

Inset games and their strategies

仲田 研登*

1 導入と準備

組合せゲーム理論は1930年代の[9][3]などによる必勝戦略の研究から始まったと言ってよいだろう。ゲームの局面の Sprague-Grundy function (SG-関数) は, 不偏ゲーム (impartial game) と呼ばれるクラスのゲームに対して, その必勝戦略を記述する基本的な関数になっている。本稿では不偏ゲームのみを扱うので, 組合せゲーム理論の目標は (少々大袈裟だが) 与えられたゲームの SG-関数を計算することだと言えるだろう。

1.1 ゲームの定義

ゲームの定義には様々なバリエーションが考えられるが, 本稿では次のように定義する。

定義 1.1. P を集合, \rightarrow を P 上の二項関係とする。このとき, 組 $(P; \rightarrow)$ が**ゲーム**であるとは, 次の条件 (1)(2) を満たすことである:

- (1) P の元の無限列 p_0, p_1, p_2, \dots で, $p_i \rightarrow p_{i+1}$ となるものは存在しない。
- (2) P の元 p に対して, $p \rightarrow q$ となる元 q は有限個しか存在しない。

P の元を**局面**と呼び, P の元 p, q が $p \rightarrow q$ の関係にあるとき, q は p の**次局面**であるということにする。また, $\varphi(p), \varphi_P(p) := \{q \in P \mid p \rightarrow q\}$ を p の**次局面集合**と呼ぶ。条件 (2) より, $\#\varphi(p) < \infty$ である。 $\varphi(p) = \emptyset$ なる局面 p を**終局面**と呼ぶ。

局面 $p = p_0 \in P$ を 2 人ゲームの開始局面として, 2 人のプレイヤーが交互に可能な手を打つ。

$$\begin{aligned} p_0 &\rightarrow p_1 && \text{(first player の手),} \\ p_1 &\rightarrow p_2 && \text{(second player の手),} \\ p_2 &\rightarrow p_3 && \text{(first player の手),} \\ &&& \dots \end{aligned}$$

可能な手は条件 (2) から有限通りである。また, 条件 (1) からこのような手順 (棋譜) は有限回の後に終局面 p_n にたどり着く ($\varphi(p_n) = \emptyset$)。もし n が奇数 (resp. 偶数) ならば first (resp. second) プレイヤーが “勝つ”。もしゲーム $(P; \rightarrow)$ と $(Q; \rightarrow)$ がグラフとして同型であれば, ゲーム $(P; \rightarrow)$ は $(Q; \rightarrow)$ と**ゲーム同型**であると言う。

*Kento Nakada, Graduate School of Education, Okayama University, Japan. e-mail address: nakada@okayama-u.ac.jp

例 1.2 (1-heap nim). \mathbb{N} を 0 を含む自然数の全体とする. $>$ を通常の順序関係 (大なり) とすれば, $(\mathbb{N}; >)$ はゲームである. このゲームを *1-heap nim* と呼ぶ.

1.2 SG 関数

次に, [9][3] に従って, SG 関数を定義する:

定義 1.3. 真部分集合 $S \subset \mathbb{N}$ に対して,

$$\text{mex } S := \min(\mathbb{N} \setminus S)$$

とおく. ゲーム $(P; \rightarrow)$ に対して, $\text{SG} = \text{SG}_P : P \rightarrow \mathbb{N}$ を帰納的に

$$\text{SG}(p) = \text{mex} \{ \text{SG}(q) \mid p \rightarrow q \}$$

で定める. 条件 (1) から帰納的に定まることに注意する. 関数 SG を P の *Sprague-Grundy 関数* と呼ぶ.

補足 1.4. より広い定義では, 条件 (2) を仮定しないものもある. この場合 SG 関数は一般に順序数を値にとることになる.

例 1.5 (1-heap nim). 1-heap nim $(\mathbb{N}; >)$ の SG 関数は恒等写像である:

$$\text{SG}(x) = x \quad (x \in \mathbb{N}).$$

次に, Sprague-Grundy 関数が必勝戦略を与えることを見る. これは次の命題から従う.

命題 1.1 ([9][3]). SG 関数は次を満たす:

- (1) $\text{SG}(p) > 0$ ならば, ある $q \in \varphi(p)$ に対して, $\text{SG}(q) = 0$.
- (2) $\text{SG}(q) = 0$ ならば, 任意の $r \in \varphi(q)$ に対して, $\text{SG}(r) > 0$.
- (3) q が終局面 ($\varphi(q) = \emptyset$) のとき, $\text{SG}(q) = 0$.

補足 1.6. 局面 $p \in P$ について, 以下の 2 条件は同値である:

- (1) 局面 p にいるプレイヤーは必勝戦略を持つ (勝ち筋がある).
- (2) $\text{SG}(p) > 0$.

