

オートマトンを用いたあみだくじの列挙

熊谷 滉士郎¹ ディプタラマ ヘンリアン¹ 吉仲 亮¹ 篠原 歩¹

概要：無限集合の要素を列挙するのは不可能であるが、これらを適切な方法で符号化した文字列を受理する決定性オートマトン (DFA) を構築することで、目標の無限集合の要素の列挙と見なすことができる。本発表では縦線数 n のあみだくじを符号化した文字列からなるあみだくじ言語を受理する DFA を提案する。この DFA は $O(4^n)$ 時間で構築可能であり、我々は $n = 30$ まで実際に求めることができた。さらにあみだくじを拡張し、最左の縦線と最右の縦線が隣り合っているとみなして横線を引くことを許す循環あみだくじを定義する。循環あみだくじ言語を受理する DFA は $O(4^n n^2)$ 時間で構築可能であるが、循環あみだくじにおける縦線の対称性に着目して状態数を大幅に減らすことで $n = 32$ までの構築に成功した。

1. はじめに

あみだくじとは日本で利用されているくじの1つである。あみだくじの基本的なルールは以下の通りである。縦線を複数本描き、任意の隣り合う2つの縦線を繋ぐ横線を任意の本数描く。縦線の上端にはくじを引く参加者の名前を、縦線の下端にはくじの結果を書く。くじを引く際は縦線を上から下に辿り、たどり着いた下端に書かれていることが結果となる。ただし縦線を上端に向かって移動してはならず、横線と出会ったら必ず横線を通らなければいけない。あみだくじの例を図1(a)に示す。

あみだくじの列挙についての研究は既に行われており、山中らによって縦線数 n 横線数 b のあみだくじの列挙・数え上げ・ランダム生成の手法が示されている [3]。また、あみだくじは Primitive Sorting Network (PSN) と関連性があることから、あみだくじを利用して最小の PSN の個数を求める研究が行われている [4], [5]。さらに、あみだくじの集合と Braid 群などの集合との関連性について研究がなされている [1]。

本論文では縦線数 n のあみだくじと循環あみだくじそれぞれの列挙手法を提案する。あみだくじは図1(a)で示したように左端の縦線と右端の縦線が隣り合っていないと考える。一方で循環あみだくじは左端の縦線と右端の縦線が隣り合っていると考える。循環あみだくじはあみだくじより対称性が高いと言える。循環あみだくじの例を図1(b)に示す。

先行研究 [3] では縦線数と横線数を固定して列挙を行っていたが、本論文では縦線数のみ固定する。横線数を定め



(a) あみだくじ

(b) 循環あみだくじ

図 1: あみだくじと循環あみだくじの図示

ないためあみだくじの個数は無限個存在し列挙は不可能である。そこで、あみだくじの符号化を受理するような DFA を構築することで列挙と代える。この手法は覆面算の列挙でも用いられおり [2], DFA を探索することによって出力に比例した時間で列挙が可能である。

2. あみだくじの定義と性質

整数 x, y に対して x 以上 y 未満の整数集合を $[x, y)$ と表記する。 x と y の最大公約数を $\gcd(x, y)$, 最小公倍数を $\text{lcm}(x, y)$ と表記する。また $x \% y$ は $x \equiv z \pmod{y}$ を満たす整数 z ($0 \leq z < y$) を表す。 $\mathbb{B} = \{0, 1\}$ とする。さらに、系列 $a = \langle a_0, \dots, a_{n-1} \rangle$ の i 番目の要素を $a[i] = a_i$ と表し、系列 a と系列 b の連結を $a \cdot b$ と表す。しばしば長さが 1 の列とその要素を同一視する。

2.1 あみだくじ

あみだくじを符号化するために、縦線と横線は格子点上に配置されているものとする。 n 本の縦線からなるあみだくじのそれぞれの段は、そこに横線があれば 1, なければ 0 と表すことで、長さ $n-1$ のビット列 $w \in \mathbb{B}^{n-1}$ で表すことができる (図 2)。これを上から順に各段に対応させながら並べて得られる行列をあみだくじ行列と呼ぶ。本論

¹ 東北大学 大学院情報科学研究科
GSIS, Tohoku University

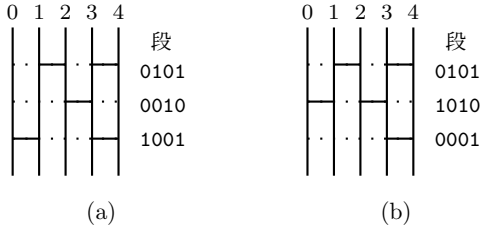


図 2: あみだくじの符号化

文ではあみだくじ行列を段の系列のように表記することもある。例えば、図 2 (a) のあみだくじは $\langle 0101, 0010, 1001 \rangle$ と表される。

ここで、すべてのビット列が正しくあみだくじの段を表すとは限らないことに注意されたい。例えば、あみだくじでは横線が隣接できないため、段を表すビット列の中に 1 が連続して現れることはない。また横線が全く含まれない段 0^m は、一般性を失うことなく削除することができる。これらを考慮して、あみだくじの段を表す長さ m のビット列の集合 $\mathcal{D}_m$ を次のように定義する。まず、表記を簡便にするためにいくつかの関数を導入する。ビット列 $w \in \mathbb{B}^m$ と整数 i に対し、範囲外の参照時には 0 を返す関数 $\text{get}(w, i)$ を定義し、これを用いて w の位置 i から始まる長さ ℓ の連続する部分列を $\text{sub}(w, i, \ell)$ と表す：

$$\text{get}(w, i) = \begin{cases} w[i] & \text{if } 0 \leq i < m \\ 0 & \text{otherwise} \end{cases}$$

$$\text{sub}(w, i, \ell) = \langle \text{get}(w, i), \dots, \text{get}(w, i + \ell - 1) \rangle$$

これらの関数を用いて、 $\mathcal{D}_m$ は次式で与えられる。

$$\mathcal{D}_m = \{w \in \mathbb{B}^m \setminus \{0^m\} \mid \text{sub}(w, i, 2) \neq 11 \text{ for } \forall i \in [0, m)\}$$

こうして定義された $\mathcal{D}_m$ から任意有限個の要素を取り出して得られる系列 $A \in \mathcal{D}_m^*$ は、常にあみだくじの符号化となっている。また行数が 0 のあみだくじ行列を ε と表す。

本論文では先行研究 [5] に倣い、全ての縦線の軌跡が一致するとき、そのときに限り 2 つのあみだくじは等価であるとみなす。あみだくじの縦線 i の軌跡とは、縦線 i からくじを引く際に辿る縦線の番号を順に並べたものであり、あみだくじ行列 $A \in \mathcal{D}_m^*$ において縦線 i の軌跡を表す関数 $\text{lot}(A, i)$ は次式で再帰的に定義される：

$$\text{lot}(\varepsilon, i) = i$$

$$\text{lot}(w \cdot A, i) = \begin{cases} i \cdot \text{lot}(A, i - 1) & \text{if } \text{get}(w, i - 1) = 1 \\ i \cdot \text{lot}(A, i + 1) & \text{if } \text{get}(w, i) = 1 \\ \text{lot}(A, i) & \text{otherwise,} \end{cases}$$

for any $w \in \mathcal{D}_m$.

