

Weyl 群の minuscule 元の最短表示総数公式

大阪大学大学院情報科学研究科 岡村修志

smv206os@ecs.cmc.osaka-u.ac.jp

1 はじめに

Kac-Moody Lie 環の Weyl 群 W の dominant minuscule 元 w (5 節参照) に対して、その最短表示の総数公式を与える定理 (Peterson) がある。この定理は Proctor[P1][P2] と Stembridge[St2] によって、d-complete poset と呼ばれる図形の標準盤の総数を与える hook 公式 (4 節参照) と同値であることが示された。d-complete poset は Young 図形や Shifted Young 図形を特殊な場合を含む拡張された図形であり、(simply-laced な場合の) dominant minuscule 元を特徴付けるものとして Proctor[P1][P2] によって導入された (2 節・5 節参照)。本稿において d-complete poset の定義 (2 節)、及びその hook の定義 (3 節) を与え、d-complete poset に対する hook 公式及び、その証明方針を概説し (4 節)、Peterson の定理と hook 公式との同値性について説明する (5 節)。

Young 図形に対する hook 公式は、歴史的には Frame-Robinson-Thrall[FRT] によって最初に与えられ、様々な証明が知られているが、その中で確率論的証明が Greene-Nijenhuis-Wilf[GNW] により与えられた。hook 公式は hook 自体が分からなくても、hook length さえ分かれば適用できる公式である (従って hook length 公式とも呼ばれる) がこの確率論的証明においては hook 自体が大きな役割を演じる。正確には、hook walk algorithm という hook が本質的な役割を果たす確率的アルゴリズムを構成し、そのアルゴリズムが与えられた形の d-complete poset の任意の標準盤を等確率生成するアルゴリズムであることを示すことで hook 公式が証明される (4 節参照)。同様の方法を用いて Sagan[Sa] により Shifted Young 図形の場合が証明された。一般の d-complete poset の場合について、その hook の定義は Kawanaka[K] により与えられた (3 節参照)。その hook を適用することで一般の場合にも hook walk algorithm が構成でき、従って Greene-Nijenhuis-Wilf の方法が拡張できる。d-complete poset の hook 公式の hook walk algorithm による確率論的証明は本稿では概説に留められるが、詳細については [O] において記述されている。

2 d-complete poset

以下、有限順序集合の事を poset と呼ぶ (連結とは限らない)。poset を図にした時は常に上、又は左に向かうほど大きくなるように表す。

Proctor によって定義された d-complete poset は Young 図形をその特殊なケースとして含む拡張された概念になっている。本節では基本的な言葉の準備と d-complete poset の定義を与え、d-complete poset の簡単な性質を説明する。この節の内容は基本的に Proctor[P2] に従う。

Young 図形は各正方形を点として見て、隣り合う点同士を線で結んだ図形と見直すことができる。この図形を集合として各点を要素に持つと考え、上や左に向かうほど大きくなるように順序が入っていると考えると poset になる。この poset を Young 図形の poset 表示と呼ぶ。以下、Young 図形は常にこの poset 表示を意味するとする。

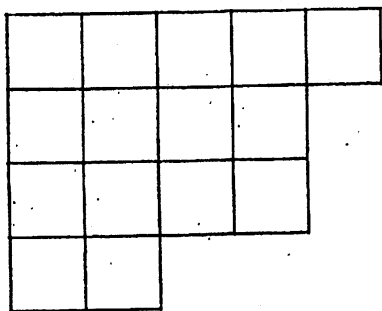


図 1: Young 図形

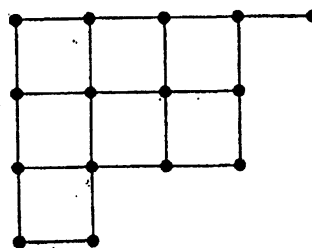


図 2: poset 表示

P を任意の poset とする。 $x, y \in P$ に対して x が y を cover する時、 $x \succ y$ と表す。 P と P' が順序同型であるとき、 $P \simeq P'$ と表す。 P の極小元を P の corner と呼ぶ。 P の順序が全順序である時 P は chain であると言う。 $a, b \in P, a < b$ に対して $[a, b] := \{c \in P \mid a \leq c \leq b\}$ とする。 P の部分 poset P' が任意の $x \in P'$ に対して「 $y \geq x$ in $P \implies y \in P'$ 」なる条件を満たす時、 P' を P の filter と言う。

