

K-hive 上の A 型結晶構造に関するアルゴリズムとその実装

Algorithms and implementations for crystals of K-hives of type A

東邦大学大学院理学研究科 成澤 翔大

SHOTA NARISAWA

GRADUATE SCHOOL OF SCIENCE, TOHO UNIVERSITY *1

東邦大学理学部 白柳 潔

KIYOSHI SHIRAYANAGI

FACULTY OF SCIENCE, TOHO UNIVERSITY *2

Abstract

K-hives are combinatorial objects, which realize the crystal basis of an irreducible highest weight module over a quantized enveloping algebra of type A . In this paper, we provide a set of algorithms to compute the crystal structure on a set of K-hives. We also implemented these algorithms and created a Python package called *khive-crystal*. Then we give some examples of performing this package.

1 はじめに

$\mathfrak{g}$ を対称化可能 Kac-Moody リー環とする. 量子展開環 $U_q(\mathfrak{g})$ は非可換非余可換な $\mathbb{C}(q)$ 上の結合代数であり, $\mathfrak{g}$ の包絡環 $U_q(\mathfrak{g})$ の q -量子化である [2, 3]. $U_q(\mathfrak{g})$ のある加群のクラスは, $q = 0$ の極限において, 結晶基底というある特殊な基底をもつ. これは色付き有向グラフの構造を持っており, 様々な計算を組合せ論として計算可能にする. また, $q = 0$ という特殊な場合の基底ではあるが, そこから元の加群の情報の多くが復元できる. すなわち, 結晶基底は量子展開環の表現論に対する組合せ論的なアプローチを提供する. 例えば, $U_q(\mathfrak{g})$ 上の加群 M_1, M_2 のテンソル積 $M_1 \otimes M_2$ の既約分解を考える. テンソル積の各成分の M_1, M_2 はそれぞれ結晶基底を持つとすると, $M_1 \otimes M_2$ も結晶基底を持つことがわかる. このとき, $M_1 \otimes M_2$ の既約分解は, $M_1 \otimes M_2$ の結晶基底のグラフの連結成分を求めることに対応する. また実際に計算する際には, 具体的な実現を使うことが多いが, 古典型の既約最高ウェイト加群の結晶基底に対しては, ヤング盤の類似を用いた実現が知られている [4]. 特に A 型に対しては, 半標準盤を用いた実現が存在する.

[8, 9] で導入された hive は, 1 辺が n の正三角形に 1 辺が 1 の正三角形を敷き詰めた図形の各頂点にラベリングしたものである. 本稿で扱う K-hive は, hive にいくつか条件を足した図形であり, 半標準盤と一対一に対応する [6]. hive には多くの情報が組み込まれているため, ヤング盤では見えにくい情報を明らかにし, 対象の構造や議論の見通しをよくするなどの利点が期待される. 例えば, K-hive は (Stretched) Kostka coefficients などに応用がある [6]. また hive に別の条件を付与した LR-hive という図形も存在し, これは (Stretched) Littlewood-Richardson coefficients などに応用がある [7, 14].

*1 7520002n@st.toho-u.jp

*2 kiyoshi.shirayanagi@is.sci.toho-u.ac.jp

[11] では A 型の既約最高ウェイト加群の結晶基底が K-hive によって実現できることを示した. 本稿では, これらの結果に対し, 結晶構造に関する写像を計算するアルゴリズムを与える [12]. またこれらのアルゴリズムを *khive-crystal* という Python のパッケージとして実装し, その実行例を示す [10].

2 準備

2.1 量子展開環と結晶基底

$n \geq 2$ に対し, $\mathfrak{sl}_n$ を A_{n-1} 型のリー環とする. $I = \{1, 2, \dots, n-1\}$, $[n] = \{1, 2, \dots, n\}$ とする. また $\mathfrak{h} = \{x \in \mathfrak{sl}_n \mid x = \text{diag}(x_i \mid i \in [n])\} \subset \mathfrak{sl}_n$ とする. $i \in [n]$ に対し, 線型写像 $\epsilon_i: \mathfrak{h} \rightarrow \mathbb{C}$ を $\epsilon_i(h) = c_i$ によって定義する. ここで, $h = \text{diag}(c_j \mid j \in [n]) \in \mathfrak{h}$ とする. また $i \in I$ に対し, $\alpha_i = \epsilon_i - \epsilon_{i+1}$ とする. $\Pi = \{\alpha_i\}_{i \in I} \subset \mathfrak{h}^*$ の元を単純ルートとよび, $\Pi^\vee = \{h_i\}_{i \in I} \subset \mathfrak{h}$ の元を単純コルートとよぶ. $i \in I$ に対し, $\Lambda_i = \epsilon_1 + \epsilon_2 + \dots + \epsilon_i \in \mathfrak{h}^*$ とし, 基本ウェイトとよぶ. $P = \bigoplus_{i \in I} \mathbb{Z}\Lambda_i$, $P^+ = \bigoplus_{i \in I} \mathbb{Z}_{\geq 0}\Lambda_i$, $P^\vee = \bigoplus_{i \in I} \mathbb{Z}h_i$ とする. P をウェイト格子とよぶ. P^+ の元を支配的ウェイトとよぶ.

q を不定元とし, $U_q(\mathfrak{sl}_n)$ を $\mathfrak{sl}_n$ の量子展開環とする. 支配的ウェイト $\lambda \in P^+$ に対し, $V(\lambda)$ を $U_q(\mathfrak{sl}_n)$ 上の最高ウェイト λ の既約最高ウェイト加群とする. $(\mathcal{L}(\lambda), \mathcal{B}(\lambda))$ を $V(\lambda)$ の結晶基底とする.

結晶基底の組み合わせ論的な性質を抽出した概念である結晶は次で定義される.

定義 1

集合 B が $U_q(\mathfrak{sl}_n)$ -結晶とは, 写像 $\text{wt}: B \rightarrow P$, $e_i, f_i: B \rightarrow B \cup \{0\}$, $\varepsilon_i, \varphi_i: B \rightarrow \mathbb{Z} \cup \{-\infty\}$ ($i \in I$) が存在して, 以下を満たすものである. $i \in I$, $b, b' \in B$ とする.

1. $\varphi_i(b) = \varepsilon_i(b) + \text{wt}(b)(h_i)$,
2. $e_i b \in B$ のとき, $\text{wt}(e_i b) = \text{wt}(b) + \alpha_i$
3. $f_i b \in B$ のとき, $\text{wt}(f_i b) = \text{wt}(b) - \alpha_i$,
4. $e_i b \in B$ のとき, $\varepsilon_i(e_i b) = \varepsilon_i(b) - 1$, $\varphi_i(e_i b) = \varphi_i(b) + 1$ if,
5. $f_i b \in B$ のとき, $\varepsilon_i(f_i b) = \varepsilon_i(b) + 1$, $\varphi_i(f_i b) = \varphi_i(b) - 1$,
6. $f_i b = b' \Leftrightarrow b = e_i b'$.

特に, $\mathcal{B}(\lambda)$ は $U_q(\mathfrak{sl}_n)$ -結晶である.