実際, $\text{SG}(p) > 0$ なる局面 u にいるプレイヤーは, $\text{SG}(q) = 0$ なる局面 q を相手に渡すことができる. $\text{SG}(q) = 0$ なる局面 q を渡された相手は, 既に終局面にいるか, さもなくば相手が返してくる手は必ず $\text{SG}(r) > 0$ なる局面である. つまりこの手続きを繰り返せば, 条件 (1) から, 相手に必ず終局面を渡すことができる.

この命題に基づいて, $\text{SG}(p) > 0$ なる局面 p を **先手勝ち局面**, $\text{SG}(p) = 0$ なる局面 p を **先手負け局面** と呼ぶ¹.

命題 1.1 が与える必勝戦略は SG の計算を要求する. しかし, SG を定義通りに計算するのは, 終局面までのすべての SG 関数の値を必要とし, 計算量が膨大になる. つまり, 多くの場合 (ひどく一般的なゲーム P の場合), SG の計算は非現実的であると言える. 一方で, いくつかのゲームについては SG 関数の計算が簡単であることが知られている. 例えば,

¹歴史的には先手負け局面 ($\text{SG}(p) = 0$ なる局面 p) を後手勝ち局面 (後手必勝局面) と呼ぶことが多いが, この呼称は初学者には分かりにくいと筆者は考えている.

- Nim
- Sato-Welter's game
- turning turtles

などである。こうして、「SG関数が(比較的)簡単に計算できるゲームのクラスは何か」という基本的な問題意識にたどり着く。

川中宣明氏は、このようなクラスを完全可解ゲームと呼ぶことを提唱し、その中でも具体的なクラスとして、平明ゲームを定義した [4]。

1.3 Nim 和など

次に、具体的に SG 関数を記述するために用いるニム和などの基本的な演算を導入する。

\mathbb{Z} で整数の全体を表わす。整数 $a \in \mathbb{Z}$ の二進展開を

$$a = [a_i] = [a_i]_{i \in \mathbb{N}} = [\dots, a_i, \dots, a_3, a_2, a_1, a_0]$$

のように表わす²。例えば、次のようになる:

$$\begin{aligned} 11 &= 1 + 2 + 0 + 2^3 + 0 + 0 + \dots = [\dots 001011], \\ -1 &= 1 + 2 + 2^2 + 2^3 + 2^4 + 2^5 + \dots = [\dots 111111], \\ -12 &= 0 + 0 + 2^2 + 0 + 2^4 + 2^5 + \dots = [\dots 110100]. \end{aligned}$$

整数 $a = [a_i]$, $b = [b_i]$, and $c = [c_i]$ in \mathbb{Z} に対して、ニム和は次のように定義される:

$$a \oplus b = c$$

ここで、右辺は繰り上がりなしの和である:

$$a_i + b_i \equiv c_i \pmod{2}, \quad i \in \mathbb{N}.$$

ニム和は例えば次のように計算される:

$$3 \oplus 5 = [\dots 00011] \oplus [\dots 00101] = [\dots 00110] = 6.$$

代数系 $(\mathbb{Z}; \oplus, 0)$ は

$$a \oplus a = 0, \quad (a \in \mathbb{Z})$$

を満たす可換群をなす。ここで、 \mathbb{N} は \mathbb{Z} の指数 2 の部分群をなすことに注意する。また、 $-$ で通常の加法に関する逆元(マイナス)を表わすことにすれば、

$$(-1) \oplus a = -a - 1$$

²より正確には、 \mathbb{Z} を二進整数環 \mathbb{Z}_2 の部分環とみて二進展開している。したがって、二進位相に関して $-1 = \sum_{i=0}^{\infty} 2^i$ となる。

となる. また, 整数 $a \in \mathbb{Z}$ に対して,

$$N(a) := a \oplus (a - 1)$$

とおき, $a, b \in \mathbb{Z}$ に対して,

$$(a | b) = N(a \oplus b)$$

とおく. [2, ch.13], [8], [10] に従って, 次が得られる:

補題 1.2. 整数 $a, b, c \in \mathbb{Z}$ について, 以下が成り立つ:

- (1) 整数 a が 2^t ($t \in \mathbb{N}$) の倍数で, 2^{t+1} の倍数でないならば,

$$N(a) = [\dots 0 \overbrace{11 \dots 1}^{t+1}] = 2^{t+1} - 1.$$

- (2) $N(a) < 0$ となる必要十分条件は $a = 0$.

- (3) “すべての $x \in \mathbb{Z}$ について $(a | x) = (b | x)$ ” ならば $a = b$.

- (4) $(a | b) = N(a - b)$.

- (5) $(a | b) = (a + c | b + c) = (a \oplus c | b \oplus c)$.

