

数式処理システムを用いた組み合わせゲームの研究と教育への応用

宮寺良平

関西学院高等部*

RYOHEI MIYADERA

KWANSEI GAKUIN

要約 この論文では組み合わせゲームの一種であるチョコレート問題について考える。チョコレート問題は数学関連の関数を多く持つ数式処理システムには適している。今回紹介するのは、L-state がやや複雑な規則性を持っているチョコレート問題である。その問題の特徴を理解していただくために、既に発表済みの L-state の公式を持つチョコレートと比較して考える。
研究全般において、数式処理システム Mathematica に大きく依存しており、その計算機能とプログラム機能が本質的に寄与している。

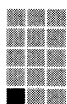
1 チョコレートゲームの紹介

まずチョコレートゲームを定義する。定義を理解するためには、常に具体的なチョコレートの形を見ながら定義を読む必要がある。まず初めには、図 1.1 のチョコレートを見ながら読んでいただきたい。

定義 1

薄いグレーの部分は甘く、濃いグレーの部分は苦くて食べることができないチョコレートがある。このチョコレートを 2 人のプレイヤーが交互に、切り取り線に沿って 2 つに割り、濃いグレーの部分を含む方を残して、薄いグレーだけの部分を食べる。その結果として食べた部分は消去される。最後に苦い部分を残されたプレイヤーが負けである。

定義 1 においては、最後に苦くないチョコを割って食べた者が勝つ。苦いチョコを残されたものは食べることができない。最後に食べた者が勝つので、ゲーム理論では正規形ゲームとよばれるものである。図 1.1 のチョコレートを考える。縦と横のブロックの数を使って 5×3 のチョコレートと呼ぶことにする。切り取り線は、白い部分で、縦か横に切ることができる。切るとチョコレートの図形が 2 つのパートに分割されるが、左下の濃いグレーの部分を含んだ部分はそのままにして、もう 1 つのパートは食べてしまう。結果として、縦に切るとすれば、 $5 \times y$ ($1 \leq y < 3$) のチョコレートが残る。例えば、縦の 1 列を切り取ると、図 1.2 のチョコレートが残る。横に切るとすれば $x \times 3$ ($1 \leq x < 5$) となり、例えば、横の 1 列を切り取ると、図 1.3、横の 2 列を切り取ると、1.4 のようになる。このチョコレートゲームをプレーする場合に現れるチョコレートの形は $x \times y$ ($1 \leq x < 5$) ($1 \leq y < 3$) となる。



*miyadera1272000@yahoo.co.jp

図 1.1

5 × 3

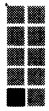


図 1.2

5 × 2

図 1.3

4 × 3

図 1.4

3 × 3

これを使ってゲームするとすれば、先手と後手のどちらが有利であるかを考える。実はあなたが先手でプレーすれば勝つことができる。方法は簡単である。図 1.4 とすれば良い。ここから相手は $x \times 3$ ($1 \leq x < 3$) か $3 \times y$ ($1 \leq y < 3$) に移る。そうすればあなたは $x \times x$ ($1 \leq x < 3$) か $y \times y$ ($1 \leq y < 3$) に移ればよい。すなわち相手がどのようにプレーしても、あなたは縦と横のブロック数を同じすれば、最終的に 1×1 のチョコの状態にして、相手に苦くて食べれない部分を残して勝てる。

定義 2

チョコレートゲームにおいては重要な 2 つのポジション（座標）がある。

- (a) あるチョコレートの状態が *W-States* (Winning position) であるとは、この状態からゲームを始めたプレイヤーは、これ以後正しい手を選ぶ限り、必ず勝つということである。
- (b) あるチョコレートの状態が *L-States* (Losing position) であるとは、この状態からゲームを始めたプレイヤーは、相手が正しい手を選び続けると、どんな手を選んでも、必ず負けるということである。

