社団法人 情報処理学会　研究報告
IPSJ SIG Technical Report

2004－AL－93　(12)
2004／1／30

対称性を考慮した整数分配のグレイコード

菊地 洋右* 来嶋 秀治*

概要　組合せグレイコードとはグレイコードの拡張である。本論文では対称性を考慮した整数分配に対する組み合わせグレイコードについて述べる。本論文での対称性を考慮した整数分配に対するグレイコード生成アルゴリズムはコード一つ当たり平均 O(1) で出力するアルゴリズムである。

A Gray code for integer compositions with mirror images identified

Yosuke Kikuchi* and Shuji Kijima*

abstract　A combinatorial Gray code is an extension of Gray code that a list with the minimum change successive elements. In this paper, we design an algorithm to generate the combinatorial Gray code for integer compositions with mirror images identified. The average running time of the algorithm is O(1) per element.

1　はじめに

通常のグレイコードは 2 進 n 桁の文字列を 1 ビットずつ異なるようにして、列挙する列挙アルゴリズムとして考えることができる。デジタル回路で使われる 2 進反転符号はグレイコードの代表例である。2 進反転符号以外にもいくつかの構成方法が提案されている [9, 17, 14]。構成法以外にもグレイコードについて様々な研究がなされている [18, 3]。

組合せグレイコードは、グレイコードを一般化したものであり、列挙される集合に対して、連続する要素を “最小の差異” で列挙したものである [13, 19]。例えば整数の分配 [6, 11]、整数の分割 [10, 12]、置換 [15]、組合せ [4]、グラフの k-分木 [1, 7, 20] など [16] に対する組合せグレイコードが存在し、その生成アルゴリズムが提案されている。例として整数の分配、分割の問題について述べよう [8]。

整数 n の d 部への分配とは、和が n となる d 項からなる非負整数列 $(x_1, \ldots, x_d)$ である。

一方、整数の分割とは整数 n を $x_1 + x_2 + \cdots + x_d = n$ と $x_1 \geq x_2 \geq \cdots \geq x_d$ を満たす非負整数列 $(x_1, \ldots, x_d)$ である。

整数の分配、分割に対してはそれぞれグレイコードが生成できる。これらのグレイコードの生成においての “最小の差異” とは連続する整数列においてちょうど二つの桁の値が 1 ずつ異なることである。整数の分配に対するグレイコードは可能かどうかは Knuth によって解かれ、Klingsberg によって分配一つあたり平均定数時間の生成アルゴリズムが与えられた [6]。下の例はその生成アルゴリズムによって得られる整数 6 の 3 部への分配に対するグレイコードである。

$(6,0,0) \rightarrow (5,1,0) \rightarrow (4,2,0) \rightarrow (3,3,0) \rightarrow (2,4,0) \rightarrow (1,5,0) \rightarrow (0,6,0) \rightarrow (0,5,1) \rightarrow (1,4,1) \rightarrow (2,3,1) \rightarrow (3,2,1) \rightarrow (4,1,1) \rightarrow (5,0,1) \rightarrow (4,0,2) \rightarrow (3,1,2) \rightarrow (2,2,2) \rightarrow (1,3,2) \rightarrow (0,4,2) \rightarrow (0,3,3) \rightarrow (1,2,3) \rightarrow (2,1,3) \rightarrow (3,0,3) \rightarrow (2,0,4) \rightarrow (1,1,4) \rightarrow (0,2,4) \rightarrow (0,1,5) \rightarrow (1,0,5) \rightarrow (0,0,6)$

整数の分割は Savage[12] によって解かれ、アルゴリズムが与えられた。さらに、Beer[2] はそのアルゴリズムを改良し、分割一つあたり平均

*東京大学大学院情報理工学系研究科数理情報学専攻
Department of Mathematical Informatics, Graduate School of Information Science and Technology, The University of Tokyo.
{ykikuchi, kijima}@simplex.t.u-tokyo.ac.jp

定数時間の生成アルゴリズムを設計した。

下の例は整数 6 の 3 部への分割に対するグレイコードである。

$(6,0,0) \to (5,1,0) \to (4,2,0) \to (3,3,0) \to (3,2,1) \to (4,1,1) \to (3,1,2) \to (2,2,2)$

先に挙げたグレイコードの他に、グラフに対するグレイコードも提案されている。これはグラフを文字列として表現し、その文字列をグレイコードとして生成するものである。

このように様々な対象に対するグレイコードが提案される一方、グレイコードを統一的に扱う手法も研究されている。列挙する対象を頂点とし、グレイコードとして連続して列挙できるものどうしを辺で結びできるグラフを、そのグレイコードに対する随伴グラフという。随伴グラフがハミルトンパスをもつことはグレイコードが構成できることと同値である。また、随伴グラフがハミルトンサイクルをもつことは巡回的グレイコードが構成できることと同値である。随伴グラフの特徴付けはなく、また与えられたグラフがハミルトンパス (サイクル) をもつかどうかを判定する問題は NP 完全であることから、$\mathrm{P} \neq \mathrm{NP}$ ならば 随伴グラフを用いて効率的なアルゴリズムを設計することはできない。そのため、個々の対象に対してグレイコードを生成するアルゴリズムを設計する必要がある。

