斐波那契词的移位的状态复杂度

Delaram Moradi[1]⋆、Pierre Popoli[1],Jeffrey Shallit[1]*和 Ingrid Vukusic[2]⋆⋆

[1] 滑铁卢大学计算机科学学院，加拿大安大略省滑铁卢市 University Ave. W. 200 号，
邮编 N2L
3
G
1delaram.moradi@uwaterloo.capierre.popoli@uwaterloo.cashallit@uwaterloo.ca

[2] 约克大学数学系，Ian Wand 大楼，Deramore Lane，约克，YO10 5GH，英国ingrid.vukusic@york.ac.uk

arXiv:2603.18858v1 [cs.FL] 19 Mar 2026

摘要。斐波那契无限词$\mathbf{f} = (f_i)_{i\geq 0} = 01001010\cdots$是词组合学中最著名的对象之一。存在一个简单的五状态自动机，对于以最低有效位优先的齐肯多夫表示给出的 i，它可以计算其第 i 项 f_i；对于最高有效位优先，则有一个二状态自动机。本文研究生成移位序列 $(f_{i+c})_{i\geq 0}$ 的自动机的状态复杂度，并证明无论采用最高有效位优先还是最低有效位优先输入，其复杂度均为 $O(\log c)$。对于非周期序列而言，这接近信息论下界。所用方法结合了状态复杂度技巧与丢番图逼近。

1 引言

状态复杂度是描述复杂度的重要主题之一。给定一个正则语言 L，我们想知道识别L 所需的最小有限自动机的规模。这一主题已被研究数十年，始于 Maslov 在 20 世纪 70 年代发表的奠基性论文 [12,13]，后来由 Yu、Zhuang 和 Salomaa 在西方推广开来 [20]。

在状态复杂度领域，一个重要的分支研究正则语言上的各种运算的状态复杂度上界，例如并、连接和 Kleene 星号。这里给定语言，比如 L_1 和 L_2，它们分别由具有 m 和 n 个状态的自动机识别；我们想知道，对这些语言 [20]施加某种一元或二元运算后，最小自动机的规模可能有多大。

Cobham [7,8] 引入了自动序列的概念：自动序列 $(a(i))_{i\geq 0}$ 是指由带输出确定性有限自动机（DFAO）生成的、取值于有限字母表 Δ上的序列。我们在某个计数系统中表示 i（例如，$k \geq 2$的底数为 k），并将表示中的数字 x输入自动机。每个状态都通过输出映射τ 映射到 Δ，而 $a(i)$则是将 τ 应用于输入 x后到达的最后一个状态所得的值。有两种不同的变体：一种从最高有效位开始读取表示（称为最高有效位优先），另一种从最低有效位开始读取（称为最低位优先）。一个序列是最低位优先自动序列，当且仅当它是最高有效位优先自动序列，不过所需的状态数可能相差悬殊。更正式地说，DFAO是一个六元组 $M = (Q, \Sigma, \delta, q_0, \Delta, \tau)$，其中 Q 是一个非空有限状态集

⋆ 研究由加拿大自然科学与工程研究委员会资助，项目编号 2024-03725。
⋆⋆ 研究由奥地利科学基金（FWF）资助，项目编号 10.55776/J4850。

；Σ 是输入字母表；$\delta\colon Q\times\Sigma\to Q$ 是转移函数，其定义域按通常方式扩展至 $Q\times\Sigma^*$；q_0 是初始状态；Δ 是非空有限输出字母表；$\tau\colon Q\to\Delta$ 是输出函数。本文假设，所有DFAO在输入中含有任意多个前导零时都给出相同且正确的输出（按最高有效位优先方式读取输入时），尾随零也必须满足同样的条件（按最低位优先方式读取时）。

当然，我们可以将这两个主题结合起来，研究自动序列上的运算的状态复杂度，例如取线性索引子序列 $(a(ci))_{i\geq 0}$，或将序列向后移若干项，得到 $(a(i+c))_{i\geq 0}$。这些主题曾在 [2], 中得到研究，之后又在近期的多篇论文中进行了探讨，例如 [21,6,22,14]。

本文聚焦于将一个序列向后移固定项数 c，从而得到新序列 $(a(i+c))_{i\geq 0}$。若 $(a(i))_{i\geq 0}$ 由一个 m状态的确定性有限自动机输出机生成，其输入为基数 k下的最低位优先输入，则我们可以使用确定性转换器即时计算逐位加法 c，构造出该序列的 $i+c$ 位移对应的自动机，其状态数为$O(m\log c)$。然而，对于 Zeckendorf 表示法中的数，这种方法行不通（见第2节），因为不存在能一次性完成加法的转换器[1]。

与十进制中的最低位优先情形相比，最高有效位优先情形可能导致更高的状态复杂度，因为同样不存在能计算和的确定性转换器（考虑在十进制中将输入 $9999\cdots9$ 加 1）。例如，最近有研究证明，对于著名的 Thue-Morse 序列 $\mathbf{t}=(t(i))_{i\geq 0}$ [3]，其确定性有限自动机输出机在最高有效位优先情形下对 $(t(i+c))_{i\geq 0}$ 的状态复杂度为 $O(c)$，但有无穷多个 [14] 使其超过 $c^{0.694}$。

然而，在最高有效位优先情形下，某些序列的移位序列状态复杂度会低得多。例如，考虑 2 的幂的特征序列 $\mathbf{p}_2=(p_2(i))_i$，其定义为：若 i 是 2 的幂，则 $p_2(i)=1$，否则为 0。可以证明，该序列 $(p_2(i+c))_{i\geq 0}$ 在最高有效位优先情形下的状态复杂度为 $O(\log c)$ [14]。这种对数增长似乎相当罕见。事实上，对数增长已接近非周期移位序列可能达到的最小复杂度膨胀，因为一个简单的计数论证表明，若自动序列 $(a(i))_{i\geq 0}$ 不是最终周期的，那么其移位序列 $(a(i+c))_{i\geq 0}$ 的状态复杂度必然会无穷多次超过 $c'(\log c)/(\log\log c)$。[3] 因此，$\mathbf{p}_2$ 的移位序列状态复杂度增长相当接近可设想的最小膨胀幅度。

本文考察一种特别著名的非周期无限词——斐波那契词 $\mathbf{f}=f(0)f(1)f(2)\cdots=01001010\cdots$ 的移位的状态复杂度。定义 $\mathbf{f}$ 的等价方式有很多，其中最著名的似乎是以下几种：

(a) 它是以 0 开始的无限不动点，由将 0 映射为 01、将 1 映射为 0 的态射生成。(b) 它是当 X_i 趋于 $i\to\infty$时，由极限产生的无限词，其中 $X_0=1$、$X_1=0$和$X_i=X_{i-1}X_{i-2}$。

[3] 具体而言，输入字母表为 $\{0,1,\ldots k-1\}$、输出字母表为 $\{0,1\}$ 的 n 状态确定性有限自动机输出机，至多有 $n^{kn-1}2^n$ 种。要么两个不同的移位相同（当且仅当该序列最终周期时会发生这种情况），要么必定存在某个移位 $c\leq n^{kn-1}2^n$，无法由具有 $\leq n$ 个状态的确定性有限自动机输出机计算。现在将 n 用 c 表示，即可得到这个界。

