$ and go to first;

IF (N) has done

$I \leftarrow \overline{I}$;

IF Φ contains logical product (regarded as $\Phi = \phi_1 \wedge \phi_2 \wedge ... \wedge \phi_q$)

for $k = 1$ to q

$I \leftarrow I$+**Orderconditionmodel**$(\phi_k, a, p, x_1, ..., x_n)$;

IF Φ contains logical union (regarded as $\Phi = \phi_1 \vee \phi_2 \vee ... \vee \phi_q$)

$J \leftarrow \langle 1 \rangle$

for $k = 1$ to q

$J \leftarrow J \cdot$ **Orderconditionmodel**$(\phi_k, a, p, x_1, ..., x_n)$;

$I \leftarrow I + J$

RETURN I;

以下、**GroebnerBasis**(I) で I の辞書式順序によるグレブナー基底を出力するとしたとき、メインのアルゴリズムは以下で表される。

Algorithm Deduction$(\Phi_1, ..., \Phi_l, \Psi_1, ..., \Psi_m, a, p, x_1, ..., x_n)$

INPUT $\Phi_1, ..., \Phi_l, \Psi_1, ..., \Psi_m$: logical condition

OUTPUT whether Ψ_i is always followed by hypothesis or not$(i = 1, ..., m)$

BEGIN;

$R \leftarrow \mathbb{F}_p[x_1, ..., x_n]$;

$I \leftarrow \langle x_1^p - x_1, ..., x_n^p - x_n \rangle$;

FOR $i = 1$ to l

$I \leftarrow I$+ **Orderconditionmodel**$(\Phi_i, a, p, x_1, ..., x_n)$;

$G \leftarrow$**GroebnerBasis**(I);

IF $G = \langle 1 \rangle$

RETURN "$\Psi_1, ..., \Psi_m$ always hold";

FOR $i = 1$ to m

$J_i \leftarrow \overline{\textbf{Orderconditionmodel}(\Psi_i, a, p, x_1, ..., x_n)}$;

$G_i \leftarrow$**GroebnerBasis**$(\langle I, J_i \rangle)$;
IF $G_i = \langle 1 \rangle$
RETURN "Ψ_i is always followed by hypothesis";
ELSE
RETURN "Ψ_i is not always followed by hypothesis";
END;

以降の問題についてもそれぞれ解法を考えるが、**Conditionmodel** については具体的な条件に差異があるものの、条件を多項式表現するという点で **Orderconditionmodel** と本質的なアルゴリズム構造は同一である。また **Deduction** では、本チャプター冒頭で述べた判断推理の解法手順を行っているにすぎない。したがって、以降の問題においてもメインとなるアルゴリズムは基本的に同様である。以上により、残りの問題についてはアルゴリズムの具体的な記述について割愛し、各問題の多項式表現の方法について重点的に記述する。

5.2　命題論理

命題論理

ある集団に、海外旅行の経験をアンケートしたところ、次のア～エのことがわかった。
ア オーストラリアに行ったことがある人は、中国に行ったことがある。
イ メキシコに行ったことがある人は、フランスとベルギーの両方へ行ったことがある。
ウ 中国に行ったことがある人は、ベルギーまたはオーストラリアへ行ったことがある。
エ ドイツに行ったことがない人は、ベルギーに行ったことがない。

このとき確実にいえることはどれか。
1. 中国に行ったことがない人は、メキシコに行ったことがない。
2. ドイツに行ったことがある人は、中国に行ったことがある。
3. フランスに行ったことがない人は、ベルギーに行ったことがない。
4. メキシコに行ったことがある人は、ドイツに行ったことがある。
5. オーストラリアに行ったことがない人は、ベルギーに行ったことがある。

各国に行ったことがあるないについて、値をそれぞれ $1, 0$ と定め、オーストラリアに対応する変数を $x(1)$、以降も同様に定義する。

ここで例えばアの場合、「オーストラリアに行ったことがある人は、中国に行ったことがある。」という条件が正しいことと、「オーストラリアに行ったことがない、または中国に行ったことがある、という人がいる」という条件が正しいことは同値である。（”ならば”の言い換え。）以上に注意すると、計算機代数ソフト singular 上で仮定イデアルは以下のコードで表せる。

country.txt

```
ring r=2,x(1..6),lp;
//x(1):オーストラリア x(2):中国 x(3):メキシコ
//x(4):ベルギー x(5):フランス x(6):ドイツ
ideal hy=
x(1)*(x(2)-1),
x(3)*(x(4)-1)*(x(1)-1),
x(2)*(x(5)*x(4)-1),
(x(6)-1)*x(4),
//仮定アイウエ
x(1)^2-x(1),
x(2)^2-x(2),
x(3)^2-x(3),
x(4)^2-x(4),
x(5)^2-x(5),
x(6)^2-x(6);
//前提
ideal g=groebner(hy);
g
```

結論に対する多項式について考える。例えば「中国に行ったことがない人は、メキシコに行ったことがない。」が確実に導かれるためには、「中国に行ったことがない、かつメキシコに行ったことがある、という人がいる」という事象に解が存在しないことがわかれば良い。