$k \geq 3$ に対して順序集合 $d_k(1)$ を次のように定める。

$$d_k(1) := \{w_1, w_2, \dots, w_{k-2}, x, y, z_1, z_2, \dots, z_{k-2}\}$$

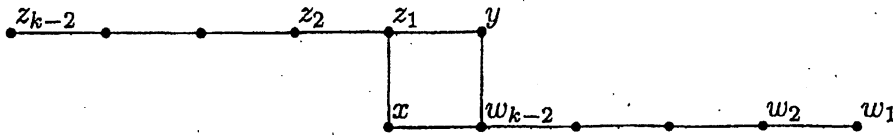
$$w_1 < w_2 < \dots < w_{k-2}$$

$$w_{k-2} < x < z_1$$

$$w_{k-2} < y < z_1$$

$$z_1 < z_2 < \dots < z_{k-2}$$

$d_k(1)$ の比較不能元 x, y を $d_k(1)$ の side と言う。又、 $d_k^-(1) := d_k(1) - \{z_{k-2}\}$ とする。

図 3: $d_k(1)$

$w, x, y, z \in P$ に対して

$$z \succ x, y \quad x, y \succ w \text{ in } P$$

なる条件が成り立つ時、 $\{w, x, y, z\}$ を P 上の diamond と言う。

$a, b \in P$ に対して $[a, b] \simeq d_k(1)$ なる時、 $[a, b]$ は P の d_k -interval であると言う。その時、 b は P 上 d_k -interval の neck であると言い、 a は P 上 d_k -interval の tail であると言う。又、 a は b を neck に持つと言い、 b は a を tail に持つと言う。 $b \in P$ がある k に対して P 上 d_k -interval の neck である時、単に b を P 上の neck と呼ぶ。

$a, b \in P$ とする。 $k \geq 4$ について $[a, b] \simeq d_k^-(1)$ の時 $[a, b]$ を P の d_k^- -interval と呼ぶ。 $w, x, y \in P$ に対して $x, y \succ w$ なる時、 $\{w, x, y\}$ を d_3^- -interval と呼ぶ。 $k \geq 3$ について d_k -interval から最大元を除いたものは d_k^- -interval である。

$k \geq 4, x, y \in P, [x, y] : d_k^-$ -interval に対して

$$\exists z \in P \text{ s.t. } y \prec z, [x, z] : d_k\text{-interval}$$

の時、 $[x, y]$ を complete d_k^- -interval と呼ぶ。同様に $k = 3, w, x, y \in P, \{w, x, y\} : d_3^-$ -interval に対して

$$\exists z \in P \text{ s.t. } \{w, x, y, z\} : d_3\text{-interval}$$

の時、 $\{w, x, y\}$ を complete d_3^- -interval と呼ぶ。 d_k^- -interval であるが、complete d_k^- -interval でないものを incomplete d_k^- -interval と呼ぶ。

$x, y, w, w' \in P, w \neq w'$ とする。 $k \geq 4$ に対して

$$\begin{aligned} w \prec x, w' \prec x, \\ [x, y] : d_{k-1}\text{-interval}, \\ [w, y], [w', y] : d_k^-\text{-interval} \end{aligned}$$

のとき d_k^- -interval $[w, y]$ と $[w', y]$ は overlap するという。又、

$$\begin{aligned} w \prec x, w \prec y, \\ w' \prec x, w' \prec y, \\ \{w, x, y\}, \{w', x, y\} : d_3^-\text{-interval} \end{aligned}$$

のとき d_3^- -interval $\{w, x, y\}$ と $\{w', x, y\}$ は overlap するという。

P が次の条件を満たす時 P を d_3 -complete と呼ぶ。

- (1) incomplete d_3^- -interval は存在しない。
- (2) $\{w, x, y, z\}$:diamond, z :最大元ならば $|\{v \in P | v < z\}| = 2$
- (3) overlap する d_3^- -interval は存在しない。

$k \geq 4$ に対して、 P が次の条件を満たす時 P を d_k -complete と呼ぶ。

- (1) incomplete d_k^- -interval は存在しない。
- (2) $[x, y]$: d_k -interval ならば $|\{v \in P | v < y\}| = 1$
- (3) overlap する d_k^- -interval は存在しない。

全ての $k \geq 3$ に対して d_k -complete である poset を d-complete poset と呼ぶ。

Young 図形 は d-complete poset である。又、後で説明するように Young 図形 の hook 公式と同様の公式が d-complete poset に対しても成り立つ。従って d-complete poset は Young 図形 のある種の自然な拡張になっていると言える。

任意の d-complete poset は'slant 既約'な成分の'slant 和'として分解され、各'slant 既約'成分は 15 種類に分類されることが Proctor によって示された。従って任意の d-complete poset は完全に視覚的に把握できる。以下、その説明をする。

命題 2.1 d-complete poset の filter は d-complete poset である。

命題 2.2 連結な d-complete poset には最大元が存在する。

最大元を持ち、それ以外の元が必ず唯一つの元によって cover される poset を rooted tree と呼ぶ。rooted tree は明らかに d-complete poset である。

以下 S を一般の d-complete poset とする。 S の元 x で $y \geq x \implies y$ は高々一つの元によって cover されるという条件を満たすものを S の top tree element と言う。 S の top tree element 全体から成る poset を S の top tree と言う。 S の top tree は明らかに S の filter である。 S の top tree の元で特に S 上の neck になっていないものを S の acyclic element と呼ぶ。