(c) 它是斜率为 $\gamma := 2 - \varphi$ 的斯图姆特征词，其中 φ 是黄金比例（$1+\sqrt{5}$）$/2 \doteq 1.61803$。换言之，它是由（$\lfloor (i+2)\gamma \rfloor - \lfloor (i+1)\gamma \rfloor$）$_{i \geq 0}$ 给出的词。(d) i 的第 f 个符号是 i 的齐肯多夫表示的最后一位数字。齐肯多夫表示将在第 2 节中回顾。

还有第五种定义f的方法，虽然不太为人所知，但事实证明它对本文至关重要。它涉及小数部分。令 $\{x\}$ 表示 x的小数部分；即 $x \bmod 1$。在全文中，我们还使用区间的圆周表示。如果$\{x\} < \{y\}$，则开区间 $(x, y) \bmod 1$ 表示为 $(\{x\}, \{y\})$。如果$\{x\} > \{y\}$，则 $(x, y) \bmod 1$ 表示为 $(\{y\}, 1) \cup [0, \{x\})$。因此，例如，$(\pi, e) \bmod 1 =$表示为$(.14159\cdots, .71828\cdots)$，而 $(e, \pi) \bmod 1 =$表示为$(.71828\cdots, 1) \cup [0, .14159\cdots)$。于是我们得到以下基本引理。

引理 1。 f的第 i 位，即 $f(i)$，当且仅当 $\{i\varphi\} \in (-\varphi, -2\varphi) \bmod 1$ 时为1。

证明。引理t $\gamma = 2 - \varphi$。于是，根据f 等价定义中的条目 (c)，我们有

$$\begin{aligned} f(i) = 1 \iff \lfloor (i+2)\gamma \rfloor - \lfloor (i+1)\gamma \rfloor = 1 &\iff \{(i+1)\gamma\} \in (1-\gamma, 1), \\ &\iff \{(i+1)(2-\varphi)\} \in (\varphi - 1, 1), \\ &\iff \{-(i+1)\varphi\} \in (\varphi - 1, 1), \\ &\iff \{(i+1)\varphi\} \in (0, 1-\varphi), \\ &\iff \{i\varphi\} \in (-\varphi, -2\varphi) \bmod 1. \end{aligned}$$

斐波那契词 f 广为人知且已得到深入研究 [4], 因此，理解其移位的状态复杂度是一个自然而重要的问题。我们的主要结果是，若将 f移位 c，则所得词可由一个仅有 $O(\log c)$ 个状态的确定性有限自动机输出机生成。此外，无论输入采用最高有效位优先还是最低位优先格式，这一界限都成立。我们还需要以下引理，它可直接由引理 1推出。

引理 2。 我们有 $f(i + c) = 1 \iff \{i\varphi\} \in (-(c+1)\varphi, \ -(c+2)\varphi) \bmod 1$。

证明。 由 引理 1 可得

$$\begin{aligned} f(i+c) = 1 &\iff \{(i+c)\varphi\} \in (-\varphi, -2\varphi) \bmod 1, \\ &\iff \{i\varphi\} \in (-(c+1)\varphi, -(c+2)\varphi) \bmod 1. \end{aligned}$$

虽然我们的主要结果关注某些自动机的状态复杂度，但证明技巧也颇有意义。我们综合运用了丢番图逼近、状态复杂度和逻辑方面的论证。特别是，证明中的部分内容借助自动定理证明器 `Walnut` [15,18]完成；只需用一阶逻辑陈述所需结果，它便能为关于自动序列的结果给出严格证明。在本次会议投稿版本中，许多细节放在附录里。

2 齐肯多夫表示与斐波那契自动序列

为了陈述结果和证明，我们需要介绍一种经典的计数系统，有时称为 Zeckendorf 数字系统 [11,23], ，不过 Ostrowski [16] 早在 1922 年就已经发表了一种更为一般的计数系统。在这个系统中，自然数表示为互不相同的斐波那契数之和 F_i ，其中 $i \geq 2$，按惯例记作 $F_0 = 0$，$F_1 = 1$，且$F_i = F_{i-1} + F_{i-2}$ 当 $i \geq 2$。为了确保表示唯一，我们规定每个斐波那契数至多使用一次，且不能使用两个连续的斐波那契数。例如，10 写作 $8 + 2$ ，而不是 $5 + 5$ 或 $5 + 3 + 2$。

我们将 n 的表示写成字母表 $\{0, 1\}$上的二进制词，记作 $(n)_F$，其形式为 $a_t a_{t-1} \cdots a_2$ ，其中 $n = \sum_{2 \leq i \leq t} a_i F_i$。例如，17 的表示为 $(17)_F = 100101$。注意，记号 $(\cdot)_F$ 采用最高有效位优先表示。类似地，对于长度为 ℓ、形式为 $x = x_{\ell-1} \cdots x_0$的二进制词 x，我们用 $[x]_F$表示相应的整数值；即 $[x]_F = \sum_{i=0}^{\ell-1} x_i F_{i+2}$。于是，对于每个整数 $n \geq 0$，都有 $[(n)_F]_F = n$；对于每个不含 11 的二进制词 x ，都有 $([x]_F)_F = x$。斐波那契数列中不能使用两个连续的斐波那契数这一要求，转化为有效的齐肯多夫表示不得包含两个连续的 1。

若序列（a（i））$_{i \geq 0}$ 满足以下条件，则称其为*Fibonacci*自动序列：存在一个确定性有限自动机输出机，以 i 的有效齐肯多夫表示作为输入，并输出 a（i）。斐波那契词**f** 是斐波那契自动序列的典型例子；它由图 1 中的自动机生成图 1。

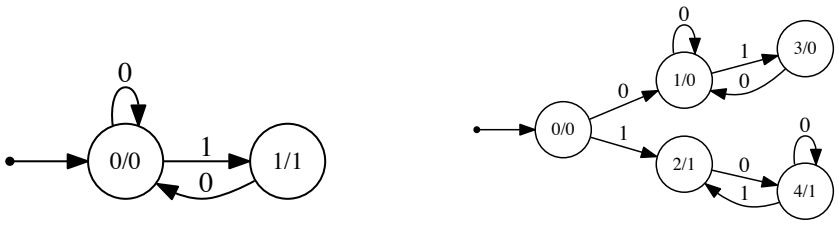

图 1。斐波那契词 **f**的确定性有限自动机输出机。左侧为最高有效位优先；右侧为最低位优先。

这里沿用惯例，状态上的标签 a/b 表示 a 是状态编号，而 b 是与该状态关联的输出。请注意，我们的约定是：构成无效齐肯多夫表示的输入（即含有两个连续 1 的输入）不会到达任何状态，也不会产生任何输出。

我们还需要用斐波那契数表示负数。尽管已有许多将负数纳入 Zeckendorf 数字系统的方案 [5,9,19,1], ，但我们需要一种此前尚未讨论过的方案。事实上，它是 Rockett 和 Sz¨usz [17]提出的一种更一般表示法的特例。具体而言，要表示 $-n$（其中 $n \geq 1$），我们首先找出最小的斐波那契数 $F_k \geq n$ ，并写出 $n = F_k - m$。然后，我们用 Zeckendorf 系统表示 m ，例如表示为 $\sum_{2 \leq i \leq t} a_i F_i$，并将 $-n$ 的表示定义为 $(-F_k) + \sum_{2 \leq i \leq t} a_i F_i$。注意，由于对所有 $k \geq 3$都有 $-F_k + F_{k-2} = -F_{k-1}$ ，因此 $t \leq k - 3$。我们可以将这种表示记作 $(-n)_F$，并将其视为字母表 $\{-1, 0, 1\}$ 上的一个词，其中首位数字为 -1 ，其余数字均为 0 或 1。这些表示的前几个例子见表 1。