ここで扱うグレイコードについて述べる。整数 n に対して $x_1 + x_2 + \cdots + x_d = n$ かつ $\sum_{l=1}^{d}\{(x_l - x_{d-l+1})(n+1)^{d-l}\} \geq 0$ を満たす非負整数列 $(x_1, x_2, \cdots, x_d)$ に対するグレイコードである。ここでの "最小の差異" は整数の分配、分割と同じである。この問題は対称性を考慮した整数の分配であり、整数の分割における条件 $x_1 \geq x_2 \geq \cdots \geq x_d$ を弱めたものと考えることができる。この対称性を考慮した整数の分配はキャタピラグラフという木の部分クラスの文字列の表現でもある [5]。現在知られているキャタピラグラフの列挙アルゴリズムはグレイコードとしての性質をもっていない。$n = 6$, $d = 3$ に対して対称性を考慮した整数の分配グレイコードの例を下に示す。

$(6,0,0) \to (5,0,1) \to (4,0,2) \to (3,0,3) \to (3,1,2) \to (4,1,1) \to (5,1,0) \to (4,2,0) \to (3,2,1) \to (2,2,2) \to (2,3,1) \to (3,3,0) \to (2,4,0) \to (1,4,1) \to (1,5,0) \to (0,6,0)$

本論文ではこの問題に対するアルゴリズムを提案する。本論文で提案するアルゴリズムではまず、生成する正数列を d 次元空間の整数格子点とみなす。次にその格子点の凸包によってできる多面体の内部の点を巡回することでグレイコードを生成する。また、このアルゴリズムは次元 d $(d \geq 3)$ に基づく再帰的アルゴリズムになっている。$d = 2, 3, 4$ に対しては個別にそのアルゴリズムを与える。

以下、第 2 章で定義と本論文で扱う文字列について例を挙げて詳しく述べ、キャタピラとの関連についても述べる。第 3 章でアルゴリズムの概要を d 次元空間の整数格子点を用いて述べる。また、$d = 3, 4$ に対するアルゴリズムも与える。さらに提案するアルゴリズムの評価をし、生成される文字列一つ当たり平均定数時間であることを示す。第 4 章はまとめである。

2　定義と用語

文字列の集合に対して、その要素列をコードと呼ぶ。

非負整数の集合を $\mathbb{Z}_+$ であらわす。自然数 n の d 部への分配 x を $x = (x_1, \ldots, x_d) \in \mathbb{Z}_+^d$ とし、x の集合を $\Omega_{\mathrm{com}}(n,d)$ であらわす。集合 $\Omega_{\mathrm{com}}(n,d)$ 上の関係 "$\succ$" を次のように定める。二つの分配 $x, y \in \Omega_{\mathrm{com}}(n,d)$ に対して、$\min\{i \mid x_i \neq y_i\}$ なる i について $x_i > y_i$ ならば $x \succ y$ と定義し、すべての i について $x_i = y_i$ ならば $x = y$ と定義する。$x \succ y$ または $x = y$ であるとき $x \succeq y$ と書く。このとき関係 $x \succeq y$ は $x, y \in \mathbb{Z}^d$ の辞書式順序である。

$x = (x_1, \ldots, x_d) \in \Omega_{\mathrm{com}}(n,d)$ に対して、(x_1, x_d) を外対とよび、$(x_2, \ldots, x_{d-1})$ を内列とよぶ。また、$x \in \Omega_{\mathrm{com}}(n,d)$ に対して「反転」を $s(x) \stackrel{\mathrm{def.}}{=} (x_d, \ldots, x_1)$ と定義する。

このとき、$\Omega_{\mathrm{com}}(n,d)$ の部分集合 $\Omega_{\mathrm{mic}}(n,d)$ を $\Omega_{\mathrm{mic}}(n,d) \stackrel{\mathrm{def.}}{=} \{x \mid x \in \Omega_{\mathrm{com}}(n,d), x \succeq s(x)\}$ で定義する。集合 $\Omega_{\mathrm{mic}}(n,d)$ を鏡像を同一視した分配の集合 (set of mirror images identified compositions) と呼ぶ。

$\Omega(n,d) \in \{\Omega_{\rm mic}(n,d),\ \Omega_{\rm com}(n,d)\}$ とし、$x,y \in \Omega(n,d)$ とする。$x = (x_1,x_2,\ldots,x_d)$, $y = (y_1,y_2,\ldots,y_d)$ であるとき、x と y との距離 $D(x,y)$ を

$$D(x,y) = \frac{1}{2}\sum_{i=1}^{d}|x_i - y_i|$$

で定義する。$x,y \in \Omega(n,d)$ に対して $D(x,y) = 1$ となる必要十分条件はある整数 p,q $(p,q \le d)$ が存在して、$|x_p - y_p| = |x_q - y_q| = 1$ かつ $|x_r - y_r| = 0, (r \ne s,t)$ である。この距離を用いて本論文での $\Omega(n,d)$ 上のグレイコードを定義しよう。$\Omega(n,d)$ の列の数を T とする。$\Omega(n,d)$ のコード $x^1,x^2,\ldots,x^T$ が Ω の組合せグレイコードであるとは $|\{x^1,x^2,\ldots,x^T\}| = T$ かつ任意の t $(1 \le t < T)$ に対して、$D(x^t,x^{t+1}) = 1$ であるときをいう。