命題 2.3 d-complete poset S_1, S_2 (S_2 は連結) と S_1 の acyclic element a, S_2 の最大元 b に対して、集合 $S_0 = S_1 \cup S_2$ に S_1, S_2 のもとの順序関係と新たな順序関係 $a > b$ によって生成される順序を入れると S_0 も d-complete poset である。

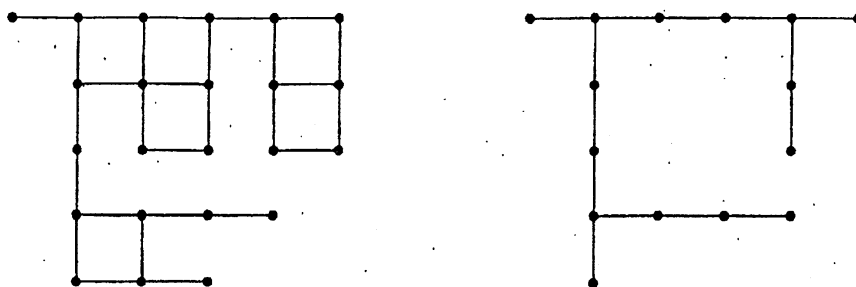


図 4: d-complete poset とその top tree の例

命題 2.3 によって構成された d-complete poset S_0 を d-complete poset S_1, S_2 の slant 和と呼び $S_1 \# S_2$ と表す。d-complete poset が連結であり、2つ以上の d-complete poset の slant 和 で表せない時 slant 既約であると言う。特に1つの点のみから成る集合 (ドットと呼ぶ) は slant 既約 d-complete poset である。ドット以外の slant 既約 d-complete poset は全て少なくとも一つの diamond を含む。

命題 2.4 任意の 連結 d-complete poset は slant 既約 d-complete poset の slant 和 として一意的に表される。

連結 d-complete poset S が slant 既約 d-complete poset $S_1, S_2, \dots, S_l$ の slant 和 で表せる時、 $S_1, S_2, \dots, S_l$ を S の slant 既約成分と呼ぶ。

定理 2.5 (slant 既約 d-complete poset の分類定理)(Proctor)
slant 既約 d-complete poset は 15 種類に分類される。

15 種類の slant 既約 d-complete poset の一覧はここでは与えない。その一覧は [P2] 及び [O] において記載されている。

3 hook

この節以降も特に注意が無ければ S は一般の d-complete poset を表すとする。hook とは S の各元に対応して定まるある条件を満たす S の部分集合である。Young 図形の hook は広く知られているが d-complete poset への hook の定義の拡張は Kawanaka[K] により与えられた。本節ではこの hook の定義を与える。

S の filter S' ($S' \neq S$) と $v \in S \setminus S'$ について $S' \cup \{v\}$ が S の filter なら v を S' の (S における) outside corner と呼ぶ。d-complete poset に hook を定義するためにいくつかのステップを踏む (図 5-8 において例示されている)。

(STEP1) 任意の d-complete poset S に対してその top tree を T と表す。 $k := |T|$ とする。 T の各元に $1 \sim k$ の値を重複無く割り当てる (どのように割り当てても良い)。この時、次のようにして S の全ての元に対して $1 \sim k$ を割り当てることができる。

(1) $S' := T$

(2-a) $S \setminus S' = \emptyset$ ならアルゴリズム終了

(2-b) $S \setminus S' \neq \emptyset$ なら S' の outside corner v を一つ取ってくる。(どのように取っても良い)

(3) v を tail に持つような S 上の neck v' が一意に存在し、それは S' の元である。この v' にラベルされている値を v にラベルする。

(4) $S' := S' \cup \{v\}$ として (2-a) へ

この対応付けによって $l: S \rightarrow \{1, 2, \dots, k\}$ が定まる。

(STEP2) $\{1, 2, \dots, k\}$ で添え字付けられた不定元の集合 $A := \{\alpha_1, \alpha_2, \dots, \alpha_k\}$ に対して $Z(A) := \{z_1\alpha_1 + \dots + z_k\alpha_k \mid z_i \in \mathbb{Z} \ (1 \leq i \leq k)\}$ として $\mathbb{Z}$ -線形空間を定め、 $\forall i \in \{1, \dots, k\}$ に対して $Z(A)$ への $\mathbb{Z}$ -線形写像 s_i を次の関係式から生成されるものとして定める。

$$s_i(\alpha_i) = -\alpha_i$$

$$s_i(\alpha_j) = \alpha_i + \alpha_j \quad (T \text{ 上で } i \text{ と } j \text{ が隣接している時})$$

$$s_i(\alpha_j) = \alpha_j \quad (T \text{ 上で } i \text{ と } j \text{ が隣接していない時})$$