そこで以下のコードを追加する。

country.txt

```
ideal re1=x(2),x(3)-1;
ideal re2=x(6)-1,x(2);
ideal re3=x(5),x(4)-1;
ideal re4=x(3)-1,x(6);
ideal re5=x(1),x(4);
//結論 1 ～ 5
ideal i1=g,re1;
ideal i2=g,re2;
ideal i3=g,re3;
ideal i4=g,re4;
ideal i5=g,re5;
//対応するイデアル
ideal g1=groebner(i1);
ideal g2=groebner(i2);
ideal g3=groebner(i3);
ideal g4=groebner(i4);
ideal g5=groebner(i5);
//そのグレブナー基底
g1;
g2;
g3;
g4;
g5;
//出力
```

結果以下のような出力になる。

```
g1[1]=x(6)+1
g1[2]=x(5)^2+x(5)
g1[3]=x(4)+1
g1[4]=x(3)+1
g1[5]=x(2)
g1[6]=x(1)
```

g2[1]=x(6)+1
g2[2]=x(5)^2+x(5)
g2[3]=x(4)^2+x(4)
g2[4]=x(3)*x(4)+x(3)
g2[5]=x(3)^2+x(3)
g2[6]=x(2)
g2[7]=x(1)
g3[1]=x(6)+1
g3[2]=x(5)
g3[3]=x(4)+1
g3[4]=x(3)^2+x(3)
g3[5]=x(2)
g3[6]=x(1)
g4[1]=1
g5[1]=x(6)^2+x(6)
g5[2]=x(5)^2+x(5)
g5[3]=x(4)
g5[4]=x(3)
g5[5]=x(2)
g5[6]=x(1)

以上により、４番目の条件「メキシコに行ったことがある人は、ドイツに行ったことがある。」が仮定から導かれることがわかる。

5.3　位置関係

位置関係

円形のテーブルに１～８の番号がついた席が番号順に時計並びに並べられている。P,Q,R,S の４人が互いに重複しないよう席に座った。以下の条件があるとき、S が座っている可能性のある席番号をすべて答えよ。
（１）P は 1 に座っている。
（２）Q の真正面には R が座っている。
（３）S は Q の左隣に座っている。（例えば Q が 1 に座っている場合、その左隣は２である。）

この問題に関しては変数の置き方について３つの方法がある。

- 例えば Q を値２としたとき、４番目に Q が座っているという条件を二値論理として $x_{24} = 1$、そうでない場合を $x_{24} = 0$ と置く。全てにおいて同様の条件を多項式表現し $\mathbb{F}_2$ 上の多項式として計算する。。
- テーブルに対して変数を $x_1, ..., x_8$ とし、４人に対応する値をそれぞれ１，２，３，４とおく。誰も座ってない場合値は０と定め、$\mathbb{F}_5$ 上の多項式として計算する。
- ４人に対して変数を $x_1, ..., x_4$ とおく。席に対応する値をそれぞれ１～８とおき、$\mathbb{F}_{11}$ 上の多項式として計算する。

後半の条件になるに従い変数の数が減っていることを確認できる。実験的に最後の置換が一番高速で行われたので、これを採用し話を進める。

（１），（２）、（３）の条件に関しては考えられるケースを多項式表現すればよい。（（２）に関しては簡潔に表現することが可能であるが、$\mathbb{F}_{11}$ 上の計算であることに注意する。）４人が重複しないように座るという条件は差積が０でないということにほかならない。

結論の計算に関しては注意が必要である。前述の例に関しては、結論が「必ず」導かれるかが焦点であったが、今回は導かれる「可能性がある」ものについて考えている。従って今回は結論に対応する多項式を仮定イデアルに付加し、自明なイデアルでないものが現れたら問題の解答となるのである。これに注意して以下のコードを用意する。

table.txt

```
ring r=11,x(1..4),lp;
ideal hy=
(x(1)-1)*(x(1)-2)*(x(1)-3)*(x(1)-4)*(x(1)-5)*(x(1)-6)*(x(1)-7)*(x(1)-8),
(x(2)-1)*(x(2)-2)*(x(2)-3)*(x(2)-4)*(x(2)-5)*(x(2)-6)*(x(2)-7)*(x(2)-8),
(x(3)-1)*(x(3)-2)*(x(3)-3)*(x(3)-4)*(x(3)-5)*(x(3)-6)*(x(3)-7)*(x(3)-8),
(x(4)-1)*(x(4)-2)*(x(4)-3)*(x(4)-4)*(x(4)-5)*(x(4)-6)*(x(4)-7)*(x(4)-8),
((x(1)-x(2))*(x(1)-x(3))*(x(1)-x(4))*(x(2)-x(3))*(x(2)-x(4))*(x(3)-x(4)))^10-1;
ideal i1=x(1)-1;
ideal i2=(x(2)-x(3)-4)*(x(3)-x(2)-4);
ideal i31=x(4)-x(2)-1;
ideal i32=x(4)-1,x(2)-8;
ideal i3=i31*i32;
ideal co=hy,i1,i2,i3;
ideal g1=co,x(4)-1;
ideal g2=co,x(4)-2;
ideal g3=co,x(4)-3;
ideal g4=co,x(4)-4;
ideal g5=co,x(4)-5;
ideal g6=co,x(4)-6;
ideal g7=co,x(4)-7;
ideal g8=co,x(4)-8;
groebner(g1);
groebner(g2);
groebner(g3);
groebner(g4);
groebner(g5);
groebner(g6);
groebner(g7);
groebner(g8);
```