例えば、図 2 (a) のあみだくじにおいて、縦線 4 の軌跡は

$$\begin{aligned} \text{lot}(\langle 0101, 0010, 1001 \rangle, 4) &= 4 \cdot \text{lot}(\langle 0010, 1001 \rangle, 3) \\ &= 4 \cdot 3 \cdot \text{lot}(\langle 1001 \rangle, 2) = 4 \cdot 3 \cdot \text{lot}(\varepsilon, 2) = \langle 4, 3, 2 \rangle \end{aligned}$$

と得られ、また縦線 0 からは $\langle 0, 1 \rangle$ という軌跡が得られる。さらに縦線 3 からは $\langle 3, 4, 3 \rangle$ という軌跡が得られる。

2 つのあみだくじ行列 $A, B \in \mathcal{D}_m^*$ が等価であるとは、任意の整数 i に対して $\text{lot}(A, i) = \text{lot}(B, i)$ を満たすときを言い、 $A \equiv B$ と表記する。例えば、図 2 (a) と図 2 (b) のあみだくじは等価であることが確かめられる。

ここで図 2 (a) の左下の横線を一段上に移動させると図 2 (b) が得られることがわかる。上下に隣接する 2 つの段 $w, v \in \mathcal{D}_m$ (上段 w , 下段 v) に対し、次式で定義される $\mathcal{P}(w, v) = \text{T}$ のとき、そのときに限り等価性を保ったまま下段 v から上段 w に移動できる横線が存在する。

$$\mathcal{P}(w, v) = \begin{cases} \text{T} & \text{if } \exists i \in [0, m), (v[i] = 1 \\ & \quad \wedge \text{sub}(w, i - 1, 3) = 000) \\ \text{F} & \text{otherwise} \end{cases} \quad (1)$$

あみだくじ行列 A と等価なもののうち、ビット列の系列としてみたときに辞書式順序で最大のものを A の標準形といい、 $\hat{A}$ と表す。補題 1 により $A = \hat{A}$ かどうかを簡単に確認することができる。

補題 1 任意のあみだくじ行列 $a = \langle w_0, \dots, w_\ell \rangle \in \mathcal{D}_m^*$ に対して次の 2 つは同値である。

- (1) $A = \hat{A}$
- (2) 任意の $i \in [0, \ell)$ に対して $\mathcal{P}(w_i, w_{i+1}) = \text{F}$ である。

証明 最初に $A < \hat{A} \implies \exists i \in [0, \ell), \mathcal{P}(w_i, w_{i+1}) = \text{T}$ を証明する。 $\hat{A} = \langle v_0, \dots, v_k \rangle$ とすると、 $a < \hat{a}$ より次の 3 つの式を満たす自然数 s, t, u ($t < m$) が存在する。

$$\begin{cases} (\forall i \in [0, s), w_i = v_i) \wedge w_s < v_s \\ (\forall i \in [0, t), w_s[i] = v_s[i]) \wedge w_s[t] = 0 \wedge v_s[t] = 1 \\ (\forall i \in [s, u), w_i[t] = 0) \wedge w_u[t] = 1 \end{cases}$$

s は A と $\hat{A}$ の最長共通接頭辞の長さを表している。 t は w_s と v_s の最長共通接頭辞の長さを表している。 u は w_s より後ろの段で初めて $w_u[t] = 1$ となる箇所を表している。

$A_s = \langle w_s, w_{s+1}, \dots, w_{\ell-1} \rangle$, $\hat{A}_s = \langle v_s, v_{s+1}, \dots, v_k \rangle$ とおく。また、 $x_t = \text{lot}(\hat{A}_s, t)$ とおくと $x_t[1] = t + 1$ である。

$\text{get}(w_j, t - 1) = 1$ を満たす整数 $j \in [s, u)$ が存在すると仮定する。 $y_t = \text{lot}(A_s, t)$ とおくと $y_t[1] = t - 1$ と求められ $y_t \neq x_t$ となり矛盾が生じる。そのため全ての j に対して $\text{get}(w_j, t - 1) = 0$ である。同様に $\text{get}(w_j, t + 1) = 1$ を満たす整数 $j \in [s, u)$ が存在しないことを示すことができる。よって全ての $j \in [s, u)$ に対して $\text{sub}(w_j, t - 1, 3) = 000$ が成立する。このとき $\mathcal{P}(w_{u-1}, w_u) = \text{T}$ である。

続いて $(\exists i \in [0, \ell), \mathcal{P}(w_i, w_{i+1}) = \text{T}) \implies A < \hat{A}$ で

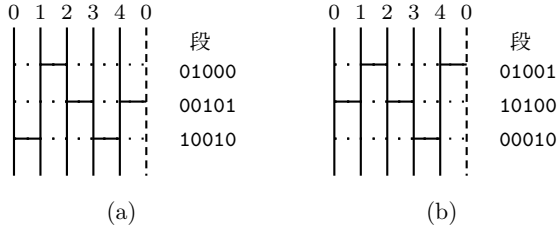


図 3: 図と段の対応関係

あるが、これは式 (1) の導入から明らかである。以上から $A < \hat{A} \iff \exists i \in [0, \ell), \mathcal{P}(w_i, w_{i+1}) = \mathbf{T}$ が成立する。故に (1) $\iff$ (2) が成立する。□

2.2 循環あみだくじ

あみだくじと同様に、 n 本の縦線からなる循環あみだくじのそれぞれの循環段を、長さ n のビット列 $w \in \mathbb{B}^n$ で表すことができる (図 3)。これを上から順に並べて得られる行列を循環あみだくじ行列と呼ぶ。循環あみだくじ行列も循環段の系列として表記できる。例えば、図 3 (a) の循環あみだくじは $\langle 01000, 00101, 10010 \rangle$ となる。