$-n$	$(-n)_F$	$-n$	$(-n)_F$
-1	(-1)	-6	$(-1)0010$
-2	$(-1)0$	-7	$(-1)0001$
-3	$(-1)00$	-8	$(-1)0000$
-4	$(-1)001$	-9	$(-1)00101$
-5	$(-1)000$	-10	$(-1)00100$

表 1. 前几个负整数的表示。

3 最低有效位优先情形中的状态复杂度

本节将证明关于移位序列 $(f(i+c))_i$（采用最低有效位优先输入）的第一个主要结果，即其状态复杂度。

定理 1. 设 $(f(i))_{i\geq 0}$ 为斐波那契词 f。对于 $c \geq 0$，存在一个最低位优先确定性有限自动机输出机，可生成 $(f(i+c))_{i\geq 0}$，且具有 $O(\log c)$ 个状态。

关于 定理 1 的证明将推迟到本节末尾。我们先介绍由最低有效位优先表示诱导的区间 $[0,1)$ 的一个划分。

定义 1. 对于每个 m，满足 $0 \leq m < F_{d+1}$，令 $I_d(m)$ 为由 $I_d(m) = [L_d(m)\varphi, R_d(m)\varphi) \bmod 1$ 定义的区间，其中若 d 为偶数，则有

$$R_d(m) = -F_{d+1} + m \quad and \quad L_d(m) = \begin{cases} -F_d + m, & if\ m < F_d; \\ -F_{d+2} + m, & if\ m \geq F_d, \end{cases}$$

而若 d 为奇数，则交换 $R_d(m)$ 和 $L_d(m)$ 的值。

令 $d \geq 2$。对于每个整数 n，令 $n^{[\leq d]}$ 表示将 n 中所有索引大于 d的数字舍去后得到的整数；也就是说，若 $n = \sum_{2\leq i\leq t} a_i F_i$，则 $n^{[\leq d]} = \sum_{i=2}^{\min\{d,t\}} a_i F_i$。类似地，我们定义 $n^{[\geq d]} = \sum_{i=\max\{2,d\}}^{t} a_i F_i$。显然，我们有 $n = n^{[\leq d]} + n^{[\geq d+1]}$。以下引理说明，对于 $I_d(m$, 区间 $0 \leq m < F_{d+1}$ 构成 $[0,1$的一个划分。证明及所有必要细节见附录 A。

引理 3。 令 $d \geq 2$。那么，对于 $0 \leq m < F_{d+1}$，区间 $I_d(m)$ 构成 $[0,1)$的一个划分。此外，对于每个非负整数 n，我们有

$$\{n\varphi\} \in I_d(m) \iff n^{[\leq d]} = m. \tag{1}$$

非正式地说，整数 n 的低位 Zeckendorf 比特决定了 $n\varphi$ 所在的区间，而该区间的长度约为 φ^{-d}。

注 1。 对于每个 $d \geq 2$，引理 3 中的每个划分 $\{I_{d+1}(m)\colon\ 0 \leq m < F_{d+2}\}$ 是 引理 3 中划分 $\{I_d(m)\colon\ 0 \leq m < F_{d+1}\}$的细化。这一点可由区间的构造以及式（1）直接看出。

注 2。令 c 为一个正整数，使得对于某个 $k \geq 2$ 有 $F_k < c \leq F_{k+1}$（或 $c = 1$ 且$k = 2$）。令 $m = F_{k+1} - c$，并注意到 $m < F_{k+1} - F_k = F_{k-1} < F_k$。因此，我们得到

$$\begin{aligned}[L_k(m)\varphi, R_k(m)\varphi) \bmod 1 &= [(-F_k + m)\varphi, (-F_{k+1} + m)\varphi) \bmod 1 \\ &= [(-F_k + m)\varphi, -c\varphi) \bmod 1,\end{aligned}$$

若 k 为偶数。若 k 为奇数，则交换区间的端点。特别地，点$\{-c\varphi\}$ 是划分 $\{I_d(m)$ 中某个区间的端点：$0 \leq m < F_{d+1}\}$ 对所有 $d \geq k$ 均成立。此外，根据定义 1可以验证，点 $\{-c\varphi\}$不是 $d \leq k-1$ 的划分中的端点。

现在我们可以证明 定理 1。

证明（定理 1）。我们给出自动机 $M_c = (Q, \Sigma, \delta, q_0, \Delta, \tau)$ 的构造，用于在最低位优先输入下生成 $f(i + c)$。令 $\Sigma = \{0, 1\}$ 和 $\Delta = \{0, 1\}$。我们创建状态集 Q 和输出函数 τ，如下所示。对于 $k \geq 0$，令 $J_k = [-(k + 1)\varphi, -(k + 2)\varphi) \bmod 1$ 且 $J_k = [-(k + 2)\varphi, -(k + 1)\varphi) \bmod 1$。

首先，将初始状态记为 q_ε。由于与该状态关联的输出必须与 $f(c)$ 相匹配，引理 1 表明，$\tau(q_\varepsilon) = 1$ 当且仅当 $\{c\varphi\} \in J_0$。

接下来，我们添加四个状态 A_0、A_1 和 R_0、R_1。其中，A_0 和 A_1 是输出为 1 的汇状态，R_0 和 R_1 是输出为 0 的汇状态；也就是说，我们有

$$\begin{aligned}&\delta(A_0, 0) = A_0,\ \delta(A_0, 1) = A_1,\ \delta(A_1, 0) = A_0,\\ &\delta(R_0, 0) = R_0,\ \delta(R_0, 1) = R_1,\ \delta(R_1, 0) = R_0,\\ &\tau(A_0) = \tau(A_1) = 1,\ \tau(R_0) = \tau(R_1) = 0.\end{aligned}$$

请注意，由于 Zeckendorf 数制的数字限制，汇状态被拆分为两个；我们不能简单地让 0 和 1 输入都形成自环。

其余状态 q_x 对应于自动机在以最低位优先的方式读入输入 $x \in \{0, 1\}^*$后到达的状态。假设 $F_k < c + 2 \leq F_{k+1}$。根据注 2，我们知道 $\{-(c + 1)\varphi\}$ 和 $\{-(c + 2)\varphi\}$ 都是划分 $\{I_k(m)$ 中的端点：$0 \leq m < F_{k+1}\}$。此外，引理 3 表明，如果我们读取一个对应于 F_2、F_3、……、F_k 的整数 n 的数字，就能知道点 $\{n\varphi\}$ 落在哪个划分区间内。这意味着我们需要读取前 $k - 1$ 个数字。此后，我们就能确定点 $\{n\varphi\}$ 落在划分 $\{I_k(m) : 0 \leq m < F_{k+1}\}$ 的哪个区间内，尤其能确定 $\{n\varphi\} \in J_c = [-(c + 2)\varphi, -(c + 1)\varphi) \bmod 1$ 是否成立。由于根据引理 2，我们有$f(n + c) = 1 \iff \{n\varphi\} \in J_c$，这意味着自动机至多在读取 $k-1$ 个数字后就会到达汇状态。具体而言，如果最低位优先表示法中的最后一个输入是 0（分别为 1），它就会到达状态 A_0或 R_0（分别为 A_1或 R_1）。因此，我们只需考虑自动机在读取长度为 $\leq k - 1$ 的输入 x 后到达的状态。