- (6) $c > 0$ (resp. $c < 0$) ならば,

$$a \oplus \sum_{h=0}^{c-1} \oplus (a | h) = a - c \quad \left(\text{resp. } a \oplus \sum_{h=c}^{-1} \oplus (a | h) = a - c \right),$$

ここで \sum^{\oplus} はニム和についての総和.

1.4 n -heap Nim

本節の締めくくりに, Nim の SG 関数を紹介しておく.

定義 1.7. 整数 $n \geq 1$ に対して, 直積集合 \mathbb{N}^n 上の二項関係 \rightarrow を

$$(x_1, \dots, x_n) \rightarrow (y_1, \dots, y_n) \quad :\Leftrightarrow \quad \exists_1 i \in \{1, 2, \dots, n\} \text{ s.t. } \begin{cases} x_j > y_j & j = i \\ x_j = y_j & j \neq i \end{cases}$$

で定める. こうして定まるゲーム $(\mathbb{N}^n; \rightarrow)$ を n -heap nim と呼ぶ.

命題 1.3 ([1]). n -heap nim の SG 関数は次で与えられる:

$$SG(\mathbf{x}) = x_1 \oplus x_2 \oplus \dots \oplus x_n, \quad (\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{N}^n).$$

2 The Sato-Welter Game

定義 2.1. 整数 $n \geq 1$ に対して, 集合 P_n を次のように定める:

$$P_n := \{ \mathbf{x} \subseteq \mathbb{N} \mid |\mathbf{x}| = n \}.$$

$\mathbf{x}, \mathbf{y} \in P_n$ に対して, 以下が成り立つとき, $\mathbf{x} \rightarrow_A \mathbf{y}$ と表わす:

- $|\mathbf{x} \cap \mathbf{y}| = n - 1$;
- $x \in \mathbf{x} \setminus \mathbf{y}, y \in \mathbf{y} \setminus \mathbf{x}$ ならば, $y < x$.

こうして得られるゲーム $(P_n; \rightarrow_A)$ を *Sato-Welter game with n balls* と呼ぶ.

Sato-Welter game with n balls は, n 個の玉 (○) を半直線上に配置するすることで視覚的に翻訳できる. このとき, プレイヤーの手は, 玉をより左にある空いている箱に移動させることに対応する. 最後の手を行なったプレイヤーが勝ちである.

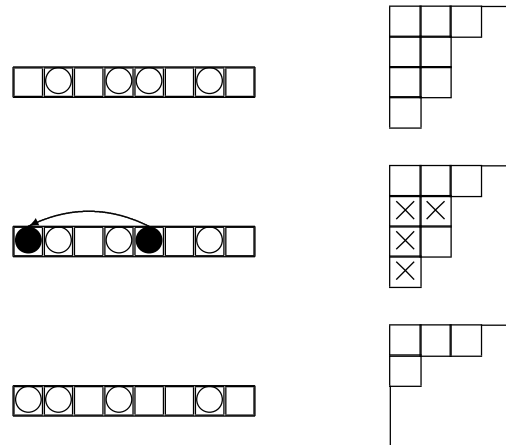
例 2.2. 例えば, 局面 $\mathbf{x} = \{3, 5\} \in P_2$ は,

$$\mathbf{x} = \{3, 5\} = \square \square \square \circ \square \circ \square \square \cdots \in P_2$$

のように表示できる (左端の箱が 0 番目であることに注意する). この局面の次局面としては, 以下のような局面が得られる:

$$\begin{aligned} \{3, 4\} &= \square \square \square \circ \circ \square \square \square \cdots, & \{2, 5\} &= \square \square \circ \square \square \circ \square \square \cdots, \\ \{3, 2\} &= \square \square \circ \circ \square \square \square \cdots, & \{1, 5\} &= \square \circ \square \square \square \circ \square \square \cdots, \\ \{3, 1\} &= \square \circ \square \circ \square \square \square \cdots, & \{0, 5\} &= \circ \square \square \square \square \circ \square \square \cdots. \\ \{3, 0\} &= \circ \square \square \circ \square \square \square \cdots, \end{aligned}$$

補足 2.3. Sato-Welter game の局面 $\mathbf{x} \in P_n$ を beta number とみなすことで³, 局面をヤング図形とみなすことができる. 例えば, 局面 $\mathbf{x} = \{3, 5\} \in P_2$ は分割 $(4, 3)$ に対応するヤング図形とみなせ, 局面 $\mathbf{x} = \{1, 3, 4, 6\} \in P_4$ は分割 $(3, 2, 2, 1)$ に対応するヤング図形とみなせる. このとき, 手は「フックを抜いて詰めること」に対応する.



補足 2.4. Sato-Welter game P_1 of 1 ball は 1-heap nim とゲーム同型である.

³ヤング図形の第 1 列のフック長を beta number と呼ぶ.