注意 1

W-state から始めるプレイヤーがミスをするると負ける可能性があるし、*L-state* から始めるプレイヤーは、相手がミスをするると勝つ可能性がある。

L-States は *P-position* (a previous-player-winning position) と呼ばれることもある。また、*W-States* は *N-position* (an next-player-winning position) と呼ばれることもある。

次に考えるのは、長方形ではなく、三角形のチョコレートである。このタイプのチョコレート問題は著者と関西学院高等部の生徒によって作られた新しい問題である。

図 1.5 のチョコレートは [1] で既に研究されたものである。しかし良く似た形の図 1.6 のチョコレートは新しいチョコレートである。図形の下に書いてある数字の組はチョコレートの座標である。これについては後で説明する。



図 1.5

{3, 6, 3}



図 1.6

{3, 6, 6}

これらのチョコレートの状態を表す座標を導入する。図 1.5 はチョコレートを左上から右下に向う線に沿って、苦いチョコの左を最大 3 回、左上から右下に向う線に沿って苦いチョコの右を最大 6 回、右上から左下に向う線に沿って苦いチョコの位置よりも右側を最大 3 回切ることが出来る。したがって座標は {3, 6, 3}

とする。

図 1.6 はチョコレートを上から右下に向う線に沿って、甘いチョコの左を最大 3 回、左上から右下に向う線に沿って甘いチョコの右を最大 6 回、右上から左下に向う線に沿って甘いチョコの位置よりも右側を最大 6 回切ることが出来る。したがって座標は $\{3, 6, 6\}$ とする。

以上のように、チョコレートの座標は、チョコレートをそれぞれ 3 つの方向に切ることができる最大回数で決まる。

本論文においては、これらのタイプのチョコレート問題を中心テーマとして扱うが、構造は同じままで任意の $x, y, z \in \mathbb{Z}_{\geq 0}$ に対して、座標 $\{x, y, z\}$ となるチョコレートを考える。

2 L-state が簡単な公式になるタイプのチョコレート

$\mathbb{Z}_{\geq 0}$ は非負整数の集合とする。ここでは図 1.5 のチョコレートを考える。チョコレートを切った場合について例をあげる。図の下に書いている番号はチョコレートの状態を表す座標である。



図 2.1
 $\{2, 4, 2\}$

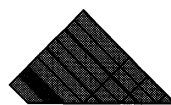


図 2.2
 $\{1, 6, 3\}$

すぐにわかることだが座標は、 $y \geq 2z$ を満たす。ただし、 $y, z \in \mathbb{Z}_{\geq 0}$ 。このように座標が不等式を満たすようなチョコレートは著者達が作り出した問題である。

チョコレート問題において重要なニム和という概念を定義する。

定義 3

$x, y \in \mathbb{Z}_{\geq 0}$ とし、2 進数表示して、 $x = \sum_{i=0}^n x_i 2^i$, $y = \sum_{i=0}^n y_i 2^i$ とする。なおここで $x_i, y_i \in \{0, 1\}$ とする。このときニム和 $x \oplus y$ を次のように定義する。

$$x \oplus y = \sum_{i=0}^n w_i 2^i, \quad (1)$$

ここで $w_i = x_i + y_i \pmod{2}$ とする。すなわち w_i は $x_i + y_i$ を 2 で割った余りである。ニム和は計算機科学の分野では排他的論理和と呼ばれることが多い。

ニム和を使うと、図 1.5 のチョコレートに関しては L-state を特徴づけることができる。

定理 4

$A_1 = \{\{x, y, z\}; x, y, z \in \mathbb{Z}_{\geq 0}, y \geq 2z \text{ かつ } x \oplus y \oplus z = 0\}$, $B_1 = \{\{x, y, z\}; x, y, z \in \mathbb{Z}_{\geq 0}, y \geq 2z \text{ かつ } x \oplus y \oplus z \neq 0\}$ とする。 A_1 は図 1.5 チョコレートの L-state の集合で、 B_1 は W-state の集合である。

