平成15年度卒業論文

DeBruijn系列を係数とする多項式の原始性判定とMathmaticaによる探索

理学部数学科
1271006F 大塚祐樹

平成16年2月10日

1　はじめに

この論文の目的は符号理論に利用される DeBruijn 系列を係数とする多項式がどのくらいの割合で原始多項式になっているか調べることにある. まず DeBruijn 系列の定義を与え, その存在を証明する. 次に原始多項式を定義し,DeBruijn 系列を係数とする多項式が原始多項式かどうか数式処理ソフト Mathmatica をもちいて調べてゆく.

最後にこの論文を書くにあたって御指導頂いた松本眞教授に感謝の意を表したい.

2 DeBruijn系列

定義 1 (DeBruijn 系列). X を有限集合,$q = \#(X)$,n を自然数とする. X 上の n 次 *DeBruijn* 系列とは, 周期 q^n の X 上の数列であって連続する n 個を見ると周期の中でどのパターンもちょうど一つずつでているもの.

例. $1110100011101000\cdots$ は $\{0,1\}$ 上の *3* 次 *DeBruijn* 系列である.

定理 1. 任意の自然数 q,n に対して *DeBruijn* 系列は存在する.

この定理を証明するためにまずグラフの一筆書きに関する次の定理を示す.

定理 2. 有向グラフ G にすべての辺をただ一度だけとおりもとの頂点に戻る道が存在するための必要十分条件は, 連結であり各頂点において出ていく辺と, 入ってくる辺の数が等しいことである.

証明. 必要性は明らか. 十分性を示す. 適当な頂点から始めて辺を一度だけ通る道を考えると, 各頂点で出て行く辺と入ってくる辺の数が等しいので次の頂点が始点と異なれば必ずその頂点から出て行く辺は残っている. だからもし, ある頂点にたどり着いたとき出て行く辺が残ってなければ, その点は始点である. そうした道で通る辺の数が最大になるものを T とする. これがすべての辺を尽くしていたら, それは一筆書きである. そこで T に含まれない辺が存在すると仮定する.G から T に含まれている辺を除いたグラフを G' とすると, 一つの頂点から出ていく辺と, 入ってくる辺は同じ数だけ取り除かれるので,G' おいても出ていく辺と, 入ってくる辺の数が等しくなる. だから G' の適当な頂点から出て G' の辺だけを通り, 同じ辺を通ることなく戻ってくる道が存在する. また,G は連結なので G' の辺でその両端の頂点のうち少なくともひとつは T に含まれるようなものが存在する. その T に含まれる頂点から出て G' の辺だけを通り, 同じ辺を通ることなく戻ってくる道を T' とする.T と T' は、共通する頂点を持つのでまず T の道をその頂点までたどりそこから T' の道をたどりその後 T の残りをたどる道を考えるとこれは G の辺を二度とることはなく T の辺の数より多くの辺を持つこととなり T は辺の数が最大に矛盾している. よって T に含まれない辺は存在しないのですべての辺をただ一度だけとおりもとの頂点に戻る道が存在する. □

証明. $X = \{a_1, a_2, \cdots, a_q\}$, とする. このとき次のような有向グラフを考える. 頂点集合は $X^n = \{(x_1, x_2, \cdots, x_n) | x_i \in X, 0 \leq i \leq n\}$ とし, 辺は任意の頂点 $X^n \ni x = (x_1, x_2, \cdots, x_n)$ から $(x_2, x_3, \cdots, x_n, a_i) \in X^n$ $(1 \leq i \leq q)$ の q 個の点にでているとする. 頂点の数は q^n, 辺の数は $q^n \times q = q^{n+1}$ である. このグラフは連結である. なぜなら任意の 2 点 $X^n \ni x = (x_1, x_2, \cdots, x_n), y = (y_1, y_2, \cdots, y_n)$ に対して $x = (x_1, x_2, \cdots, x_n) \to (x_2, x_3, \cdots, x_n, y_1) \to \cdots \to (x_n, y_1, \cdots, y_{n-1}) \to (y_1, y_2, \cdots, y_n) = y$ とできるからである. また x に入る辺を出す点は $(a_i, x_1, x_2, \cdots, x_{n-1})$ $(1 \leq i \leq q)$ の q 個である. よってこのグラフは連結であり, 各頂点に出入りする辺の数が等しいので全ての辺を一度だけ通って始点に戻ってくることができる. そのような道をひとつ選び, その頂点を順に並べたものを $z_1, z_2, \cdots, z_{q^{n+1}}$ とし、X 上の周期 q^{n+1} の数列 $\{b_k\}$ を $1 \leq i \leq q^{n+1}$ のとき,b_i は z_i の第一成分であるようなものとする.$1 \leq j \leq q^{n+1}, i \neq j$ のとき $(b_i, b_{i+1}, \cdots, b_{i+n+1}) = (b_j, b_{j+1}, \cdots, b_{j+n+1})$ と仮定すると $z_i = z_j, z_{i+1} = z_{j+1}$ となり同じ辺は一度しか通らないことに矛盾する. よって $(b_i, b_{i+1}, \cdots, b_{i+n+1}) \neq (b_j, b_{j+1}, \cdots, b_{j+n+1})$ となるので $\{b_k\}$ は X 上の n+1 次 DeBruijn 系列である.1 次 DeBruijn 系列の存在は自明なので定理は示された. □