さて、局面 $\mathbf{x} = \{x_1, x_2, \dots, x_n\} \in P_n$ に対して、

$$\alpha_n(\mathbf{x}) := \sum_i^{\oplus} x_i \oplus \sum_{i < j}^{\oplus} (x_i | x_j)$$

とおく. $\alpha_n(x) \geq 0$ となることに注意する. 関数 α_n は x_1, \dots, x_n に関して対称になっていることに注意して、以下、 $\alpha_n(\mathbf{x}) = \alpha_n(x_1, x_2, \dots, x_n)$ とも表わすことにする. Sato [8] と Welter [10] は独立に次を示した:

定理 2.1 (Sato [8], and Welter [10]). Sato-Welter game $(P_n; \rightarrow_A)$ の SG 関数は次で与えられる:

$$\text{SG}(\mathbf{x}) = \alpha_n(\mathbf{x}), \quad \mathbf{x} \in P_n.$$

例 2.5. 例えば、局面 $\mathbf{x} = \{3, 5\} \in P_2$ の SG 値 $\alpha_2(\mathbf{x})$ は

$$\alpha_2(\mathbf{x}) = 3 \oplus 5 \oplus (3 | 5) = 5 \neq 0$$

となる. したがって、局面 \mathbf{x} は先手勝ち局面である. 必勝戦略に沿った手は (この場合はただ一つだが)

$$\{3, 5\} \rightarrow_A \{3, 2\}$$

である.

3 The Turning turtles

定義 3.1. 集合 P_{odd} を次のように定める:

$$P_{\text{odd}} := \bigcup_{n:\text{odd}} P_n = \bigcup_{n:\text{odd}} \{ \mathbf{x} \subseteq \mathbb{N} \mid |\mathbf{x}| = n \}. \quad (3.1)$$

$\mathbf{x} \in P_n, \mathbf{y} \in P_m$ に対して、以下が成り立つとき、 $\mathbf{x} \rightarrow_D \mathbf{y}$ と表わす:

- $n = m$ and $\mathbf{x} \rightarrow_A \mathbf{y}$; or • $n = m + 2$ and $\mathbf{x} \supset \mathbf{y}$.

こうして得られるゲーム $(P_{\text{odd}}; \rightarrow_D)$ を *turning turtles* と呼ぶ.

turning turtles は、カメを半直線上に配置することで視覚的に翻訳できる. カメは‘起きている’と‘寝ている’の2通りの状態をとることができ、奇数匹のカメが起きているとする. このとき、プレイヤーの手は、2匹のカメを選んで裏返すことに対応する (ただし2匹のうち右のカメは起きているカメを選ばなければならない). 最後の手を行なったプレイヤーが勝ちである. ただし、起きているカメと寝ているカメを描き分けるのは大変なので、本稿では Sato-Welter ゲームとの類似で、起きているカメを玉 (○) で、寝ているカメを空いている箱で表すことにする.

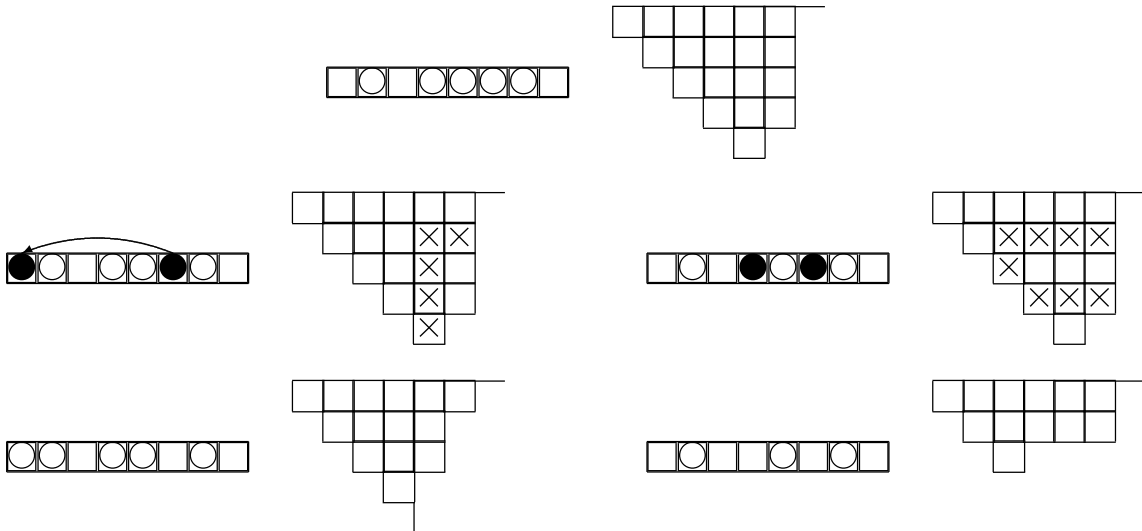
例 3.2. 例えば、局面 $\mathbf{x} = \{2, 3, 5\} \in P_{\text{odd}}$ は、

$$\mathbf{x} = \{2, 3, 5\} = \boxed{} \boxed{} \boxed{\circ} \boxed{\circ} \boxed{} \boxed{\circ} \boxed{} \boxed{} \cdots \in P_{\text{odd}}$$