出力は以下のようになる。

_[1]=1
_[1]=1
_[1]=x(4)-3
_[2]=x(3)+5
_[3]=x(2)-2
_[4]=x(1)-1
_[1]=x(4)-4
_[2]=x(3)+4
_[3]=x(2)-3
_[4]=x(1)-1
_[1]=x(4)-5
_[2]=x(3)+3
_[3]=x(2)-4
_[4]=x(1)-1
_[1]=1
_[1]=x(4)+4
_[2]=x(3)-2
_[3]=x(2)+5
_[4]=x(1)-1
_[1]=x(4)+3
_[2]=x(3)-3
_[3]=x(2)+4
_[4]=x(1)-1

従ってＳは３，４，５，７，８番目に座っている可能性があり、１，２，６に座っている可能性はない。

5.4　真偽判定

真偽判定

ある幼稚園で、砂場で遊んでいた A、B、C、D、部屋で遊んでいた E、F、G の 7 人の中に、逆上がりができる子が 2 人いることが分かっている。そこで、A〜G に尋ねたところ、それぞれ以下の発言をした。ただし、7 人のうち、本当のことを言っているのは 2 人だけで、あとの 5 人は間違ったことを言っている。

A：B は逆上がりできるよ。

B：A は間違ったことを言っているよ。

C：A も B も 2 人とも間違っていることを言っているよ。

D：砂場で遊んでいた子の中には逆上がりできる子はいないよ。

E：私は逆上がりができない。

F：逆上がりができるのは 2 人とも砂場で遊んでいた子だよ。

G：E と F の少なくともどっちかは本当のことを言っているよ。

このとき確実にいえることはどれか。

1. 本当のことを言っているのは、1 人が砂場で遊んでいた子であり、1 人は部屋で遊んでいた子である。
2. D は本当のことを言っており、E は間違ったことを言っている。
3. B は逆上がりができ、間違ったことは言っていない。
4. F が逆上がりできるならば、G は逆上がりできない

本ジャンルの問に関して難しいところは、「逆上がりできるできない」、「主張が正しい間違い」の 2 点にそれぞれ二値論理の構造が現れてしまうところにある。
これを解決するために、以下のステップで多項式表現していく。

- $\mathbb{F}_2$ 上で考える。
- それぞれの子に対して変数を $x_1, ..., x_7$ とし、逆上がりができるとき $x_i = 1$、そうでないとき $x_i = 0$ とする。
- それぞれの子に対して主張に対応する多項式を $p_1, ..., p_7$ する。（例えば「A：B は逆上がりできるよ。」という条件に対しては $p_1 = x_2 - 1$ という形で多項式表現す

る。）各 x_i に $0,1$ を代入したとき、p_i は $0,1$ の値を取るが、$p_i = 0$ なら主張は真、$p_i = 1$ なら主張は偽と考える。

$x_2 - 1$ は B が逆上がりできるという条件を多項式表現していて、p_1 は右辺の付値に対応して値が定まるような多項式と考えるのである。この場合、$x_2 = 1$ なら A の主張は正しく、$x_2 = 0$ なら A の主張は間違っているということになる。

このことを踏まえてそれぞれの条件を多項式表現していくが、「7 人のうち、本当のことを言っているのは 2 人だけ」という条件を多項式表現するためには次の補題が必要である。

補題 5.3. $\mathbb{F}_2[x_1, ..., x_n]$ において、x_1〜x_n はそれぞれ 0 か 1 のいずれか一方のみの値を取るとする。また、x_1〜x_n に対して i 次の対称式を $S_i(x_1, ..., x_n)$ とする。このとき、
x_1〜x_n に 1 がちょうど i 個ある
$\iff S_i(x_1, ..., x_n) = 1$ かつ $S_{n-i}(x_1 - 1, ..., x_n - 1) = 1$

Proof. $\Rightarrow$）自明である。
$\Leftarrow$）対偶を考える。x_1〜x_n に 1 が $i-1$ 個以下しかないとき、$S_i(x_1, ..., x_n) = 0$ が成立する。（i 次対称式において必ず 0 に対応する変数が出てきてしまうため。）

また、x_1〜x_n に 1 が $i+1$ 個以上あるときは、x_1〜x_n に 0 が $n-i-1$ 個以下しかないため、$\mathbb{F}_2$ において $x_1 - 1$ から $x_n - 1$ に 1 が $n-i-1$ 個以下しかない。従って $S_{n-i}(x_1 - 1, ..., x_n - 1) = 0$ が成立する。

故に x_1〜x_n のうち 1 であるものの個数が i 個でないならば $S_i(x_1, ..., x_n) = 0$ または $S_{n-i}(x_1 - 1, ..., x_n - 1) = 0$ が成立する。 □