为了估算所得自动机的规模，我们统计给定长度为 $1 \leq \ell \leq k - 1$ 的这些词 x 中，有多少会使自动机到达非汇状态。首先，考虑 $\ell = 1$ 的情况，此时只有两个可能的输入。我们区分以下两种情况：

– 若 $\{-(c+1)\varphi\}$ 和 $\{-(c+2)\varphi\}$ 属于 J_0，则添加一个状态 q_0。这种情况对应于满足 $f(c-1) = f(c-2) = 0$ 的 c。从 q_ε 出发的转移定义为：若 $f(c+1) = 1$，则 $\delta(q_\varepsilon, 0) = q_0$ 和 $\delta(q_\varepsilon, 1) = A_1$；若 $f(c+1) = 0$，则 $\delta(q_\varepsilon, 1) = R_1$。– 否则，添加两个状态 q_0 和 q_1，从 q_ε 出发的转移对 $a \in \{0,1\}$ 定义为 $\delta(q_\varepsilon, a) = q_a$。

现在假设 $2 \le \ell \le k-1$。对于输入字 $x \in \{0,1\}^\ell$，令 $m = [x]_F$ 和 $I_{\ell+1}(m)$ 为 定义 1中定义的区间。我们区分以下两种情况：

– 如果 $I_{\ell+1}(m) \subseteq J_c$，那么根据引理 2，自动机 M_c 在读入 x后到达汇状态 A_i，其中 i 是 x的最后一个字母。同样地，如果 $I_{\ell+1}(m) \subseteq J_c$，那么 M_c 在读入 $\overline{x}$后到达汇状态 R_i，其中 i 是 x的最后一个字母。– 否则，两个点 $\{-(c+2)\varphi\}$ 或 $\{-(c+1)\varphi\}$ 中恰有一个包含在 $I_{\ell+1}(m)$中。在这种情况下，我们向 Q添加一个新状态 q_x，并且 q_x 的输出为 $\tau(q_x) = 1$，当且仅当 $\{m\varphi\} \in J_c$。由于集合 $\{I_{\ell+1}(m)\colon\ 0 \le m < F_{\ell+2}\}$ 根据引理 3 构成一个划分，因此对于每个长度 ℓ，这种情况至多发生两次。

因此，对于每个给定的长度 $1 \le \ell \le k-1$，至多增加两个非汇状态。这些状态上的转移函数定义为 $\delta(q_x, a) = q_{xa}$。因此，Q 中的状态总数至多为 $1 + 4 + 2(k-1) = 2k+3$，确实是 $O(\log c)$，因为 $F_k < c+2 \le F_{k+1}$。

例 *1*。 我们用 $c = 10$来说明上一定理的证明。在图 2上半部分最底下一行中，可以看到 $\{-12\varphi\}$ 和 $\{-11\varphi\}$ 是读入 5 个数字后得到的划分的端点。由于 $c+2 = 12 \le 13 = F_7$，注 释 2 保证读入 5 个数字后确实会出现这种情况。 图 2 下半部分说明，且仅当 $\{n\alpha\} \in [-11\varphi, -12\varphi) \bmod 1$ 时，我们才希望输出 1；这对应于顶行的蓝色区域。完全包含在 $[-11\varphi, -12\varphi) \bmod 1$ 中的区间标为蓝色。蓝色区间表示读入相应数字后，我们可以转移到某个汇状态 A_0, A_1。类似地，完全包含在 $[-12\varphi, -11\varphi) \bmod 1$ 中的区间标为橙色，并与汇状态 R_0, R_1相关联。包含某个“临界点” $\{-11\varphi\}, \{-12\varphi\}$ 的区间标为黑色。这些区间对应于下方列出的“中间状态”。

总的来说，这些状态如下：

– 初始状态 q_ε，输出为 0，因为 $\{0\varphi\} \notin [-11\varphi, -12\varphi)$。– 中间状态： $q_0, q_1, q_{01}, q_{10}, q_{100}, q_{010}, q_{1000}, q_{0100}$。每个中间状态的输出都可以明确计算出来，并可在图 2 的下半部分看到（通过橙色/蓝色的点 $\{\varphi\}, \{0\}, \{2\varphi\}$）。事实上，只需计算每个 q_x 对应的 $\{[x^R]_F\varphi\}$ 值，其中 z^R 是 z 的逆像，并检查它是否属于区间 $[-11\varphi, -12\varphi) \bmod 1$，以输出 1。– 两对汇状态： A_0, A_1 和 R_0, R_1。

该自动机展示在图 3中。

下一个定理给出了序列 $(f(i+c))_{i \ge 0}$的状态复杂度的精确公式。特别是，该公式意味着 定理 1 中的上界是最优的。关于 定理 2 的证明见附录A 节。

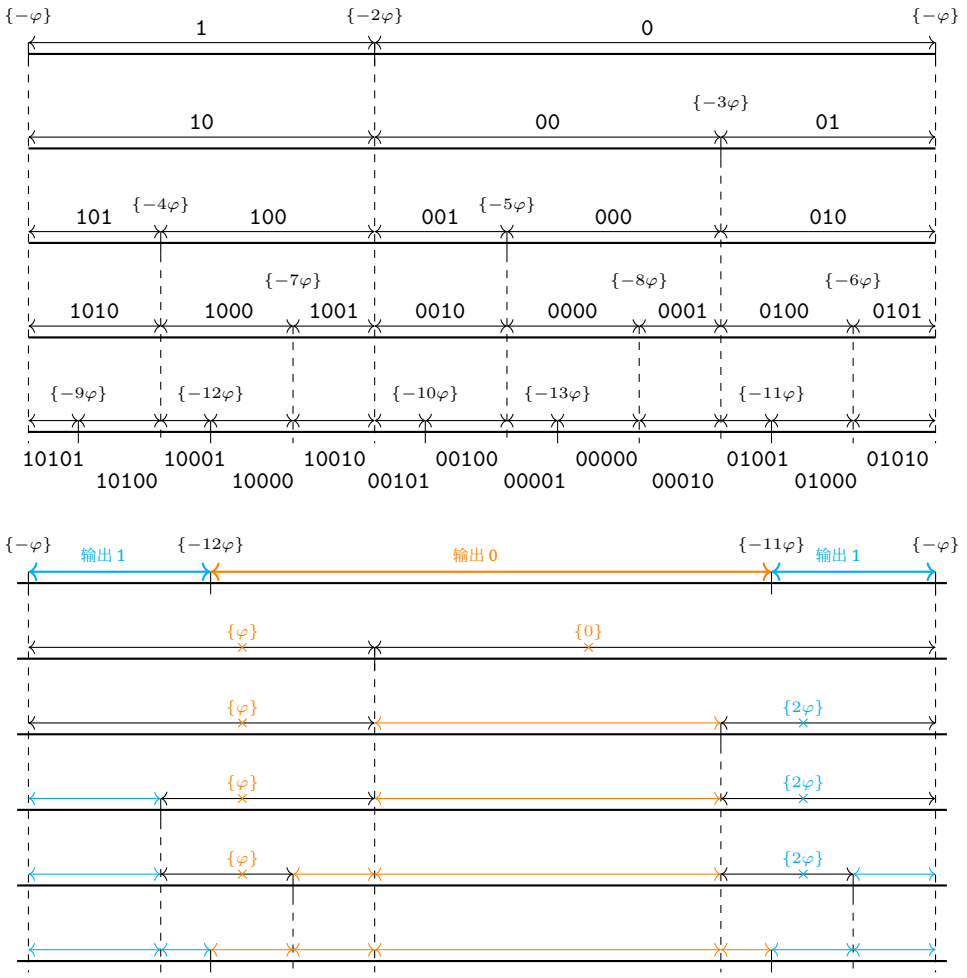

图 2。 $c = 10$ 的划分引理示意图及自动机构造。

定理 2。设 $(f(i))_{i\geq 0}$ 为斐波那契词 f。对于 $c \geq 5$，生成 $(f(i+c))_{i\geq 0}$ 的最小低位优先确定性有限自动机输出机的状态数为 $2(|(c)_F| + 1) + 1 - g(c)$，其中 $g(c) = 1$ 当 $f(c-1) = f(c-2) = 0$ 时成立，而 $g(c) = 0$ 在其他情况下成立。特别是，将 c 映射为 $(f(i+c))_{i\geq 0}$ 的状态复杂度的函数是斐波那契正则的。