のように表示できる (左端の箱が 0 番目であることに注意する). この局面の次局面としては、以下のような局面が得られる:

$$\begin{aligned}
 \{2, 3, 4\} &= \square\square\bigcirc\bigcirc\bigcirc\square\square\square\cdots, & \{1, 3, 5\} &= \square\bigcirc\square\bigcirc\square\bigcirc\square\square\cdots, \\
 \{2, 3, 1\} &= \square\bigcirc\bigcirc\bigcirc\square\square\square\square\cdots, & \{0, 3, 5\} &= \bigcirc\square\square\bigcirc\square\bigcirc\square\square\cdots, \\
 \{2, 3, 0\} &= \bigcirc\square\bigcirc\bigcirc\square\square\square\square\cdots, & \{5\} &= \square\square\square\square\bigcirc\square\square\cdots, \\
 \{2, 1, 5\} &= \square\bigcirc\bigcirc\square\square\bigcirc\square\square\cdots, & \{3\} &= \square\square\square\bigcirc\square\square\square\cdots, \\
 \{2, 0, 5\} &= \bigcirc\square\bigcirc\square\square\bigcirc\square\square\cdots, & \{2\} &= \square\square\bigcirc\square\square\square\square\cdots.
 \end{aligned}$$

補足 3.3. turning turtles の局面 $\mathbf{x} \in P_{\text{odd}}$ を相異分割 (strict partition) とみなすことで、局面をシフトヤング図形とみなすことができる. 例えば、局面 $\mathbf{x} = \{2, 3, 5\} \in P_{\text{odd}}$ は相異分割 $(5, 3, 2)$ に対応するシフトヤング図形とみなせ、局面 $\mathbf{x} = \{1, 3, 4, 5, 6\} \in P_{\text{odd}}$ は相異分割 $(6, 5, 4, 3, 1)$ に対応するシフトヤング図形とみなせる. このとき、手は「フックを抜いて詰めること」に対応する⁴.



(0 番と 5 番のカメを裏返す (左のカメが寝ている)) (3 番と 5 番のカメを裏返す (左のカメが起きている))
 (5 番の玉を 0 番の箱に動かすように見える) (3 番と 5 番の玉を取り除くように見える)

定理 3.1 ([1]). turning turtles $(P_{\text{odd}}; \rightarrow_D)$ の SG 関数は次で与えられる:

$$SG(\mathbf{x}) = x_1 \oplus x_2 \oplus \cdots \oplus x_n, \quad \mathbf{x} = \{x_1, x_2, \dots, x_n\} \in P_{\text{odd}}.$$

例 3.4. 例えば、局面 $\mathbf{x} = \{2, 3, 5\} \in P_{\text{odd}}$ の SG 値は、

$$\alpha(\mathbf{x}) = 2 \oplus 3 \oplus 5 = 4 \neq 0$$

となる, したがって、局面 \mathbf{x} は先手勝ち局面である. 必勝戦略に沿った手は (この場合はただ一つだが)

$$\{2, 3, 5\} \rightarrow_D \{2, 3, 1\}.$$

⁴シフトヤング図形のフックをバーと呼ぶこともある.

3.1 Turning turtles のバリエーション

turning turtles と呼ばれるゲームには、少なくとも以下の3通りのバリエーションがある。

- (1) 起きているカメを奇数匹として、一度に裏返すカメを2匹とするもの
- (2) 起きているカメを偶数匹として、一度に裏返すカメを2匹とするもの
- (3) 起きているカメを任意匹として、一度に裏返すカメを2匹か1匹とするもの。

本節で定義したのは(1)であるが、(2)や(3)も(1)と同型であることは簡単に示される。

定義 3.5. 集合 P_{even} を次のように定める:

$$P_{\text{even}} := \bigcup_{n:\text{even}} P_n = \bigcup_{n:\text{even}} \{ \mathbf{x} \subseteq \mathbb{N} \mid |\mathbf{x}| = n \}. \quad (3.2)$$

$\mathbf{x} \in P_n, \mathbf{y} \in P_m$ に対して、以下が成り立つとき、 $\mathbf{x} \rightarrow_{\text{even}} \mathbf{y}$ と表わす:

- $n = m$ and $\mathbf{x} \rightarrow_{\text{A}} \mathbf{y}$; or • $n = m + 2$ and $\mathbf{x} \supset \mathbf{y}$.