ここで以下のようなコードを用意し問題を解いてみる。（singular では”chern.lib”を入れることで、与えられたリストに対する対称式が生成可能である。）

children.txt

```
LIB ”chern.lib”;
ring r=2,x(1..7),lp;
//x(1)〜x(7)：A〜G
//逆上がりができる：1、できない：0
//0:true 1:false
poly p1=x(2)-1;
poly p2=p1-1;
poly p3=(p1-1)*(p2-1)+(p1-1)+(p2-1);
//p1=0 かつ p2=0
poly p4=(x(1)-1)*(x(2)-1)*(x(3)-1)*(x(4)-1)-1;
poly p5=x(5);
poly p61=
x(1)*(x(2)+x(3)+x(4))+x(2)*(x(3)+x(4))+x(3)*x(4)-1;
poly p62=(x(1)-1)*((x(2)-1)+(x(3)-1)+(x(4)-1))+(x(2)-1)*((x(3)-1)+(x(4)-1))
+(x(3)-1)*(x(4)-1)-1;
poly p6=p61*p62+p61+p62;
//x(1)〜x(4) の間に 1 が 2 つ以上 2 つ以下
poly p7=p5*p6;
list l11=(p1,p2,p3,p4,p5,p6,p7);
list l12=(p1-1,p2-1,p3-1,p4-1,p5-1,p6-1,p7-1);
poly s5=symm(l11)[5]-1;
// 5 次の対称式
poly sn2=symm(l12)[2]-1;
// 2 次の対称式
list l21=(x(1),x(2),x(3),x(4),x(5),x(6),x(7));
list l22=(x(1)-1,x(2)-1,x(3)-1,x(4)-1,x(5)-1,x(6)-1,x(7)-1);
poly s2=symm(l21)[2]-1;
poly sn5=symm(l22)[5]-1;
ideal i=sn2,s5,s2,sn5,x(1)^2-x(1),x(2)^2-x(2),x(3)^2-x(3),x(4)^2-x(4),
x(5)^2-x(5),x(6)^2-x(6),x(7)^2-x(7);
ideal g=groebner(i,”fglm”);
```

```
list l31=(p1,p2,p3,p4);
list l32=(p1-1,p2-1,p3-1,p4-1);
poly ss1=symm(l31)[1]-1;
poly ssn3=symm(l32)[3]-1;
list l41=(p5,p6,p7);
list l42=(p5-1,p6-1,p7-1);
poly sr1=symm(l41)[1]-1;
poly srn2=symm(l42)[2]-1;
ideal re1=(ss1-1)*(ssn3-1)*(sr1-1)*(srn2-1),g;
ideal g1=groebner(re1,"fglm");
g1;
ideal re2=(p4-1)*p5,g;
ideal g2=groebner(re2,"fglm");
g2;
ideal re3=x(2)*(p2-1),g;
ideal g3=groebner(re3,"fglm");
g3;
ideal re4=x(6)-1,x(7)-1,g;
ideal g4=groebner(re4,"fglm");
g4;
```

出力は以下のようになる。

```
g1[1]=x(7)^2+x(7)
g1[2]=x(6)+x(7)+1
g1[3]=x(5)+1
g1[4]=x(4)
g1[5]=x(3)
g1[6]=x(2)
g1[7]=x(1)
g2[1]=1
g3[1]=x(7)^2+x(7)
g3[2]=x(6)+x(7)+1
```

g3[3]=x(5)+1
g3[4]=x(4)
g3[5]=x(3)
g3[6]=x(2)
g3[7]=x(1)
g4[1]=1

従って 2.4 の主張が正しいことがわかる。

上記問題は、複雑とはいえ人間による手計算で短時間内に導出可能である。（例えば、A,B,C の主張から B と C の主張の真偽が一致していないことがわかる。）これは主張が「逆上がりできるか否か」という単純な点に着目していることによるものである。しかし、次に述べる真偽判定の例は必然的に三値の値を必要とし、人間の手計算によっては膨大な時間のかかる問題である。

真偽判定

男 A.B.C.D と女 E.F.G はそれぞれ 0 ～ 2 のカードのうち 1 種類を 1 枚ずつ持っている
このうち、0 のカードを持っているのは 2 人だけであることがわかっている。
7 人が以下のように主張した。
A「B は 1 のカードを持っている」
B「女のうち 1 人だけ 2 のカードを持っている」
C「A も B も嘘をついている」
D「男に 0 のカードを持っている人はいない」
E「私は 1 のカードを持っていない」
F「男で 1 のカードを持っている人は 2 人だけいる」
G「E と F の少なくとも一方は本当のことを言っている」
ただし、本当のことを言っているのはこのうち 2 人のみである。

次のうち、必ず正しい主張はどれか
1　E は 2 のカードを持っている
2　C と D は嘘をついている
3　1 のカードは 3 枚ある
4　G が 1 のカードを持っているならば、D は 2 のカードを持っている
5　A が嘘をついているならば、D は本当のことを言っている
6　2 のカードは 1 枚もない

3 枚のカードがあるため、有限体の枠組みとしては $\mathbb{F}_3$ が妥当であると考えられる。方針を以下に記す。

- それぞれの人に対して変数を $x_1, ..., x_7$ とし、0 を持っているとき $x_i = 0$、1 を持っているとき $x_i = 1$、2 を持っているとき $x_i = 2$ とする。
- それぞれの人に対して主張に対応する多項式を $p_1, ..., p_7$ する。各 x_i に $0, 1, 2$ を代入したとき、p_i は $0, 1, 2$ の値を取るが、$p_i = 0$ なら主張は真、$p_i = 1, 2$ なら主張は偽と考える。

