山口大学教育学部研究論叢　第 70 巻　197 頁～ 206 頁　2021 年 1 月

大学入試問題（実数解の個数判定）の解法への Quantifier Elimination の応用について

傍田　裕子 *・北本　卓也

Application of Quantifier Elimination to University Entrance Examination (Counting Real Roots)

SOBATA Yuko*, KITAMOTO Takuya

（Received September 25, 2020）

概 要

計算機の急速な発展と普及に伴い，計算機を教育や学習に活用する様々な試みが行われている。数式処理システムについても価格・使い勝手の両面で個人でも気軽に利用できるものになり，その活用方法が模索されている。本稿では，方程式の実数解の個数に関する大学入試問題を数式処理システムの QE ツールを用いて解く方法について述べる。数式処理システムは Mathematica，QE ツールは Mathematica の組み込みシンボルである Reduce を用いる。
キーワード：数式処理システム，QE，Mathematica

1　はじめに

1.1　QE（Quantifier Elimination）について

数式処理のアルゴリズムである限量子消去（quantifier elimination : QE）は，∃（存在する），∀（すべての）といった限量子がついた一階述語論理式を入力として，それと等価で限量子がない論理式を出力する。問題の内容を一階述語の論理式で書き表すことにより QE を利用して問題を解くことができる。QE の日本語による解説としては [1] が詳しく，同書では大学入試問題の解法も解説されている。一階述語論理の言語には個体の個数について述べる表現力があり，方程式の実数解の個数に関する問題を一階述語論理の言語で表して QE を適用することで問題を解くことができる。

1.2　実数解の個数に関する問題

大学入試問題でよく扱われる題材の一つに，方程式の実数解の個数に関する問題がある。与えられた方程式の実数解の個数を調べたり，実数解が指定された個数になるための条件を調べたりする問題である。人間が手で解く場合は，文字の置き換えを用いて 2 次関数に帰着させたり，微分法を利用して方程式に現れる関数の増減を調べ，グラフを利用して求めるなどするが，本稿では数式処理システムの QE ツールを用いて解く方法について述べる。大学入試では解を表す変数のほかにも変数を含む方程式（すなわち，係数に文字を含む方程式）の解の個数についての問題も出題される。本稿ではこのような方程式も含めて解の個数についての問題を扱う。

QE ツールを適用する数式は代数方程式およびそれに準ずるもの（多項式，有理関数，それらの絶対値などで表された方程式）を基本とし，不等式，非等式も取り扱う。また，本研究では数式処理システムは Mathematica，QE ツールは Mathematica の組み込みシンボルである Reduce コマンドを用いる。

2　一階述語言語での個数の表現

一階述語言語は数字を用いることなく「少なくとも 2 つある」「ちょうど 3 つある」など個体の数量を表す命題を記述することができる（参考文献 [2] ）。

* 放送大学教養学部（情報コース）

2.1　「少なくとも k 個ある」を表す論理式

数式 $\phi(x)$ をみたす x が少なくとも 1 個ある，すなわち「$\phi(x)$ をみたす x が存在する」のは

$$\exists x(\phi(x))$$

で表され,「数式 $\phi(x)$ をみたす x が少なくとも 2 個ある」のは，2 個（例えば x_1 と x_2）存在し，それらが異なるものであることから

$$\exists x_1 \exists x_2(\phi(x_1) \wedge \phi(x_2) \wedge x_1 \fallingdotseq x_2)$$

と表すことができる。

同様に考えると，数式 $\phi(x)$ をみたす x が少なくとも k 個（$k=1,2,\dots$）存在することを表す一階述語論理式を s_k とすると，$\phi(x)$ をみたす x が k 個（例えば $x_1,x_2,\dots,x_k$）存在し，それらが相異なるものであることから，s_k は次のように表現できる。

$$\begin{aligned} s_1 &\equiv \exists x_1(\phi(x_1)) \\ s_2 &\equiv \exists x_1 \exists x_2(\phi(x_1) \wedge \phi(x_2) \wedge x_1 \fallingdotseq x_2) \\ s_3 &\equiv \exists x_1 \exists x_2 \exists x_3(\phi(x_1) \wedge \phi(x_2) \wedge \phi(x_3) \\ &\qquad \wedge x_1 \fallingdotseq x_2 \wedge x_1 \fallingdotseq x_3 \wedge x_2 \fallingdotseq x_3) \\ &\vdots \end{aligned}$$

すなわち，数式 $\phi(x)$ をみたす x が少なくとも k 個存在することは

$$s_k \equiv \exists x_1 \exists x_2 \dots \exists x_k(a_k \wedge b_k)$$
$$\left(\text{ただし}, a_k \equiv \bigwedge_{1\leqq i\leqq k} \phi(x_i),\ b_k \equiv \bigwedge_{1\leqq i<j\leqq k} x_i \fallingdotseq x_j\right) \tag{1}$$

と表される。

2.2　「ちょうど k 個ある」ことを表す論理式

次に,「ちょうど k 個ある」ことを表す論理式を考える。

2.2.1　論理式での表し方（その 1）

「数式 $\phi(x)$ をみたす x がちょうど k 個ある」は,
「数式 $\phi(x)$ をみたす x が少なくとも k 個ある」
（式 (1) の s_k）
かつ「数式 $\phi(x)$ をみたす x はそれ以外にはない」
と同値であるので，これを論理式で表せばよい。