定義 2

B を $U_q(\mathfrak{sl}_n)$ -結晶とする. B の結晶グラフとは, B の要素を頂点とし, $f_i b = b'$ ($i \in I, b, b' \in B$) のとき, $b \xrightarrow{i} b'$ と辺を定めることによって得られる色付有向グラフである.

結晶のテンソル積は次で定義される.

定義 3

B_1, B_2 を結晶とする. B_1 と B_2 のテンソル積 $B_1 \otimes B_2$ とは, 集合 $B_1 \times B_2$ に対し以下の方法で結晶構造を定めたものをいう.

1. $\text{wt}(b_1 \otimes b_2) = \text{wt}(b_1) + \text{wt}(b_2)$,
2. $\varepsilon_i(b_1 \otimes b_2) = \max(\varepsilon_i(b_1), \varepsilon_i(b_2) - \text{wt}(b_1)(h_i))$,

$$3. \varphi(b_1 \otimes b_2) = \max(\varphi(b_2), \varphi(b_1) + \text{wt}(b_2)(h_i)),$$

$$4. e_i(b_1 \otimes b_2) = \begin{cases} e_i b_1 \otimes b_2 & \varphi_i(b_1) \geq \varepsilon_i(b_2), \\ b_1 \otimes e_i b_2 & \varphi_i(b_1) < \varepsilon_i(b_2), \end{cases}$$

$$5. f_i(b_1 \otimes b_2) = \begin{cases} f_i b_1 \otimes b_2 & \varphi_i(b_1) > \varepsilon_i(b_2), \\ b_1 \otimes f_i b_2 & \varphi_i(b_1) \leq \varepsilon_i(b_2). \end{cases}$$

結晶の射は次で定義する. 以下では次の意味で, ある K-hive の集合が既約最高ウェイト加群の結晶基底と同型になることを述べる.

定義 4

B_1, B_2 を $U_q(\mathfrak{sl}_n)$ -結晶とする. $\Psi: B_1 \rightarrow B_2$ が結晶の射であるとは, 写像 $\Psi: B_1 \cup \{0\} \rightarrow B_2 \cup \{0\}$ が以下を満たすことをいう.

1. $b \in B_1$ に対し, $\Psi(b) \in B_2$ のとき, $\text{wt}(\Psi(b)) = \text{wt}(b)$, $\varepsilon_i(\Psi(b)) = \varepsilon_i(b)$, $\varphi_i(\Psi(b)) = \varphi_i(b)$,
2. $b \in B_1$ に対し, $\Psi(b), \Psi(e_i b), \Psi(f_i b) \in B_2$ のとき, $f_i \Psi(b) = \Psi(f_i b)$, $e_i \Psi(b) = \Psi(e_i b)$,
3. $\Psi(0) = 0$.

結晶の射 $\Psi: B_1 \rightarrow B_2$ は, $\Psi: B_1 \cup \{0\} \rightarrow B_2 \cup \{0\}$ が単射のとき, 埋め込みとよぶ. また結晶の射 $\Psi: B_1 \rightarrow B_2$ は, $\Psi: B_1 \cup \{0\} \rightarrow B_2 \cup \{0\}$ が全単射のとき, 同型射とよぶ. 同型射 $\Psi: B_1 \rightarrow B_2$ が存在するとき, $B_1 \cong B_2$ とかく.

2.2 K-hives

hive は T.Tao と A.Knutson によって導入された, 1 辺が $n \in \mathbb{Z}_{>0}$ の正三角形に 1 辺が 1 の正三角形を敷き詰め, その各頂点に実数をラベリングした図形である [8, 9]. hive にはあらかじめ多くの情報が組み込まれており, またさまざまな形式を持つ. ここでは主に hive にいくつかの条件を付与した K-hive という図形の upright gradient representation という形式を使用する. 詳しくは [14] を参照せよ.

$n \in \mathbb{Z}_{\geq 0}$ とする. $\tilde{\lambda} = (\tilde{\lambda}_1, \tilde{\lambda}_2, \dots, \tilde{\lambda}_n) \in \mathbb{Z}_{\geq 0}^n$ とする. $m \in \mathbb{Z}_{\geq 0}$ に対し, $\tilde{\lambda}_1 + \dots + \tilde{\lambda}_n = m$ を満たすとき, $\tilde{\lambda}$ を m の composition という. また composition $\tilde{\lambda}$ が, $\tilde{\lambda}_1 \geq \dots \geq \tilde{\lambda}_n \geq 0$ を満たすとき, $\tilde{\lambda}$ を m の partition という. m の partition $\tilde{\lambda}$ が $\tilde{\lambda}_i = k$ ($1 \leq i \leq l \leq n$), $\tilde{\lambda}_i = 0$ ($l < i \leq n$) を満たすとき, $\tilde{\lambda} = (k^m)$ とかく. 特に, 混同の恐れがないときは (0^n) を単に 0 とかく. また, $\ell(\tilde{\lambda})$ で $\tilde{\lambda}$ の長さを表す.

$\mu \in P$ に対し, $\tilde{\mu}_1 \epsilon_1 + \dots + \tilde{\mu}_n \epsilon_n = \mu$ となる composition $\tilde{\mu}$ が存在する. また $\lambda \in P^+$ に対し, $\tilde{\lambda}_1 \epsilon_1 + \dots + \tilde{\lambda}_n \epsilon_n = \lambda$ となる partition $\tilde{\lambda}$ が存在する. $\xi \in P$ を $V(\lambda)$ のウェイトとすると, $\xi = \lambda - \sum_{i \in I} k_i \alpha_i \in P$ ($k_i \in \mathbb{Z}$) とかける. このとき ξ に対し, $\tilde{\xi}_1 \epsilon_1 + \tilde{\xi}_2 \epsilon_2 + \dots + \tilde{\xi}_n \epsilon_n = \xi$, $\sum_{k=1}^n \tilde{\xi}_k = \sum_{k=1}^n \tilde{\lambda}_k$ を満たす composition $\tilde{\xi}$ が存在する. 以下記号を濫用して, $\mu \in P$ を表す composition $\tilde{\mu}$ を μ と書く. 同様に $\lambda \in P^+$ を表す partition $\tilde{\lambda}$ を λ と書く.

定義 5

$n \in \mathbb{Z}_{\geq 0}$ とする. $\alpha = (\alpha_i)_i, \beta = (\beta_i)_i, \gamma = (\gamma_i)_i \in \mathbb{Z}_{\geq 0}^n$, $(U_{ij})_{1 \leq i < j \leq n} \in \mathbb{Z}^{n(n-1)/2}$ とする. upright gradient representation ([14]) のサイズ n の integer hive とは, $(\alpha, \beta, \gamma, (U_{ij})_{1 \leq i < j \leq n})$ で

$$\beta_k = (\gamma_k + \sum_{i=1}^{k-1} U_{ik}) + (\alpha_k - \sum_{j=k+1}^n U_{kj}). \quad (1)$$

をみたすものをいう.

簡単のため, 以下 $(U_{ij})_{1 \leq i < j \leq n}$ を $(U_{ij})_{i < j}$ とかく. integer hive $(\alpha, \beta, \gamma, (U_{ij})_{i < j})$ は以下のような図形で表す.