ここで、すべての段が循環あみだくじの循環段となるわけではないことに注意されたい。循環あみだくじでは左端の縦線と右端の縦線が隣り合っていると考えるため、例えば 1001 は段であるが循環段ではない。段の性質に加えこのことを考慮して、循環段を表す長さ n のビット列の集合 $\overline{\mathcal{D}}_n$ を次のように定義する。まず、循環するビット列 $w \in \mathbb{B}^n$ の位置 i から始まる長さ ℓ の連続する部分列を

$$\overline{\text{sub}}(w, i, \ell) = \langle w[i\%n], \dots, w[(i + \ell - 1)\%n] \rangle$$

と表し、 $\overline{\mathcal{D}}_n$ を次式で与える。

$$\overline{\mathcal{D}}_n = \{w \in \mathbb{B}^n \setminus \{0^n\} \mid \overline{\text{sub}}(w, i, 2) \neq 11 \text{ for } \forall i \in [0, n)\}$$

この $\overline{\mathcal{D}}_n$ から任意有限個の要素を並べた系列 $A \in \overline{\mathcal{D}}_n^*$ は、常に循環あみだくじの符号化となっている。

循環あみだくじ行列 $A \in \mathcal{D}_m^*$ において縦線 i の軌跡を表す関数 $\overline{\text{lot}}(A, i)$ を次式で再帰的に定義する：

$$\begin{aligned} \overline{\text{lot}}(\varepsilon, i) &= i\%n \\ \overline{\text{lot}}(w \cdot A, i) &= \begin{cases} i\%n \cdot \overline{\text{lot}}(A, i-1) & \text{if } w[(i-1)\%n] = 1 \\ i\%n \cdot \overline{\text{lot}}(A, i+1) & \text{if } w[i\%n] = 1 \\ \overline{\text{lot}}(A, i) & \text{otherwise,} \end{cases} \end{aligned}$$

for any $w \in \overline{\mathcal{D}}_n$.

例えば図 3 (a) の循環あみだくじの縦線 0 の軌跡は、

$$\begin{aligned} \overline{\text{lot}}(\langle 01000, 00101, 10010 \rangle, 0) &= \overline{\text{lot}}(\langle 00101, 10010 \rangle, 0) \\ &= 0 \cdot \overline{\text{lot}}(\langle 10010 \rangle, -1) = 0 \cdot 4 \cdot \overline{\text{lot}}(\varepsilon, -2) = \langle 0, 4, 3 \rangle \end{aligned}$$

と得られ、また縦線 1 からは $\langle 1, 2, 3, 4 \rangle$ という軌跡が得ら

れる。さらに縦線 4 からは $\langle 4, 0, 1 \rangle$ という軌跡が得られる。

2つの循環あみだくじ行列 $A, B \in \overline{\mathcal{D}}_n^*$ が平行等価であるとは、任意の整数 i に対して $\overline{\text{lot}}(A, i) = \overline{\text{lot}}(B, i)$ を満たすときを言い、 $A \equiv B$ と表記する。例えば、図 3 (a) と図 3 (b) のあみだくじは並行等価である。

循環あみだくじにおいて横線を一段上にずらすような移動を行えるかどうかの判定を次式で定義する：

$$\overline{\mathcal{P}}(w, v) = \begin{cases} \mathbf{T} & \text{if } \exists i \in [0, n), (v[i] = 1 \\ & \wedge \overline{\text{sub}}(w, i-1, 3) = 000) \\ \mathbf{F} & \text{otherwise} \end{cases} \quad (2)$$

上下に隣接する2つの循環段 $w, v \in \overline{\mathcal{D}}_n$ (上段 w , 下段 v) に対し、 $\overline{\mathcal{P}}(w, v) = \mathbf{T}$ のとき、そのときに限り等価性を保ったまま下段 v から上段 w に移動できる横線が存在する。

循環あみだくじ行列 A と等価なもののうち、ビット列の系列としてみたときに辞書式順序で最大のものを A の平行標準形といい、 $\hat{A}$ と表す。補題 1 と同様に、次の補題を証明することができる。

補題 2 任意の循環あみだくじ行列 $A = \langle w_0, \dots, w_\ell \rangle \in \overline{\mathcal{D}}_n^*$ に対して次の2つは同値である。

- (1) $A = \hat{A}$
- (2) 任意の $i \in [0, \ell)$ に対して $\overline{\mathcal{P}}(w_i, w_{i+1}) = \mathbf{F}$ である。

循環あみだくじは左端の縦線と右端の縦線が隣り合っていると考えるため図 4 に示すように縦線が左右に無限に続くと考えることができる。ここで、循環あみだくじを代表する部分として左右に無限に続く図から連続するちょうど n 本の縦線を含む部分を取り出すことを考える。このようにして取り出した部分は全て同じ循環あみだくじを代表するものとして等価とみなす。この性質を回転の性質と呼び、この等価性を回転等価と呼ぶ。回転の例を図 4 に示す。

回転等価を定義するために、まず、循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ において縦線 i 個分の回転を表す関数 $\text{rot}(A, i)$ を

$$\text{rot}(\langle w_0, \dots, w_\ell \rangle, i) = \langle \overline{\text{sub}}(w_0, i, n), \dots, \overline{\text{sub}}(w_\ell, i, n) \rangle$$

と定義する。例えば、図 4 の左側の循環あみだくじを1つ回転させた循環あみだくじは、

$$\text{rot}(\langle 01010, 00100, 00001 \rangle, 1) = \langle 10100, 01000, 00010 \rangle$$

である。2つの循環あみだくじ行列 $A, B \in \overline{\mathcal{D}}_n^*$ が回転等価であるとは、 $A = \text{rot}(B, i)$ を満たす整数 i が存在するときを言い、 $A \cong B$ と表記する。例えば、図 4 の左側の循環あみだくじと右側の循環あみだくじは回転等価である。

循環あみだくじにおける等価とは平行等価と回転等価の組み合わせである。すなわち、 $\hat{A} \cong \hat{B}$ が成立するとき循環あみだくじ行列 A, B が等価と言い $A \approx B$ と表記する。