こうして得られるゲーム $(P_{\text{even}}; \rightarrow_{\text{even}})$ も *turning turtles* と呼ぶ。

命題 3.2. $f : P_{\text{even}} \rightarrow P_{\text{odd}}$ を

$$f(\{x_1, \dots, x_n\}) = \begin{cases} \{x_1, \dots, x_n\} \cup \{0\} & 0 \notin \{x_1, \dots, x_n\} \\ \{x_1, \dots, x_n\} \setminus \{0\} & 0 \in \{x_1, \dots, x_n\} \end{cases}$$

で定めると、 f はゲーム同型。

定義 3.6. 集合 P_{B} を次のように定める:

$$P_{\text{B}} := \bigcup_{n \in \mathbb{N}} P_n = \bigcup_{n \in \mathbb{N}} \{ \mathbf{x} \subseteq \mathbb{N} \mid |\mathbf{x}| = n \}. \quad (3.3)$$

$\mathbf{x} \in P_n, \mathbf{y} \in P_m$ に対して、以下が成り立つとき、 $\mathbf{x} \rightarrow_{\text{B}} \mathbf{y}$ と表わす:

- $n = m$ and $\mathbf{x} \rightarrow_{\text{A}} \mathbf{y}$; • $n = m + 1$ and $\mathbf{x} \supset \mathbf{y}$; or • $n = m + 2$ and $\mathbf{x} \supset \mathbf{y}$.

こうして得られるゲーム $(P_{\text{B}}; \rightarrow_{\text{B}})$ も *turning turtles* と呼ぶ。

命題 3.3. $f : P_{\text{B}} \rightarrow P_{\text{odd}}$ を

$$f(\{x_1, \dots, x_n\}) = \begin{cases} \{0, x_1 + 1, \dots, x_n + 1\} & n : \text{even} \\ \{x_1 + 1, \dots, x_n + 1\} & n : \text{odd} \end{cases}$$

で定めると、 f はゲーム同型。

本稿で3種(1)(2)(3)のturning turtlesのうち(1)を採用しているのは、この定義が図形化の際に整合性が高いからである。

4 主結果

本節では, この研究の主たる対象である inset game を導入し, その SG 関数を紹介する. まず, turning turtles P_{odd} が Sato-Welter game の “patchwork” として定義されたことに注目して, inset game を定義する.

定義 4.1. 整数 $N \geq 2$ に対して, 集合 $P_{N,1}$ を次のように定める:

$$P_{N,1} := P_N \cup P_1.$$

このとき, 局面 $\mathbf{x} \in P_n, \mathbf{y} \in P_m$ ($n, m \in \{1, N\}$) に対して, 手 $\mathbf{x} \rightarrow \mathbf{y}$ を次で定める:

- $n = m$ and $\mathbf{x} \rightarrow_{\text{AY}} \mathbf{y}$; or • $n = N, m = 1$ and $\mathbf{x} \supset \mathbf{y}$.

こうして定まるゲーム $(P_{N,1}; \rightarrow)$ を inset game と呼ぶことにする.

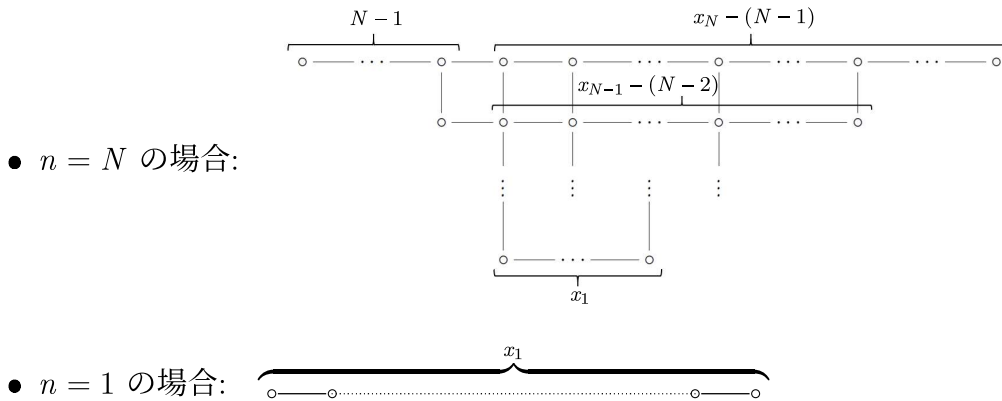
例 4.2. 例えば, $N = 4$ とすると, 局面 $\mathbf{x} = \{1, 2, 4, 5\} \in P_{4,1}$ は,

$$\mathbf{x} = \{1, 2, 4, 5\} = \boxed{} \boxed{\circ} \boxed{\circ} \boxed{} \boxed{\circ} \boxed{\circ} \boxed{} \boxed{} \cdots \in P_{4,1}$$