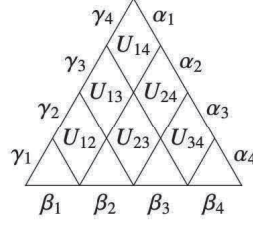


図 1: サイズ 4 の Integer hive

注意 1

[8, 9, 14] では, hive という用語は hive に rhombus inequalities と呼ばれる条件を追加したものを指す. ここでは, [5, 6, 7] の用語に従う.

$i \in [n]$ に対し,

$$U_{ii} = \beta_i - \sum_{k=1}^{i-1} U_{ki} \quad (2)$$

と定める. また

$$U_{ij} = 0 \quad \text{if } i > j \text{ or } j > n \text{ or } i < 1.$$

とする. ここでは, integer hive にいくつか条件を付与した K-hive を主に考える.

定義 6

$m, n \in \mathbb{Z}_{\geq 0}$ とする. $\alpha, \beta, \gamma \in \mathbb{Z}_{\geq 0}^n$ とする. $1 \leq i < j \leq n$ に対し, $L_{ij} = \sum_{k=1}^{j-1} U_{ik} - \sum_{k=1}^j U_{i+1,k}$ とする. integer hive $H = (\alpha, \beta, \gamma, (U_{ij})_{i < j})$ が以下を満たすとき, K-hive とよぶ.

1. α は m の partition,
2. β は m の composition,
3. $\gamma = (0^n)$,
4. $U_{ij} \geq 0 \quad (1 \leq i < j \leq n)$,
5. $L_{ij} \geq 0 \quad (1 \leq i < j \leq n)$,
6. $\beta_i \geq \sum_{k=1}^{i-1} U_{ki} \quad (i \in [n])$.

また

$$\begin{aligned} \mathcal{H}^{(n)}(\alpha, \beta, 0) &= \{H = (\alpha, \beta, 0, (U_{ij})_{i < j}) \mid H \text{ is a K-hive}\}, \\ \mathbb{H}(\alpha) &= \bigcup_{\beta} \mathcal{H}^{(n)}(\alpha, \beta, 0) \end{aligned}$$

とする. ここで和は m の composition 全体を渡る.

注意 2

$H = (\alpha, \beta, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\alpha)$ とする. このとき, $U_{ii} = \alpha_i - \sum_{j=i+1}^n U_{ij}$ が成り立つことがわかる. また, $\alpha_i = 0$ のとき, $U_{kl} \geq 0$ ($1 \leq k \leq l \leq n$) より, $U_{ij} = 0$ ($j \in [n]$) を得る.

例 1

$n = 4$, $\lambda = (3, 2, 1, 0)$, $\mu = (2, 3, 1, 0)$ とする. また $(U_{12}, U_{13}, U_{14}, U_{23}, U_{24}, U_{34}) = (1, 0, 0, 0, 0, 0)$ とする. このとき, $H \in \mathcal{H}^{(4)}(\lambda, \mu, 0) \subset \mathbb{H}(\lambda)$ は図 2 のようになる.

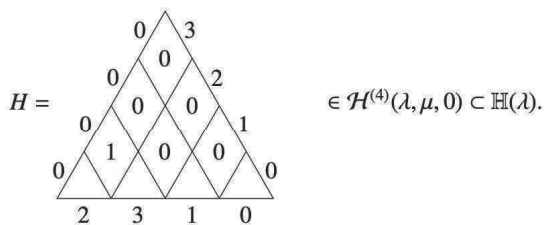


図 2: K-hive

2.3 K-hives 上の結晶構造

$\mathbb{H}(\lambda)$ ($\lambda \in P^+$) は既約最高ウェイト加群の結晶基底と結晶として同型になるが, ここではその $\mathbb{H}(\lambda)$ 上の結晶構造について解説する. $\mathbb{H}(\lambda)$ 上の結晶構造は次の 2 つの方法で導入される. 1 つの方法は, $\mathbb{H}(\lambda)$ の $\otimes_k \mathbb{H}(\Lambda_k)$ の中への結晶としての埋め込みを与えることで導入される. もう 1 つの方法は, より直接的な組合せ論的な表示を考えることで与えられる. 詳しくは [11] を参照せよ.

まず基本的な場合として, $\mathbb{H}(\Lambda_k)$ ($k \in I$) の結晶構造を考える.

定義 7 ([11])

$\nu \in I$ とする. 写像 $\text{wt}: \mathbb{H}(\Lambda_\nu) \rightarrow P$, $e_i, f_i: \mathbb{H}(\Lambda_\nu) \rightarrow \mathbb{H}(\Lambda_\nu) \cup \{0\}$, $\varepsilon_i, \varphi_i: \mathbb{H}(\Lambda_\nu) \rightarrow \mathbb{Z}_{\geq 0}$ ($i \in I$) を以下の方法で定める. $H = (\Lambda_\nu, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\Lambda_\nu)$ とする.

1. $\text{wt}(H) := \sum_{k=1}^{n-1} (\mu_k - \mu_{k+1}) \Lambda_k \in P$,
2. $\varepsilon_i(H) := \max(\mu_{i+1} - \mu_i, 0)$,
3. $\varphi_i(H) := \max(\mu_i - \mu_{i+1}, 0)$,
4. $\mu' = \sum_{k=1}^n \mu'_k \epsilon_k \in P$ とする. ここで $\mu'_i = \mu_i + 1$, $\mu'_{i+1} = \mu_{i+1} - 1$, $\mu'_k = \mu_k$ ($k \neq i, i+1$) である. また $U_{k_0, i+1} > 0$ となる $k_0 \in \{1, 2, \dots, i+1\}$ が存在するとき, $U'_{k_0, i} = U_{k_0, i} + 1$, $U'_{k_0, i+1} = U_{k_0, i+1} - 1$ とする. $k \neq k_0$, $l \neq i, i+1$ に対しては, $U'_{kl} = U_{kl}$ とする. このとき, $i \in I$ に対し, $e_i: \mathbb{H}(\Lambda_\nu) \rightarrow \mathbb{H}(\Lambda_\nu) \cup \{0\}$ を以下のように定める:

$$e_i H = \begin{cases} (\Lambda_\nu, \mu', 0, (U'_{kl})_{k < l}) & \varepsilon_i(H) > 0, \\ 0 & \varepsilon_i(H) = 0, \end{cases}$$

5. $\mu' = \sum_{k=1}^n \mu'_k \epsilon_k \in P$ とする. ここで $\mu'_i = \mu_i - 1$, $\mu'_{i+1} = \mu_{i+1} + 1$, $\mu'_k = \mu_k$ ($k \neq i, i+1$) である. また $U_{k_0, i} > 0$ となる $k_0 \in \{1, 2, \dots, i\}$ が存在するとき, $U'_{k_0, i} = U_{k_0, i} - 1$, $U'_{k_0, i+1} = U_{k_0, i+1} + 1$ とする. $k \neq k_0$, $l \neq i, i+1$ に対しては, $U'_{kl} = U_{kl}$ とする. このとき, $i \in I$ に対し, $f_i: \mathbb{H}(\Lambda_\nu) \rightarrow \mathbb{H}(\Lambda_\nu) \cup \{0\} (i \in I)$ を以下のように定める.

$$f_i H = \begin{cases} (\Lambda_\nu, \mu', 0, (U'_{kl})_{k < l}) & \varphi_i(H) > 0, \\ 0 & \varphi_i(H) = 0. \end{cases}$$

注意 3

$\nu \in I$ に対し, $H = (\Lambda_\nu, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\Lambda_\nu)$ とする. このとき, $i \in I$ に対し, $U_{ki} > 0$ となる $k \leq i$ が存在すれば, 一意であることがわかる. また同様に, $i+1 \in I$ に対し, $U_{k, i+1} > 0$ となる $k \leq i+1$ が存在すれば, 一意であることがわかる. 詳しくは [11] を参照せよ.