$\phi(x)$ をみたす相異なる k 個の x を $x_1,x_2,\dots,x_k$ とすると,「数式 $\phi(x)$ をみたす x はそれ以外にはない」のは,「$\phi(x)$ をみたす x は $x_1,x_2,\dots,x_k$ のいずれかと一致する」ということである。したがって,「数式 $\phi(x)$ をみたす x がちょうど k 個ある」ことを表す一階述語論理式を t_k とすると，t_k は次のように表現できる。

$$\begin{aligned} t_1 &\equiv \exists x_1(\phi(x_1) \wedge \forall x(\phi(x) \rightarrow x = x_1)) \\ t_2 &\equiv \exists x_1 \exists x_2(\phi(x_1) \wedge \phi(x_2) \wedge x_1 \fallingdotseq x_2 \\ &\qquad \wedge \forall x(\phi(x) \rightarrow (x = x_1 \vee x = x_2))) \\ t_3 &\equiv \exists x_1 \exists x_2 \exists x_3(\phi(x_1) \wedge \phi(x_2) \wedge \phi(x_3) \wedge x_1 \fallingdotseq x_2 \wedge x_1 \fallingdotseq x_3 \\ &\qquad \wedge x_2 \fallingdotseq x_3 \wedge \forall x(\phi(x) \rightarrow (x = x_1 \vee x = x_2 \vee x = x_3))) \\ &\vdots \end{aligned}$$

すなわち，数式 $\phi(x)$ をみたす x がちょうど k 個存在することは

$$t_k \equiv \exists x_1 \exists x_2 \dots \exists x_k(a_k \wedge b_k \wedge \forall x(\phi(x) \rightarrow c_k))$$
$$\left(\text{ただし}, a_k \equiv \bigwedge_{1\leqq i\leqq k} \phi(x_i),\ b_k \equiv \bigwedge_{1\leqq i<j\leqq k} x_i \fallingdotseq x_j,\ c_k \equiv \bigvee_{1\leqq i\leqq k} x = x_i\right) \tag{2}$$

と表される。

2.2.2　論理式での表し方（その 2）

「ちょうど k 個ある」ことは,「少なくとも k 個存在するが，$k+1$ 個は存在しない」と考えこともできる。したがって,「ちょうど k 個ある」ことは，式 (2) のように表す以外にも,「少なくとも k 個存在する」ことを表す論理式を s_k として

$$t_k \equiv s_k \wedge \neg s_{k+1} \tag{3}$$

と表すこともできる。

QE の計算では，小さな問題に分けて処理する方が計算時間が短くなる傾向があり，式 (2) の立式より，上記の式 (3) による立式に QE を適用するほうが計算時間が短くなることが期待できる。実際，Wolfram Cloud [3] での計算では式 (2) による立式を (3) に変えることで計算時間が短くなる問題が多数見られた。このことから，本稿では「ちょうど k 個存在する」を立式する際は式 (3) を用いることにする。

3　QE を用いた「実数解の個数に関する必要十分条件」の計算方法

前節で示したように解の個数の条件は一階述語言語の数式で表せるので，これに QE を適用することで実数解の個数に関する必要十分条件を求めることができる。システムの文法に従って一階述語言語の数式を書き換えたものに QE ツールを適用すればよく，例えば，Mathematica の Reduce 関数で実数係数の方程式の実数解を考える場合は，解についての条件を表す一階述語論理式を e として

```
Reduce[e, Reals]
```

とすることで求める条件が計算できる（「,Reals」とすることですべての変数が実数として扱われる）。

以下では，数式処理システムの QE ツールを用いて実数解の個数に関する問題を解く方法について，解が 1 変数の場合（解が実数の場合）と 2 変数の場合（解が 2 つの実数の組の場合）を述べる。

3.1　1 変数の場合

x の方程式 $f(x)=0$ の実数解の個数を求める場合など解が実数値（2 個以上の実数の組ではなく，1 つの実数）の場合は，前節の論理式をそのまま使うことができるが，$x_1,x_2,\ldots,x_k$ が実数のとき，相異なる k 個の実数 $x_1,x_2,\ldots,x_k$ が存在することと，$x_1<x_2<\cdots<x_k$ となる $x_1,x_2,\ldots,x_k$ が存在することは同じことなので，k 個の解 $x_1,x_2,\ldots,x_k$ が相異なる数であることを示す式 (1) の

$$b_k \equiv \bigwedge_{1\leqq i<j\leqq k} x_i \neq x_j$$

のかわりに

$$x_1<x_2<\cdots<x_k$$

を用いることができる。例えば，x の方程式 $f(x)=0$ が少なくとも 3 つの実数解をもつ条件は，このことと **2.1** の式 (1) より

$$\exists x_1\exists x_2\exists x_3(f(x_1)=0\wedge f(x_2)=0\wedge f(x_3)=0\wedge x_1<x_2<x_3)$$

と表される。

以上より，1 変数の方程式の実数解の個数について，次の**定理 3.1** が成り立つ。

定理 3.1. 数式 $\phi(x)$ をみたす実数 x が少なくとも k 個存在する条件を s_k とすると，s_k（$k=1,2,\ldots$）は

$$s_1\equiv\exists x_1(\phi(x_1)),$$

$$s_k\equiv\exists x_1\exists x_2\ldots\exists x_k(a_k\wedge b_k)\quad(k\geqq 2\text{ のとき})$$

$$\left(\text{ただし}, a_k\equiv\bigwedge_{1\leqq i\leqq k}\phi(x_i),\ b_k\equiv x_1<x_2<\cdots<x_k\right)$$