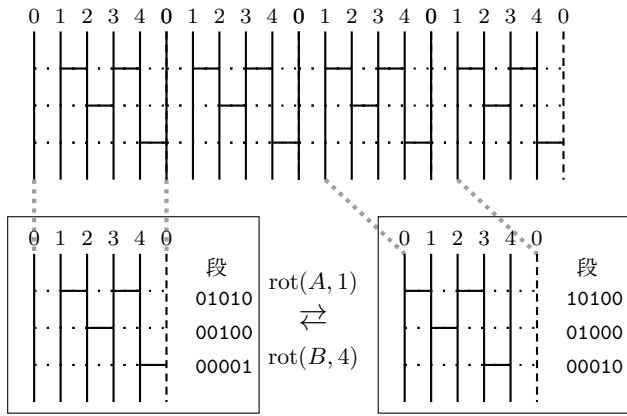


図 4: 縦線数 5 の循環あみだくじの回転の例

2.3 周期

この章では証明で用いる回転関数の諸性質と周期について述べる。次の補題は回転関数の定義から容易に確かめられる。

補題 3 任意の循環あみだくじ行列 $A_1, A_2 \in \overline{\mathcal{D}}_n^*$ と整数 i, j に対して以下の等式が成り立つ。

- (1) $\text{rot}(A_1 \cdot A_2, i) = \text{rot}(A_1, i) \cdot \text{rot}(A_2, i)$
- (2) $\text{rot}(\text{rot}(A_1, i), j) = \text{rot}(A_1, i + j)$
- (3) $A_1 = \text{rot}(A_2, i) \iff \text{rot}(A_1, j) = \text{rot}(A_2, i + j)$

循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ に対して、 $A = \text{rot}(A, i)$ を満たす整数 i を A の周期と呼ぶ。0 は常に周期となる。また、 A の 1 以上で最小の周期を基本周期と呼び、 p_A と表す。循環段 $w \in \overline{\mathcal{D}}_n$ に対しては循環あみだくじ $\langle w \rangle$ の周期と基本周期を w の周期と基本周期とする。この節では周期が必ず基本周期の倍数となることと、循環あみだくじ行列の周期の高速な計算方法を示す。

補題 4 循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ に対し、 i を A の周期とすると、任意の整数 k に対して $A = \text{rot}(A, ki)$ である。

証明 $k = 0, 1$ に対しては $A = \text{rot}(A, i)$ から明らかである。まず $k > 0$ に対して帰納法を用いる。 $A = \text{rot}(A, ki)$ と仮定すると、 $k + 1$ に対しても

$$\text{rot}(A, (k + 1)i) = \text{rot}(\text{rot}(A, ki), i) = \text{rot}(A, i) = A$$

が成り立つ。さらに、補題 3(3) より、 $A = \text{rot}(A, ki)$ から

$$\text{rot}(A, -ki) = \text{rot}(A, ki - ki) = A$$

を得る。よって負の数に対しても成り立つ。□

補題 5 循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ に対し、 i と j を A の周期とすると $\text{gcd}(i, j)$ もまた A の周期である。

証明 ベズーの補題により、 $xi + yj = \text{gcd}(i, j)$ を満たす整数 x と y が必ず存在する。補題 4 より、 $A = \text{rot}(A, xi) =$

$\text{rot}(A, yj)$ が成立するので、

$$\begin{aligned} \text{rot}(A, \text{gcd}(i, j)) &= \text{rot}(A, xi + yj) \\ &= \text{rot}(\text{rot}(A, xi), yj) = \text{rot}(A, yj) = A \end{aligned}$$

を得る。□

補題 6 循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ の任意の周期 i に対して $i \% p_A = 0$ が成立する。

証明 任意の周期 x に対して $\text{gcd}(p_A, x) \geq p_A$ である。この式を満たす x は p_A の倍数のみである。よって $i \% p_A = 0$ が成立する。□

以上より基本周期の倍数のみが周期であることが示された。続いて、循環あみだくじの基本周期の計算の高速化に有用な性質を証明する。

補題 7 任意の循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ と循環段 $w \in \overline{\mathcal{D}}_n$ に対して $p_{A \cdot w} = \text{lcm}(p_A, p_w)$ が成立する。

証明 $A \cdot w$ の周期 i に対して、周期の定義から

$$A \cdot w = \text{rot}(A \cdot w, i) = \text{rot}(A, i) \cdot \overline{\text{sub}}(w, i, n)$$

となり、 $A = \text{rot}(A, i)$ かつ $w = \overline{\text{sub}}(w, i, n)$ となる。補題 6 から、 $i \% p_A = 0$ かつ $i \% p_w = 0$ であり、すなわち $i \% \text{lcm}(p_A, p_w) = 0$ と求められる。基本周期は 1 以上で最小の周期であるため $p_{A \cdot w} = \text{lcm}(p_A, p_w)$ である。□

補題 7 から、すべての $w \in \overline{\mathcal{D}}_n$ の周期を求めておくことで任意の循環あみだくじの周期を高速に計算することができる。

3. あみだくじ言語を受理する DFA

決定性有限オートマトン (DFA) を 5 つ組 $M = \langle Q, \Sigma, \delta, q_0, F \rangle$ で定義する。ここに Q は状態の有限集合、 Σ は有限アルファベット、 $\delta : Q \times \Sigma \rightarrow Q$ は状態遷移を表す部分関数、 $q_0 \in Q$ は初期状態、そして $F \subseteq Q$ は受理状態の集合である。状態 $q \in Q$ と文字 $w \in \Sigma$ に対して $\delta(q, w) = r$ であるとき、 q から r に w で遷移すると言い、 q から r への辺は辺ラベル w を持つ。 δ の定義域を $Q \times \Sigma^*$ に自然に拡張し、系列 $A \in \Sigma^*$ に対して $\delta(q_0, A) \in F$ のとき、 M は A を受理すると呼ぶ。 M が受理する言語を $L(M) = \{A \in \Sigma^* \mid \delta(q_0, A) \in F\}$ と定義する。

すでに述べたとおり、任意の系列 $A \in \mathcal{D}_m^*$ は正しくあみだくじの符号化となっているが、我々は重複を避けるために標準形のみを漏れなく受理する DFA を構築し、その状態最小化を行う。このような DFA M が得られれば、その開始状態 q_0 から系統的に辺を探索することで、 M が受理する系列の一部または全部を列挙することが可能となる。

3.1 あみだくじを列挙する DFA

縦線数 n のあみだくじに対する DFA を $C_n = \langle D_m, D_m, \delta, 1^m, D_m \rangle$ と以下のように定義する. ここに $m = n - 1$ とし, 部分関数 $\delta : D_m \times D_m \rightarrow D_m$ を次式で定義する.

$$\delta(q, w) = \begin{cases} w & \text{if } \mathcal{P}(q, w) = F \\ \text{undefined} & \text{otherwise} \end{cases}$$

定理 1 DFA C_n は縦線数 n のあみだくじを列挙する. すなわち, $L(C_n) = \{A \in D_{n-1}^* \mid A = \hat{A}\}$ である.