基本的なコードについては前問の真偽判定と同一であるが、「7 人のうち 0 が 2 つ」という条件に関してはやや工夫が必要である。今回は $x = 1, 2$ のとき $x^2 = 1$ であることを利用する。また計算量の観点から多項式は常に被約形にして使用する。

012.txt

```
LIB "chern.lib";
ring r=3,x(1..7),lp;
//0:true 1,2:false
ideal i=x(1)^3-x(1),x(2)^3-x(2),x(3)^3-x(3),x(4)^3-x(4),
x(5)^3-x(5),x(6)^3-x(6),x(7)^3-x(7);

poly p1=x(2)-1;
list v257=((x(5)-2)^2,(x(6)-2)^2,(x(7)-2)^2);
list vn257=(reduce(((x(5)-2)^2-1)^2,i),reduce(((x(6)-2)^2-1)^2,i),
reduce(((x(7)-2)^2-1)^2,i));
poly p21=reduce(symm(v257)[2]-1,i);
poly p22=reduce(symm(vn257)[1]-1,i);
poly p2=reduce(p21^2*p22^2+2*p21^2*p22+2*p21*p22^2+2*p21*p22,i);
//p21=0 かつ p22=0 のときのみ 0　ウカシェヴィッチの三値論理参照
poly p3=reduce(p1^2*p2^2-1,i);
poly p4=x(1)^2*x(2)^2*x(3)^2*x(4)^2-1;
poly p5=(x(5)-1)^2-1;
list v114=((x(1)-1)^2,(x(2)-1)^2,(x(3)-1)^2,(x(4)-1)^2);
list vn114=(reduce(((x(1)-1)^2-1)^2,i),reduce(((x(2)-1)^2-1)^2,i),
reduce(((x(3)-1)^2-1)^2,i),reduce(((x(4)-1)^2-1)^2,i));
poly p61=reduce(symm(v114)[2]-1,i);
poly p62=reduce(symm(vn114)[2]-1,i);
poly p6=reduce(p61^2*p62^2+2*p61^2*p62+2*p61*p62^2+2*p61*p62,i);
poly p7=reduce(p5*p6,i);

list v017=(x(1)^2,x(2)^2,x(3)^2,x(4)^2,x(5)^2,x(6)^2,x(7)^2);
list vn017=(reduce((x(1)^2-1)^2,i),reduce((x(2)^2-1)^2,i),reduce((x(3)^2-1)^2,i),
reduce((x(4)^2-1)^2,i),reduce((x(5)^2-1)^2,i),reduce((x(6)^2-
1)^2,i),reduce((x(7)^2-1)^2,i));
poly hy11=reduce(symm(v017)[5]-1,i);
poly hy12=reduce(symm(vn017)[2]-1,i);
ideal hy1=hy11,hy12;
```

```
list p017=(reduce(p1^2,i),reduce(p2^2,i),reduce(p3^2,i),reduce(p4^2,i),
reduce(p5^2,i),reduce(p6^2,i),reduce(p7^2,i));
list pn017=(reduce((p1^2-1)^2,i),reduce((p2^2-1)^2,i),reduce((p3^2-1)^2,i),
reduce((p4^2-1)^2,i),reduce((p5^2-1)^2,i),reduce((p6^2-1)^2,i),reduce((p7^2-
1)^2,i));
poly hy21=reduce(symm(p017)[5]-1,i);
poly hy22=reduce(symm(pn017)[2]-1,i);
ideal hy2=hy21,hy22;

ideal j=i,hy1,hy2;
ideal g=groebner(j,"fglm");

poly re1=x(5)-2;
ideal g1=re1^2-1,g;
groebner(g1,"fglm");
poly re2=reduce(p3^2*p4^2-1,i);
ideal g2=re2^2-1,g;
groebner(g2,"fglm");
list v117=((x(1)-1)^2,(x(2)-1)^2,(x(3)-1)^2,(x(4)-1)^2,
(x(5)-1)^2,(x(6)-1)^2,(x(7)-1)^2);
list vn117=(reduce(((x(1)-1)^2-1)^2,i),reduce(((x(2)-1)^2-1)^2,i),
reduce(((x(3)-1)^2-1)^2,i),reduce(((x(4)-1)^2-1)^2,i),reduce(((x(5)-1)^2-
1)^2,i),reduce(((x(6)-1)^2-1)^2,i),reduce(((x(7)-1)^2-1)^2,i));
poly re31=reduce(symm(v117)[4]-1,i);
poly re32=reduce(symm(vn117)[3]-1,i);
ideal g3=(reduce(re31^2,i)-1)*(reduce(re32^2,i)-1),g;
groebner(g3,"fglm");
ideal g4=x(7)-1,(x(4)-2)^2-1,g;
groebner(g4,"fglm");
ideal g5=reduce(p1^2-1,i),reduce(p4^2-1,i),g;
groebner(g5,"fglm");
ideal g6=(x(1)-2)*(x(2)-2)*(x(3)-2)*(x(4)-2)*(x(5)-2)*(x(6)-2)*(x(7)-2),g;
groebner(g6,"fglm");
```