と表される。

また,「ちょうど k 個存在する」条件については，**定理 3.1** と **2.2** の式 (3) より，次の**定理 3.2** が成り立つ。

定理 3.2. 数式 $\phi(x)$ をみたす実数 x がちょうど k 個（$k=1,2,\ldots$）存在する条件を t_k とすると，t_k は**定理 3.1** の s_k を用いて

$$t_k\equiv s_k\wedge\neg s_{k+1}$$

と表される。

数式 $\phi(x)$ の実数解の個数に関する条件を求める問題を解くには，**定理 3.1**，**定理 3.2** を用いて条件を一階述語論理式に表し，これに QE を適用すればよい。**コード 1** の atl 関数，**コード 2** の exac 関数は，それぞれ，**定理 3.1**，**定理 3.2** を利用して実数解の個数に関する条件を求める関数である。これらを用いると，変数を x，x の満たすべき数式を f として，実数解が少なくとも k 個ある条件は `atl[f,x,k]`，実数解がちょうど k 個ある条件は `exac[f,x,k]` により求められる。

```
atl[phi_ , x_, kosuu_ ] :=
Module [{xL, xkcond , phiL , i, e, f},
If[kosuu == 0, Return [True ]];
xkcond = True;
For[i = 2, i <= kosuu , i++, xkcond = And[
  xkcond , xL[i - 1] < xL[i]]];
phiL = phi /.x ->xL[1];
For[i = 2, i <= kosuu , i++, phiL = And[phiL ,
   phi /. x -> xL[i]]];
f = phiL && xkcond ;
e=Exists [Evaluate[Table[xL[i],{i,kosuu}]], f
  ];
Return[Reduce [e, Reals ]];
]
```

コード 1: atl 関数

```
exac[phi_,x_, kosuu_]:=Module[{atlk1,aklk2,r},
atlk1=atl[phi,x, kosuu];
aklk2=atl[phi,x, kosuu+1];
r=Reduce[atlk1 && Not[aklk2],Reals];
Return[r]
]
```

コード 2: exac 関数

2017 年度弘前大学（前期）理工第 1 問 (2) は，人間が解く場合は $f(x)=\dfrac{1}{x}-\dfrac{1}{x^3}$ または，$f(x)=a$ を同値変形して得た関数 ax^3-x^2+1 の増減を調べて解くが，QE を利用したこれらの関数（atl 関数，exac 関数）を用いて解くことができる。

問題 3.1 (2017 年度弘前大学（前期）理工第 1 問 (2))**.** a を定数とする。次の方程式の異なる実数解の個数を求めよ。

$$\frac{1}{x}-\frac{1}{x^3}=a$$

【解答】 $f(x)=\dfrac{1}{x}-\dfrac{1}{x^3}$ とおくと

$$f(x)=a\Leftrightarrow ax^3-x^2+1=0$$

より $f(x)=a$ は高々3 次の方程式だから，実数解は高々 3 個である。したがって，**コード 2** の exac 関数を用いて解がちょうど 0 個，1 個，2 個，3 個になる条件を求める。

```
f=1/x-1/x^3;
ans=Table[{i,exac[f==a,x,i]},{i,0,3}]
```

$$\left\{\{0,\ \mathrm{False}\},\ \left\{1,\ a<-\frac{2}{3\sqrt{3}}\,\|\,a>\frac{2}{3\sqrt{3}}\right\},\right.$$
$$\left.\left\{2,\ a==0\,\|\,a==-\frac{2}{3\sqrt{3}}\,\|\,a==\frac{2}{3\sqrt{3}}\right\},\ \left\{3,\ -\frac{2}{3\sqrt{3}}<a<0\,\|\,0<a<\frac{2}{3\sqrt{3}}\right\}\right\}$$

図 1: 問題 3.1 の出力

のように入力とすると（$f(x)$ を f とした），出力（図 1）より，実数解の個数は

$a<-\dfrac{2}{3\sqrt{3}}$ または $a>\dfrac{2}{3\sqrt{3}}$ のとき 1 個,

$a=0$ または $a=-\dfrac{2}{3\sqrt{3}}$ または $a=\dfrac{2}{3\sqrt{3}}$ のとき 2 個,

$-\dfrac{2}{3\sqrt{3}}<a<0$ または $0<a<\dfrac{2}{3\sqrt{3}}$ のとき 3 個

とわかる。

（補足） `f==a` のかわりに `a*x^3-x^2+1==0` を用いてもよい（出力は同じ）。また，実数解が高々3 個であることは `atl[f==a,x,4]` を計算することでもわかる（`False` が出力されて 4 個以上の解をもたないとわかる）。

2016 年度山口大学（前期）理系 α 第 1 問は，人間が解く場合は (1)，(2) の誘導によりグラフをかいて調べるが，**コード 2** の exac 関数を用いて直接 (3) を解くことができる。

> **問題 3.2** (2016 年度山口大学（前期）理系 α 第 1 問). 関数 $f(x)=|x^3-3x^2-3x+1|$ について，次の問いに答えなさい。
>
> (1) 方程式 $f(x)=0$ の実数解をすべて求めなさい。
>
> (2) $f(x)$ の増減，極値を調べ，$y=f(x)$ のグラフをかきなさい。ただし，グラフの変曲点と凹凸は調べなくてよい。
>
> (3) a を実数の定数とする。x についての方程式 $f(x)=a$ がちょうど 4 個の異なる実数解をもつように，a の値の範囲を求めなさい。