証明 任意のあみだくじ行列 $A = \langle w_0, \dots, w_\ell \rangle \in \overline{D}_n^*$ に対して $A = \hat{A} \iff A \in L(C_n)$ を証明する. $A = \hat{A}$ のとき, 補題 1 よりすべての整数 $i \in [0, \ell]$ に対して $\mathcal{P}(w_i, w_{i+1}) = F$ であるので, 初期状態から A に沿って遷移する状態はすべて定義されており, またすべての状態が受理状態であるため, A は C_n に受理される. 逆に $A \neq \hat{A}$ のとき, 補題 1 よりある整数 i が存在して, $\mathcal{P}(w_i, w_{i+1}) = T$ であるため, 該当する状態は未定義となり, A は C_n に受理されない. $\square$

例えば C_4 において, 状態 101 と 010 の任意の段を読み込んだ遷移先は全く同じである. そのため C_n は最小状態数の DFA ではない.

遷移関数の定義と式 (1) から $\delta(q, w)$ を求めるときに各 i に対して $\text{sub}(q, i, 3) = 000$ かどうか確認される. その他に q が直接参照されることは無いため, $\text{sub}(q, i, 3) = 000$ である場合 0, そうでない場合 1 を i 番目のビットに保存したビット列を作成する. この処理は次式に示す関数 $\text{ext} : \mathbb{B}^m \rightarrow \mathbb{B}^m$ によって求められる.

$$\text{ext}(w)[i] = \begin{cases} 0 & \text{if } \text{sub}(w, i-1, 3) = 000 \\ 1 & \text{otherwise} \end{cases}$$

また, δ の条件分岐部分は次式で求められる.

$$\hat{\mathcal{P}}(q, w) = \begin{cases} T & \text{if } q[i] = 1 \vee w[i] = 0 \text{ for } \forall i \in [0, m] \\ F & \text{otherwise} \end{cases}$$

縦線数 n のあみだくじに対する DFA を $M_n = \langle Q, D_m, \delta, 1^m, Q \rangle$ と定義する. ここに $m = n - 1$, $Q = \{\text{ext}(w) \mid w \in D_m\}$ とし, 部分関数 $\delta : Q \times D_m \rightarrow Q$ を次式で定義する.

$$\delta(q, w) = \begin{cases} \text{ext}(w) & \text{if } \hat{\mathcal{P}}(q, w) \\ \text{undefined} & \text{otherwise} \end{cases}$$

縦線数 5 のあみだくじ言語を受理する M_4 を図 5 に示す. 構築アルゴリズムは $O(|Q| \cdot |D_m|)$ 時間である. $|Q| = O(2^n)$ かつ $|D_m| = O(2^n)$ より $O(4^n)$ 時間と求められる.

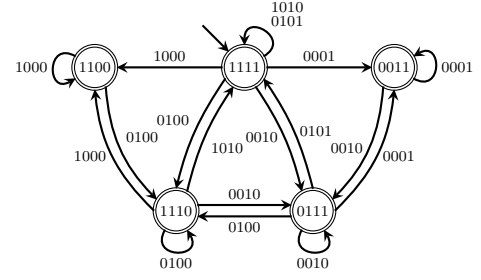


図 5: M_4

3.2 循環あみだくじを列挙する DFA

あみだくじに対するものと同様に, 循環段 $w \in \overline{D}_n$ と整数 $i \in [0, n)$ に対して $\text{sub}(w, i, 3) = 000$ である場合 0, そうでない場合 1 を i 番目のビットに保持したビット列を作成する. この処理は次式に示す関数 $\text{ext} : \mathbb{B}^n \rightarrow \mathbb{B}^n$ によって求められる.

$$\text{ext}(w)[i] = \begin{cases} 0 & \text{if } \text{sub}(w, i-1, 3) = 000 \\ 1 & \text{otherwise} \end{cases}$$

循環段 $w \in \overline{D}_n$ に対して, 次式で示される $\hat{w}$ を基本段と定義する.

$$\hat{w} = \max \left\{ v \in \overline{D}_n \mid \begin{array}{l} v \cong w \\ \wedge (\forall j \in \mathbb{N}, \text{ext}(v) \geq \text{ext}(\text{sub}(w, j, n))) \end{array} \right\}$$

$\hat{w}$ は w と回転等価な循環段のうち $\text{ext}(\hat{w})$ が辞書順最大であるような循環段である. また, $\phi(w)$ を次式で定義する.

$$\phi(w) = \min\{i \in \mathbb{N} \mid \hat{w} = \text{sub}(w, i, n)\}$$

任意の整数 j に対して $j \% p_w = \phi(w)$ であることと $\hat{w} = \text{sub}(w, j, n)$ が成立することは同値である.

縦線数 n の循環あみだくじに対する DFA を $\overline{M}_n = \langle Q, \overline{D}_n, \delta, q_0, Q \rangle$ と以下のように定義する. ここに, $Q \subseteq \{\text{ext}(w) \mid w \in \overline{D}_n\} \times [1, n+1)$ は次式で定義される $\delta : Q \times \overline{D}_n \rightarrow Q$ によって初期状態 $q_0 = (1^n, 1)$ から到達可能なものとする.

$$\delta((q, k), w) = \begin{cases} (\text{ext}(w), \text{lcm}(k, p_w)) & \text{if } \phi(w) \in [0, \text{gcd}(k, p_w)) \\ & \wedge \hat{\mathcal{P}}(q, w) \\ \text{undefined} & \text{otherwise} \end{cases}$$

補題 7 より循環あみだくじ行列 $A \in \overline{D}_n^*$ を読んだ遷移先の状態 $(q, k) = \delta(q_0, A)$ の k は基本周期 p_A と一致する.

定理 2 DFA $\overline{M}_n$ は循環あみだくじを列挙する.

定理を証明するためにいくつか補題を示す.

補題 8 $L(\overline{M}_n) \subseteq \{A \in \overline{D}_n^* \mid A = \hat{A}\}$ である.