命題 8 ([11])

$\nu \in I$ とする. $\mathbb{H}(\Lambda_\nu)$ は定義 7 で定義した写像 wt , e_i , f_i , φ_i , ε_i によって, $U_q(\mathfrak{sl}_n)$ -結晶となる.

また $\mathbb{H}(\Lambda_k)$ ($k \in I$) が Stembridge axioms ([13, 1]) を満たすことから以下を得る.

補題 9

$k \in I$ に対し, 結晶として $\mathbb{H}(\Lambda_k) \cong B(\Lambda_k)$ が成り立つ.

次に, $\mathbb{H}(\lambda)$ の $\bigotimes_k \mathbb{H}(\Lambda_k)$ への埋め込み Ψ を定義する. 埋め込み Ψ は, $\mathbb{H}(\mu) \otimes \mathbb{H}(\Lambda_k)$ ($\mu \in P^+$, $k \in I$) への埋め込み Ψ_λ を繰り返し適用することで定義される.

定義 10 ([11])

$\lambda = \sum_{i \in I} m_i \Lambda_i \in P^+$, $N = \sum_{i \in I} m_i$ とする. $l_N = \max\{i \in I \mid m_i \neq 0\}$ とする. $H = (\lambda, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$ に対し, $H_N = (\Lambda_{l_N}, \mu^{(N)}, 0, (U_{ij}^{(N)})_{i < j})$ を次のように定める:

$$U_{ij}^{(N)} = \begin{cases} 1 & \text{if } j = \min\{j \in [n] \mid U_{ij} > 0\}, \\ 0 & \text{otherwise,} \end{cases}$$

$$\mu_k^{(N)} = \begin{cases} 1 & \text{if there exists } j \in [n] \text{ such that } U_{kj}^{(N)} > 0, \\ 0 & \text{otherwise.} \end{cases}$$

また H と H_N に対し, $H^{(N-1)} = (\lambda^{(N-1)}, \xi^{(N-1)}, 0, (V_{ij}^{(N-1)})_{i < j})$ を次のように定める: $\lambda^{(N-1)} = \lambda - \Lambda_{l_N}$, $\xi^{(N-1)} = \mu - \mu^{(N)}$, $V_{ij}^{(N-1)} = U_{ij} - U_{ij}^{(N)}$ ($1 \leq i < j \leq n$).

補題 11 ([11])

$\lambda \in P^+$ に対し, $\Psi_\lambda: \mathbb{H}(\lambda^{(N-1)}) \otimes \mathbb{H}(\Lambda_{l_N}) \rightarrow \mathbb{H}(\lambda)$ を $\Psi_\lambda(H) = H^{(N-1)} \otimes H_N$ と定義すると, Ψ_λ は単射.

Ψ_λ を繰り返し適用することにより, $\mathbb{H}(\lambda)$ から $\bigotimes_k \mathbb{H}(\Lambda_k)$ への単射を得る.

命題 12 ([11])

$\lambda \in P^+$ とする. このとき単射

$$\Psi: \mathbb{H}(\lambda) \rightarrow \bigotimes_{i \in I} \mathbb{H}(\Lambda_i)^{\otimes m_i}.$$

が存在する.

命題 13

Ψ は f_i, e_i ($i \in I$) の作用で不変である.

定義 14 ([11])

$\lambda = \sum_{i \in I} m_i \Lambda_i \in P^+$ とする. $\mathbb{H}(\lambda)$ 上の結晶構造は Ψ が結晶の射となるように定義する.

定義 15 ([11])

$\lambda \in P^+$ に対し, $H_\lambda \in \mathbb{H}(\lambda)$ を $H_\lambda = (\lambda, \lambda, 0, (0)_{i < j})$ と定義する.

定理 16 ([11])

$\lambda \in P^+$ とする. このとき結晶の同型射 $\Phi: \mathbb{H}(\lambda) \rightarrow B(\lambda)$ で $\Phi(H_\lambda) = b_\lambda$ となるものが存在する.

またより直接的に組合せ論的な方法で, $\mathbb{H}(\lambda)$ 上の結晶構造を計算することもできる.

定理 17 ([11])

$\lambda = \sum_{i \in I} m_i \Lambda_i$ とする. $H = (\lambda, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$ に対し, 写像 $\text{wt}, f_j, e_j, \varphi_j, \varepsilon_j$ ($j \in I$) は以下のよう計算できる. $j \in I$ を固定する.

1. $\text{wt}(H) = \sum_{i \in I} (\mu_i - \mu_{i+1}) \Lambda_i$,
2. $k \in \{1, 2, \dots, j\}$ に対し, $\varphi_j^{(k)}(H) = \max\{\varphi_j^{(k-1)}(H) + U_{k,j} - U_{k+1,j+1}, 0\}$ とする. ここで $\varphi_j^{(0)}$ は 0 とみなす. このとき, $\varphi_j(H) = \varphi_j^{(j)}(H)$ が成り立つ,
3. $k \in \{1, 2, \dots, j+1\}$ に対し, $\varepsilon_j^{(k)}(H) = \max\{\varepsilon_j^{(k-1)}(H) + U_{j+2-k,j+1} - U_{j+1-k,j}, 0\}$ とする. ここで $\varepsilon_j^{(0)}$ は 0 とみなす. このとき, $\varepsilon_j(H) = \varepsilon_j^{(j+1)}(H)$ が成り立つ,
4. $\varphi_j(H) = 0$ のとき, $f_j H = 0$ が成り立つ. $\varphi_j(H) \neq 0$ のとき,

$$k_{f_j H} = \min\{k \in [n] \mid \forall l \geq k, \varphi_j^{(l)}(H) > 0\}$$

とする. このとき, $f_j H = (\lambda, \mu', 0, (U'_{kl})_{k < l})$ となる. ここで

$$\begin{aligned} \mu' &= \sum_{k \neq j, j+1} \mu_k \epsilon_k + (\mu_j - 1) \epsilon_j + (\mu_{j+1} + 1) \epsilon_{j+1}, \\ U'_{kl} &= \begin{cases} U_{kl} - 1 & \text{if } k = k_{f_j H}, l = j, \\ U_{kl} + 1 & \text{if } k = k_{f_j H}, l = j + 1, \\ U_{kl} & \text{else,} \end{cases} \end{aligned}$$