この定理の証明は [1] にある。

3 L-state が簡単な公式にならないタイプのチョコレート

ここでは図 1.6 のチョコレートを研究する。チョコレートを切った場合について例をあげる。図の下に書いている番号はチョコレートの状態を表す座標である。



図 3.1

{2,3,3}



図 3.2

{3,6,1}

すぐにわかることだが座標は、 $y \geq z$ を満たす。
 次は図 1.6 のチョコレートの L-state を計算する Mathematica プログラムである。L-state を表す法則や公式を見つけるためには、計算機によってできるだけ多くの L-state のデータを出し、それを整理しながら法則を見つけるのが一般的な方法である。

```
ss = 100; k=2; a1 =
  Flatten[Table[{a, b, c}, {a, 0, ss}, {b, 0, ss}, {c, 0, ss}], 2];
allcases = Select[a1, (1/k) (#[[2]]) >= #[[3]] &];
move[z_] := Block[{p}, p = z;
  Union[Table[{p[[1]], p[[2]], t3}, {t3, 0, p[[3]] - 1}],
    Table[{p[[1]], t2, Min[Floor[(1/k) (t2)], p[[3]]]}, {t2, 0,
      p[[2]] - 1}],
    Table[{t1, p[[2]], p[[3]]}, {t1, 0, p[[1]] - 1}]]];
Mex[L_] := Min[Complement[Range[0, Length[L]], L]];
Gr[pos_] := Gr[pos] = Mex[Map[Gr, move[pos]]];
pposition1 = Select[allcases, Gr[#] == 0 &];
```

上のプログラムによって L-state のデータを作り、それらを第 1 座標で整理することで次のような式によって、データを表すことが出来た。ただし、数学的な証明がまだ未完成であるために、予想として扱う。

```
AA[n_] :=
  Union[Join[
    Table[{6 n - 1, 4 n + 2 k - 1, 4 n - 4 k - 1}, {k, 0, n - 1}],
    Table[{6 n - 1, 4 n + 2 k, 4 n - 4 k - 2}, {k, 0, n - 1}],
    Table[{6 n - 1, 6 n + 2 k - 1, 4 k}, {k, 0, n - 1}],
    Table[{6 n - 1, 6 n + 2 k, 4 k + 1}, {k, 0, n - 1}]]];
BB[n_] :=
  Union[Join[Table[{6 n, 4 n + 2 k, 4 n - 4 k}, {k, 0, n}],
    Table[{6 n, 4 n + 2 k + 1, 4 n - 4 k - 3}, {k, 0, n - 1}],
    Table[{6 n, 6 n + 2 k + 1, 4 k + 2}, {k, 0, n - 1}],
    Table[{6 n, 6 n + 2 k + 2, 4 k + 3}, {k, 0, n - 1}]]];
CC[n_, m_] :=
  Union[Join[
    Table[{6 n + 1, 4 n + 2 k + 1, 4 n - 4 k - 1}, {k, 0, n - 1}],
    Table[{6 n + 1, 4 n + 2 k + 2, 4 n - 4 k - 2}, {k, 0, n - 1}],
    Table[{6 n + 1, 6 n + 2 k + 1, 4 k}, {k, 0, n}],
    Table[{6 n + 1, 6 n + 2 k + 2, 4 k + 1}, {k, 0, n - 1}],
    Table[{6 n + 1, k + 8 n + 2, k + 4 n + 1}, {k, 0, n + m}]]];
DD[n_] :=
  Union[Join[
    Table[{6 n + 2, 4 n + 2 k + 1, 4 n - 4 k + 1}, {k, 0, n}],
    Table[{6 n + 2, 4 n + 2 k + 2, 4 n - 4 k}, {k, 0, n}],
    Table[{6 n + 2, 6 n + 2 k + 3, 4 k + 2}, {k, 0, n - 1}],
    Table[{6 n + 2, 6 n + 2 k + 4, 4 k + 3}, {k, 0, n - 1}]]];
EE[n_] :=
  Union[Join[
    Table[{6 n + 3, 4 n + 2 k + 2, 4 n + 2 - 4 k}, {k, 0, n}],
    Table[{6 n + 3, 4 n + 2 k + 3, 4 n - 4 k - 1}, {k, 0, n - 1}],
    Table[{6 n + 3, 6 n + 2 k + 3, 4 k}, {k, 0, n}],
    Table[{6 n + 3, 6 n + 2 k + 4, 4 k + 1}, {k, 0, n}]]];
```

```

FF[n_,m_] :=
Union[Join[
  Table[{6 n + 4, 4 n + 2 k + 3, 4 n - 4 k + 1}, {k, 0, n}],
  Table[{6 n + 4, 4 n + 2 k + 4, 4 n - 4 k}, {k, 0, n}],
  Table[{6 n + 4, 6 n + 2 k + 5, 4 k + 2}, {k, 0, n}],
  Table[{6 n + 4, 6 n + 2 k + 6, 4 k + 3}, {k, 0, n - 1}],
  Table[{6 n + 4, k + 8 n + 6, k + 4 n + 3}, {k, 0, n + m}]]];