DeBruijn 系列の数については次の公式が知られている.

定理 3. 位数 q の集合 X 上の n 次 *DeBruijn* 系列の数は $((q-1)!)^{q^{n-1}} q^{q^{n-1}-n}$ で与えられる.

3　原始多項式

定理 4. K を位数 q の有限体,$f(t)$ を K 上の n 次既約多項式とし $K[t]$ の　$\bmod f(t)$ 剰余類全体

$$K_f = K[t]/(f(t))$$

を考えると K_f は位数 q^n の体である.

証明. K_f が体であることを示すには 0 以外の元に逆元が存在することを示せばよい. $g(t)$ を $f(t)$ の倍数でない多項式とすると $f(t)$ が既約なことより $\gcd(f(t), g(t)) = 1$ なので $h(t)g(t) \equiv 1 \pmod{f(t)}$ である $h(t)$ が存在する. よって K_f は体. 代表元として $n-1$ 次以下の多項式すべてが取れるので位数は q^n である. □

定理 5. 位数 q の体の乗法群は位数 $q-1$ の巡回群である.

補題 1. ϕ をオイラー関数,n を自然数とするとき

$$n = \sum_{d|n} \phi(d)$$

が成り立つ.

証明. d が n の約数であるとき巡回群 $\mathbb{Z}/n\mathbb{Z}$ はただひとつの部分群 C_d をもつ.Φ_d を C_d の生成元からなる集合とすると,$\mathbb{Z}/n\mathbb{Z}$ の任意の元はある C_d の生成元となるから

$$n = \#(\mathbb{Z}/n\mathbb{Z}) = \sum_{d|n} \#(\Phi_d) = \sum_{d|n} \phi(d)$$

□

補題 2. H を位数 n の有限群とする.n の任意の約数 d に対して H の元 x で $x^d = 1$ を満たすものが高々 d 個しかなければ H は巡回群である.

証明. d を n の約数とする. もし, 位数 d の $x \in H$ があれば x によって生成される部分群 $\langle x \rangle = \{1, x, \cdots, x^{d-1}\}$ は位数 d の巡回群である. 仮定により $y \in H$ は $y^d = 1$ を満たせば $\langle x \rangle$ に属する. 特に y の位数が d ならば y は $\langle x \rangle$ の生成元であり, その逆も成り立つ.$\langle x \rangle$ の生成元の個数は $\phi(d)$ であるので H の位数 d の元の個数 r_d は 0 または $\phi(d)$ に等しい. ところが, ある n の約数 d について $r_d = 0$ ならば $n = \sum_{d|n} \phi(d)$ より H の元の個数は n より小さくなり仮定に反する. 特に $d = n$ とおけば,H の位数 n の元 x があり H は巡回群となる □

定理 5 の証明. H を位数 q の体の乗法群,$n = q-1$ とすると $x^d = 1$ が高々 d 個の根しか持たないので補題 2 より H は巡回群となる. □

定義 2 (原始多項式). K を位数 q の有限体とする.$f(t)$ を K 上の m 次既約多項式とする. 乗法群 $(K[t]/f(t))^*$ は上記定理により巡回群であるが t がその生成元となるとき $f(t)$ は原始多項式であるという.

定理 6. K 上の m 次既約多項式 $f(t) \in K[t]$ が原始多項式であるための必要十分条件は,$q^m - 1$ のすべての素因数 p に対して $t^{(q^m-1)/p} \not\equiv 1 \pmod{\phi(t)}$ となることである.