5. $\varepsilon_j(H) = 0$ のとき, $e_j H = 0$ が成り立つ. $\varepsilon_j(H) \neq 0$ のとき,

$$k_{e_j H} = \min\{k \in [n] \mid \forall l \geq k, \varepsilon_j^{(l)}(H) > 0\}$$

とする. このとき, $e_j H = (\lambda, \mu', 0, (U'_{kl})_{k < l})$ となる. ここで

$$\begin{aligned} \mu' &= \sum_{k \neq j, j+1} \mu_k \epsilon_k + (\mu_j + 1) \epsilon_j + (\mu_{j+1} - 1) \epsilon_{j+1}, \\ U'_{kl} &= \begin{cases} U_{kl} + 1 & \text{if } k = j + 2 - k_{e_j H}, l = j, \\ U_{kl} - 1 & \text{if } k = j + 2 - k_{e_j H}, l = j + 1, \\ U_{kl} & \text{else,} \end{cases} \end{aligned}$$

3 K-hive 上の結晶構造に関するアルゴリズム

ここでは, $\mathbb{H}(\lambda)$ ($\lambda \in P^+$) 上の結晶構造を計算するアルゴリズムを与える. ここで $\mathbb{H}(\lambda)$ 上の結晶構造は次の2つの方法で計算できることに注意する. 1つの方法は, 定義 14 に基づいた方法であり, もう一つの方法は, 定理 17 に基づいた方法である.

アルゴリズムを考える際には, $H = (\lambda, \mu, \nu, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$ を key が $\lambda, \mu, \nu, (U_{ij})_{i < j}$ のハッシュテーブルとして考える. ここで λ の value は配列 $[\lambda_1, \lambda_2, \dots, \lambda_n]$, μ の value は配列 $[\mu_1, \mu_2, \dots, \mu_n]$, ν の value は配列 $[0, 0, \dots, 0]$, $(U_{ij})_{i < j}$ の value は 2 次元配列 $[[U_{12}, U_{13}, \dots], [U_{23}, \dots], \dots, [U_{n-1, n}]]$ である.

定義 14 に基づいたアルゴリズムを考えるために, まず $\mathbb{H}(\Lambda_k)$ 上の結晶構造を考える. $\mathbb{H}(\Lambda_k)$ ($k \in I$) 上の作用 f_i は Algorithms 1 によって計算される. また e_i の作用も同様の方法で計算できる. ここで $\text{wt}, \varphi_i, \varepsilon_i$ ($i \in I$) は定義 7 から直ちに計算されることに注意する.

Algorithm 1 Algorithm for f_i on $\mathbb{H}(\Lambda_k)$

Input: $H = (\Lambda_k, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\Lambda_k)$, $i \in I$

Output: $f_i H \in \mathbb{H}(\Lambda_k)$

```

1: if  $\max(\mu_i - \mu_{i+1}, 0) = 0$  then
2:   return 0
3: end if
4: Take  $k_0$  from  $\{k \in [i] \mid U_{k,i} > 0\}$ 
5:  $\mu_i := \mu_i - 1$ 
6:  $\mu_{i+1} := \mu_{i+1} + 1$ 
7:  $U_{k_0,i} := U_{k_0,i} - 1$ 
8:  $U_{k_0,i+1} := U_{k_0,i+1} + 1$ 
9: return  $(\Lambda_k, \mu, 0, (U_{ij})_{i < j})$ 

```

$\lambda \in P^+$ に対し, Ψ_λ は Algorithm 2 によって計算される. $a, b \in \mathbb{Z}_{\geq 0}$ に対し, $[a, b]_{\mathbb{Z}} = \{c \in \mathbb{Z} \mid a \leq c \leq b\}$ とする.

例 2

$n = 4$, $\lambda = (3, 2, 1, 0)$, $\mu = (2, 3, 1, 0)$ とする. $H = (\lambda, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$ とする, ここで $1 \leq i < j \leq 4$ に対し, $(i, j) = (1, 2)$ のとき $U_{ij} = 1$, それ以外の場合は $U_{ij} = 0$ とする. このとき, $\Psi_\lambda(H)$ は Algorithm 2 によって, 次のように計算される. $\nu = \ell(\lambda) = 3$ とする. $\lambda^{(2)} = (\lambda_1^{(2)}, \lambda_2^{(2)}, \dots, \lambda_n^{(2)})$ とする, ここで $k \in [\nu]$ のとき $\lambda_k^{(2)} = 1$, それ以外の場合は $\lambda_k^{(2)} = 0$ とする. $1 \leq i < j \leq 4$ に対し, $U_{ij}^{(2)} = U_{ij}$ とする. $\min\{l \in [4] \mid U_{1l} > 0\} = 1$ より, $U_{11}^{(2)} = 1$ また $U_{12}^{(2)} = U_{13}^{(2)} = U_{14}^{(2)} = 0$ とする. $\min\{l \in [4] \mid U_{2l} > 0\} = 2$ より, $U_{22}^{(2)} = 1$ また $U_{23}^{(2)} = U_{24}^{(2)} = 0$ とする. $\min\{l \in [4] \mid U_{3l} > 0\} = 3$ より, $U_{33}^{(2)} = 1$ また $U_{34}^{(2)} = 0$ とする. また

$$\begin{aligned} \mu_1^{(2)} &= U_{11}^{(2)} = 1, & \mu_2^{(2)} &= U_{12}^{(2)} + U_{22}^{(2)} = 1, \\ \mu_2^{(2)} &= U_{13}^{(2)} + U_{23}^{(2)} + U_{33}^{(2)} = 1, & \mu_4^{(2)} &= U_{14}^{(2)} + U_{24}^{(2)} + U_{34}^{(2)} + U_{44}^{(2)} = 0, \end{aligned}$$

$$\begin{aligned} \lambda_1^{(1)} &= \lambda_1 - \lambda_1^{(2)} = 2, & \lambda_2^{(1)} &= \lambda_2 - \lambda_2^{(2)} = 1, \\ \lambda_3^{(1)} &= \lambda_3 - \lambda_3^{(2)} = 0, & \lambda_4^{(1)} &= \lambda_4 - \lambda_4^{(2)} = 0. \end{aligned}$$