【解答】 Mathematica の組み込みシンボル Abs（`Abs[z]` で z の絶対値を返す）を用いて

```
f=Abs[x^3-3x^2-3x+1];
ans=exac[f==a,x,4]
```

とすると，求める a の範囲は $4\sqrt{2}-4<a<4\sqrt{2}+4$ とわかる（入出力：図 2）。

（補足） Abs[] を用いない場合は，`f==a` のかわりに，`g=x^3-3x^2-3x+1` として，

```
(g>=0 && g==a) || (g<a && g==-a)
```

を用いればよい。

```
In[16]:=  f = Abs[x^3 - 3 x^2 - 3 x + 1];
          ans = exac[f == a, x, 4]
Out[17]=  -4 + 4 √2 < a < 4 + 4 √2
```

図 2: 問題 3.2 の入出力

3.2　2 変数の場合

x，y についての方程式の実数解など，解が 2 つの実数の組の場合も同様であるが，解が「相異なる」の表し方に注意が必要である。すなわち，x，y の方程式 $\varphi(x,y)$ の 2 つの解 (x_1,y_1)，(x_2,y_2) について

(x_1,y_1) と (x_2,y_2) が同じ解 $\Leftrightarrow x_1=x_2\wedge y_1=y_2$

(x_1,y_1) と (x_2,y_2) が異なる解 $\Leftrightarrow x_1\fallingdotseq x_2\vee y_1\fallingdotseq y_2$

であり，例えば，x，y の方程式 $f(x,y)=0$ が少なくとも 3 つの実数解をもつ条件は，これと **2.1** の式 (1) より

$$\exists x_1\exists y_1\exists x_2\exists y_2\exists x_3\exists y_3(f(x_1,y_1)=0\wedge f(x_2,y_2)=0\wedge$$
$$f(x_3,y_3)=0\wedge(x_1\fallingdotseq x_2\vee y_1\fallingdotseq y_2)\wedge(x_1\fallingdotseq x_3\vee y_1\fallingdotseq y_3)$$
$$\wedge(x_2\fallingdotseq x_3\vee y_2\fallingdotseq y_3))$$

である。

以上より，$\varphi(x,y)$ をみたす x，y の組の個数について，次の**定理 3.3** が成り立つ

> **定理 3.3.** $\varphi(x,y)$ をみたす x，y の組が少なくとも k 個存在することを表す一階述語論理式を u_k とおくと，u_k（$k=1,2,\ldots$）は
>
> $$u_1\equiv\exists x_1\exists y_1(\varphi(x_1,y_1)),$$
> $$u_k\equiv\exists x_1\exists y_1\exists x_2\exists y_2\ldots\exists x_k\exists y_k({a_k}'\wedge{b_k}')$$
> $$(k\geqq 2\ \text{のとき})$$
> $$\left(\text{ただし},\ {a_k}'\equiv\bigwedge_{1\leqq i\leqq k}\varphi(x_i,y_i),\right.$$
> $$\left.{b_k}'\equiv\bigwedge_{1\leqq i<j\leqq k}(x_i\fallingdotseq x_j\vee y_i\fallingdotseq y_j)\right)$$
>
> と表される。

また,「ちょうど k 個存在する」条件については，**定理 3.3** と **2.2** の式 (3) より，次の**定理 3.4** が成り立つ。

定理 3.4. $\varphi(x,y)$ をみたす x，y の組がちょうど k 個（$k=1,2,\ldots$）存在する条件を v_k とすると，v_k は**定理 3.3** の u_k を用いて

$$v_k \equiv u_k \wedge \neg u_{k+1}$$

と表される。

同じ「k 個の相異なる解」であっても，2 変数の方程式の場合は 1 変数の方程式の場合と比べて，変数の個数，式の個数ともに多くなるため，計算量が大きくなる。論理式中の束縛変数の個数は 2 倍となる。また,「相異なること」を調べる部分は，1 変数のときは x_i と x_{i+1}（$i=1,2,\ldots,k-1$）の $k-1$ 組の 2 数の大小を調べればよかったが，2 変数では k 個のうちの 2 つの数の組の選び方が $\dfrac{k(k-1)}{2}$ 通り，さらに各組で最大 2 回調べる必要がある。

コード 3 の atlxy 関数，**コード 4** の exacxy 関数は，それぞれ，**定理 3.3**，**定理 3.4** を利用して実数解の個数に関する条件を求める関数である。これらを用いると，変数を x，y とし，x，y のみたすべき数式を f とすると，f をみたす実数 x，y の組が少なくとも k 個ある条件は `atlxy[f,x,y,k]`，ちょうど k 個ある条件は `exacxy[f,x,y,k]` により求められる。

```
atlxy [phi_ , x_, y_, kosuu_ ] := Module [{xL,
    yL, xykcond , phiL ,i, j, e, f},
If[kosuu == 0, Return [True ]];
xykcond = True;
For[i = 2, i <= kosuu , i++,
For[j = 1, j < i, j++,
xykcond = And[xykcond , xL[j] != xL[i] || yL[j
   ] != yL[i]]
] ];
phiL = phi /. {x->xL[1], y->yL[1]};
For[i = 2, i <= kosuu , i++, phiL = And[phiL ,
    phi /. {x->xL[i], y ->
yL[i]}]];
f = phiL && xykcond ;
e=Exists [Evaluate[Join[Table[xL[i],{i,kosuu
   }],Table[yL[i],{i,kosuu}]]], f];
Return[Reduce [e, Reals ]];
]
```