証明 $A = \langle w_0, \dots, w_\ell \rangle \in \overline{\mathcal{D}}_n^*$ とおく. $A \neq \hat{A}$ のとき, 補題 2 よりある整数 i が存在して $\hat{P}(\text{ext}(w_i), w_{i+1}) = \top$ である. よって初期状態から A を読んで着く状態は未定義であるため $A \notin L(\overline{M}_n)$ である. $\square$

補題 9 循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ と循環段 $w \in \overline{\mathcal{D}}_n$, 整数 i, j に対して次の 2 つは同値である.

- (1) $A \cdot \text{sub}(w, i, n) \cong A \cdot \text{sub}(w, j, n)$
- (2) $i \equiv j \pmod{\gcd(p_A, p_w)}$

証明 $B_k = A \cdot \text{sub}(w, k, n)$, $g = \gcd(p_A, p_w)$ と表す.

まず (1) $\implies$ (2) を示す. $B_i \cong B_j$ より $B_i = \text{rot}(B_j, k)$ を満たす整数 k が存在する. すなわち,

$$\begin{aligned} A \cdot \text{sub}(w, i, n) &= \text{rot}(A \cdot \text{sub}(w, j, n), k) \\ &= \text{rot}(A, k) \cdot \text{sub}(w, j + k, n) \end{aligned}$$

である. 整理すると,

$$0 \equiv k \pmod{p_A} \quad \text{かつ} \quad i \equiv j + k \pmod{p_w}$$

となり, k を消去すると整数 x, y が存在して, $i - j = xp_A - yp_w$ が得られる. 左辺が g の倍数である時に限り x, y が必ず存在するため, $i \equiv j \pmod{g}$ が成立する.

続いて (2) $\implies$ (1) を示す. 最初に任意の整数 k に対して $\forall x \in \mathbb{Z}$, $B_k \cong B_{k+xp_A}$ が成立することを示す.

$$\begin{aligned} B_k &\cong \text{rot}(B, xp_A) = \text{rot}(A, xp_A) \cdot \text{sub}(w, k + xp_A, n) \\ &= A \cdot \text{sub}(w, k + xp_A, n) = B_{k+xp_A} \end{aligned}$$

続いて $\forall y \in \mathbb{Z}$, $B_k = B_{k+yp_w}$ が成立することを示す.

$$B_{k+yp_w} = A \cdot \text{sub}(w, k + yp_w, n) = A \cdot \text{sub}(w, k, n) = B_k$$

以上から $B_k \cong B_{k+xp_A+yp_w}$ である. ここで, 任意の整数 z に対して $zg = xp_A + yp_w$ を満たす整数 x, y が存在する. すなわち, $B_k \cong B_{k+zg}$ が成立する.

$i \equiv j \pmod{g}$ より $i = j + zg$ を満たす整数 z が存在する. $B_j \cong B_{j+zg} = B_i$ となるため $B_i \cong B_j$ である. $\square$

補題 10 循環あみだくじ行列 $A \in \overline{\mathcal{D}}_n^*$ に対して $\hat{B}$ が $\overline{M}_n$ に受理され, かつ, $\hat{A} \cong \hat{B}$ を満たす $B \in \overline{\mathcal{D}}_n^*$ が存在する.

証明 $\hat{A} = \langle w_0, \dots, w_{\ell-1} \rangle$ とする. $\hat{A}$ の接頭辞を $\hat{A}_k = \langle w_0, \dots, w_{k-1} \rangle$ ($0 \leq k \leq \ell$) とおく.

循環あみだくじ $B \in \overline{\mathcal{D}}_n^\ell$ を次に示すように定義する. 任意の整数 $i \in [0, \ell)$ に対して, $j = \phi(w_i) \% \gcd(p_{\hat{A}}, p_{w_i})$ とし, $B[i] = \text{sub}(\hat{w}_i, j, n)$ とする. B は定義から $\delta(q_0, B) \neq \text{undefined}$ であるため $\overline{M}_n$ に受理され, 補題 9 より $B \cong \hat{A}$ であるから命題は成立する. $\square$

補題 11 $\overline{M}_n$ に受理される循環あみだくじ行列 $A_1, A_2 \in$

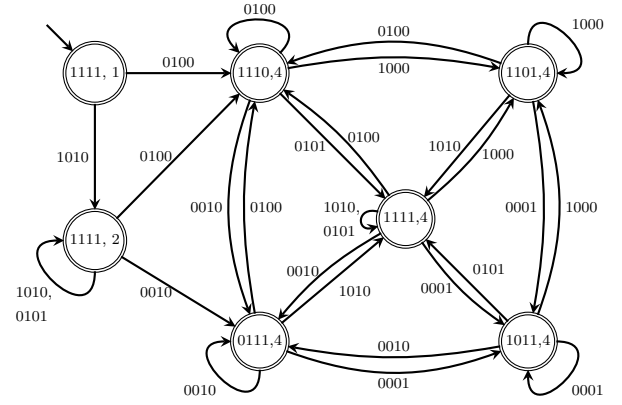


図 6: $\overline{M}_4$

$\overline{\mathcal{D}}_n^*$ に対して $\hat{A}_1 \cong \hat{A}_2$ ならば $A_1 = A_2$ である.

証明 補題 8 より $A_1 = \hat{A}_1$ かつ $A_2 = \hat{A}_2$ なので $A_1 \cong A_2$ である.

$A_1 = A_2 = \varepsilon$ のとき明らかに $A_1 = A_2$ である. その他のとき $A_1 = B_1 \cdot w_1$, $A_2 = B_2 \cdot w_2$ とする. $B_1 = B_2$ と仮定すると, 補題 9 からある整数 i ($i \% \gcd(p_{B_1}, p_{w_1}) = 0$) が存在して $w_1 = \text{sub}(w_2, i, n)$ である. δ_0 の定義から $i = 0$ と求められるため $w_1 = w_2$ である. よって $A_1 = A_2$ が成立する. 以上から段数に関する帰納法より $A_1 = A_2$ と求められる. $\square$

補題 10 と補題 11 より定理 2 が示された. $\overline{M}_4$ を図 6 に示す. 構築アルゴリズムは $O(|Q| \cdot |\overline{\mathcal{D}}_n| \log |Q|)$ 時間である. $|Q| = O(2^n)$ かつ $|\overline{\mathcal{D}}_n| = O(2^n)$ であるから $O(4^n n^2)$ 時間と求められる.

4. DFA の圧縮

図 6 の DFA $\overline{M}_4$ には, 4 つの対象な状態が存在することが確認できる. すなわち, $(1110, 4)$, $(1101, 4)$, $(1011, 4)$, $(0111, 4)$ は回転対称であり, それらの状態からの遷移も対称性を保存している. 本節ではこのような $\overline{M}_n$ の持つ対称性に注目し, 対象な状態を統合してあみだくじを列挙する DFA を圧縮する手法について述べる.