Algorithm 2 Algorithm for Ψ_λ **Input:** $H = (\lambda, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$ **Output:** $\Psi_\lambda(H)$

```

1: for  $k = 1, 2, \dots, n$  do ▷ Compute  $\lambda^{(2)}$ 
2:   if  $k \in [1, \ell(\lambda)]_{\mathbb{Z}}$  then
3:      $\lambda_k^{(2)} = 1$ 
4:   else
5:      $\lambda_k^{(2)} = 0$ 
6:   end if
7: end for
8:  $\lambda^{(2)} := (\lambda_1^{(2)}, \lambda_2^{(2)}, \dots, \lambda_n^{(2)})$ 
9:  $(U_{ij}^{(2)})_{i < j} := (U_{ij})_{i < j}$  ▷ Compute  $(U_{ij}^{(2)})_{i < j}$ 
10: for  $i = 1, 2, \dots, n - 1$  do
11:   for  $j = i + 1, i + 2, \dots, n$  do
12:     if  $j = \min\{l \in [n] \mid U_{il} > 0\}$  then
13:        $U_{ij}^{(2)} := 1$ 
14:     else
15:        $U_{ij}^{(2)} := 0$ 
16:     end if
17:   end for
18: end for
19: for  $k = 1, 2, \dots, n$  do ▷ Compute  $\mu^{(2)}$ 
20:    $\mu_k^{(2)} := \sum_{l=1}^i U_{li}^{(2)}$ 
21: end for
22:  $\mu^{(2)} := (\mu_1^{(2)}, \mu_2^{(2)}, \dots, \mu_n^{(2)})$ 
23: for  $k = 1, 2, \dots, n$  do ▷ Compute  $\lambda^{(1)}$ 
24:    $\lambda_k^{(1)} := \lambda_k - \lambda_k^{(2)}$ 
25: end for
26:  $\lambda^{(1)} := (\lambda_1^{(1)}, \lambda_2^{(1)}, \dots, \lambda_n^{(1)})$ 
27:  $(U_{ij}^{(1)})_{i < j} := (U_{ij})_{i < j}$  ▷ Compute  $(U_{ij}^{(1)})_{i < j}$ 
28: for  $i = 1, 2, \dots, n - 1$  do
29:   for  $j = i + 1, i + 2, \dots, n$  do
30:      $U_{ij}^{(1)} := U_{ij} - U_{ij}^{(2)}$ 
31:   end for
32: end for
33: for  $i = 1, 2, \dots, n$  do ▷ Compute  $\mu^{(1)}$ 
34:    $\mu_i^{(1)} = \sum_{l=1}^i U_{li}^{(1)}$ 
35: end for
36: return  $(\lambda^{(1)}, \mu^{(1)}, 0, (U_{ij}^{(1)})_{i < j}) \otimes (\lambda^{(2)}, \mu^{(2)}, 0, (U_{ij}^{(2)})_{i < j})$ 

```

とする. $1 \leq i \leq j \leq 4$ に対し, $U_{ij}^{(1)} = U_{ij} - U_{ij}^{(2)}$ とする. また

$$\begin{aligned} \mu_1^{(1)} &= U_{11}^{(1)} = 1, & \mu_2^{(1)} &= U_{12}^{(1)} + U_{22}^{(1)} = 2, \\ \mu_2^{(1)} &= U_{13}^{(1)} + U_{23}^{(1)} + U_{33}^{(1)} = 0, & \mu_4^{(1)} &= U_{14}^{(1)} + U_{24}^{(1)} + U_{34}^{(1)} + U_{44}^{(1)} = 0. \end{aligned}$$

とする。このとき、 $\Psi_\lambda(H) = (\lambda^{(1)}, \mu^{(1)}, 0, (U_{ij}^{(1)})) \otimes (\lambda^{(2)}, \mu^{(2)}, 0, (U_{ij}^{(2)}))$ を得る。図 3 を参照せよ。

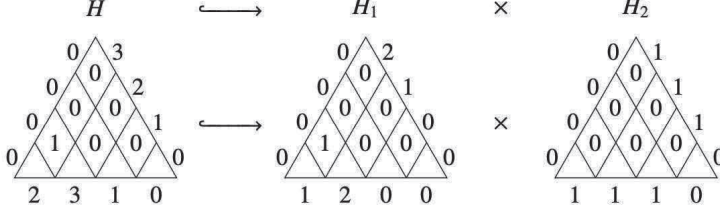


図 3: Ψ_λ の作用

Ψ は Algorithm 3 によって計算される。

Algorithm 3 Algorithm for Ψ

Input: $H = (\lambda, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$

Output: $\Psi(H)$

- 1: $H_1 \otimes H_2 := \Psi_\lambda(H)$
 - 2: $N = 2$
 - 3: **while** $H_1 \notin \mathbb{H}(\Lambda_k)$ for any $k \in I$ **do**
 - 4: $K_1 \otimes K_2 := \Psi(H_1)$
 - 5: $H := K_1 \otimes K_2 \otimes H_2 \otimes \cdots \otimes H_N$
 - 6: $N = N + 1$
 - 7: Rename H as $H = H_1 \otimes H_2 \otimes \cdots \otimes H_N$
 - 8: **end while**
 - 9: **return** $\bigotimes_{k \in N} H_k$
-

$\mathbb{H}(\lambda)$ 上の作用 f_i, e_i ($i \in I$) を計算するために、Algorithm 4 を考える。

定義 14 により、 $\mathbb{H}(\lambda)$ 上の結晶構造は $\mathbb{H}(\lambda)$ の $\bigotimes_k \mathbb{H}(\Lambda_k)$ の中への埋め込みを利用して定義される。したがって、 $\text{wt}, \varphi_i, \varepsilon_i, f_i, e_i$ ($i \in I$) は以下のアルゴリズムによって計算される。 $\lambda \in P^+$ に対し、 $H \in \mathbb{H}(\lambda)$ とする。 $\Psi(H) = H_1 \otimes H_2 \otimes \cdots \otimes H_N$ とする。ここで $\Psi(H)$ は Algorithm 3 によって計算される。このとき、 $\text{wt}(H)$ は $\text{wt}(H) = \sum_{k=1}^N \text{wt}(H_k)$ によって計算される。ここで $\text{wt}(H_k)$ は $\mathbb{H}(\Lambda_{l_k})$ に対する wt のアルゴリズムで計算される。また、 $\varphi_i(H)$ は $\varphi_i(H) = \varphi_i(H_1 \otimes H_2 \otimes \cdots \otimes H_N)$ と計算される。ここで $\varphi_i(H_1 \otimes H_2 \otimes \cdots \otimes H_N)$ は定義 3 と $\mathbb{H}(\Lambda_k)$ に対する φ_i のアルゴリズムで計算される。同様に、 $\varepsilon_i(H)$ も計算できる、また、 $f_i(H)$ は $\Psi^{-1}(f_i(H_1 \otimes H_2 \otimes \cdots \otimes H_N))$ によって計算できる。ここで $f_i(H_1 \otimes H_2 \otimes \cdots \otimes H_N)$ は定義 3 と Algorithm 1 により計算される。同様に、 $e_i(H)$ も計算できる。

また、 $\mathbb{H}(\lambda)$ 上の結晶構造は、定理 17 により組合せ論的に計算できる。Algorithms 5, 6 は、定理 17 に基づき、 φ_i, f_i ($i \in I$) をそれぞれ計算する。ここで wt は、 $H = (\lambda, \mu, 0, (U_{ij})_{i < j})$ に対し、単に $\sum_{k \in I} (\mu_k - \mu_{k+1}) \Lambda_k$ と計算できることに注意する。また ε_i, e_i ($i \in I$) についても同様に計算できる。