コード $X = x^1,x^2,\ldots,x^T$ が与えられたとき、$(-1)X = x^T,x^{T-1},\ldots,x^1$ と定義する。ただし、$(-1)(-1)X = X$ であり、$\overbrace{(-1)\cdots(-1)}^{\sigma個}X = (-1)^\sigma X$ と記述する。コード $X = x^1,x^2,\ldots,x^T$, $Y = y^1,y^2,\ldots,y^T$ に対して、X と Y の連結 $X \oplus Y$ を $X \oplus Y = x^1,x^2,\ldots,x^T,y^1,y^2,\ldots,y^T$ と定義する。

2 項からなる列 $x = (x_1,x_2)$ と列 $y = (y_1,y_2,\ldots,y_d)$ が与えられたとき $x \circ y = (x_1,y_1,y_2,\ldots,y_d,x_2)$ と定義する。また、2 項からなる列 $x = (x_1,x_2)$ とコード $Y = Y^1,Y^2,\ldots,Y^T$ に対して、$x \circ Y = x \circ Y^1, x \circ Y^2,\ldots,x \circ Y^T$ と定義する。最後に、2 項からなるコード $X = X^1,X^2,\ldots,X^S$ とコード $Y = Y^1,Y^2,\ldots,Y^T$ に対して、$X \odot Y = (x^1 \circ Y) \oplus (x^2 \circ Y) \oplus \cdots \oplus (x^S \circ Y)$ と定義する。

3 グレイコード生成アルゴリズム

本章では $\Omega_{\rm mic}(n,d)$ のグレイコード生成アルゴリズムについて述べる。$d = 1$ の時は自明にアルゴリズムが存在する。$\Omega_{\rm mic}(n,2)$ に対しては、$(n,0) \to (n-1,1) \to \cdots \to (\lceil n/2\rceil,\lfloor n/2\rfloor)$ という単純なグレイコードが存在する。以下では $d \ge 3$ について議論する。

提案するアルゴリズムは d に関する 2 ごとの再帰的アルゴリズムである。再帰の基は $d = 3,4$ であり、アルゴリズムの構成には $\Omega(m,d-2)$, $(0 \le m \le n)$ のグレイコード生成アルゴリズムを用いる。

以下では $\Omega_{\rm mic}(n,d)$ に対するグレイコード生成アルゴリズムを $A_{\rm mic}(n,d)$ で表し、得られるグレイコードを $G_{\rm mic}(n,d)$ と表記する。同様に、$\Omega_{\rm com}(n,d)$ のグレイコード生成アルゴリズムを $A_{\rm com}(n,d)$ で表し、得られるグレイコードを $G_{\rm com}(n,d)$ と表記する。

本論文では $\Omega_{\rm mic}(n,d)$ および $\Omega_{\rm com}(n,d)$ に対して、以下の性質を持つグレイコードを生成するアルゴリズムを提案する。

性質 1 分配の列が $x^1 = (n,0,\ldots,0)$ で始まり $x^T = (\overbrace{0,\ldots,0,n}^{\lceil d/2\rceil\ 項},0,\ldots,0)$ で終わる。

次の節では、このような性質を持つアルゴリズムが再帰的に構成できることを示す。

3.1 $d \ge 5$ のとき $A_{\rm cat}(n,d)$ の再帰アルゴリズム

$n = 0$ に対して、$A_{\rm com}(0,d)$ は $(0,\ldots,0)$ を返して終了する。

次に、$n \ge 1,\ d \ge 5$ として $G_{\rm mic}(n,d)$ の構成する場合を考えよう。仮定として $2 < h < d$ なる h に対して、任意の n について $A_{\rm com}, A_{\rm mic}$ によって性質1を満たすグレイコード $G_{\rm com}(n,h), G_{\rm mic}(n,h)$ が構成できるとする。

再帰的なアルゴリズムとして、外対 (x_1,x_d) を固定し、その外対を満たすようなすべての要素を列挙する。すなわち、$k = n - x_1 - x_d$ とおいて、外対が $x_1 = x_d$ を満たすとき、$A_{\rm mic}(k,d-2)$ を動かし、外対が $x_1 \ne x_d$ のときは $A_{\rm com}(k,d-2)$ を動かす。

Fix (x_1,x_d).
For $(x_2,\ldots,x_{d-1})$,
  if $x_1 = x_d$ then run $A_{\rm mic}(n-x_1-x_d,d-2)$,
  else run $A_{\rm com}(n-x_1-x_d,d-2)$.

次に外対 (x_1,x_d) を固定していく順番を考え

る。アルゴリズム $A_{\mathrm{mic}}(n,d)$ に対しては次の順で固定する。

For $(k=0;k\leq n;k++)$
　if($k\equiv_2 0$) then
　　start (x_1,x_d) := $(n-k,0)$ to $(\lceil(n-k)/2\rceil,\lfloor(n-k)/2\rfloor)$,
　else
　　start $(x_1,x_d):=(\lceil(n-k)/2\rceil,\lfloor(n-k)/2\rfloor)$ to $(n-k,0)$.