4 最高有效位优先情形下的状态复杂度

现在，我们来讨论最高有效位优先格式下的同一问题。要确定一个生成移位斐波那契词 $(f(i+c))_i$ 的最高有效位优先格式 DFAO，可以利用 定理 1 中基于最低位优先格式的构造。为此，我们将自动机中的所有转移反转，并将得到的新自动机确定化。这一操作可能导致状态数指数级膨胀，因此对于最高有效位优先格式，我们只能得到 (c^e) 个状态的上界，其中 $e > 0$。不过，本节将证明以下强得多的定理。

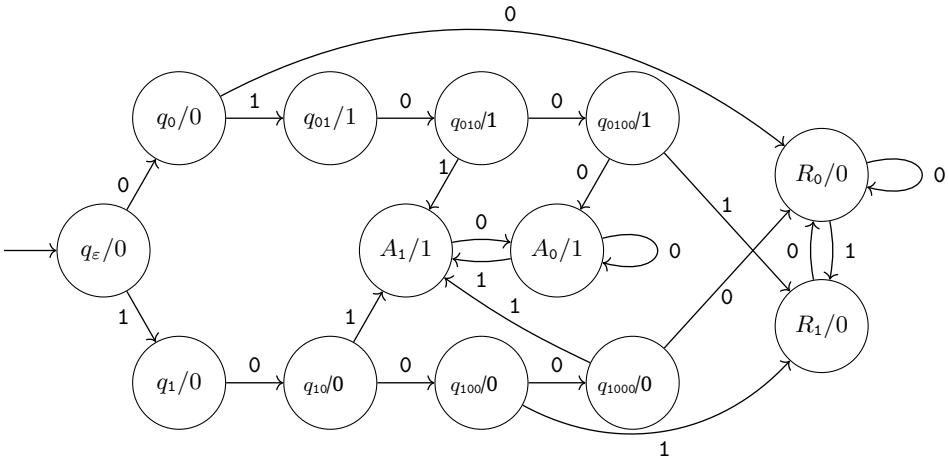

图 3. $(f(i+10))_{i\geq 0}$ 的最低有效位优先 DFAO。

定理 3。 对于所有整数 $c \geq 0$，都存在一个含有 $O(\log c)$ 个状态的斐波那契 *DFAO*，能够以最高有效位优先格式生成移位斐波那契词 $(f(i+c))_{i\geq 0}$。

定理 3 的证明推迟到本节末尾。和之前一样，我们首先引入一个与最高有效位优先表示相关的划分。设 $n = \sum_{2\leq i\leq t} a_i F_i$ 是 n的斐波那契表示。对于 $a \in \{0,1\}$，我们记作

$$n^{(a)} = \sum_{i=3}^{t+1} a_{i-1}F_i + aF_2. \tag{2}$$

换句话说，操作 $n^{(a)}$ 会在 n的齐肯多夫展开右端追加比特 a。注意，如果 $a_2 = 1$，那么 $n^{(1)}$ 不是有效的斐波那契表示。对于负数 $m = -F_k + \sum_{2\leq i\leq t} a_i F_i$，我们类似地定义

$$m^{(a)} = -F_{k+1} + \sum_{i=3}^{t+1} a_{i-1}F_i + aF_2, \tag{3}$$

并且，如果 $a = 1$，上述表示也可能不是有效表示。

n	$(n)_F$	$(n^{(0)})_F$	$(n^{(1)})_F$
1	1	10	-
2	10	100	101
3	100	1000	1001
4	101	1010	-
5	1000	10000	10001
6	1001	10010	-

表 2。 $(\cdot)^{(a)}$ 操作示例。

我们还需要将这些表示（正数和负数）向右移一位，在最高有效位优先表示中丢弃最右侧的位。我们将此操作记作

符号 $'$表示，如 n'。因此，如果 $n = \sum_{2\le i\le t} a_i F_i$，那么 $n' = \sum_{2\le i\le t-1} a_{i+1}F_i$。类似地，如果 $-n = (-F_k) + \sum_{2\le i\le t} a_i F_i$，那么 $(-n)' = (-F_{k-1}) + \sum_{2\le i\le t-1} a_{i+1}F_i$。注意，对于 $m = -1 = -F_2$，此定义给出 $(-1)' = -F_1 = -1$。该操作的简要表格见 表3。

n	$(n)_F$	n'	$(n')_F$	$(-n)'$	$((-n)')_F$
1	1	0	0	-1	(-1)
2	10	1	1	-1	(-1)
3	100	2	10	-2	$(-1)0$
4	101	2	10	-3	$(-1)00$
5	1000	3	100	-3	$(-1)00$
6	1001	3	100	-4	$(-1)001$
7	1010	4	101	-5	$(-1)000$
8	10000	5	1000	-5	$(-1)000$
9	10001	5	1000	-6	$(-1)0010$

表3. $'$ 运算示例。

定义2。 设 $\mathcal{P}$ 是集合 $[0,1)$ 模1的区间划分。若对每个区间 $I \in \mathcal{P}$，都存在两个区间 I_0、$I_1 \in \mathcal{P}$，使得对所有非负整数 n，均有以下性质，则称 $\mathcal{P}$与最高位优先斐波那契表示相一致：

$$\{n\varphi\} \in I \implies \begin{cases} \left\{n^{(0)}\varphi\right\} \in I_0, & \text{if } f(n) = 1; \\ \left\{n^{(0)}\varphi\right\} \in I_0 \text{ and } \left\{n^{(1)}\varphi\right\} \in I_1, & \text{otherwise.} \end{cases}$$

注意，在上述定义中，条件 $f(n) = 1$ 等价于其齐肯多夫表示中的最低有效位 n 为**1**；参见第1节中斐波那契词的定义（d）。

引理 4。 设 $\mathcal{P}_A$ 和 $\mathcal{P}_B$ 是 $[0,1)$ 的两个与最高位优先斐波那契表示相一致的划分。那么，将 $\mathcal{P}_A$ 和 $\mathcal{P}_B$ 两者的端点合并所得到的划分$\mathcal{P}_{A,B}$ 也与最高位优先斐波那契表示相一致。

证明。 考虑区间$I \in \mathcal{P}_{A,B}$。那么，即使 I 的两个端点都来自同一个划分， I 也可以写成某些区间 $I_A \in \mathcal{P}_A$ 和 $I_B \in \mathcal{P}_B$的交集；换言之， $I = I_A \cap I_B$。因此，根据假设，对于 $a \in \{0,1\}$，存在区间$I_{A,a} \in \mathcal{P}_A, I_{B,a} \in \mathcal{P}_B$，使得对所有满足 $\{n\varphi\} \in I$的 n，都有 $\{n^{(a)}\varphi\} \in I_{A,a} \cap I_{B,a} \in \mathcal{P}_{A,B}$。因此，新划分与最高位优先斐波那契表示相一致。