のように表示できる (左端の箱が 0 番目であることに注意する). この局面の次局面としては, 以下のような局面が得られる:

$$\begin{aligned} \{1, 2, 4, 3\} &= \boxed{} \boxed{\circ} \boxed{\circ} \boxed{\circ} \boxed{} \boxed{} \boxed{} \cdots, & \{0, 2, 4, 5\} &= \boxed{\circ} \boxed{} \boxed{\circ} \boxed{} \boxed{\circ} \boxed{} \boxed{} \cdots, \\ \{1, 2, 4, 0\} &= \boxed{\circ} \boxed{\circ} \boxed{} \boxed{\circ} \boxed{} \boxed{} \boxed{} \cdots, & \{5\} &= \boxed{} \boxed{} \boxed{} \boxed{} \boxed{\circ} \boxed{} \boxed{} \cdots, \\ \{1, 2, 3, 5\} &= \boxed{} \boxed{\circ} \boxed{\circ} \boxed{\circ} \boxed{} \boxed{\circ} \boxed{} \cdots, & \{4\} &= \boxed{} \boxed{} \boxed{} \boxed{\circ} \boxed{} \boxed{} \cdots, \\ \{1, 2, 0, 5\} &= \boxed{\circ} \boxed{\circ} \boxed{} \boxed{} \boxed{} \boxed{\circ} \boxed{} \cdots, & \{2\} &= \boxed{} \boxed{} \boxed{\circ} \boxed{} \boxed{} \boxed{} \cdots, \\ \{1, 0, 4, 5\} &= \boxed{\circ} \boxed{\circ} \boxed{} \boxed{} \boxed{\circ} \boxed{\circ} \boxed{} \cdots, & \{1\} &= \boxed{\circ} \boxed{} \boxed{} \boxed{} \boxed{} \boxed{} \cdots. \end{aligned}$$

補足 4.3. Sato-Welter game や turning turtles をヤング図形やシフトヤング図形で表わしたように, inset game も図形で表わせる. この図形を inset と呼ぶ⁵. 局面 $\mathbf{x} = \{x_1, x_2, \dots, x_n\} \in P_{N,1}$ ($x_1 < x_2 < \dots < x_n$) に対応した inset は次のような Hasse 図で定まる図形である:



⁵研究の背景としては話は逆で, inset と呼ばれる図形が先にあり, それをリボン表示したものが上述のものである.

$n = N$ の場合の図形の‘右半分’はヤング図形になっていることに注意する. また, $N = 2$ の場合は高々2行のヤング図形, $N = 3$ の場合は高々3行のシフトヤング図形になっている (4.1, 4.2節で述べる). したがって, 本質的に新しいのは $N \geq 4$ の場合である.

ところで, inset というのは R. A. Proctor によって導入された d-complete poset の 15 種類の既約成分の一つに付けられた呼称である [7]:

- (1) shapes, (2) shifted shapes, (3) birds,
- (4) insets, (5) tailed insets, (6) banners, (7) nooks,
- (8) swivels, (9) tailed swivels, (10) tagged swivels, (11) swivel shifteds,
- (12) pumps, (13) tailed pumps, (14) near bats, (15) bat.

ここで, shapes, shifted shapes というのはそれぞれヤング図形, シフトヤング図形のことである. 川中の発想は, Sato-Welter game や turning turtles といったよく知られた組み合わせゲームの研究対象を, d-complete poset の文脈の中で眺めてみよう, というところにある. その結果について詳細は [4] に譲ることにするが, ここでは, その記述が図形に基づいていることだけ言及しておこう. 一般の d-complete poset に対して「フック」および「フックを抜いて詰める」の定義については川中の [4] や筆者の [5] を参照されたい.

本研究ではリボン表示に注目している. この点が川中の [4] と異なるところである. 旧来の Sato-Welter game や turning turtles の SG 関数がリボン表示に基づいて記述されていることに注目すると, その他の d-complete poset に対しても SG 関数がリボン表示の基づいて計算できるはずだ, と考えるのは自然なことである.

さて, 主結果を述べよう:

定理 4.1 ([6]). inset game $(P_{N,1}; \rightarrow)$ の SG 関数は次で与えられる:

$$\text{SG}(\mathbf{x}) = \begin{cases} \sum_i^{\oplus} (-1|\alpha_{N-1}(\mathbf{x}^{(i)})) \oplus \alpha_N(\mathbf{x}) & n = N \\ x_1 & n = 1, \end{cases} \quad \mathbf{x} = \{x_1, x_2, \dots, x_n\} \in P_{N,1}.$$

ここで, $\mathbf{x}^{(i)} = \mathbf{x} \setminus \{x_i\}$ であり, α_n は Sato-Welter game の SG 関数である.