生成される $\Omega_{\mathrm{mic}}(n,d)$ のコード G の構造は次のようになっている。$G\ni(x_1,x_2,\ldots,x_d)$ とする。

1. 外対の和 x_1+x_d に対して降順に生成。
2. $n\geq 2$ に対して、外対の和が $n-1$ である最初の部分コードは n が偶数のとき $(x_1,x_d)\circ(-1)G_{\mathrm{com}}(1,d-2)$, $n\equiv 3 \pmod 4$ のとき $(x_1,x_d)\circ(-1)G_{\mathrm{mic}}(1,d-2)$, $n\equiv 1 \pmod 4$ のとき $(x_1,x_d)\circ G_{\mathrm{mic}}(1,d-2)$ となる。
3. q を (x_1,x_d) を固定していく順に対応するカウンターとし、$q=1$ は $(\lceil\frac{n-1}{2}\rceil,\lfloor\frac{n-1}{2}\rfloor)$ のときである。外対の和 x_1+x_d を固定した場合を考える。外対が $(a,b),(a',b')$ の順で、λ, λ' を内列のコードとする ($\lambda\in\{(\pm1)G_{\mathrm{com}}(n-x_1-x_d,d-2)\}$)。まず、外対は $G_{\mathrm{mic}}(x_1+x_d,2)$ 順に生成されるとする。このとき $(a,b)\circ(-1)^q\lambda$ が生成された直後に生成される部分コードは $(a',b')\circ(-1)^{q+1}\lambda'$ である。ただし、 $\lambda'=\lambda$ ($a>b+2$ のとき), $\lambda'=G_{\mathrm{mic}}(n-x_1-x_d,d-2)$ ($a=b+2$ のとき) とする。つぎに、外対は $(-1)G_{\mathrm{mic}}(x_1+x_d,2)$ 順に生成されるとする。$(a,b)\circ(-1)^q\lambda$ が生成された直後に生成される部分コードは $(a',b')\circ(-1)^{q+1}\lambda'$ である。ただし、$\lambda'=G_{\mathrm{com}}(n-x_1-x_d,d-2)$。
4. 3. において外対の和 $x_1+x_d=r$ である最後の部分コードを $(e,f)\circ\sigma$ とする。このとき、$e=f+1$ かつ $\sigma=(-1)^zG_{\mathrm{com}}(n-r,d-2)$ ならば $x_1+x_d=r-1$ である最初のコードは $(e-1,f)\circ(-1)^{z+1}G_{\mathrm{mic}}(n-r+1,d-2)$ である。$e>f+1$ かつ $\sigma=(-1)^zG_{\mathrm{com}}(n-r,d-2)$ ならば $x_1+x_d=r-1$ である最初のコードは $(e-1,f)\circ(-1)^{z+1}G_{\mathrm{com}}(n-r+1,d-2)$ である。$e=f$ かつ $\sigma=(-1)^zG_{\mathrm{com}}(n-r,d-2)$ ならば $x_1+x_d=r-1$ である最初のコードは $(e,f-1)\circ(-1)^{z+1}G_{\mathrm{com}}(n-r+1,d-2)$ である。

定理 1 *1-4 の構造をもつように構成された $\Omega_{mic}(n,d)$ のコードが性質 1 を満たす $G_{mic}(n,d)$ である。*

証明 $\Omega_{\mathrm{mic}}(n,d)\ni x=(x_1,x_2,\ldots,x_d)$ とする。$x_1>x_d$ のとき、$x=(x_1,x_d)\circ(x_2,\ldots,x_{d-1})\in(x_1,x_d)\circ G_{\mathrm{com}}(n-x_1-x_d,d-2)$ となり、$x_1=x_d$ のとき、$x=(x_1,x_d)\circ(x_2,\ldots,x_{d-1})\in(x_1,x_d)\circ G_{\mathrm{mic}}(n-x_1-x_d,d-2)$ となる。$A_{\mathrm{mic}}(n,d)$ は外対を固定しているアルゴリズムであるから、$\Omega_{\mathrm{mic}}(n,d)$ の列を重複なくかつ、もれなく列挙している。

3. において外対が (a,b) で固定されたとき、λ 内の連続する列が距離 1 であることから $(a,b)\circ(-1)^q\lambda$ の連続する列の距離は 1 である。$((a,b)\circ(-1)^q\lambda)\oplus((a',b')\circ(-1)^{q+1}\lambda'$ において $(-1)^q\lambda)$ の最後尾の列と $(-1)^{q+1}\lambda'$ の先頭の列は一致し、距離は 0 である。$(a,b),(a',b')$ は $G_{\mathrm{mic}}(a+b,2)$ の連続する列であるから $D((a,b),(a',b'))=1$。よって $((a,b)\odot(-1)^q\lambda)\oplus((a',b')\odot(-1)^{q+1}\lambda')$ の連続する列の距離は 1 である。

また、4. より $(G_{\mathrm{mic}}(n,2)\odot(-1)^q\lambda)\oplus((-1)(G_{\mathrm{mic}}(n-1,2)\odot(-1)^{q+1}\lambda'))$ の連続する列の距離は 1 である。よって生成されるコード G はグレイコードである。