$v_1, v_2, \dots, v_n$ を $S = \{v_1, v_2, \dots, v_n\}$ で $v_i > v_j \Rightarrow i < j$ なるように取る。 $i_k := l(v_k) \ (1 \leq k \leq n)$ とする。各 s_{i_k} は $Z(A)$ 上の $\mathbb{Z}$ -線形写像である。以上の準備の下、 S の hook の定義をする。

(STEP3) (d-complete poset における hook の定義)

$\forall v \in S$ に対し v の S 上の hook $H_S(v)$ を次のように定義する。 S が明らかかな時は単に H_v とも表す。

(1) $H_S(v) := \emptyset$

(2) k を $v = v_k$ なる k として定める。(一意に定まる)

(3) $\alpha := \alpha_{i_k}$

(4) $\alpha_{\text{new}} := s_{i_k}(\alpha)$

(5) $\alpha_{\text{new}} \neq \alpha$ なら $H_S(v) := H_S(v) \cup \{v_k\}$

(6) $\alpha := \alpha_{\text{new}}$

(7-a) $k < n$ なら $k := k+1$ として (4) へ

(7-b) $k = n$ なら アルゴリズム終了

一見この定義が well-defined であることは自明ではないが実際には well-defined である。

例 図5の slant 既約 d-complete poset S に対して v における hook を求める。まず、STEP1により S に図6のように $\{1, 2, \dots, 6\}$ をラベルする。STEP2で与えた条件を満たす座標を図7のように与える。この時、 $H_S(v)$ はSTEP3により

$$s_2 s_6 s_3 s_2 s_1 s_4 s_3 s_2 s_6 s_5 s_4 s_3 s_2(\alpha_2)$$

を計算することによって求めることができる。

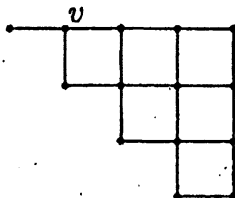


図 5: d-complete poset の例

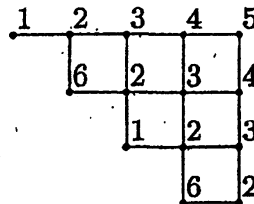


図 6: STEP1 によるラベル

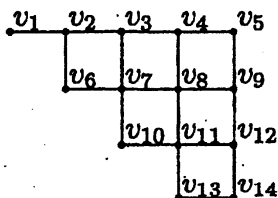


図 7: STEP2 による S の座標

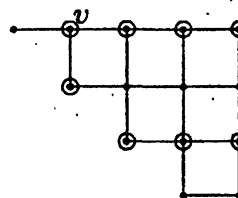


図 8: $H_S(v)$

STEP2 における座標の取り方は沢山あるがどの取り方をしても同じ hook を導くことに注意。

以下、hook に関していくつかの性質を述べる。

命題 3.1

- (i) v' が v の hook に含まれる $\implies v \geq v'$
- (ii) $\forall v \in S$ に対して v 及び、 v に cover される元は v の hook に入る。

系 v が S の corner $\iff h_S(v) = 1$

命題 3.2 S の任意の filter S' 及び $v \in S'$ に対して

$$H_{S'}(v) = H_S(v) \cap S'$$

命題 3.3 v が S において neck でないなら $H_S(v) = \{v' \in S | v' \leq v\}$ である。

命題 3.2 の結果と合わせて、次の系が得られる。

系 1 $w < v$ in S に対して v が $[w, v]$ において neck でないなら $[w, v] \subset H_S(v)$ である。

また、(本当は少し説明が必要だが) 次の系も得られる。

系 2 $w < v$ in S に対して v と w が S の異なる slant 既約成分 S_1, S_2 に属するなら $S_2 \subset H_S(v)$ である。

従ってこの系により、slant 既約 d-complete poset の hook が分かっているれば、直ちに一般の d-complete poset の hook も導くことができることが分かる。