```

次のような予想ができる。

$\bigcup_{n \in \mathbb{Z}_{\geq 0}} (AA[n] \cup BB[n] \cup CC[n, m] \cup DD[n] \cup EE[n] \cup FF[n, m])$ は L-state の集合となる

この予想の証明に関しては、まだ完成していないが、大まかな方法論だけはできている。それを紹介する。ただし、以下のような議論を精密にして、なお読者が理解できるようにすることはかなりの工夫が必要であり、まだ未完成である。

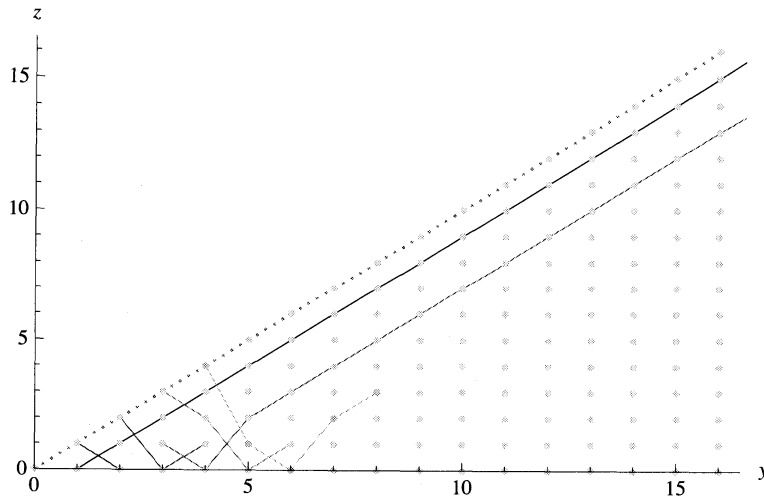


図 3.3

証明のポイント

図 3.3 はこのチョコレートゲームに現れる state を表示したものであるが、第 1 座標を省いて、第 2 座標と第 3 座標で点をプロットしている。横軸は第 2 座標の y 座標で、縦軸は第 3 座標の z 軸である。点線は直線 $z = y$ である。各点は、直線で結ばれた点の集合は L-state の点の第 2, 3 座標をプロットしたものである。そして、同じ濃度を持つ色の点同士は、第 1 座標が同じものである。1 番色の濃い点の集まりは第 1 座標が 1 で、順に、色の薄い点の集まりになっていくが、第 1 座標は 2, 3, 4, 5, 6 となっていく。