次に性質 1 を満たすことを示す。1,3 より G の先頭の列は $(n,0,\ldots,0)$。2. より列 $(\lceil\frac{n}{2}\rceil,\ldots,\lfloor\frac{n}{2}\rfloor)$ の直後の列から G の最後尾の列 z までの列の個数は n が偶数のとき $\frac{n}{2}\left(\frac{n}{2}+1\right)$ であり、n が奇数のとき $\left(\frac{n+1}{2}\right)^2$ である。よって $n\equiv 1 \pmod 4$ のとき列 $(\lceil\frac{n}{2}\rceil,\ldots,\lfloor\frac{n}{2}\rfloor)$ の直後を $(-1)G_{\mathrm{mic}}(n-1,2)\odot G_{\mathrm{com}}(1,d-2)$ とし、$n\not\equiv 1 \pmod 4$ かつ $n=1$ のとき列 $(\lceil\frac{n}{2}\rceil,\ldots,\lfloor\frac{n}{2}\rfloor)$ の直後を $(-1)G_{\mathrm{mic}}(n-1,2)\odot G_{\mathrm{mic}}(1,d-2)$ とし、そして $n\not\equiv 1 \pmod 4$ のとき列 $(\lceil\frac{n}{2}\rceil,\ldots,\lfloor\frac{n}{2}\rfloor)$ の直後を $(-1)G_{\mathrm{mic}}(n-1,2)\odot(-1)G_{\mathrm{com}}(1,d-2)$ とすれば、$z=(\overbrace{0,\ldots,n}^{\lceil n/2\rceil\text{ 項}},0,\ldots,0)$ となる。

$\mathcal{Q.E.D.}$

3.2 整数分配に対する再帰アルゴリズム

次に A_{com} について議論を行おう。A_{com} は Knuth によるアルゴリズムと同じ手法(revolving door algorith) に基づいている。しかし、Knuth のアルゴリズムによって生成されるグレイコードは性質 1 を満たしていない。そこで、性質 1 を満たすように工夫する必要がある。A_{mic} と同様に外対 (x_1, x_d) を決め、$(x_2, \ldots, x_{d-1})$ について再帰的にアルゴリズムを呼び出すことで性質 1 を満たすグレイコード生成アルゴリズムを設計する。

このアルゴリズムの設計において A_{mic} と異なる点が 3 つある。

i. 固定される外対によらず、常にサブルーチンは $A_{\mathrm{com}}(k, d-2)$ だけという点。

ii. 外対についても $A_{\mathrm{com}}(n-k, 0)$ だけを用いるという点。

iii. サブルーチンの $(-1)^j A_{\mathrm{com}}(k, d-2)$ 符号 (j の値) が A_{mic} と異なるという点。

iii. の符号 j について $(0, \ldots, 0, n)$ の直後の列から $(\overbrace{0, \ldots, 0, n}^{\lceil d/2 \rceil \text{項}}, 0, \ldots, 0)$ までの固定される外対の個数は $M = n(n+1)/2$ となる。よって $n \equiv 0, 3 \pmod 4$ のとき $(0, \ldots, 0, n)$ の直後を $(-1)G_{\mathrm{com}}(n-1, 2) \odot (-1)G_{\mathrm{com}}(1, d-2)$ で始め、$n \equiv 1, 2 \pmod 4$ のとき $\lfloor n/2 \rfloor \equiv_2 0$ のとき、$(0, \ldots, 0, n)$ の直後を $(-1)G_{\mathrm{com}}(n-1, 2) \odot (-1)^2 G_{\mathrm{com}}(1, d-2)$ で始める。$A_{\mathrm{mic}}(n, d)$ に対してこの変更と i. と ii. の変更を行うことで性質 1 を満たすグレイコードを満たす $A_{\mathrm{com}}(n, d)$ が設計できる。

ここで、A_{com} の基について考えよう。再帰は d に関して 2 ごとの再帰である。したがって、d が奇数の場合と偶数の場合を考える必要がある。まず d が奇数のとき、$d = 1$ を再帰の基に取る。すなわち、$A_{\mathrm{com}}(n', 1)$, $(-1)A_{\mathrm{com}}(n', 1)$ ともに、列 (n') を返して終わる。このとき、任意の n に対して、$A_{\mathrm{com}}(n, 3)$ は性質 1 を満たすことが分かる。したがって先ほどの再帰の議論により、任意の n および任意の奇数 d に対する $A_{\mathrm{com}}(n, d)$ を得る。

次に、d が偶数の場合について考える。まず、$d = 2$ に対して、性質 1 を満たすアルゴリズムは存在しない。そこで、$A_{\mathrm{com}}(n', 2)$ は $(0, n')$ から順に生成し $(n', 0)$ で終わるアルゴリズムであると定義しよう。そして、偶数 d に対する $A_{\mathrm{com}}(n, d)$ の再帰の基を $d = 4$ にする。$A_{\mathrm{com}}(n, 4)$ に再帰的アルゴリズムを当てはめると、$A_{\mathrm{com}}(n, 4)$ は性質 1 を満たす。したがって $d \geq 6$ に対して、再帰の議論が行え、任意の偶数 $d \geq 4$ に対する $A_{\mathrm{com}}(n, d)$ を得る。

この節の議論から、任意の n と任意の $d \geq 3$ について $\Omega_{\mathrm{com}}(n, d)$ に対し、性質 1 を満たすグレイコード生成アルゴリズム $A_{\mathrm{com}}(n, d)$ が設計できた。

3.3 再帰の基

この節では $A_{\mathrm{mic}}(n, d)$ の再帰の基について議論する。前節 3.2 の $A_{\mathrm{com}}(n, d)$ に対する議論と同様、再帰の基として $d = 3$ と $d = 4$ について考えれば十分である。