コード 3: atlxy 関数

```
exacxy[phi_,x_, y_,kosuu_]:=Module[{atlxyk1,
   aklxyk2,r},
atlxyk1=atlxy[phi,x,y, kosuu];
aklxyk2=atlxy[phi,x, y,kosuu+1];
r=Reduce[atlxyk1 && Not[aklxyk2],Reals];
Return[r]
]
```

コード 4: exacxy 関数

これらを用いて，2020 年度東北大学（前期）理系第 2 問を解くことができる。

問題 3.3 (2020 年度東北大学（前期）理系第 2 問)**.** a を 0 でない実数とする。xy 平面において，円 C : $x^2-2ax+y^2-4y+4=0$，直線 L : $-4x+3y+a=0$，直線 M :$3x+4y-7a=0$ を考える。

(1) L と M の交点が C 上にあるような a の値を求めよ。

(2) C と L が異なる 2 つの共有点をもつような a の値の範囲を求めよ。

(3) 集合 $\{\mathrm{P}|$ 点 P は C と L の共有点 $\} \cup \{\mathrm{P}|$ 点 P は C と M の共有点 $\}$ の要素の個数が 3 となるような a の値をすべて求めよ。

【**解答**】円 C，直線 L, 直線 M の方程式

$$\text{円 } C \quad : x^2-2ax+y^2-4y+4=0 \tag{4}$$

$$\text{直線 } L : -4x+3y+a=0 \tag{5}$$

$$\text{直線 } M : 3x+4y-7a=0 \tag{6}$$

について，式 (4)，式 (5)，式 (6) をそれぞれ $c(x,y)$，$l(x,y)$，$m(x,y)$ とおくと，(1) の L と M の交点が C 上にあるのは

$$c(x,y) \wedge l(x,y) \wedge m(x,y)$$

をみたす (x,y) が存在するときであり，(2) の C と L が異なる 2 つの共有点をもつのは

$$c(x,y) \wedge l(x,y)$$

をみたす相異なる (x,y) が 2 組存在するときである。また，(3) は

$$(c(x,y) \wedge l(x,y)) \vee (c(x,y) \wedge m(x,y))$$

をみたす相異なる (x,y) がちょうど 3 組存在するときである。したがって，求める a の範囲は，**コード 5** の入力により（c，l，m はそれぞれ $c(x,y)$，$l(x,y)$，$m(x,y)$ である），(1) $a=1$，(2) $a<-3$ または $a>\dfrac{3}{4}$，(3) $a=-8,\ \dfrac{8}{9},\ 1$ とわかる（入出力：図 3)。

```
c = x^2-2*a*x+y^2-4y+4== 0;
l = -4*x+3*y+a== 0;
m = 3x+4y-7a== 0;
ans1 = atlxy[c&&l&&m,x,y,1]
ans2 = atlxy[c&&l,x,y,2 ]
ans3 = exacxy[(c&&l)||(c&&m),x,y,3 ]
```

コード 5: 問題 3.3 の解答（入力）

```
In[26]:= c = x^2 - 2 * a * x + y^2 - 4 y + 4 == 0;
         l = - 4 * x + 3 * y + a == 0;
         m = 3 x + 4 y - 7 a == 0;

In[29]:= ans1 = atlxy [c && l && m, x, y, 1]

Out[29]= a == 1

In[30]:= ans2 = atlxy [c && l, x, y, 2 ]

Out[30]= a < -3 || a > 3/4

In[31]:= ans3 = exacxy [(c && l) || (c && m), x, y, 3 ]

Out[31]= a == -8 || a == 8/9 || a == 1
```

図 3: 問題 3.3 の入出力

4　三角方程式の解の個数への応用

三角関数のような超越関数は基本的には QE の適用外であり，例えば次の 2013 年東大理系第 2 問のような問題は QE で解くことはできない [4] 。

> **問題 4.1** (2013 年度東京大学（前期）理系第 2 問)**.**
> a を実数とし，$x>0$ で定義された関数 $f(x)$，$g(x)$ を次のように定める。
> $$f(x)=\frac{\cos x}{x},\ g(x)=\sin x+ax$$
> このとき $y=f(x)$ のグラフと $y=g(x)$ のグラフが $x>0$ において共有点をちょうど 3 つもつような a をすべて求めよ。

しかし，処理を工夫することにより QE が適用できる形にできる場合がある。本節では与えられた x の方程式が $\sin x$, $\cos x$ の多項式で表される場合について，QE に適用させるための 2 つの方法を紹介する。

4.1　方法 1 : 2 変数の代数方程式に帰着させる

x の方程式が $\sin x$, $\cos x$ の多項式で表される場合は $\cos x=c$，$\sin x=s$ と置き換えて $c^2+s^2=1$ の条件を付加することで c，s の多項式の問題に帰着させることができ x の範囲が $0\leqq x<2\pi$ の場合（この形でなくても円 1 周分ならよい），x の値と対応する (c,s) の組は 1 対 1 に対応するので，置き換えで得た c，s の方程式の解 (c,s) の組の個数（同じ個数）についての問題に帰着できる。

例えば**問題 4.2** の 2020 年度北海道大学（前期）文系第 2 問 (3) は，人間が解く場合は文字の置き換えで 2 次関数（1 変数）に帰着させるが（(1) で $t=\sin\theta-\cos\theta$ の置き換えの誘導がある），ここでは上記の方法で解く。