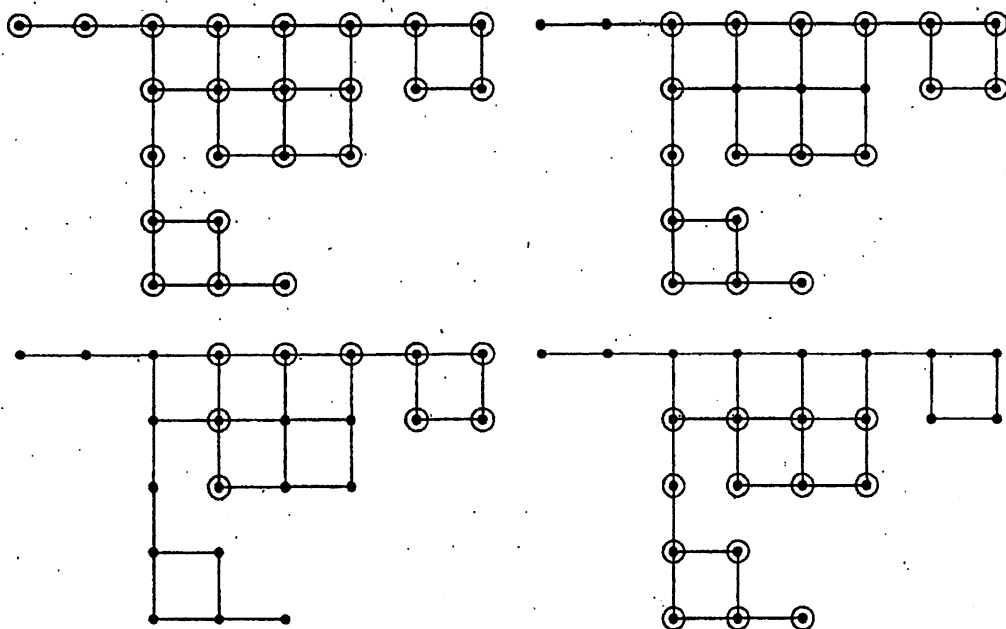


図 9: 一般の d-complete poset における hook の例

$\forall v \in S$ に対して $h_S(v) := |H_S(v)|$ を v の S 上の hook length と呼ぶ。 S が明らかな時は単に h_v とも表す。

命題 3.4 $b \in S$ に対して

(i) b が S 上の neck でない時、

$$h_b = |\{v \in S | b \geq v\}|$$

(ii) b が S 上の neck である時、 $[a, b]$ が d_k -interval となるような $a \in S$ が一意に存在するが、この時 x, y を $[a, b]$ の side とすると次の式が成り立つ。

$$h_b = h_x + h_y - h_a$$

(ii) は自明ではないが、(i) は命題 3.3 により明らか。命題 3.4 により、任意の d-complete poset に対して、その hook の形を構成しなくても帰納的に hook length だけを計算することができる。歴史的には d-complete poset に対しては hook よりも hook length が先に定義された (5 節参照)。

4 d-complete poset の hook 公式

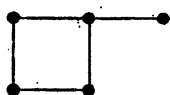
本節において hook 公式 及び、任意の d-complete poset に対してその標準盤を等確率生成するアルゴリズムの説明をし、等確率性の証明を与えるために何を示すべきかを明らかにする。

以下、 $n := |S|$ とする。全単射 $\sigma : S \rightarrow \{1, 2, \dots, n\}$ が次の条件を満たす時、 σ を S の標準盤と呼ぶ。

$$v, v' \in S \text{ について } v \geq v' \Rightarrow \sigma(v) \leq \sigma(v')$$

この時、 f_S を S の標準盤の総数とする。

例



上の図形に対して標準盤は次のように与えられる。

1	2	3	1	2	4	1	3	4	1	2	5	1	3	5
4	5		3	5		2	5		3	4		2	4	

よって $f_S = 5$ である。

定理 4.1 (hook 公式) 任意の d-complete poset S に対して次の式が成り立つ。

$$f_S = \frac{n!}{\prod_{v \in S} h_v} \quad (n = |S|) \quad (1)$$

この定理の証明のために hook walk algorithm を定義する。このアルゴリズムが d-complete poset の標準盤を等確率で生成するアルゴリズムであることが後の定理から分かる。

定義 (hook walk algorithm)

- (1) $i := 1$
- (2) S の任意の元を $\frac{1}{n-i+1}$ の確率で取り出し、それを v_1 と表す。
- (3) $j := 1$
- (4-a) v_j が S の corner なら (5) へ

(4-b) v_j が S の corner でないなら、 v_{j+1} を $H_{v_j} \setminus \{v_j\}$ の中から $\frac{1}{h_{v_j}-1}$ の確率で取り出す。 $j := j+1$ として (4-a) へ

(5) v_j に $n-i+1$ をラベルする。 $S := S \setminus \{v_j\}$

(6-a) $S = \emptyset$ ならアルゴリズム終了

(6-b) $S \neq \emptyset$ なら $i := i+1$ として (2) へ

hook の定義により、このアルゴリズムは必ず有限試行で停止する。このアルゴリズムによって S の一つの標準盤を得る。 S の任意の標準盤 σ に対して hook walk algorithm によって σ が選ばれる確率を $Prob_S(\sigma)$ で表す。

上のアルゴリズムの (1)-(5) の部分は S の一つの corner を選ぶ単独のアルゴリズムと見ることができる。このアルゴリズムを corner hook walk algorithm と呼ぶ。 S の任意の corner ω に対して corner hook walk algorithm によって ω が選ばれる確率を $prob_S(\omega)$ と表す。