まず、$A_{\mathrm{mic}}(n, 3)$ を考える。3.2 節における $A_{\mathrm{com}}(n, 3)$ と同様の議論が行える。$A_{\mathrm{mic}}(n, 3)$ に関しては素朴な再帰を考えることができる。すなわち、$A_{\mathrm{mic}}(n, 1)$ は $A_{\mathrm{com}}(n, 1)$ 同様に n を返して終わるので、d の再帰を 1 まで行う。したがって d が奇数の場合は A_{com}, A_{mic} とも、$d = 1$ まで再帰を行うことができる。図 1 は $A_{\mathrm{mic}}(8, 3)$ によって生成される列をあらわしている。$\Omega_{\mathrm{mic}}(8, 3)$ の列は 3 次元空間の平面 $x_1 + x_2 + x_3 = 8$ 上の整数格子点で $x_1 \geq 0$, $x_2 \geq 0$, $x_3 \geq 0$ を満たす。$A_{\mathrm{mic}}(8, 3)$ は $\Omega_{\mathrm{mic}}(8, 3)$ の列を $(8,0,0)$, $(7,0,1)$, $(6,0,2)$, $(5,0,3)$, $(4,0,4)$, $(4,1,3)$, $(5,1,2)$, $(6,1,1)$, $(7,1,0)$, $(6,2,0)$, $(5,2,1)$, $(4,2,1)$, $(3,2,3)$, $(3,3,2)$, $(4,3,1)$, $(5,3,0)$, $(4,4,0)$, $(3,4,1)$, $(2,4,2)$, $(2,5,1)$, $(3,5,0)$, $(2,6,0)$, $(1,6,1)$, $(1,7,0)$, $(0,8,0)$ の順で生成する。次に $A_{\mathrm{mic}}(n, 4)$ について考える。$\Omega_{\mathrm{com}}(n, 2)$ に対して性質 1 を満たさなかったように $\Omega_{\mathrm{mic}}(n, 2)$ についても $n \geq 2$ について性質 1 を満たすグレイコードは存在しない。さらに

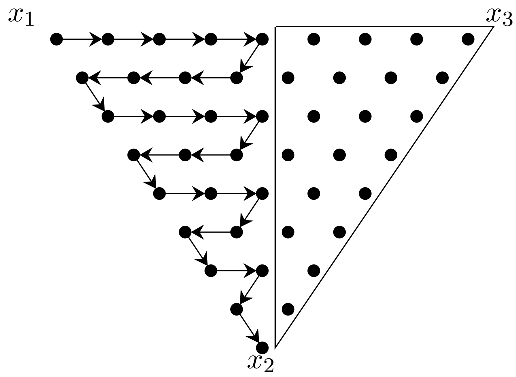

図 1: $n=8$, $d=3$ のグレイコードアルゴリズムの動き

$A_{\mathrm{mic}}(n,4)$ の場合もうひとつ大きな問題が存在する。それは、$x_1=x_4$ の場合、内側の対 (x_2,x_3) はすべてで $(n,0),(n-1,1),\ldots,(\lceil n/2\rceil,\lfloor n/2\rfloor)$ である。すなわちゼロを含む列は $(n,0)$ だけとなってしまうので、前の反復で A_{com} からきて $(n,0)$ で A_{mic} を進めると A_{mic} の終わりは非ゼロの対となるので、単純に次の反復で A_{com} に移ることはできない。

そこで $A_{\mathrm{mic}}(n,4)$ に対して特別なアルゴリズムを考える。このアルゴリズムでは先頭 x_1 を順に固定し、(x_2,x_3,x_4) を列挙する。x_1 が固定されたとき、(x_2,x_3,x_4) の満たすべき条件は

$$x_2+x_3+x_4=n-x_1, \tag{1}$$

$$x_4 \le x_1, \tag{2}$$

$$x_2 \ge x_3 (x_1=x_4\text{の場合}) \tag{3}$$

の三つである。この条件を満たすような (x_2,x_3,x_4) の列挙アルゴリズムを $B(x_1)$ とする。

以下では $x_1>0$ に対して、$B(x_1,n)$ を考える。まず、$x_1 \ge \lceil n/2\rceil$ の時を考える。このとき制約 (2) は、自動的に不等式で満たされる。すなわち制約 (2) だけを考慮すればよい。このとき、$A_{\mathrm{com}}(n',3)$ と同様のアルゴリズムを考える。すなわち、(x_2,x_4) を外対とみなし、(x_2,x_3,x_4) を全列挙する。

次に $1<x_1\le\lfloor n/2\rfloor$ を考える。このとき、制約 (2), (3) によって、三角形から除去される点がある。これらの点の集合を禁止領域と呼ぶ。このとき、禁止領域に入らないように進む。図 1, 2, 3 では線で囲まれた点が禁止領域である。図 2 は $n=12$, $d=4$ で $x_1=4$ のときの例である。このとき列挙すべき整数格子点は $x_2+x_3+x_4=8$, $x_2\ge 0$, $x_3\ge 0$, $4\ge x_4\ge 0$, $x_1=x_4$ のとき $x_2\ge x_3$ を満たす。