> **問題 4.2** (2020 年度北海道大学文系（前期）第 2 問)**.**
> 関数
> $$f(\theta)=\frac{1}{\sqrt{2}}\sin 2\theta-\sin\theta+\cos\theta\quad(0\leqq\theta\leqq\pi)$$
> を考える。
>
> (1)，(2)　略
>
> (3) a を実数の定数とする。$f(\theta)=a$ となる θ がちょうど 2 個であるような a の範囲を求めよ。

【解答】$\sin 2\theta=2\sin\theta\cos\theta$ より

$$f(\theta)=\sqrt{2}\sin\theta\cos\theta-\sin\theta+\cos\theta \tag{7}$$

なので，$\cos\theta=c$，$\sin\theta=s$ とおくと

$$\begin{aligned}&f(\theta)=a\wedge 0\leqq\theta\leqq\pi\\ \Leftrightarrow&\sqrt{2}s\cdot c-s+c=a\wedge c^2+s^2=1\wedge s\geqq 0\end{aligned} \tag{8}$$

となる。したがって，式 (8) をみたす (c,s) の組がちょうど 2 個である条件を求めればよいから，**コード 4** の exacxy 関数を用いて

```
e= Sqrt[2]*s*c-s+c==a && c^2+s^2==1 && s>=0;
exacxy[e,c,s,2]
```

とすると，求める a の範囲は

$$-\frac{3}{\sqrt{2}}<a\leqq -1 \text{ または } 1\leqq a<\frac{3}{2\sqrt{2}}$$

とわかる（入出力：図 4)。

この方法は置き換えの方法は簡単であるが，x の方程式が c，s の 2 変数の方程式になるため，**3.2** で述べたように計算量が増えるのが難点である。

4.2　方法 2 : $\sin\theta$，$\cos\theta$ を有理式で表す

もう 1 つの方法は，$\tan\dfrac{x}{2}=t$ となる実数 t を用いて t の方程式に帰着させる方法である。

x の範囲が $0\leqq x<2\pi$ のとき，$x\fallingdotseq\pi$ であれば

$$\cos x=\frac{1-t^2}{1+t^2},\ \sin x=\frac{2t}{1+t^2} \tag{9}$$

とおけて t の方程式に帰着でき，x（$0\leqq x<\pi$, $\pi<x<2\pi$）と実数 t は 1 対 1 に対応するので，元の方程式の $x=\pi$ 以外の解の個数と置き換えた方程式の解 t の個数は等しい。式 (9) の置き換えで $\sin x$，$\cos x$ の多項式は t の有理関数の形となるが，QE ツールが対応していなければ，$f(x),g(x)$ が x の多項式のとき

$$\frac{f(x)}{g(x)}=0\Leftrightarrow f(x)=0\wedge g(x)\fallingdotseq 0$$

となることを用いて適用可能な形にできる（有理関数のままでも QE が適用できる場合もある）。

ただし，この方法では $x=\pi$ を解に持つ場合を別に調べなければならないので注意が必要である。先ほどの**問**

```
In[39]:= e = Sqrt[2]*s*c - s + c == a && c^2+s^2 == 1 && s ≥ 0;
         exacxy[e, c, s, 2]
Out[40]= -3/√2 < a ≤ -1 || 1 ≤ a < 3/(2 √2)
```

図 4: 問題 4.2 の入出力（方法 1）

```
In[41]:= f = Sqrt[2]*Sin[th]*Cos[th] - Sin[th] + Cos[th];
         fp = f /. th → Pi;
         g = f /. {Cos[th] → (1 - t^2)/(1 + t^2), Sin[th] → 2 t/(1 + t^2)};
         r1 = exac[g == a && t ≥ 0, t, 2] && fp ≠ a;
         r2 = exac[g == a && t ≥ 0, t, 1] && fp == a;
         Reduce[r1 || r2, Reals]
Out[46]= -3/√2 < a ≤ -1 || 1 ≤ a < 3/(2 √2)
```

図 5: 問題 4.2 の入出力（方法 2）

題 **4.2** をこの方法で解くと次のようになる。

【解答】（**問題 4.2** の別解）

式 (7) より $f(\theta)=\sqrt{2}\sin\theta\cos\theta-\sin\theta+\cos\theta$ なので，$\theta\fallingdotseq\pi$ のとき $f(\theta)$ において $\cos\theta=\dfrac{1-t^2}{1+t^2}$，$\sin\theta=\dfrac{2t}{1+t^2}$ とした式を $g(t)$ とおくと

$$g(t)=\sqrt{2}\cdot\frac{2t}{1+t^2}\cdot\frac{1-t^2}{1+t^2}-\frac{2t}{1+t^2}+\frac{1-t^2}{1+t^2}$$

であり

$$f(\theta)=a\ \wedge\ 0\leqq\theta<\pi$$
$$\Leftrightarrow g(t)=a\wedge t\geqq 0$$

となる。よって，$f(\theta)=a$ となる θ がちょうど 2 個であるのは

「$g(t)=a$ となる t（$t\geqq 0$）がちょうど 2 個かつ $f(\pi)\fallingdotseq a$)」

または

「$g(t)=a$ となる t（$t\geqq 0$）がちょうど 1 個かつ $f(\pi)=a$)」

だから，exac 関数（**コード 2**）を用いて次の入力で解くことができる（入出力結果は図 5)。

```
f=Sqrt[2]*Sin[th]*Cos[th]-Sin[th]+Cos[th];
fp=f/.th ->Pi;
g=f/. {Cos[th]->(1-t^2)/(1+t^2),Sin[th]->2t
  /(1+t^2)};
r1 = exac[g==a && t>=0,t,2] && fp!=a;
r2 = exac[g==a && t>=0,t,1] && fp==a;
Reduce[r1 || r2,Reals]
```