本節では集合 X 上の順列 $\pi: X \rightarrow X$ に対して, $w \in X$ の順列 π による像を $\pi(w)$ と表記する. また, 2 つの順列 π, π' の合成を $\pi \circ \pi'$ と表記し, X 上の全ての順列からなる集合を Π_X とする.

4.1 順列ラベル付き DFA

DFA の拡張として Nozaki ら [2] が定義した順列ラベル付き DFA (DFA-p) を紹介する. DFA-p は 6 つ組 $\langle Q, \Sigma, \delta, \gamma, q_0, F \rangle$ であり, $\langle Q, \Sigma, \delta, q_0, F \rangle$ は通常の DFA を構成する. $\gamma: Q \times \Sigma \rightarrow \Pi_\Sigma$ は順列遷移関数であり, その定義域は δ に一致する. DFA-p 上の計算においては状態の他

に Σ 上の順列 $\pi \in \Pi_\Sigma$ を記憶し、入力文字を π で読み替えて遷移する. 例えば、状態 $q \in Q$ で順列 $\pi \in \Pi_\Sigma$ を記憶し $w \in \Sigma$ が与えられると、状態は $\delta(q, \pi(w))$ に遷移し、さらに順列を $\gamma(q, \pi(w)) \circ \pi$ に更新する. また、 $\delta(q, w) = r$ かつ $\gamma(q, w) = \pi$ であるとき、 q から r への辺は辺ラベル (w, π) を持つという.

4.2 循環あみだくじ言語を受理する DFA-p

$\overline{M}_n$ とは異なる縦線数 n の循環あみだくじに対する DFA を $\overline{C}_n = \langle Q, \overline{D}_n, \delta_i, (1^n, 1), Q \rangle$ と以下のように定義する. ここに、 $Q \subseteq \{\text{ext}(w) \mid w \in \overline{D}_n\} \times [1, n+1]$ とし、遷移関数 δ_i を次式で定義する. 添字の i は状態から一意に決定され、 δ_0 は $\overline{M}_n$ の遷移関数と一致する.

$$\delta_i((q, k), w) = \begin{cases} (\text{ext}(w), \text{lcm}(k, p_w)) & \text{if } (\exists j \in [i, i + \gcd(k, p_w)), \\ & j \% p_w = \phi(w)) \wedge \widehat{P}(q, w) \\ \text{undefined} & \text{otherwise} \end{cases}$$

状態 $(q_1, k_1), (q_2, k_2) \in Q$ がシフト同値であるとは $k_1 = k_2$ かつ $q_1 = \text{sub}(q_2, i, n)$ を満たす整数 i が存在することを言い $(q_1, k_1) = S((q_2, k_2), i)$ と表記する. 循環段 w と整数 i, k に対する状態 $S((\text{ext}(\widehat{w}), k), i)$ は遷移関数 δ_i を用いて遷移を行うと定義する. 補題 11 から $\overline{C}_n$ は循環あみだくじを列挙する.

補題 12 循環段 $w, v \in \overline{D}_n$ と整数 i, k に対して状態 $q = (\text{ext}(\widehat{w}), k) \in Q$ とすると $S(\delta_0(q, v), i) = \delta_i(S(q, i), \text{sub}(v, i, n))$ が成立する.

証明 $r = \text{ext}(\widehat{w})$ とおくと $S(q, i) = (\text{sub}(r, i, n), k)$ である. また、 $u = \text{sub}(v, i, n)$ と置く.

$\widehat{P}(r, v) = \widehat{P}(\text{sub}(r, i, n), u)$ かつ $\phi(u) \equiv \phi(v) + i \pmod{p_v}$ かつ v と u の周期は等しいため $\delta_0(q, v) = \text{undefined} \iff \delta_i(S(q, i), u) = \text{undefined}$ が成立する. さらに、 $\text{ext}(u) = \text{sub}(\text{ext}(v), i, n)$ であるため $S(\delta_0(q, v), i) = \delta_i(S(q, i), \text{sub}(v, i, n))$ が成立する. $\square$

列数 n の循環あみだくじに対する DFA-p を $\widetilde{M}_n = \langle \widetilde{Q}, \overline{D}_n, \delta, \gamma, (1^n, 1), \widetilde{Q} \rangle$ と定義する. ここに、 $\widetilde{Q} \subseteq \{\text{ext}(\widehat{w}) \mid w \in \overline{D}_n\} \times [1, n+1]$ であり、 δ は次式で定義され、 $\gamma(q, w) = \pi_{\phi(w)}$ である. ただし順列 $\pi_k \in \Pi_{\overline{D}_n}$ は $\pi_k(w) = \text{sub}(w, -k, n)$ を満たす.

$$\delta((q, k), w) = \begin{cases} (\text{ext}(\widehat{w}), \text{lcm}(k, p_w)) & \text{if } \phi(w) \in [0, \gcd(k, p_w)) \\ & \wedge \widehat{P}(q, w) \\ \text{undefined} & \text{otherwise} \end{cases}$$

定理 3 DFA-p $\widetilde{M}_n$ は循環あみだくじを列挙する.

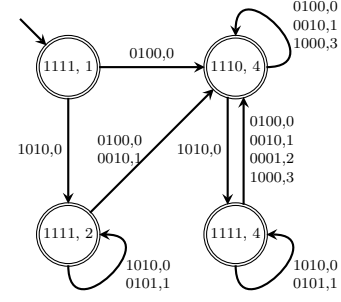


図 7: $\widetilde{M}_4$

証明 $\overline{C}_n$ から $\widetilde{M}_n$ が変形して得られることを示す.

状態 $q \in \widetilde{Q}$ に対して、 $\overline{C}_n$ では状態 $S(q, k) \in Q$ で $w \in \overline{D}_n$ が与えられた時、補題 12 より次の状態は $S(\delta_0(q, \pi_k(w)), k)$ で求められる. また、 $\delta_0(q, w) = (x, y) \neq \text{undefined}$ とすると $(x, y) = S((\text{ext}(\widehat{w}), y), \phi(w))$ が成立する. すなわち、ある状態 $r \in \widetilde{Q}$ が存在して $S(\delta_0(q, \pi_k(w)), k) = S(r, k + \phi(\pi_k(w)))$ が成立する.