$x_1=1$ については $(-1)B(x_1)$ $(x_1=1)$ の整数格子点のたどり方を変更し、列 $(\lfloor\frac{n-1}{2}\rfloor+1,\lceil\frac{n-1}{2}\rceil-1,0)$ で終わるようにする。そして、$x_1=0$ については $(-1)B(x_1)$ $(x_1=1)$ の整数格子点のたどり方を列 $(\lceil\frac{n}{2}\rceil,\lfloor\frac{n}{2}\rfloor,0)$ から始まり、列 $(n,0,0)$ で終わるようにする。x_1 の値も含め 4 桁で $x_1=1,0$ のたどり方を記述すると $(1,0,n-1,0)$, $(1,1,n-2,0)$, $(1,2,n-3,0)$, $\ldots$, $(1,\lfloor\frac{n-1}{2}\rfloor,\lceil\frac{n-1}{2}\rceil,0)$, $(1,\lfloor\frac{n-1}{2}\rfloor,\lceil\frac{n-1}{2}\rceil-1,1)$, $(1,\lfloor\frac{n-1}{2}\rfloor+1,\lceil\frac{n-1}{2}\rceil-2,1)$, $(1,\lfloor\frac{n-1}{2}\rfloor+2,\lceil\frac{n-1}{2}\rceil-3,1)$, $\ldots$, $(1,n-2,0,1)$, $(1,n-1,0,0)$, $(1,n-2,1,0)$, $(1,n-3,2,0)$, $\ldots$, $(1,\lfloor\frac{n-1}{2}\rfloor+1,\lceil\frac{n-1}{2}\rceil-1,0)$, $(0,\lceil\frac{n}{2}\rceil,\lfloor\frac{n}{2}\rfloor,0)$, $(0,\lceil\frac{n}{2}\rceil+1,\lfloor\frac{n}{2}\rfloor-1,0)$, $(0,\lceil\frac{n}{2}\rceil+2,\lfloor\frac{n}{2}\rfloor-2,0)$, $\ldots$, $(0,n,0,0)$ となる。これらの変更により、4 桁の列で $x_1=0,1$ であるものをグレイコードとして生成することができる。

最後に x_1 を n から順に 1 ずつ減らし、x_1 が偶数のときは $B(x_1)$ を x_1 が奇数のときは $(-1)B(x_1)$ を x_1 が 0 のときは $B(x_1)$ を実行すると $A_{\mathrm{mic}}(n,4)$ を得る (図 3 参照)。

3.4 計算量

定理 2 アルゴリズム $A_{mic}(n,d)$ は平均 $O(1)$ 時間で一つの列を生成する。

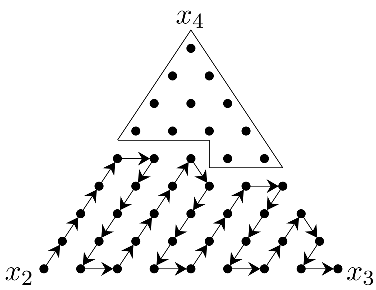

図 2: $n = 12$, $d = 4$ で、x_1 が 4 に固定されているときのグレイコードアルゴリズム A_{mic} の動き

証明 桁数 d に関する帰納法で証明する。まず、桁数が 3 の場合、$A_{\mathrm{mic}}(n,3)$,$A_{\mathrm{com}}(n,3)$ による 1 要素あたりの生成時間は明らかに n に依存しない。桁数が 4 の場合も同様である。桁数が $d-2$ の時、1 要素あたりの平均生成時間は n,d に依存しないと仮定する。桁数 d に対するアルゴリズム $A(n,d)$ では、$A(n,d-2)$ を再帰的に呼び出す。ここで、アルゴリズム $A(n,d-2)$ を呼び出すのに必要な時間は n,d に依存しない。仮定より、$A(n,d-2)$ は列一つ当たり平均 $O(1)$ で生成するアルゴリズムであった。したがって、桁数 d に対しても平均生成時間は n,d に依存しない。 $\mathcal{Q.E.D.}$

4 まとめ

本論文では鏡像を同一視した分配に対する組合せグレイコードの生成アルゴリズムを提案した。提案したアルゴリズムは列一つ当たり平均 $O(1)$ で生成する。このコードは巡回的ではなく、最後の列と最初の列の距離は n である。この距離が 1 になるような巡回グレイコードについては提案したアルゴリズムを改良することで得られる。

参考文献

[1] D. R. van Baronaigien, A loopless Gray-code algorithm for listing k-ary trees, *J. Algorithms*, 35 (2000) 100–107.

[2] J. Beer, Implementation of an algorithm to list Gray code sequences of partitions, Manuscript, North Carolina State Univ., 1990.

[3] G. S. bhat and C. D. Savage, Balanced Gay codes, *Electronic J. of Combin.*, 3(R25), 1996.

[4] P. Eades and B. mcKay, An algorithm for generating subsets of fixed size with a strong minimal change property, *Inform. Process. Lett.*, 19 (1984) 131–133.

[5] Y. Kikuchi, H. Tanaka, S. Nakano and Y. Shibata, How to obtain the complete list of caterpillars, *Proc. of COCOON 2003*, Lecture Notes in Comp. Sci., 2697, (2003) 329–338.

[6] P. Klingsberg, A Gray code for compositions, *J. Algorithms*, 3 (1982) 41–44.

[7] J. F. Korsh and P. LaFollette, Loopless generation of Gray codes for k-ary trees *Inform. Process. Lett.*, 70 (1999) 7–11.