証明. 定義より $f(t)$ が原始多項式 $\Longleftrightarrow 1 \leq i \leq q^m - 2$ に対して,$t^i \not\equiv 1 \pmod{f(t)}$ である.
しかし元の位数は群の位数の約数なので i が $q^m - 1$ の約数でなければ $t^i \not\equiv 1 \pmod{f(t)}$.
よって,$f(t)$ が原始多項式 $\Longleftrightarrow i$ を $q^m - 1$ の約数とするとき,$t^i \not\equiv 1 \pmod{f(t)}$ である.
さらに $t^i \equiv 1 \pmod{f(t)}$ ならば,i' を i の倍数としたとき $t^{i'} \equiv 1 \pmod{f(t)}$ なので i' について考えれば十分なので,
$f(t)$ が原始多項式 $\Longleftrightarrow q^m - 1$ のすべての素因数 p に対して $t^{(q^m-1)/p} \not\equiv 1 \pmod{f(t)}$. □

4 DeBruijn 系列を係数とする多項式

2 元体 $\mathbb{F}_2 = \{0, 1\}$ 上の n 次 DeBruijn 系列を考える. 定義より n 次 DeBruijn 系列は必ず 0 が n 個並ぶところがあり,0 が n+1 個並ぶところはない. だから n 次 DeBruijn 系列として 1 で始まり周期の終わりの n 個が 0 でのものを考えれば十分である. $\{a_k\}$ を,n 次 DeBruijn 系列とするときに次のような多項式を考える.

$$f(t) = a_1 + a_2 t + a_3 t^2 + \cdots + a_{2^n-n} t^{2^n-n-1}.$$

これが原始多項式になるか調べたい. しかし $n = 1$ のときは,$f(t) = 1$ となりあまり意味をなさない.$n \geq 2$ のときは一周期 2^n 個の中で 0 と 1 の数は等しいので 1 の個数は 2^{n-1} 個であり,$n \geq 2$ よりこれは偶数なので $f(1) = 0$ となり $f(t)$ は因数 $(t + 1)$ を持つことになり既約でない. よって原始多項式にならない. 1 の個数が偶数であるために既約でないので 1 を一つ減らしてみることを考える. これは先ほどのグラフから辺をひとつ取り除くことになるので, 一筆書きができるようにするためには, 自分自身に戻る辺を抜くしかない. だから 1 が n 個並ぶところから 1 をひとつ取り除くこととなる. こうしてできた周期 $2^n - 1$ の数列を改めて $\{a_k\}$ とし

$$f(t) = a_1 + a_2 t + a_3 t^2 + \cdots + a_{2^n-n-1} t^{2^n-n-2}.$$

が原始多項式か調べる.
3 次 DeBruijn 系列は 11101000,10111000 の二つがあり,$1 + t + t^3, 1 + t^2 + t^3$ の多項式が得られる. これはどちらも既約である.$2^3 - 1 = 7$ なので原始多項式である. 4 次と 5 次について mathematica をもちいてすべてを探したところ 4 次は 16 個の DeBruijn 系列のうち 2 個が既約でかつ原始的であり,5 次は 2048 個の DeBruijn 系列のうち 332 個が既約でそのうち 316 個が原始的であった. 6,7 次はすべてを調べるには時間が膨大に必要なのでランダムに 5000 個だけ調べたところ 6 次は 347 個が既約でそのうち 160 個が原始的であり,7 次は既約が 150 個でそのうち 150 が原始的であった.DeBruijn 系列が原始的である割合は,3 次から順に 1,0.125,0.154,0,032,0.03 となっている. また, 既約なものが原始的な割合は,3 次から順に 1,1,0.95,0.46,1 となっている.6 次のときだけ既約なものが原始的な割合がこうも違うのは興味深いことである.