最終ページに出力を記述する。(ただし仮定イデアルの生成元は紙数を大幅に食う量であるため、今回はグレブナー基底についてのみ明記する。)

結果的に g5[1]=1 と出力され、他のイデアルについては否定に解が存在した。従って、正しい結論は 5 番目のみである。確かに「A が嘘をついている」という条件を仮定イデアルに追加しグレブナー基底を計算すると、A,B,C,D に関してはそれぞれ 0 以外を持っているという多項式が現れた。

「試合(総当たり戦)」については、真偽判定で用いた対称式の理論を用いることで容易にコードを作成することができるが、具体的なコードについては割愛する。

6 Conclusion

判断推理の問題をアルゴリズム計算するためには、多項式表現するための条件をある程度限定する必要があるが、表現可能なものについては多項式モデルにより自動で計算できることがわかった。一方で枠組みとなる有限体が容易に定まらないような問題に関しては、自動解法の手法化に難があると考えている。(今回の多値論理モデルにおいては変数の個数や取りうる値が確定していることから実現できているが、そうとは限らない問題例も多々ある。)これは与えられた問題に関して、高階の述語論理が数式で表現可能か否かという問題に帰着できると考えている。現段階では、一階述語論理式までは限量記号除去により多項式表現できるため、計算機で実装可能であることが判明している。

また本文中でも触れているが、計算の効率化については課題が多々ある。特に「次数を落とすこと」と「変数を減らすこと」の双方を最も適切に選ぶための条件に関しては十分な研究余地があると考えている。

7 Acknowledgements

本論文の執筆にあたり、毎週のセミナーおよび各種質問に対し丁寧に対応してくださった指導教員の楫先生にこの場を借りて感謝致します。また研究室同期の前田氏を始め、本論文に関しご意見をくださった楫研究室並びに永井研究室の皆様方にも深く感謝しております。ありがとうございました。

参考文献

[1] D.Cox, J.Little and D.Oshea. *Ideals,Varieties and Algorithms*, Springer-Verlag International Switzerland 2015.

[2] D.Cox, J.Little and D.Oshea. *Using Algebraic Geometry*, Second edition, Springer Science+Business Media, Inc 2005.

[3] Zhenyu HUANG. *Parametric Equation Solving and Quantifier Elimination in Finite Fields with The Characteristic Set Method*. Springer-Verlag Berlin Heidelberg 2012.

[4] Sicun Gao, Andre Platzer, and Edmund M. Clarke. *Quantifier Elimination over Finite Fields Using Grobner Bases*. pp. 140–157, Springer-Verlag Berlin Heidelberg 2011.

[5] J. CHAZARAIN, A. RISCOS, J. A. ALONSO, E. BRIALES. *Multi-valued Logic and Grobner Bases with Applications to Modal Logic*. University of Nice, Department of Mathematics, Pare Valrose, 06034 Nice, France University of Sevilla, Faculty of Mathematics, 41012 Seoilla, Spain. April 1988.

[6] 穴井宏和, 横山和弘 QE の計算アルゴリズムとその応用ー数式処理による最適化 東京大学出版会 2011.