定义 3。 设 $r \le -2$ 是一个表示为 $r = -F_R + \sum_{i=2}^{R-3} b_i F_i$ 的负整数。我们定义 $[0,1)$ 的划分 $\mathcal{P}(r)$ 如下。首先，令 $r_0 = r$，并递归定义$r_k = r'_{k-1}$，其中 $1 \le k \le R-2$；或者等价地

$$r_k = -F_{R-k} + \sum_{i=2}^{R-3-k} b_{i+k}F_i, \quad 0 \le k \le R-2.$$

然后我们定义 $\mathcal{P}(r)$ 为集合 $[0,1)$ 按模1划分成区间，其中端点恰好是点 $\{r_k\varphi\}$ *for* $0 \le k \le R-2$。注意，我们总有$r_{R-2} = -1$、 $r_{R-3} = -2$，并且当 $r \le -3$ 时有 $r_{R-4} = -3$。

例*2.* 我们用示例 $r = -14 = -F_8 + F_5 + F_3$来说明划分 $\mathcal{P}(r)$:

$$\begin{aligned}
r_0 &= -F_8 + F_5 + F_3 = -14, & \{-14\varphi\} &= 0.347\cdots,\\
r_1 &= -F_7 + F_4 + F_2 = -9, & \{-9\varphi\} &= 0.437\cdots,\\
r_2 &= -F_6 + F_3 = -6, & \{-6\varphi\} &= 0.291\cdots,\\
r_3 &= -F_5 + F_2 = -4, & \{-4\varphi\} &= 0.527\cdots,\\
r_4 &= -F_4 = -3, & \{-3\varphi\} &= 0.145\cdots,\\
r_5 &= -F_3 = -2, & \{-2\varphi\} &= 0.763\cdots,\\
r_6 &= -F_2 = -1, & \{-\varphi\} &= 0.381\cdots.
\end{aligned}$$

那么，划分 $\mathcal{P}(-14)$ 由以下 7 个区间组成:

$$[-3\varphi, -6\varphi) \bmod 1,\ [-6\varphi, -14\varphi) \bmod 1,\ [-14\varphi, -\varphi) \bmod 1, [-\varphi, -9\varphi) \bmod 1,$$
$$[-9\varphi, -4\varphi) \bmod 1,\ [-4\varphi, -2\varphi) \bmod 1,\ [-2\varphi, -3\varphi) \bmod 1.$$

现在我们陈述本节的主要引理。其证明见附录 B。

引理 5. 设 $r \le -2$ 是一个负整数，且 $\mathcal{P}(r)$ 是来自 定义 *3*的划分。那么$\mathcal{P}(r)$与最高位优先斐波那契表示相一致。

现在我们可以转而证明定理 3。

（定理 3 的证明。）考虑划分 $\mathcal{P}$，它由定义 3 中两个划分 $\mathcal{P}(-(c+1))$ 和 $\mathcal{P}(-(c+2))$ 的端点合并构造而成。因此，根据引理 5 和引理 4，划分 $\mathcal{P}$ 与最高位优先斐波那契表示一致。在 $\mathcal{P}$中的区间里，令 J 为包含 0 的区间。我们如下定义确定性有限自动机输出机 $M_c = (Q, \Sigma, \delta, q_0, \Delta, \tau)$：

– $Q = \mathcal{P}$, $\Sigma = \{0,1\}$, $\Delta = \{0,1\}$, $q_0 = J$,– $\delta(I, a) = I_a$ 其中 $I_a \in \mathcal{P}$ 是对每个满足 $\{n\varphi\} \in I$ 的 n 所落入的区间，– $\tau(I) = \begin{cases} 1, & \text{若 } I \subseteq (-(c+1)\varphi, -(c+2)\varphi) \bmod 1; \\ 0, & \text{否则。} \end{cases}$.

函数的正确性 δ 源于 $\mathcal{P}$ 与最高位优先斐波那契表示一致。函数的正确性 τ 源于 引理 2 以及这样一个事实：根据 定义 3中给出的构造，$\{-(c+1)\alpha\}$ 和 $\{-(c+2)\alpha\}$ 都是 $\mathcal{P}$的端点。

因此，根据 $\mathcal{P}(-(c+1))$ 和 $\mathcal{P}(-(c+2))$ 的构造，$\mathcal{P}$ 中有 $O(\log c)$ 个端点，因此 $\mathcal{P}$ 中有 $O(\log\ c)$ 个区间。这证明 M_c 中有 $O(\log c)$ 个状态。

例 *3.* 作为一个例子，考虑 $c = 10$。图中所示的自动机 图 4 展示了我们的构造以及相应的区间。由于 $-12 = -13 + 1 = -F_7 + F_2$，划分 $\mathcal{P}(-12)$ 中的端点对应于整数 $\{-12, -8, -5, -3, -2, -1\}$; 并且

类似地，由于 $-11 = -13+2 = -F_7+F_3$，划分 $\mathcal{P}(-11)$ 对应于整数$\{-11,-7,-5,-3,-2,-1\}$。因此，我们得到如下由八个区间组成的划分 $\mathcal{P}$：

$$\mathcal{P} = \big\{[-5\varphi,-8\varphi) \bmod 1,\ [-8\varphi,-3\varphi) \bmod 1,\ [-3\varphi,-11\varphi) \bmod 1,\ [-11\varphi,-\varphi) \bmod 1$$
$$[-\varphi,-12\varphi) \bmod 1,\ [-12\varphi,-7\varphi) \bmod 1,\ [-7\varphi,-2\varphi) \bmod 1,\ [-2\varphi,-5\varphi) \bmod 1\big\},$$

而证明定理 3 中的构造给出了图 4所示的自动机。

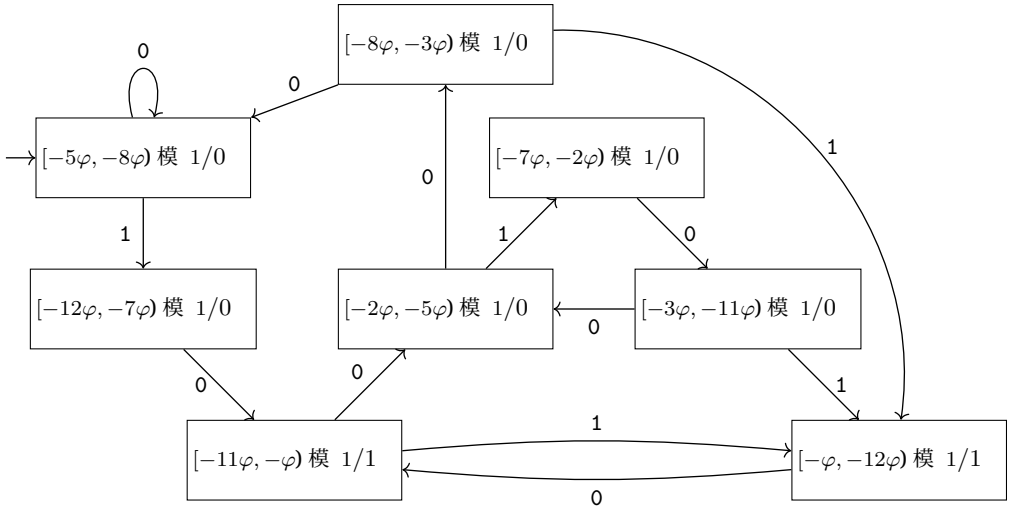

图 4. $(f(i+10))_{i\geq 0}$ 的最高有效位优先确定性有限自动机输出机。

5 结束语

考察斐波那契自动序列变换的状态复杂度，会提出一些颇具挑战性的问题，这些问题处于形式语言理论、词组合学和数论的交界处。我们还可以考察许多其他问题。

我们注意到，这些结果显然可以推广：有些情形可推广到二次无理数，另一些情形则可更一般地推广到无理数。类似地，我们也可以对特里波那契词（态射 $0 \to 01, 1 \to 02, 2 \to 0$ 的不动点）提出同类问题，而其分析更令人费解。我们希望在未来的论文中探讨这些问题。