定理 4.2 S の任意の標準盤 σ に対して $Prob_S(\sigma)$ は σ に依存せず一様に次の式で与えられる。

$$Prob_S(\sigma) = \frac{\prod_{v \in S} h_v}{n!}$$

明らかにこの定理から定理 4.1 が導かれる。従って、定理 4.2 を示すことを考える。 S の corner ω に対し $W_S(\omega)$ を

$$W_S(\omega) := \{v \in S \mid \omega \in H_S(v), v \neq \omega\}$$

と定め、 S が明らかな時は W_ω と表す。

定理 4.3 S の任意の filter S' とその任意の corner ω に対して次の式が成り立つ。

$$prob_{S'}(\omega) = \frac{1}{|S'|} \prod_{v \in W_{S'}(\omega)} \left(1 + \frac{1}{h_{S'}(v) - 1}\right)$$

S に対して定理 4.3 が成り立つなら定理 4.2 も成り立つことを示すためにまず次の補題を示す。

補題 4.4 d-complete poset S に対して ω を S の corner とし、 $S' = S \setminus \{\omega\}$ と置くと、次の式が成り立つ。

$$\frac{\prod_{v \in S} h_S(v)/n!}{\prod_{v \in S'} h_{S'}(v)/(n-1)!} = \frac{1}{n} \prod_{v \in W_S(\omega)} \left(1 + \frac{1}{h_S(v) - 1}\right)$$

証明)

$$v \notin W_S(\omega) \Rightarrow h_{S'}(v) = h_S(v)$$

$$v \in W_S(\omega) \Rightarrow h_{S'}(v) = h_S(v) - 1$$

この関係に従って左辺の分子分母をキャンセルすると右辺になる。□

主張 4.5 S に対して定理 4.3 が成り立つなら定理 4.2 も成り立つ。

(証明) S の任意の標準盤 σ を固定して考える。 σ によって i がラベルされている S の元を v_i と表し、 $S_i = \{v_1, v_2, \dots, v_i\}$ とする ($i = 1, 2, \dots, n$)。この時 S_i は S の filter であり、 v_i は S_i の corner である。hook walk algorithm により

$$Prob_S(\sigma) = prob_{S_n}(v_n) \cdot prob_{S_{n-1}}(v_{n-1}) \cdots prob_{S_1}(v_1)$$

よって定理 4.3 が成り立つ時、補題により

$$\begin{aligned} Prob_S(\sigma) &= \frac{\prod_{v \in S_n} h_{S_n}(v)/n!}{\prod_{v \in S_{n-1}} h_{S_{n-1}}(v)/(n-1)!} \cdot \frac{\prod_{v \in S_{n-1}} h_{S_{n-1}}(v)/(n-1)!}{\prod_{v \in S_{n-2}} h_{S_{n-2}}(v)/(n-2)!} \cdots 1 \\ &= \frac{\prod_{v \in S_n} h_{S_n}(v)}{n!} \end{aligned}$$

よって定理 4.2 が成り立つ。□

以上より定理 4.3 を示せば S について hook 公式を示せたことになるが全ての d-complete poset に対して hook 公式を示したいので定理 4.3 を次のように修正する。

定理 4.3' 任意の d-complete poset S とその任意の corner ω に対して次の式が成り立つ。

$$prob_S(\omega) = \frac{1}{n} \prod_{v \in W_S(\omega)} \left(1 + \frac{1}{h_S(v) - 1}\right). \quad (2)$$

今までの議論によりこれが言えれば全ての d-complete poset に対して hook 公式が言えたことになる。この (2) を S の corner ω における corner hook 公式と呼ぶ。又は単に (S, ω) の corner hook 公式と呼ぶ。

注意 S の任意の corner に対して corner hook 公式が示せても S の hook 公式が示せるわけではない。 S の hook 公式を示すためには S の任意の filter S' の任意の corner に対して corner hook 公式を示さねばならない。

5 Peterson の定理

本節では d-complete poset に対応する概念である (simply-laced な) Weyl 群の dominant minuscule 元を定義し、d-complete poset の標準盤の総数が hook 公式で与えられると言う定理と同値な Peterson の定理を紹介する。本節では d-complete poset を P で表す。

任意の Dynkin 図形 Γ を固定する。 Γ に対応する Kac-Moody Lie 環を考える。以下、real root を単に root と呼ぶ。root α の height を $ht(\alpha)$ と表す。 $\{\alpha_1, \dots, \alpha_l\}$ を対応する simple system とし、 $1 \leq i \leq l$ で s_i を α_i に対応

する simple reflection とする。 Φ^+ を positive root system, Φ^- を negative root system とし、 $w \in W$ に対して $\Phi(w) := \Phi \cap w\Phi^-$ とする。よく知られるように w の最短表示の長さ $l(w) = |\Phi(w)|$ であり、 $w = s_{i_n} s_{i_{n-1}} \cdots s_{i_1}$ (最短表示) に対して

$$\Phi(w) = \{s_{i_n} \cdots s_{i_{k+1}} \alpha_{i_k} \mid 1 \leq k \leq n\}$$

である。 $\beta_k := s_{i_n} \cdots s_{i_{k+1}} \alpha_{i_k}$ とする。

$w = s_{i_n} s_{i_{n-1}} \cdots s_{i_1} \in W$ を最短表示とする。 integral weight λ に対して

$$p = 1, 2, \dots, n \text{ で } \langle s_{i_{p-1}} \cdots s_{i_2} s_{i_1}(\lambda), \alpha_{i_p}^\vee \rangle = 1$$