[7] 公務員試験数的処理解法テクニック教室の KOMARO https://komaro.net/

[8] SPI 無料学習サイト https://saisokuspi.com/

> <"012.txt";
g[1]=x(7)^3-x(7)
g[2]=x(6)*x(7)^2+x(6)*x(7)+x(7)^2+x(7)
g[3]=x(6)^2-x(6)*x(7)+x(6)+x(7)^2+x(7)
g[4]=x(5)-1
g[5]=x(4)^2*x(7)^2+x(4)^2*x(7)+x(4)*x(7)^2+x(4)*x(7)
g[6]=x(4)^2*x(6)+x(4)^2*x(7)+x(4)^2-x(4)*x(6)*x(7)-x(6)*x(7)-x(6)-x(7)-1
g[7]=x(4)^3-x(4)
g[8]=x(3)*x(4)*x(7)^2+x(3)*x(4)*x(7)
g[9]=x(3)*x(4)*x(6)*x(7)
g[10]=x(3)^2+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)-x(3)*x(6)*x(7)+x(4)^2-
x(4)*x(6)*x(7)+x(6)*x(7)-x(6)-x(7)+1
g[11]=x(2)*x(7)-x(7)
g[12]=x(2)*x(6)-x(6)
g[13]=x(2)*x(4)^2-x(2)-x(4)^2+1
g[14]=x(2)^2-1
g[15]=x(1)*x(7)-x(3)*x(4)^2*x(7)-x(3)*x(6)*x(7)-x(3)*x(7)^2+x(3)*x(7)-x(4)*x(6)*x(7)-
x(4)*x(7)^2+x(4)*x(7)+x(6)*x(7)+x(7)^2+x(7)
g[16]=x(1)*x(6)+x(3)*x(4)^2*x(7)+x(3)*x(4)^2+x(3)*x(6)+x(3)*x(7)^2-
x(3)+x(4)*x(6)*x(7)-x(4)*x(6)+x(4)*x(7)^2+x(4)*x(7)-x(6)*x(7)-x(7)^2-x(7)
g[17]=x(1)*x(4)^2-x(1)-x(3)*x(4)^2-x(3)*x(6)*x(7)+x(3)+x(4)*x(6)*x(7)+x(6)*x(7)
g[18]=x(1)*x(3)-x(1)*x(4)-x(2)*x(3)+x(2)*x(4)+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)-
x(3)*x(6)*x(7)-x(3)*x(6)-x(3)*x(7)-x(4)^2-x(4)*x(6)*x(7)+x(4)*x(6)+x(4)*x(7)+x(6)*x(7)-
x(6)-x(7)+1
g[19]=x(1)*x(2)-x(1)*x(4)-x(2)*x(3)-x(3)*x(4)^2-x(3)*x(4)*x(6)-x(3)*x(4)*x(7)+x(3)*x(4)-
x(3)*x(6)*x(7)+x(3)*x(6)+x(3)*x(7)+x(3)+x(4)^2+x(4)*x(6)*x(7)+x(4)*x(6)+x(4)*x(7)+x(6
)*x(7)-x(6)-x(7)-1
g[20]=x(1)^2-x(3)*x(4)*x(6)-x(3)*x(4)*x(7)+x(3)*x(6)*x(7)+x(4)*x(6)*x(7)-1
g1[1]=x(7)^3-x(7)
g1[2]=x(6)*x(7)^2+x(6)*x(7)+x(7)^2+x(7)
g1[3]=x(6)^2-x(6)*x(7)+x(6)+x(7)^2+x(7)
g1[4]=x(5)-1
g1[5]=x(4)^2*x(7)^2+x(4)^2*x(7)+x(4)*x(7)^2+x(4)*x(7)

g1[6]=x(4)^2*x(6)+x(4)^2*x(7)+x(4)^2-x(4)*x(6)*x(7)-x(6)*x(7)-x(6)-x(7)-1
g1[7]=x(4)^3-x(4)
g1[8]=x(3)*x(4)*x(7)^2+x(3)*x(4)*x(7)
g1[9]=x(3)*x(4)*x(6)*x(7)
g1[10]=x(3)^2+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)-x(3)*x(6)*x(7)+x(4)^2-
x(4)*x(6)*x(7)+x(6)*x(7)-x(6)-x(7)+1
g1[11]=x(2)*x(7)-x(7)
g1[12]=x(2)*x(6)-x(6)
g1[13]=x(2)*x(4)^2-x(2)-x(4)^2+1
g1[14]=x(2)^2-1
g1[15]=x(1)*x(7)-x(3)*x(4)^2*x(7)-x(3)*x(6)*x(7)-x(3)*x(7)^2+x(3)*x(7)-x(4)*x(6)*x(7)-
x(4)*x(7)^2+x(4)*x(7)+x(6)*x(7)+x(7)^2+x(7)
g1[16]=x(1)*x(6)+x(3)*x(4)^2*x(7)+x(3)*x(4)^2+x(3)*x(6)+x(3)*x(7)^2-
x(3)+x(4)*x(6)*x(7)-x(4)*x(6)+x(4)*x(7)^2+x(4)*x(7)-x(6)*x(7)-x(7)^2-x(7)
g1[17]=x(1)*x(4)^2-x(1)-x(3)*x(4)^2-x(3)*x(6)*x(7)+x(3)+x(4)*x(6)*x(7)+x(6)*x(7)
g1[18]=x(1)*x(3)-x(1)*x(4)-x(2)*x(3)+x(2)*x(4)+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)-
x(3)*x(6)*x(7)-x(3)*x(6)-x(3)*x(7)-x(4)^2-x(4)*x(6)*x(7)+x(4)*x(6)+x(4)*x(7)+x(6)*x(7)-
x(6)-x(7)+1
g1[19]=x(1)*x(2)-x(1)*x(4)-x(2)*x(3)-x(3)*x(4)^2-x(3)*x(4)*x(6)-x(3)*x(4)*x(7)+x(3)*x(4)-
x(3)*x(6)*x(7)+x(3)*x(6)+x(3)*x(7)+x(3)+x(4)^2+x(4)*x(6)*x(7)+x(4)*x(6)+x(4)*x(7)+x(6
)*x(7)-x(6)-x(7)-1
g1[20]=x(1)^2-x(3)*x(4)*x(6)-x(3)*x(4)*x(7)+x(3)*x(6)*x(7)+x(4)*x(6)*x(7)-1
g2[1]=x(7)
g2[2]=x(6)
g2[3]=x(5)-1
g2[4]=x(4)^2-1
g2[5]=x(3)^2-1
g2[6]=x(2)^2-1
g2[7]=x(1)*x(3)-x(1)*x(4)-x(2)*x(3)+x(2)*x(4)
g2[8]=x(1)*x(2)-x(1)*x(4)-x(2)*x(3)+x(3)*x(4)
g2[9]=x(1)^2-1
g3[1]=x(7)^2+x(7)
g3[2]=x(6)*x(7)