この方法では**方法 1** とは異なり 1 変数の方程式となるので計算量の減少が期待できる。ただし，π を解に持つ場合の場合分けが必要であるのと，次数が高くなってしまうという欠点がある。

4.3　方法 1 と方法 2 の計算時間

問題 4.2 では，**方法 1**（2 変数の方程式にする）と**方法 2**（1 変数の方程式にする）のどちらも即座に計算が終わるため計算時間の差はあまり実感できないが，例えば次の**問題 4.3**（2020 年度東京大学（前期）理系第 6 問 (1)）のように解の個数が多い場合は差が大きくなる。

> **問題 4.3** (2020 年度東京大学（前期）理系第 6 問 (1))**.**
> A，α を実数とする。θ の方程式
> $$A\sin 2\theta-\sin(\theta+\alpha)=0$$
> を考える。$A>1$ のとき，この方程式は $0\leqq\theta<2\pi$ の範囲に少なくとも 4 個の解を持つことを示せ。

【解答】問題文の方程式の左辺は

$$A\sin 2\theta-\sin(\theta+\alpha)$$
$$=2A\sin\theta\cos\theta-\sin\theta\cos\alpha-\cos\theta\sin\alpha$$

なので，$\sin\alpha$，$\cos\alpha$ をそれぞれ `sinA`，`cosA`（`sinA`，`cosA` は `(sinA)`^{2}`+(cosA)`^{2}`=1` をみたす定数）として，**方法 1**，**方法 2** で解くことができ（入出力結果はそれぞれ，図 6，図 7)，**方法 1** では約 3 秒かかったが**方法 2** では 0.155 秒で，計算時間がかなり小さくなった（Wolfram Cloud [3] による計算。)。

```
In[90]:= f = 2 A * Sin[th] * Cos[th] - Sin[th] * cosA - Cos[th] * sinA == 0;
         g = f /. {Sin[th] → s, Cos[th] → c};
         Timing [
         e1 = atlxy [g && s ^ 2 + c ^ 2 == 1 && sinA ^ 2 + cosA ^ 2 == 1 && A > 1, c, s, 4];
         Reduce [ForAll [{sinA , cosA , A}, sinA ^ 2 + cosA ^ 2 == 1 && A > 1, e1], Reals ]
         ]
Out[92]= {3.092 , True}
```

図 6: 問題 4.3 の入出力（方法 1）

```
In[93]:= f = 2 A * Sin[th] * Cos[th] - Sin[th] * cosA - Cos[th] * sinA == 0;
         fp = f /. th → Pi;
         h = f /. {Cos[th] → ((1 - t ^ 2) / (1 + t ^ 2)), Sin[th] → (2 t / (1 + t ^ 2))};
         Timing [
         e2 = atl[h && A > 1 && (sinA) ^ 2 + (cosA) ^ 2 == 1, t, 4] ||
             (atl[h && A > 1 && (sinA) ^ 2 + (cosA) ^ 2 == 1, t, 3] && (fp && (sinA) ^ 2 + (cosA) ^ 2 == 1));
         Reduce [ForAll [{A, sinA , cosA }, (sinA) ^ 2 + (cosA) ^ 2 == 1 && A > 1, e2], Reals ]
         ]
Out[96]= {0.155 , True}
```

図 7: 問題 4.3 の入出力（方法 2）

さらに，**方法 2** では分母を払うと t の 4 次式になることから，解が多くて 5 個（t の方程式の分が 4 個と π の分）であることも容易にわかる。

5　入力の工夫

5.1　小問題に分ける

2.2.2 でも少し述べたが，QE は計算時間が大きいという欠点があり，現在最も効率の良い QE アルゴリズムでも，計算量が論理式中の変数の個数に対し一般に 2 重指数である [5] 。計算量を減らす工夫として，変数の個数や式の個数，次数を減らす方法がよく知られているが，ここでは小問題に分ける方法について議論する。

5.2　条件の数式が論理和の形の場合

解 x のみたすべき数式が論理和 $A \vee B$ で表されている場合

$$A \vee B \Leftrightarrow A \vee (\neg A \wedge B)$$

であることと $A \wedge (\neg A \wedge B)$ をみたす x は存在しないことから

$$\begin{aligned} & A \vee B \text{ をみたす } x \text{ がちょうど } k \text{ 個} \\ \Leftrightarrow & \bigvee_{0 \leqq i \leqq k} ((A \text{ をみたす } x \text{ がちょうど } k-i \text{ 個}) \\ & \wedge (\neg A \wedge B \text{ をみたす } x \text{ がちょうど } i \text{ 個}) \end{aligned} \tag{10}$$

とできる（解が複数の実数の組の場合も同様である）。これを利用して小問題に分けて計算すると計算時間が短くなることが期待できる。

実際，**3.2** で扱った**問題 3.3** の (3) を式 (10) を用いて解くと，**3.2** で示したそのまま計算する方法（`exacxy[(c&&l)||(c&&m),x,y,3 ]`）では約 19 秒かかった計算が 1 秒あまりで終わり，計算時間が小さくなった（入出力結果は図 8。前半は **3.2** の方法，後半は式 (10) を利用した方法。**コード 5** の c，l，m および**コード 4** の関数 exacxy を用いる）。