なる条件が成り立つ時、 w を λ -minuscule と呼ぶ。この時 λ に対して $s_{i_1}, s_{i_2}, \dots, s_{i_n}$ の順に作用させていくと

$$\lambda \mapsto \lambda - \alpha_{i_1} \mapsto \lambda - \alpha_{i_1} - \alpha_{i_2} \mapsto \cdots \mapsto \lambda - \alpha_{i_1} - \alpha_{i_2} - \cdots - \alpha_{i_n}$$

である。 $w \in W$ がある λ で λ -minuscule である時、 w を minuscule と言い、特に、 λ が dominant である時、 dominant minuscule という。 $w \in W$ に対して w の最短表示は一意的ではない。 w の λ -minuscule 性は w の最短表示の仕方に依存する性質に見えるが実は $w = s_{i_1} \cdots s_{i_n} = s_{j_1} \cdots s_{j_n}$ に対して一方の表示に対して λ -minuscule ならもう一方の表示についても λ -minuscule である (cf. [St2], Prop. 2.1)。

以上の準備の下、次の定理を紹介する。

定理 5.1 (Peterson) 任意の dominant minuscule 元 $w \in W$ に対して、 $r(w)$ を w の最短表示の総数とすると次の式が成り立つ。

$$r(w) = \frac{l(w)!}{\prod_{\beta \in \Phi(w)} ht(\beta)}$$

Peterson の定理は 1989 年頃 Peterson によって証明されたと言われており、[St3] において引用されている。この定理は Proctor[P1][P2] と Stembridge[St2] の結果により、d-complete poset の hook 公式 と同値であることが言える。以下、それを説明する。

まず、任意の d-complete poset P に対して 対応する simply-laced な dominant minuscule 元 w を構成する。 P に対して 3 節の hook を構成するアルゴリズムの step1 と全く同じ方法で P にラベルをする (3 節図 6 参照)。次に step2 と同じ方法で P に座標を付ける (すなわち P の標準盤を任意に一つ取る)。この時、 v_k におけるラベルを i_k と表し、 $w = s_{i_n} \cdots s_{i_1}$ とすると w は次のような dominant integral weight λ に対して λ -minuscule である。 P の top tree T に対して、その極大元のラベルを $i_1, \dots, i_m$ とする。この時、 T を (順序関係を無視して) simply laced な Dynkin 図形とみなした時の fundamental weight $\omega_{i_1}, \dots, \omega_{i_m}$ に対して $\lambda = \omega_{i_1} + \cdots + \omega_{i_m}$ と取ればよい。

この対応関係について、step2 の P の座標付けは方法 P の標準盤の総数 f_P 通りあるが、どの座標付けを用いても同じ dominant minuscule 元に対応し、しかもそれら全てが w の相異なる最短表示を与え尽くす。従ってこの P と w に対して $f_P = r(w)$ である。又、 $|P| = l(w) = |\Phi(w)|$ であった。今、 $v_k \mapsto \beta_k$ ($1 \leq k \leq n$) と対応させる。実はこの時、 $h_P(v_k) = ht(\beta_k)$ である。従って

$$\frac{|P|!}{\prod_{v \in P} h_P(v)} = \frac{l(w)!}{\prod_{\beta \in \Phi(w)} ht(\beta)}$$

である。以上の議論により次の主張が成り立つ。

主張 Peterson の定理 $\Rightarrow$ d-complete poset の hook 公式

証明) Peterson の定理を仮定する。任意の d-complete poset P に対して、上の方法で P に対応する simply-laced な dominant minuscule 元 w を取る。この時、仮定及び上の議論により

$$f_P = r(w) = \frac{l(w)!}{\prod_{\beta \in \Phi(w)} ht(\beta)} = \frac{|P|!}{\prod_{v \in P} h_P(v)}$$

従って、hook 公式が成り立つ。

次に、任意の (simply-laced とは限らない) dominant minuscule 元 w に対して、対応する d-complete poset を構成する。Stembridge[St2] に従う。 $w = s_{i_n} \cdots s_{i_1}$ を最短表示とする。 $P := \{v_1, v_2, \dots, v_n\}$ に対して、 P 上の順序 $\preceq$ を次のようにして定める。 $v_p, v_q \in P$ が、 $p < q$ かつ $(s_{i_p} s_{i_q} \neq s_{i_q} s_{i_p}$ 又は $i_p = i_q)$ の時 $v_p \succ v_q$ とし、その transitive closure として関係 $\preceq$ を定めると、これは順序関係として well-defined である (cf.[St2], sec.3)。このようにして w に対応して定まる poset $(P, \preceq)$ は d-complete poset になる。