g3[3]=x(6)^2+x(6)
g3[4]=x(5)-1
g3[5]=x(4)^2*x(6)+x(4)^2*x(7)+x(4)^2-x(6)-x(7)-1
g3[6]=x(4)^3-x(4)
g3[7]=x(3)^2+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)+x(4)^2-x(6)-x(7)+1
g3[8]=x(2)*x(7)-x(7)
g3[9]=x(2)*x(6)-x(6)
g3[10]=x(2)*x(4)^2-x(2)-x(4)^2+1
g3[11]=x(2)^2-1
g3[12]=x(1)*x(7)-x(3)*x(4)^2*x(7)-x(3)*x(7)-x(4)*x(7)
g3[13]=x(1)*x(6)+x(3)*x(4)^2*x(7)+x(3)*x(4)^2+x(3)*x(6)-x(3)*x(7)-x(3)-x(4)*x(6)
g3[14]=x(1)*x(4)^2-x(1)-x(3)*x(4)^2+x(3)
g3[15]=x(1)*x(3)-x(1)*x(4)-x(2)*x(3)+x(2)*x(4)+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)-x(3)*x(6)-
x(3)*x(7)-x(4)^2+x(4)*x(6)+x(4)*x(7)-x(6)-x(7)+1
g3[16]=x(1)*x(2)-x(1)*x(4)-x(2)*x(3)-x(3)*x(4)^2-x(3)*x(4)*x(6)-
x(3)*x(4)*x(7)+x(3)*x(4)+x(3)*x(6)+x(3)*x(7)+x(3)+x(4)^2+x(4)*x(6)+x(4)*x(7)-x(6)-
x(7)-1
g3[17]=x(1)^2-x(3)*x(4)*x(6)-x(3)*x(4)*x(7)-1
g4[1]=x(7)-1
g4[2]=x(6)+1
g4[3]=x(5)-1
g4[4]=x(4)
g4[5]=x(3)^2+x(3)
g4[6]=x(2)-1
g4[7]=x(1)+x(3)+1
g5[1]=1
g6[1]=x(7)^3-x(7)
g6[2]=x(6)*x(7)^2+x(6)*x(7)+x(7)^2+x(7)
g6[3]=x(6)^2-x(6)*x(7)+x(6)+x(7)^2+x(7)
g6[4]=x(5)-1
g6[5]=x(4)^2*x(7)^2+x(4)^2*x(7)+x(4)*x(7)^2+x(4)*x(7)
g6[6]=x(4)^2*x(6)+x(4)^2*x(7)+x(4)^2-x(4)*x(6)*x(7)-x(6)*x(7)-x(6)-x(7)-1
g6[7]=x(4)^3-x(4)

g6[8]=x(3)*x(4)*x(7)^2+x(3)*x(4)*x(7)

g6[9]=x(3)*x(4)*x(6)*x(7)

g6[10]=x(3)^2+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)-x(3)*x(6)*x(7)+x(4)^2-
x(4)*x(6)*x(7)+x(6)*x(7)-x(6)-x(7)+1

g6[11]=x(2)*x(7)-x(7)

g6[12]=x(2)*x(6)-x(6)

g6[13]=x(2)*x(4)^2-x(2)-x(4)^2+1

g6[14]=x(2)^2-1

g6[15]=x(1)*x(7)-x(3)*x(4)^2*x(7)-x(3)*x(6)*x(7)-x(3)*x(7)^2+x(3)*x(7)-x(4)*x(6)*x(7)-
x(4)*x(7)^2+x(4)*x(7)+x(6)*x(7)+x(7)^2+x(7)

g6[16]=x(1)*x(6)+x(3)*x(4)^2*x(7)+x(3)*x(4)^2+x(3)*x(6)+x(3)*x(7)^2-
x(3)+x(4)*x(6)*x(7)-x(4)*x(6)+x(4)*x(7)^2+x(4)*x(7)-x(6)*x(7)-x(7)^2-x(7)

g6[17]=x(1)*x(4)+x(1)+x(2)*x(3)*x(4)+x(2)*x(3)-x(3)*x(4)^2+x(3)*x(6)*x(7)+x(3)-
x(4)^2+x(4)*x(6)+x(4)*x(7)+x(6)+x(7)+1

g6[18]=x(1)*x(3)+x(1)+x(2)*x(3)*x(4)+x(2)*x(4)-
x(3)*x(4)^2+x(3)*x(4)*x(6)+x(3)*x(4)*x(7)-x(3)*x(6)-x(3)*x(7)+x(3)+x(4)^2-
x(4)*x(6)*x(7)-x(4)*x(6)-x(4)*x(7)+x(6)*x(7)-1

g6[19]=x(1)*x(2)+x(1)+x(2)*x(3)*x(4)+x(3)*x(4)^2-x(3)*x(4)*x(6)-
x(3)*x(4)*x(7)+x(3)*x(4)+x(3)*x(6)+x(3)*x(7)-x(3)+x(4)*x(6)*x(7)-x(4)*x(6)-
x(4)*x(7)+x(6)*x(7)

g6[20]=x(1)^2-x(3)*x(4)*x(6)-x(3)*x(4)*x(7)+x(3)*x(6)*x(7)+x(4)*x(6)*x(7)-1