点で結ばれていない点は、すべて第 1 座標が 6 である W-state である。

ここで重要なことは、W-state からうまく動くと L-state へ行くことである。

例えば $\{6, 10, 7\}$ は第 2 座標の y を減らすことで、 $\{6, 4, 4\}$ へ行ける。この図においては、第 1 座標は書いていないから、 $\{10, 7\}$ において、 y を 4 に減らすと、図形の性質から z も 4 となり、 $\{4, 4\}$ となる。これは不等式 $y \geq z$ を使うと考えても良い。 $\{6, 6, 3\}$ は、 z を減らして $\{6, 6, 0\}$ へ行くことができる。

このように、第 1 座標が 6 であるもののうちで、第 2, 3 座標を減らして L-state へ行けるのは、第 1 座標が 6 である L-state の右に位置する第 1 座標が 6 である W-state である。

次に第1座標が6であるL-stateの左に位置するW-stateを考える。それらの点の場合は、第1座標を減らすことでL-stateへ行くことができる。その理由は図3.4のように、L-stateは三角形の形に点を敷き詰めていくからである。結果として、 $\{6,3,2\}$ は、 $\{1,3,2\}$ へ行くことができる。次にL-stateからはどのように動いてもW-stateへ行くことが重要である。そのようになる理由を考える。

まず、 $\{x,y,z\}$ から移動するとき、 x を減らすと、L-stateへは行かない。何故なら、図3.4を見てもわかるように、第1座標の異なるL-state同士は、第1座標を省いて平面にプロットしても重ならないからである。 y を減らすと、L-stateへは行かない。何故なら、図3.4を見てもわかるように、第1座標の同じL-state同士は、第2座標に関しても同じものはないからである。

z を減らすと、L-stateへは行かない。何故なら、図3.4を見てもわかるように、第1座標の同じL-state同士は、第3座標に関しても同じものはないからである。

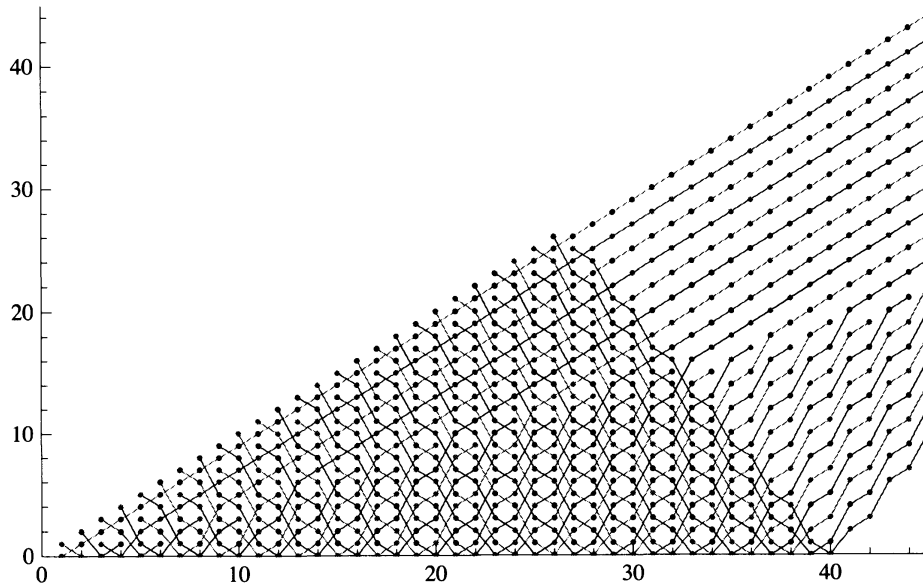


図 3.4

この研究は大阪大学の大学院生である峰松大介氏、関西学院の大学生である稗田卓人氏、高校生である藤井亮平氏、内藤瑤一郎氏のアイデアに依る部分が多かった。ここで感謝したい。

参 考 文 献

- [1] Ryohei Miyadera, Shunsuke Nakamura and Ryo Hanafusa, New Chocolate Games -Variants of the Game of Nim-, Proceeding of Annual International Conference on Computational Mathematics, Computational Geometry Statistics 2012.