$S(r, k + \phi(\pi_k(w)))$ の第一引数と第二引数を別々に考える. 第一引数は $r = (\text{ext}(\widehat{w}), \text{lcm}(k, p_w))$ と求められる. 第二引数は $k + \phi(\pi_k(w))$ であるが、 $\pi_{\phi(\pi_k(w))} \circ \pi_k = \pi_{\phi(\pi_k(w)) + k}$ が成立するので順列遷移関数を用いることができる. 従って、 $\overline{C}_n$ は $\overline{M}_n$ から変形して得られる. $\square$

実装上、枝ラベルには順列ではなく順列の添字を持たせる. $\widetilde{M}_4$ を図 7 に示す. 構築アルゴリズムは圧縮前と変わらず $O(4^n n^2)$ 時間であるが、 $\overline{M}_n$ の構築と比べて約 $1/n$ の時間で構築可能である. ただし、 $\overline{M}_n$ と $\overline{C}_n$ は異なる言語を保持するため $\overline{M}_n$ と $\widetilde{M}_n$ とでは保持する言語が異なる.

5. 実験

5.1 実験方法

$M_n, \overline{M}_n, \widetilde{M}_n$ の構築アルゴリズムを C++ version 17 (gcc version 7.4.0) で実装し、-O3 オプションをつけて最適化を行った. また実行環境は Ubuntu 18.04.2 LTS, Xeon CPU E5-2609 2.4GHz, 主記憶 256 GB である.

5.2 結果

$M_n, \overline{M}_n, \widetilde{M}_n$ それぞれの受理状態数と枝数を表 1, 2 に示す. M_n は $n = 31$ で、 $\overline{M}_n$ は $n = 30$ でメモリ不足により構築に失敗したが、 $\widetilde{M}_n$ は $n = 32$ までの構築に成功した. $n \geq 33$ については実験を行っていない. 構築時間は M_{30} と $\overline{M}_{29}$ は約 4 時間、 $\widetilde{M}_{32}$ は約 30 分であった.

表から M_n と $\overline{M}_{n-1}$ の状態数・枝数がほぼ一致していることがわかる. これは、縦線数 n のあみだくじに対するあみだくじ行列の行数が $n-1$ であり、縦線数 $n-1$ の循環あみだくじに対する循環あみだくじ行列の行数も $n-1$ であるため近い値になっていると考えられる. また、 n が大

表 1: 各 DFA の状態数

n	M_n	$\overline{M}_n$	$\widetilde{M}_n$
2	1	2	2
3	1	2	2
4	3	7	4
5	5	7	3
6	9	16	6
7	14	23	5
8	23	44	10
9	36	57	9
10	57	94	14
11	89	134	14
12	140	230	28
13	219	327	27
14	343	536	45
15	536	809	59
16	838	1297	92
17	1309	1957	117
18	2045	3126	188
19	3194	4771	253
20	4989	7549	395
21	7792	11659	563
22	12170	18308	848
23	19007	28384	1236
24	29685	44596	1895
25	46361	69238	2774
26	72405	108450	4200
27	113079	168916	6268
28	176602	264257	9487
29	275809	411860	14204
30	430746	-	21549
31	-	-	32407
32	-	-	49162

表 2: 各 DFA の枝数

n	M_n	$\overline{M}_n$	$\widetilde{M}_n$
2	1	3	3
3	2	4	4
4	8	28	16
5	19	32	16
6	50	134	54
7	114	284	68
8	281	797	200
9	634	1434	258
10	1458	3408	565
11	3278	7190	850
12	7484	17660	2443
13	16856	36674	3338
14	38137	86666	8031
15	85868	189497	14791
16	193676	436508	33527
17	435566	953298	59634
18	980270	2182095	139307
19	2204150	4820222	263018
20	4957830	10952158	594998
21	11146424	24408548	1208164
22	25063478	55135908	2608939
23	56345743	123210888	5420952
24	126676510	278407070	12073708
25	284764654	622671742	25109582
26	640153355	1403325537	54797036
27	1439014052	3147132745	117394187
28	3234812798	7085780102	256017856
29	7271524531	15898174368	549361392
30	16345684104	-	1202724990
31	-	-	2594425076
32	-	-	5672128714

きくなるにつれ M_n に対し $\overline{M}_{n-1}$ の状態数の方が少なくなっている。これは例えば長さ 5 のビット列 $w = 10100$ と $v = 10010$ のように, $\text{ext}(w) \neq \text{ext}(v)$ である一方で $\overline{\text{ext}}(w) = \overline{\text{ext}}(v)$ が成立する場合があるからである。枝数についても同様に $\overline{M}_{n-1}$ の方が少ないが, これは状態数が少ないことが影響していると思われる。

また, $\widetilde{M}_n$ の状態数, 枝数ともに $\overline{M}_n$ のおよそ $1/n$ になっていることがわかる。これは $\widetilde{M}_n$ で n 種類の順列を用いて圧縮を行ったためと考えられる。

表 3 に循環あみだくじの個数を示す。個数は縦線数や段数に対して指数的に増加する。

参考文献

- [1] Allocca, M. P., Dougherty, S. T. and Vasquez, J. F.: Signed permutations and the braid group, *Rocky Mountain Journal of Mathematics*, Vol. 47, No. 2, pp. 391–402 (2017).
- [2] Nozaki, Y., Hendrian, D., Yoshinaka, R. and Shinohara, A.: Enumeration of cryptarithms using deterministic finite automata, *International Conference on Implementa-*

表 3: 縦線数 n , 段数 d の循環あみだくじの個数

$n \backslash d$	1	2	3	4	5	6	7
2	1	2	4	8	16	32	64
3	1	3	9	27	81	243	729
4	2	8	34	150	674	3054	13890
5	2	14	92	596	3848	24824	160112
6	4	33	301	2818	26559	250763	2368862
7	4	64	912	12624	173392	2376528	32554064

tion and Application of Automata, Springer, pp. 286–298 (2018).

- [3] Yamanaka, K. and Nakano, S.: Enumeration, counting, and random generation of ladder lotteries, *IEICE Transactions on Information and Systems*, Vol. 100, No. 3, pp. 444–451 (2017).
- [4] Yamanaka, K., Nakano, S., Matsui, Y., Uehara, R. and Nakada, K.: Efficient enumeration of all ladder lotteries and its application, *Theoretical Computer Science*, Vol. 411, No. 16–18, pp. 1714–1722 (2010).
- [5] 田中勇真: あみだくじ数え上げ問題に対する高速解法, オペレーションズ・リサーチ: 経営の科学, Vol. 59, No. 10, pp. 595–600 (2014).