5 Mathematica によるプログラミング

Mathematica で使用したプログラムを以下に示す.

```
In[1]:=
<<DiscreteMath‘Combinatorica‘
```

```
In[2]:=
<<Algebra‘PolynomialPowerMod‘

In[3]:=
alldb[n_]:=
  Module[{a=Table[{1,1},{2^(n-1)}],
   adj=ToAdjacencyLists[DeBruijnGraph[2,n-1]],li={},sli={}},
   a[[1,1]]=0;a[[2^(n-1),2]]=0;
    v[k_]:=(li=Append[li,k];
        If[Length[li]==2 2^(n-1)-2,sli=Append[sli,li],If[
            a[[k,1]]==1&&a[[k,2]]==1,(a[[k,1]]=0;v[adj[[k,1]]];
              li=Drop[li,-1];a[[k,1]]=1;a[[k,2]]=0;v[adj[[k,2]]];
              li=Drop[li,-1];a[[k,2]]=1),
            If[a[[k,1]]==1&&a[[k,2]]==0,(a[[k,1]]=0;v[adj[[k,1]]];
                li=Drop[li,-1];a[[k,1]]=1),
              If[a[[k,1]]==0&&a[[k,2]]==1,(a[[k,2]]=0;
                  v[adj[[k,2]]];li=Drop[li,-1];a[[k,2]]=1)]]]]);v[1];
    Map[RotateLeft,Mod[sli+1,2]]]

In[4]:=
rc[x_]:=x[[Random[Integer,Length[x]-1]+1]]

In[5]:=
ro[x_]:=Module[{li=x,li2=Rest[x]},
    Do[While[Intersection[Join[Flatten[Table[li[[m]],{m,1,k}]]],
            li2[[1]]]=={},li2=RotateLeft[li2]];li[[k+1]]=li2[[1]];
      li2=Rest[li2],{k,1,Length[x]-1}];li]

In[6]:=
dbs[x_]:=Module[{li=x},
    Do[While[FreeQ[li[[1]],First[li[[k]]]],li[[k]]=RotateLeft[li[[k]]]];
      li[[1]]=Flatten[
          Insert[li[[1]],li[[k]],Position[li[[1]],First[li[[k]]]]]],{k,2,
        Length[x]}];Mod[li[[1]]+1,2]]

In[7]:=
rdb[n_]:=Module[{ac0,ac1,c,a=Table[{1,1},{2^(n-1)}],
      adj=ToAdjacencyLists[DeBruijnGraph[2,n-1]],li={},sli={},k=1},a[[1,1]]=0;
```

```
    a[[2^(n-1),2]]=0;
    Do[If[a[[k,1]]==1||a[[k,2]]\[Equal]1,li=Append[li,k];
        Which[a[[k,1]]==1&&a[[k,2]]==1,c=Random[Integer];
          a[[k,c+1]]=0;k=adj[[k,c+1]],a[[k,1]]==1&&a[[k,2]]==0,
          a[[k,1]]=0;k=adj[[k,1]],a[[k,1]]==0&&a[[k,2]]==1,
          a[[k,2]]=0;k=adj[[k,2]]],sli=Append[sli,li];ac0=Count[a,{1,0}];
        ac1=Count[a,{0,1}];
        If[ac0+ac1!=0,c=Random[Integer,ac0+ac1-1];
          If[c<ac0,l=rc[Position[a,{1,0}]][[1]];li={l};a[[l,1]]=0;
            k=adj[[l,1]],l=rc[Position[a,{0,1}]][[1]];li={l};a[[l,2]]=0;
            k=adj[[l,2]];]]],{2^n-1}];RotateLeft[dbs[ro[sli]]]]

In[8]:=
dblist[n_,m_]:=
  Module[{li={},s},
    While[Length[li]!=m,s=rdb[n];If[FreeQ[li,s],li=Append[li,s]]];li]

In[9]:=
pl[x_]:=(a=x;b=Flatten[a.Table[{t^k},{k,0,Length[a[[1]]]-1}]];
    c=Select[b,Length[FactorList[#,Modulus->2]]==2&];
    d=Sort[(2^Exponent[b[[1]],t]-1)/
          Map[First,FactorInteger[2^Exponent[b[[1]],t]-1]]];i=0;
    Do[Do[If[PolynomialPowerMod[t,d[[k]],{c[[l]],2}]==1,++i;Break[]],{k,
          1,Length[d]}],{l,1,Length[c]}];{Length[a],Length[c],Length[c]-i,i})
In[10]:=
pl[alldb[3]]

Out[10]=
{2,2,2,0}

In[11]:=
pl[alldb[4]]

Out[11]=
{16,2,2,0}

In[12]:=
pl[alldb[5]]
```

```
Out[12]=
{2048,332,316,16}

In[13]:=
pl[dblist[6,5000]]

Out[13]=
{5000,347,160,187}

In[14]:=
pl[dblist[7,5000]]

Out[14]=
{5000,150,150,0}
```