参考文献

1. S. Labb ´ ´ *RAIRO Inform.* e 和 *Théor. App.* 57 12, 2023
J. Lep˘sova。斐波那契类比二补数计数系统。, : 。2. J.-P.AlloucheandJ.Shallit. *Automatic Sequences: Theory, Applications, Generalizations*。剑桥大学出版社，剑桥，2003。

3. J.-P. Allouche 和 J. O. Shallit。无处不在的普鲁埃-蒂厄-莫尔斯序列。载于 C. Ding、T. Helleseth 和 H. Niederreiter 编， *Sequences and Their Applications*， *Proceedings of SETA '98*，第 1–16 页。施普林格出版社，1999。4. J. Berstel。斐波那契词——综述。载于 G. Rozenberg 和 A. Salomaa 编， *The Book of L*，第 13–27 页。施普林格出版社，1986。5. M. W. Bunder。使用负斐波那契数的齐肯多夫表示。 *Fibonacci Quart.*，30 115，1992 111：–。6. ´E. Charlier、N. Rampersad、M. Rigo 和 L. Waxweiler。在线性计数系统中识别 $m\mathbb{N}$ 的最小自动机。 *Integers*，11B：论文编号 A4，24，2011。7. A. Cobham。有限自动机可识别的数集对基数的依赖性。 *Math. Systems Theory*，3 186 192，1969：–。8. A. Cobham。一致标记序列。 *Math. Systems Theory*，6：164–192，1972。9. P. Hajnal。允许负数字的计数系统简记。 *Bull. Inst. Combin. Appl.*，97 66，2023 54：–。10. R. Knott。斐波那契数与黄金比例公式。https://r-knott.surrey.ac.uk/Fibonacci/fibFormulae.html。访问日期：2026年1月16日。11. C. G. Lekkerkerker。用斐波那契数之和表示自然数。 *Simon Stevin*，29 190 195，1952：–。12. A. N. Maslov。有限自动机状态数的估计。 *Dokl. Akad. Nauk SSSR*，

194 6 1266 1268, ( )：–。1970 *Soviet Math. Dokl.* **11** 5 1970 1373 1375 俄文。英文译本见 ( ) ( )，–。13. A. N. Maslov。语言的循环移位操作。 *Problemy Peredachi Informatsii*，9(4)：87–87，1973。俄文。英文译本见 *Problems Info. Trans.* **9** (1973)，333–338。14. D. Moradi、N. Rampersad 和 J. Shallit。 k-自动序列的线性子序列复杂度。Arxiv 预印本 arXiv:2512.10017 [cs.FL], 2025。15. H. Mousavi。在 Walnut 中进行自动定理证明。ArXiv 预印本 arXiv:1603.06017 [cs.FL], 2021。16. A. Ostrowski。关于丢番图逼近理论的注记。 *Abh. Math. Sem. Hamburg*，

1 77 98, 250 251, 1922 *Collected Mathematical Papers* 3, 57 80 ：—。重印于《》，第 卷，第 页。–。17. A. M. Rockett 和 P. Sz¨usz。 *Continued Fractions*。世界科学出版社，1992。18. J. Shallit。 *The Logical Approach To Automatic Sequences: Exploring Combinatorics on Words with* Walnut， *London Mathematical Society Lecture Note Series* 第 482 卷。剑桥大学出版社，2023。19. J. Shallit 和 S. L. Shan。通过自动机理论证明斐波那契表示性质的一般方法。载于

Proceedings of the 16th International Conference on Automata and Formal Languages，*Electron. Proc. Theor. Comput. Sci. (EPTCS)* 第 386 卷，第 228–242 页。EPTCS，2023。20. S. Yu、Q. Zhuang 和 K. Salomaa。正则语言若干基本运算的状态复杂度。

Theoret.，：–。*Comput. Sci.* 125 315 328, 1994

21. H. Zantema。自动序列的复杂度。载于 A. Leporati 等编， *LATA 2020*，*Lecture Notes in Computer Science* 第 12038 卷，第 260–271 页。施普林格出版社，2020。22. H. Zantema 和 W. Bosma。自动序列的复杂度。 *Inform. Comput.*，288：104710，2022。23. E. Zeckendorf。用斐波那契数或卢卡斯数之和表示自然数。 *Bull. Soc. Roy. Liége*，41：179–182，1972。

A 最低位优先细节

A.1 丢番图逼近与引理3的证明

我们首先回顾斐波那契数和黄金比例的一些基本性质。由斐波那契数的比内公式可以推导出（例如，参见 [10] ），

$$\{F_i\varphi\} = \left\{(-1)^i\varphi^{-i}\right\}. \tag{4}$$

令 $d \geq 2$。对于每个整数 $n = \sum_{2\leq i\leq t} a_i F_i$，回顾 $n^{[\leq d]} = \sum_{i=2}^{\min\{d,t\}} a_i F_i$，以及 $n^{[\geq d]} = \sum_{i=\text{最大值}\{2,d\}}^{t} a_i F_i$。由于 $n = n^{[\leq d]} + n^{[\geq d+1]}$，并由式（4），我们得到

$$\{n\varphi\} = \left(n^{[\leq d]}\varphi + \sum_{i=d+1}^{t}(-1)^i a_i\varphi^{-i}\right) \bmod 1. \tag{5}$$

我们希望估计项 $\sum_{i=d+1}^{t}(-1)^i a_i\varphi^{-i}$；它可以理解为一个误差项。注意，对于奇数指标 i，我们得到负项；对于偶数指标 i，则得到正项。下面这个简单恒等式及其导出的界共同表明，我们确实可以用显而易见的方式估计误差项。我们有

$$\sum_{i=0}^{\infty}\varphi^{-(d+1+2i)} = \varphi^{-d} = \left\{(-1)^d F_d\varphi\right\}, \tag{6}$$

特别地，

$$\sum_{i=0}^{\infty}\varphi^{-(d+1+2i)} + \sum_{i=0}^{\infty}\varphi^{-(d+2+2i)} \leq \varphi^{-2} + \varphi^{-3} = \varphi^{-1} < 1. \tag{7}$$

因此，由式（5）、（6）和（7）可得

$$\{n\varphi\} \in \begin{cases} \left(n^{[\leq d]}\varphi - \varphi^{-d}, n^{[\leq d]}\varphi + \varphi^{-(d+1)}\right) \bmod 1, & \text{if } d \text{ is even;} \\ \left(n^{[\leq d]}\varphi - \varphi^{-(d+1)}, n^{[\leq d]}\varphi + \varphi^{-d}\right) \bmod 1, & \text{if } d \text{ is odd.} \end{cases} \tag{8}$$