Algorithm 4 Algorithm for Ψ^{-1}

Input: $H = H_1 \otimes H_2 \otimes \cdots \otimes H_N \in \bigotimes_k \mathbb{H}(\Lambda_k)$, $H_k = (\lambda^{(k)}, \mu^{(k)}, 0, (U_{ij}^{(k)})_{i < j}) \in \mathbb{H}(\lambda^{(k)})$.**Output:** $\Psi^{-1}(H) \in \mathbb{H}(\lambda)$

```

1: for  $i = 1, 2, \dots, n$  do
2:    $\lambda_i := \sum_{k=1}^N \lambda_i^{(k)}$ 
3: end for
4:  $\lambda := (\lambda_1, \lambda_2, \dots, \lambda_n)$ 
5: for  $i = 1, 2, \dots, n$  do
6:    $\mu_i := \sum_{k=1}^N \mu_i^{(k)}$ 
7: end for
8:  $\mu := (\mu_1, \mu_2, \dots, \mu_n)$ 
9: for  $i = 1, 2, \dots, n-1$  do
10:   for  $j = i+1, i+2, \dots, n$  do
11:      $U_{ij} := \sum_{k=1}^N U_{ij}^{(k)}$ 
12:   end for
13: end for
14: return  $(\lambda, \mu, (0^n), (U_{ij})_{i < j})$ 

```

Algorithm 5 Algorithm for φ_i on $\mathbb{H}(\lambda)$

Input: $H = (\lambda, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$, $i \in I$ **Output:** $\varphi_i(H) \in \mathbb{Z}_{\geq 0}$

```

 $\varphi_i(H) := 0$ 
for  $k = 1, 2, \dots, i$  do
   $\varphi_i(H) := \max(U_{ki} - U_{k+1, i+1} + \varphi_i(H), 0)$ 
end for
return  $\varphi_i(H)$ 

```

Algorithm 6 Algorithm for f_i on $\mathbb{H}(\lambda)$

Input: $H = (\lambda, \mu, 0, (U_{ij})_{i < j}) \in \mathbb{H}(\lambda)$, $i \in I$ **Output:** $f_i H \in \mathbb{H}(\lambda)$

```

1: if  $\varphi_i(H) = 0$  then
2:   return 0
3: end if
4:  $F := [0]$ 
5: for  $k = 1, 2, \dots, i$  do
6:    $F := F.append(\max(U_{ki} - U_{k+1, i+1} + F[k-1], 0))$ 
7: end for
8:  $k_{f_i H} := 1$ 
9: for  $k = i, i-1, \dots, 1$  do
10:   if  $F[k] < 0$  then
11:      $k_{f_i H} := k-1$ 
12:     break
13:   end if
14: end for
15:  $\mu_i := \mu_i - 1$ 
16:  $\mu_{i+1} := \mu_{i+1} + 1$ 
17:  $U_{k_{f_i H}, i} := U_{k_{f_i H}, i} - 1$ 
18:  $U_{k_{f_i H}, i+1} := U_{k_{f_i H}, i+1} + 1$ 
19: return  $(\lambda, \mu, 0, (U_{ij})_{i < j})$ 

```

▷ Set an array

例 3

$n = 4$, $\lambda = \mu = \Lambda_1 + \Lambda_3$ とする. ここで $\Lambda_1 + \Lambda_3$ は, *partition* $(2, 1, 1, 0)$ で表せることに注意する. $H = (\lambda, \mu, 0, (0)_{k < l}) \in \mathbb{H}(\lambda)$ とする. このとき, $\mathbb{H}(\lambda)$ 上の f_1 の作用は, *Algorithm 6* によって, 次のように計算される. $i = 1$ とする. $F = [0]$ とする. このとき $U_{11} - U_{22} + F[0] = 1$ より, $F = [0, 1]$ と更新する. $k_{f_i H} = 1$ とする. $F[1] = 1 > 0$ より, $k_{f_i H} = 1$ を得る. よって, $\mu_1 := \mu_1 - 1 = 1$, $\mu_2 := \mu_2 + 1 = 2$, $U_{11} := U_{11} - 1 = 1$, $U_{12} := U_{12} + 1 = 1$, また $f_1 H = (\lambda, \mu, 0, (U_{ij})_{i < j})$ を得る. 図 4 を参照せよ.

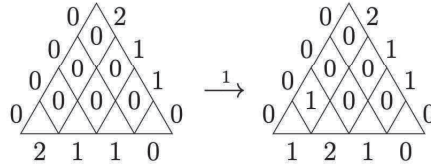


図 4: $U_q(\mathfrak{sl}_4)$ -結晶 $\mathbb{H}(\Lambda_1 + \Lambda_3)$ 上の f_1 の作用

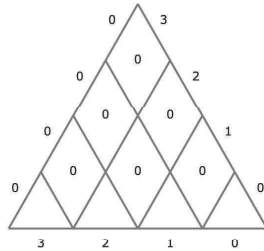
4 *khive-crystal* の実行例

$\mathbb{H}(\lambda)$ の結晶構造に関するアルゴリズムは, *khive-crystal* という Python パッケージとして実装をまとめた [10]. ここではその実行例を示す.

khive-crystal では, K-hive の定義は *khive* という AP を利用して行う. さらに, K-hive は *view* という API でグラフを表示できる. 以下に実行例を示す.

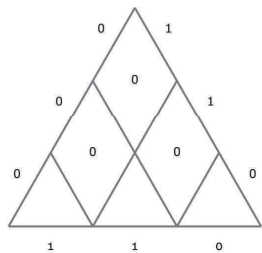
```
>> from khive.crystal import khive, view
>> H = khive(
...     n=4, alpha=[3, 2, 1, 0], beta=[3, 2, 1, 0], gamma=[0, 0, 0, 0], Uij=[[0, 0, 0], [0, 0], [0]]
... )
>> H
KHive(n=4, alpha=[3, 2, 1, 0], beta=[3, 2, 1, 0], gamma=[0, 0, 0, 0], Uij=[[0, 0, 0], [0, 0], [0]])

>> view(H)
```

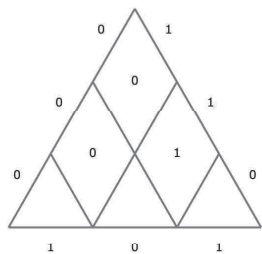


以下のコードは、 $U_q(\mathfrak{sl}_3)$ -結晶 $\mathbb{H}(\Lambda_2)$ の結晶構造を Algorithms 1 によって計算する.

```
>> from khive_crystal import e, epsilon, f, khive, phi, view
>> H = khive(n=3, alpha=[1, 1, 0], beta=[1, 1, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]])
>> view(H)
```

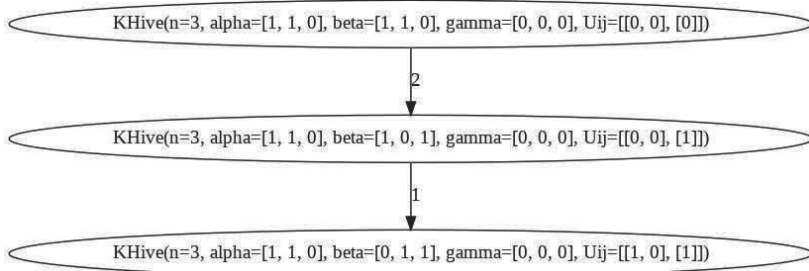


```
>> f(i=1)(H)
# None
>> view(f(i=2)(H))
```



$\mathbb{H}(\Lambda_2)$ の結晶グラフは *crystal_graph* によって計算できる. ここで *khives* は $\mathbb{H}(\Lambda_2)$ を定義する API である.