[8] A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms*, 2nd ed., Academic Press, New York, 1978.

[9] M. Ramras, A new method of generating hamilton cycles on the n-cube, Discrete Math, 85 (1990) 329–331.

[10] D. Rasmussen, C. D. Savage and D. B. West, Gray code enumeration of families of integer partitions, *J. Combin. Theory Ser. A*, 70 (1995), 201–229.

[11] F. Ruskey and C. D. Savage, A Gray code for combinations of a multiset, *Europ. J. Combinatorics*, 17 (1996) 493–500.

[12] C. D. Savage, Gray code sequences of partitions, *J. Algorithms*, 10 (1989), 577–595.

[13] C. Savage, A survey of combinatorial Gray codes, *SIAM Review*, 39 (1997) 605–629.

[14] C. D. Savage and P. Winkler, Monotone Gray codes and the middle two levels problem, *J. Combin. Theory Ser. A*, 70 (1995) 230–248.

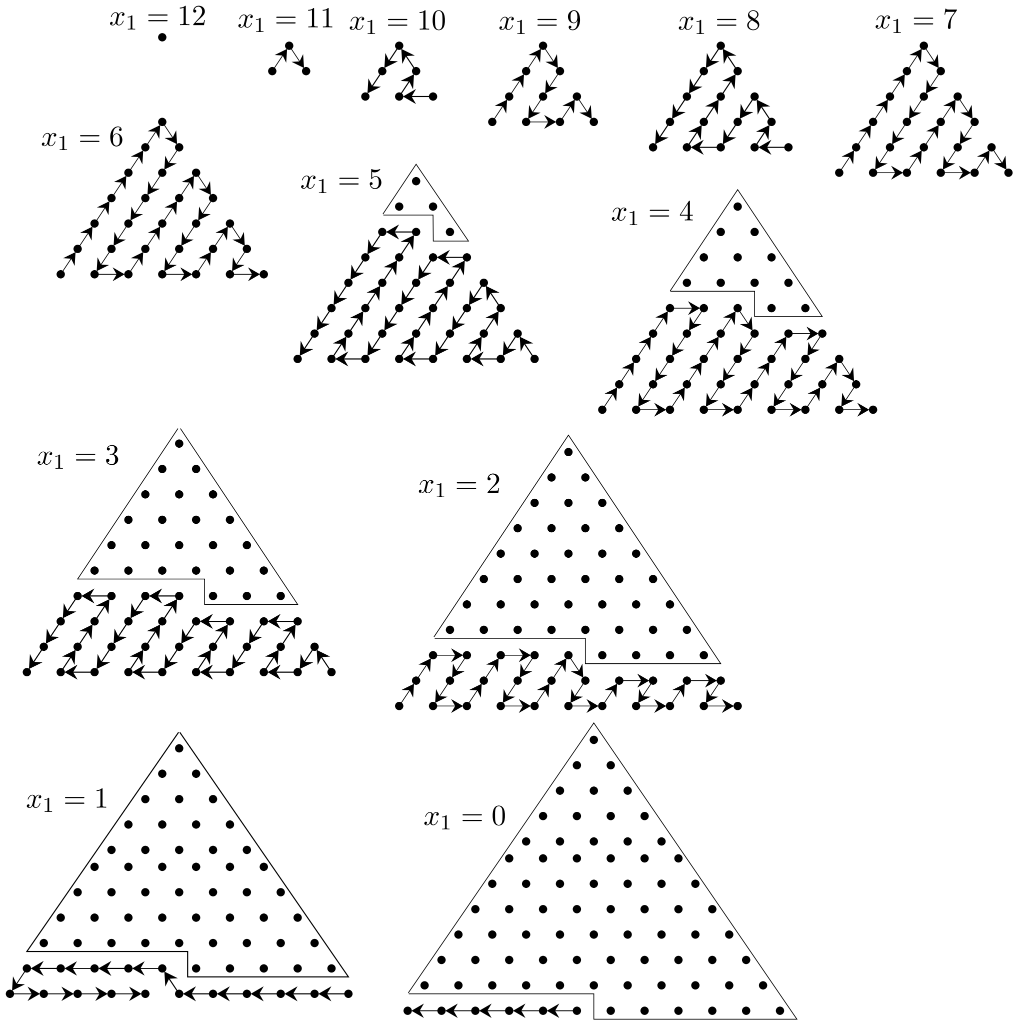

図 3: $n = 12$, $d = 4$ で、x_1 が 12 から 0 まで変化するときのグレイコードアルゴリズム A_{mic} の動き

[15] R. Sedgewick, permutation generation methods, *Computing Surveys*, 9 (1977) 137–164.

[16] T. Ueda, Gray codes for necklaces, *Discrete math.*, 219 (2000) 235–248.

[17] V. E. Vickers and J. Silverman, A technique for generating specialized Gray codes, *IEEE Trans. on Computers*, C-29 (1980) 329–331

[18] D. G. Wagner and J. West, Construction of uniform Gray codes, Congressus numerantium, 80 (1991) 217–223.

[19] H. S. Wilf, Combinatorial algorithms: An update, SIAM, Pennsylvania, 1989.

[20] L. Xiang, K. Ushijima and C. Tang, Efficient loopless generation of Gray codes for k-ary trees, *Inform. Process. Lett.*, 76 (2000) 169–174.