5.3　3 つ以上の論理和で表された条件

3 個以上の式の論理和については，n 個の論理和について

$$\begin{aligned} & \bigvee_{1 \leqq i \leqq n} A_i \\ \Leftrightarrow & A_1 \vee (\neg A_1 \wedge A_2) \vee (\neg A_1 \wedge \neg A_2 \wedge A_3) \vee \ldots \\ & \cdots \vee (\neg A_1 \wedge \neg A_2 \cdots \wedge \neg A_{n-1} \wedge A_n) \end{aligned}$$

により排反な論理和に分けられるのでこれを利用して小問題に分けられる。しかし，単純にすべて分割すると分割の個数が多くなるのでさらに工夫する必要がある。

```
In[83]:= ans3 = exacxy [(c && l) || (c && m), x, y, 3 ] // Timing
Out[83]= {19.715 , a == -8 || a == 8/9 || a == 1}

In[84]:= f1 = c && l; f2 = c && m; kosuu = 3;
         Timing [
         g = Not[f1] && f2;
         e = False ;
         For[i = 0, i ≤ kosuu , i++,
            e = Or[e , Reduce [exacxy [f1 , x, y, kosuu - i] && exacxy [g, x, y, i], Reals ]]];
         Reduce [e , Reals ]
         ]
Out[85]= {1.108 , a == -8 || a == 8/9 || a == 1}
```

図 8: 問題 3.3 (3) の入出力（**3.2** の方法と **5.2** の方法）

6　解の個数を求める他の手法について

6.1　Mathematica の CountRoots

Mathematica の組み込みシンボル CountRoots は，x の多項式 f について `CountRoots[f,x]` とすると f=0 の実数解の個数を返すが，重複度も含めて数えるので（例えば `CountRoots[(x-1)^2,x]` とすると,「2」が出力される） [6] ，相異なる実数解の個数が必要な場面ではこれだけでは必要な値が得られない。また，文字係数にも対応していないので，係数に文字 a を含む方程式が与えられて「相異なる実数解を 2 個持つ場合の a の条件を求めよ」というタイプの問題には対応できない.

6.2　スツルムの定理

1 変数の代数方程式については，実数解の個数（重解も 1 個と数える）の数え上げについてスツルム（Sturm）の定理が知られている。スツルム列と呼ばれる多項式列について，解の個数を調べたい範囲の端点での値の符号変化を調べることで実数解の個数を求める。スツルム列およびスツルムの定理についての詳しい説明や証明については [7]， [8] などを参照されたい。係数が有理数（係数に文字を含まない）の代数方程式の場合には，スツルムの定理のアルゴリズムを数式処理システムに実装することで実数解の個数を求めることも可能であるが，そのまま実装したのでは係数の有理数が複雑になって計算量が大きくなり，実用的ではない。

7　まとめ

本稿で与えた定理と数式処理システムの QE ツールを活用して個数に関する必要十分条件を求める問題を解くことができた。実数解が「存在する」ということは論理式で容易に表現できるが，個数を含めると難しくなる。特に「ちょうど k 個ある」ことは論理式も複雑になり，計算量も大きくなるが，本稿では,「ちょうど k 個ある」を「少なくとも k 個存在し，かつ，k+1 個以上は存在しない」に読み替えることで，論理式を簡明にし，短い時間で計算結果を得ることができた。本稿で述べた方法は，簡単な入力で確認できるので大学入試問題の解答の確認などにも役立てることができる。

個数に関する問題を表した一階述語論理式からもわかるように，個数を知る，つまり「数える」ためには，何を数えるか（みたすべき条件，数える変数の種類）を明確にした上で，条件をみたすものについて，それらが「同じ」かどうかを判定する必要があるが，1 変数の方程式の実数解と 2 変数（2 つの実数の組が解）の方程式の実数解とで,「同じ/異なる」の判定の部分の立式が異なり，前者のほうがは後者のほうが数式の個数も少なく，計算時間も短くなる傾向があることを見た。実数には大小関係があり，それを活用した立式ができるためである。一列に並べることができると数えやすいのは人間にとっても同じなのは興味深い。

「数える」のような基本的な活動も含めて，問題を解く上で人間が何をしているかに注目することは数式処理システムを利用する際の立式や計算速度向上のヒントになると思われる。また，逆に，数式処理システムでの計算のための工夫が，人間が問題を考える上でのヒントになったり，人間がしている処理の仕方や問題のとらえ方を明らかにすることにつながることも期待できる。数式

処理システムを，ただの便利な道具でなく，新たな発見のためのパートナーにする方法を模索していきたい。

参考文献

[1] 穴井宏和, 横山和弘（2011）『QE の計算アルゴリズムとその応用 』東京大学出版会

[2] 加藤浩，土屋俊（2014）『記号論理学』放送大学教材

[3] WOLFRAM CLOUD,
https://www.wolframcloud.com/

[4] 新井紀子，東中竜一郎（2018）『人工知能プロジェクト「ロボットは東大に入れるか」第三次 AI ブームの到達点と限界』

[5] 穴井宏和（2020）「数式・記号処理から見た AI の原理」『数学文化』No.34（2020 年 8 月 30 日発行），pp.34-45，日本評論社

[6] Wolfram 言語＆システムドキュメントセンター
組込みシンボル CountRoots，
https://reference.wolfram.com/language/ref/CountRoots.html

[7] 高木貞治（1965 年）『代数学講義　改訂新版』共立出版

[8] 長坂耕作, 岩根秀直, 北本卓也, 讃岐勝, 照井章, 鍋島克輔（2019）『計算機代数の基礎理論』共立出版