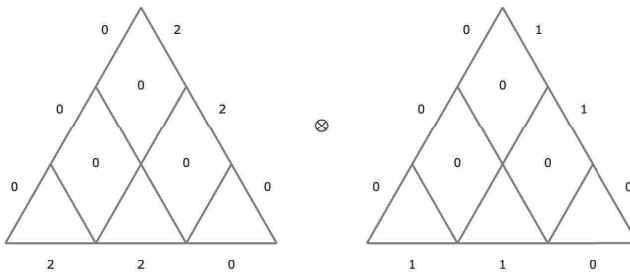
```
>> from khive_crystal import khives, crystal_graph
>> crystal_graph(khives(n=3, alpha=[1, 1, 0]))
```



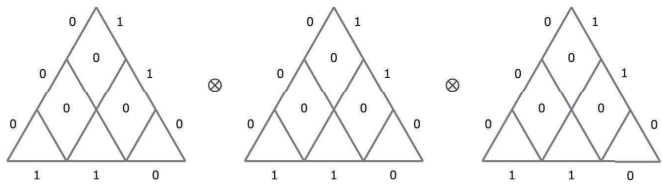
ここでこの結晶グラフの表示はオープンソースのグラフ可視化ソフトウェア *Graphviz* を利用している。

$\mathbb{H}(\lambda)$ ($\lambda \in P^+$) の結晶構造に関する *khive-crystal* の実行例を示すために、まず Algorithms 2, 3, 4 の実行例を示す。これらはそれぞれ *psi_lambda*, *psi*, *psi_inv* として実装されている。以下のコードは、 $\mathbb{H}((3, 3, 0))$ に対する、 $\Psi_{(3,3,0)}$ と Ψ の実行例である。

```
>> from khive_crystal import khive, psi, psi_lambda, view
>> H = khive(n=3, alpha=[3, 3, 0], beta=[3, 3, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]])
>> psi_lambda(H)
[
    KHive(n=3, alpha=[2, 2, 0], beta=[2, 2, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]]),
    KHive(n=3, alpha=[1, 1, 0], beta=[1, 1, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]])
]
>> view(psi_lambda(H))
```

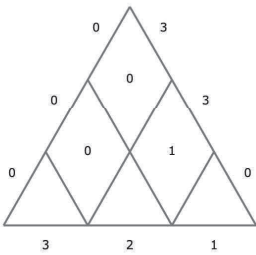


```
>> psi(H)
[
    KHive(n=3, alpha=[1, 1, 0], beta=[1, 0, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]]),
    KHive(n=3, alpha=[1, 1, 0], beta=[1, 0, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]]),
    KHive(n=3, alpha=[1, 1, 0], beta=[1, 1, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]])
]
>> view(psi(H))
```



次に, $\mathbb{H}(\lambda)$ 上の f_i の作用の実行例を示す. 以下のコードは $\mathbb{H}((3, 3, 0))$ の元に対する f_2 の作用の実行例である.

```
>> from khive.crystal import khive, psi, psi_inv, view
>> H = khive(n=3, alpha=[3, 3, 0], beta=[3, 3, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]])
>> psi_inv(f(i=2)(psi(H))) # =  $f_i(H)$ 
```



$\mathbb{H}(\lambda)$ ($\lambda \in P^+$) 上の作用 f_i は結晶構造は Algorithms 6 によっても計算できる.

```
>> from khive.crystal import khive, e, epsilon, f, phi
>> H = khive(n=3, alpha=[3, 3, 0], beta=[3, 1, 0], gamma=[0, 0, 0], Uij=[[0, 0], [0]])
>> phi(i=2)(H)
3
>> view(f(i=2)(H))
```


5 おわりに

本稿では, K-hive 上の A 型結晶構造に対し, それらを計算するアルゴリズムを与えた. また与えたアルゴリズムを Python パッケージとして実装し, その実行例を示した. K-hive 上の結晶構造は, 他の古典型やアフィン A 型などの場合に拡張できる可能性がある. 今後の課題としては, この実装を利用して, それらの拡張にアプローチすること, またその成果をアルゴリズム化や実装することなどが挙げられる.

Acknowledgements

This work was supported by the Research Institute for Mathematical Sciences, an International Joint Usage/Research Center located in Kyoto University.

参考文献

- [1] D Bump and A Schilling. *Crystal bases*. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ, 2017. Representations and combinatorics.
- [2] V. G. Drinfeld. Hopf algebras and the quantum Yang-Baxter equation. *Dokl. Akad. Nauk SSSR*, 283(5):1060–1064, 1985.
- [3] M. Jimbo. A q -difference analogue of $U(\mathfrak{g})$ and the Yang-Baxter equation. *Lett. Math. Phys.*, 10(1):63–69, 1985.
- [4] M. Kashiwara and T. Nakashima. Crystal graphs for representations of the q -analogue of classical Lie algebras. *J. Algebra*, 165(2):295–345, 1994.
- [5] R. C. King, C. Tollu, and F. Toumazet. Stretched Littlewood-Richardson and Kostka coefficients. In *Symmetry in physics*, volume 34 of *CRM Proc. Lecture Notes*, pages 99–112. Amer. Math. Soc., Providence, RI, 2004.
- [6] R. C. King, C. Tollu, and F. Toumazet. The hive model and the factorisation of kostka coefficients. *Sém. Lothar. Combin.*, 54A:B54Ah, 22 pp, 2006.
- [7] R. C. King, C. Tollu, and F. Toumazet. The hive model and the polynomial nature of stretched Littlewood-Richardson coefficients. *Sém. Lothar. Combin.*, 54A:B54Ad, 19 pp, 2006.
- [8] A. Knutson and T. Tao. The honeycomb model of $GL_n(\mathbb{C})$ tensor products. I. Proof of the saturation conjecture. *J. Amer. Math. Soc.*, 12(4):1055–1090, 1999.
- [9] Allen Knutson and Terence Tao. Apiary views of the berenstein-zelevinsky polytope, and klyachko's saturation conjecture. *arXiv preprint arXiv:9807160v1*, 1998.
- [10] S Narisawa. khive-crystal. <https://github.com/snrsw/khive-crystal>, 0.1.0.
- [11] S. Narisawa and K. Shirayanagi. Crystal structure on k-hives of type A . *Communications in Algebra*, 50(12):5266–5283, 2022.
- [12] S. Narisawa and K. Shirayanagi. Algorithms for the crystal structure on k-hives of type A . *Communications of JSSAC*, in press.
- [13] J. R. Stembridge. A local characterization of simply-laced crystals. *Trans. Amer. Math. Soc.*, 355(12):4807–4823, 2003.

- [14] I. Terada, R. C. King, and O. Azenhas. The symmetry of Littlewood-Richardson coefficients: a new hive model involutory bijection. *SIAM J. Discrete Math.*, 32(4):2850–2899, 2018.