この対応関係について、 w の任意の最短表示に対して $v_1, v_2, \dots, v_n$ は P の一つの標準盤を与える。又、どの最短表示を用いても同じ d-complete poset に対応し、しかもそれら全てが P の相異なる標準盤を与え尽くす。(それは w の 'fully-commutative' 性による。一般に fully commutative 元の任意の最短表示に対して上と同様にして得られる poset は常に同じであり ([St1] Prop.2.2)、任意の minuscule 元は fully commutative である ([St2] Prop.2.1)。) 従ってこの w と P に対して $r(w) = f_P$ である。又、 $|P| = l(w) = |\Phi(w)|$ であった。今、 $v_k \mapsto \beta_k$ ($1 \leq k \leq n$) と対応させる。この時、simply-laced の場合は先ほどと同様に $h_P(v_k) = ht(\beta_k)$ であるが、multiply-laced の場合、この等式は一般には成り立たない。しかし、multiply-laced の場合の dominant minuscule 元の分類論 [St2], Th.4.2 に従って multi set として

$$\{h_P(v) | v \in P\} = \{ht(\beta) | \beta \in \Phi(w)\}$$

であることが容易に分かる。

例 B_4 型 Weyl 群 $W = \langle s_1, s_2, s_3, s_4 \rangle$ が α_1 を short root に持つものとする。すると $w = s_1 s_2 s_1 s_3 s_2 s_1 s_4 s_3 s_2 s_1$ は dominant minuscule である。この時、各 hook length と height は下図のようになる。

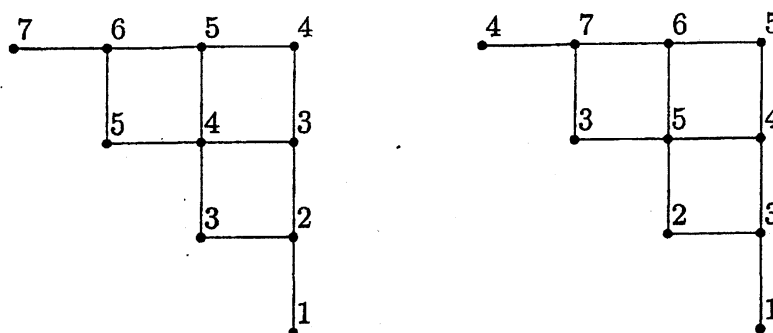


図 10: 対応する hook length と height

従っていずれの場合も

$$\frac{|P|!}{\prod_{v \in P} h_P(v)} = \frac{l(w)!}{\prod_{\beta \in \Phi(w)} ht(\beta)}$$

である。以上の議論により次の主張が成り立つ。

主張 d-complete poset の hook 公式 $\Rightarrow$ Peterson の定理

証明) d-complete poset の hook 公式を仮定する。任意の dominant minuscule 元 w に対して、上の方法で w に対応する d-complete poset P を取る。この時、仮定及び上の議論により

$$r(w) = f_P = \frac{|P|!}{\prod_{v \in P} h_P(v)} = \frac{l(w)!}{\prod_{\beta \in \Phi(w)} ht(\beta)}$$

従って、Peterson の定理が成り立つ。

参考文献

- [FRT] J.S.Frame, G. de B.Robinson, and R.M.Thrall, The hook graphs of the symmetric group, *Canad.J.Math.*6 (1954), 316-325.
- [GNW] C.Greene, A.Nijenhuis, and H.S.Wilf, A probabilistic proof of a formula for the number of Young Tableaux of a given shape, *Adv.in Math.*31 (1979), 104-109.
- [K] N.Kawanaka, Sato-Welter game and Kac-Moody Lie algebras, 数理解析研究所講究録 1190 (2001), 95-106.

- [O] S.Okamura, 一般化された標準ヤング盤を等確率でランダムに生成するアルゴリズム, 大阪大学大学院理学研究科修士論文 (2003 年 3 月)
- [P1] R.A.Proctor, Minuscule elements of Weyl groups, the numbers game, and d-complete posets, J.Algebra 213 (1999), 272-303.
- [P2] R.A.Proctor, Dynkin diagram classification of λ -minuscule Bruhat lattices and of d-complete posets, J.Algebraic Combin.9 (1999), 61-94.
- [Sa] B.E.Sagan, On selecting a random shifted Young tableau, J.Algorithms 1 (1980), 213-234.
- [St1] J.R.Stembridge, On the fully commutative elements of Coxeter groups, J.Algebraic Combin.5 (1996), 353-385.
- [St2] J.R.Stembridge, Minuscule Elements of Weyl Groups, J.Algebra 235 (2001), 722-743.
- [St3] J.R.Stembridge, Quasi-Minuscule Quotients and Reduced Words for Reflections, J.Algebraic Combin.9 (2001), 275-293.