例 4.4. 例えば, 局面 $\mathbf{x} = \{1, 2, 4, 5\} \in P_{4,1}$ の SG 値は,

$$\begin{aligned} \alpha_{4,1}(\mathbf{x}) &= (-1|\alpha_3(2, 4, 5)) \oplus (-1|\alpha_3(1, 4, 5)) \\ &\quad \oplus (-1|\alpha_3(1, 2, 5)) \oplus (-1|\alpha_3(1, 2, 4)) \oplus \alpha_4(1, 2, 4, 5) \\ &= (-1|0) \oplus (-1|7) \oplus (-1|1) \oplus (-1|4) \oplus 6 = 10. \end{aligned}$$

したがって, 局面 \mathbf{x} は先手勝ち局面である. 必勝戦略に沿った手は (この場合はただ一つだが)

$$\{1, 2, 4, 5\} \rightarrow \{0, 1, 4, 5\}.$$

4.1 Case $N = 2$

$N = 2$ の場合の inset は高々2行のヤング図形になる. これを確かめるためには, 単射 $f : P_{3,1} \rightarrow P_2$ を

$$f(\{x_1, x_2\}) = \{x_1 + 1, x_2 + 1\}, \quad f(\{x_1\}) = \{0, x_1 + 1\}.$$

で定めればよい. すると, 次が得られる:

命題 4.2. 写像 f は $P_{2,1}$ から像 $f(P_{2,1})$ へのゲーム同型を与える。特に、局面 $\mathbf{x} \in P_{2,1}$ の SG 値は、 $\alpha_{2,1}(\mathbf{x}) = \alpha_2(f(\mathbf{x}))$ を満たす。

後半部分は、直接計算でも以下のように確認できる：

$$\begin{aligned}
\alpha_{2,1}(\{x_1, x_2\}) &= (-1 \mid \alpha_1(\{x_1\})) \oplus (-1 \mid \alpha_1(\{x_2\})) \oplus \alpha_2(\{x_1, x_2\}) \\
&= (-1 \mid x_1) \oplus (-1 \mid x_2) \oplus x_1 \oplus x_2 \oplus (x_1 \mid x_2) \\
&= (0 \mid x_1 + 1) \oplus (0 \mid x_2 + 1) \oplus x_1 \oplus x_2 \oplus (x_1 + 1 \mid x_2 + 1) \\
&= (x_1 + 1) \oplus (x_2 + 1) \oplus (x_1 + 1 \mid x_2 + 1) \\
&= \alpha_2(\{x_1 + 1, x_2 + 1\}) = \alpha_2(f(\{x_1, x_2\})), \\
\alpha_{2,1}(\{x_1\}) &= x_1 \\
&= 0 \oplus (x_1 + 1) \oplus (0 \mid x_1 + 1) \\
&= \alpha_2(\{0, x_1 + 1\}) = \alpha_2(f(\{x_1\})).
\end{aligned}$$

4.2 Case $N = 3$

$N = 3$ の場合の inset は高々3行のシフトヤング図形になる。ゲームとしては、起きているカメが3匹か1匹の turning turtles になる。これを確かめるためには、単射 $f : P_{3,1} \rightarrow P_{\text{odd}}$ を

$$f(\{x_1, x_2, x_3\}) = \{x_1, x_2, x_3\}, \quad f(\{x_1\}) = \{x_1\}.$$

で定めればよい。すると次が得られる：

命題 4.3. 写像 f は $P_{3,1}$ から像 $f(P_{3,1})$ へのゲーム同型を与える。特に、 $\alpha_{3,1}(\mathbf{x}) = \alpha(f(\mathbf{x}))$ 。

References

- [1] E.R. Berlekamp, J.H. Conway, and R.K. Guy, *Winning Ways for Your Mathematical Play*, Academic Press, 1982.
- [2] J.H. Conway, *On Numbers and Games*, Academic Press, 1976.
- [3] P. M. Grundy, *Mathematics and games*, *Eureka* **2** (1939), 6–8.
- [4] N. Kawanaka, *Games and algorithms with hook structure (in japanese)*, vol. 63, Suugaku, The Mathematical Society of Japan, 2011.
- [5] K. Nakada, *Colored hook formula for a generalized Young diagram*, *Osaka J. of Math.* **54** (2008), no. 4, 1085–1120.
- [6] ———, *Inset games and strategies*, *Séminaire Lotharingien de Combinatoire* **85B article #73** (2021).
- [7] R. A. Proctor, *Dynkin diagram classification of λ -minuscule Bruhat lattices and of d -complete posets*, *J. Algebraic Combin.* **6** (1999), 61–294.

- [8] M. Sato, *On Maya game, lecture notes taken by h. enomoto (in japanese)*, vol. 15, Suugaku-no-Ayumi, 1970, Special Issue “Mikio Sato”.
- [9] R. P. Sprague, *Über mathematische Kampfspiele*, Tohoku Mathematical Journal **41** (1935–36), 438–444.
- [10] C. P. Welter, *The theory of a class of games on a sequence of squares, in terms of advancing operation in a special group*, Indag. Math. **16** (1954), 194–200.