此外，如果 $a_d = 1$，那么 $a_{d+1} = 0$，因此实际上 $\varphi^{-(d+1)}$ 不可能出现在误差项中。注意，根据齐肯多夫表示的贪心性质，当且仅当 $n^{[\leq d]} \geq F_d$ 时，我们有 $a_d = 1$ 。因此，如果 $n^{[\leq d]} \geq F_d$，事实上有

$$\{n\varphi\} \in \begin{cases} \left((n^{[\leq d]} - F_{d+2})\varphi, (n^{[\leq d]} - F_{d+1})\varphi\right) \bmod 1, & \text{if } d \text{ is even;} \\ \left((n^{[\leq d]} - F_{d+1})\varphi, (n^{[\leq d]} - F_{d+2})\varphi\right) \bmod 1, & \text{if } d \text{ is odd.} \end{cases} \tag{9}$$

我们现在可以证明引理 3。

证明（引理 3）。由（8）和（9），以及点 $\{n\varphi\}$ 在 (0,1) 中稠密这一事实，显然半开区间 $I_d(m)$，其中 $0 \le m \le F_{d+1} - 1$，覆盖 $[0, 1)$。

还需证明这些区间两两不相交。只需计算它们的总长度即可。根据构造，长度为

$$\left\{(-1)^d(-F_{d+1} + F_d)\varphi\right\} = \left\{(-1)^{d-1}F_{d-1}\varphi\right\} = \varphi^{-(d-1)},$$

的区间恰有 $F_{d+1} - F_d = F_{d-1}$ 个，长度为

$$\left\{(-1)^d(-F_{d+1} + F_{d+2})\varphi\right\} = \left\{(-1)^d F_d\varphi\right\} = \varphi^{-d}.$$

因此，总长度等于 $F_d\varphi^{-(d-1)} + F_{d-1}\varphi^{-d} = 1$，由比内公式可得（或者参见例如[10]）。因此，这些区间必定两两不相交，并构成一个划分。此外，根据上述论证，(1) 显然也成立。

A.2 定理2的证明

证明（定理 2）。设 $c \ge 5$。我们证明，定理 1的证明中的构造几乎是最优的。也就是说，我们证明所构造的自动机 $M_c = (Q, \Sigma, \delta, q_0, \Delta, \tau)$ 经过至多一次修改后即为最小自动机，并统计集合 Q中的状态数。回忆一下， Q包含

– 初始状态 q_ε,– 四个汇状态 A_0, A_1, R_0, R_1,– 一定数量的中间状态 q_x，其中字符串 x 对应于某些区间。

更精确地描述中间状态，我们有：字符串 x 对应于满足 $\{-(c+1)\varphi\} \in I_d(m)$ 或 $\{-(c+2)\varphi\} \in I_d(m)$，且$m = [x^R]_F$的整数对 (d, m)，其中 z^R 是 z的逆像，而 x 可以用 0 补齐，使得$|x| = d - 1$。我们区分以下两种情况。

情况 1：存在某个 $k \ge 1$ 使得 $F_k < c + 1 < c + 2 \le F_{k+1}$。那么，根据注 2，我们知道 $\{-(c+1)\varphi\}, \{-(c+2)\varphi\}$ 恰好在阶段 $d = k$成为划分 $\{I_d(m) : 0 \le m < F_{d+1}\}$ 的端点。这意味着我们只在阶段 $d = 2, 3, \ldots k - 1$添加状态。回想一下，在自动机的构造中，在第一个阶段（$d = 2$），即读入一个数字之后，我们会添加一个或两个状态。事实上，当且仅当 $f(c-1) = f(c-2) = 0$时，我们才只添加一个状态。由于根据定义，在这种情况下 $g(c) = 1$，而在其他情况下 $g(c) = 0$，因此我们在阶段 $d = 2$恰好添加 $2 - g(c)$ 个状态。在所有其他阶段（$d = 3, \ldots, k-1$），我们添加两个状态。因此，总共得到 $5+(2-g(c))+2(k-3) = 2k+1-g(c)$ 个状态。由于 $F_k \le c < F_{k+1}$，我们有 $|(c)_F| = k - 1$，状态数确实为 $2(|(c)_F| + 1) + 1 - g(c)$，正如所需。

显然，所有状态都是可达的。因此，要证明最小性，我们只需证明不存在两个等价的状态。也就是说，我们需要证明，对于所有状态 $q \ne q' \in Q$，都存在字符串 u， $u' \in \Sigma^*$，使得 $\tau(\delta(q, u)) \ne \tau(\delta(q', u))$。

令 m_1，m_2 为满足 $\{-(c+1)\varphi\} \in I_{k-1}(m_1)$ 且 $\{-(c+2)\varphi\} \in I_{k-1}(m_2)$ 的唯一整数。这些区间称为“临界区间”，因为在下一阶段，它们各自会分成两个子区间，而每个子区间都完全包含在 $[-\varphi, -2\varphi) \bmod 1$ 或 $[-2\varphi, -\varphi) \bmod 1$ 中。令 x_1，x_2 为对应的“临界串”，其中 $|x_1| = |x_2| = k-2$ 且 $[x_1^R]_F = m_1$，$[x_2^R]_F = m_2$。于是，根据构造，字符串 $x_1\mathtt{0}$ 和 $x_1\mathtt{1}$ 必须是有效输入，并且必须产生不同的输出；字符串 $x_2\mathtt{0}$ 和 $x_2\mathtt{1}$ 也同样如此。

现在我们证明任意两个状态都不等价。首先，很明显，任何汇状态 A_0、A_1、R_0、R_1 都不可能与其他状态等价。

现在设 $q = q_x$ 和 $q' = q_{x'}$ 是对应于（可能为空）字符串 $x \neq x'$ 的两个中间状态。根据 Q 的构造，字符串 x、x' 都是 x_1 或 x_2 的前缀。设 $xy = x_i$ 和 $x'y' = x_j$ 分别为某些（可能为空）的词 y、y' 和 i、$j \in \{1, 2\}$（其中可能有 $i = j$）。不失一般性，假设 $|x| \leq |x'|$，因此 $|y| \geq |y'|$ 也有 $|x_1| = |x_2|$。接下来根据 $y = y'$ 是否成立，分两种情况讨论。

情况 a： 假设 $y \neq y'$。那么我们知道，读入 $x'y$ 后会到达一个汇状态。设 b 为该汇状态的输出，即 $\tau(\delta(q_\varepsilon, x'y)) = b$，并且对于每个数字 $a \in \{\mathtt{0}, \mathtt{1}\}$，都有 $\tau(\delta(q_\varepsilon, x'ya)) = b$。由于 $x_i = xy$ 是一个临界字符串，我们可以选取一个数字 $c \in \{\mathtt{0}, \mathtt{1}\}$，使得 $\tau(\delta(q_\varepsilon, xyc)) = 1 - b$。因此，$\tau(\delta(q_{x'}, yc)) = b$ 且 $\tau(\delta(q_x, yc)) = 1 - b$，所以两个状态 $q = q_x$ 和 $q' = q_{x'}$ 不等价。

情况 b：假设 $y = y'$，即 $x_i = xy$ 和 $x_j = x'y$。由于 $x \neq x'$，我们有 $i \neq j$，这意味着我们有对应于两个不同临界串x_1, x_2的前缀的状态。根据式（4），$|x_1| = |x_2|$ 的奇偶性决定了在读入x_1 或 x_2 后再读入 0，意味着进入相应临界点左侧还是右侧的子区间。然而，对于两个临界点中的一个，向左意味着我们输出 0，而对于另一个临界点，向左意味着我们输出 1。因此，$\tau(\delta(q_x, y\mathtt{0}) \neq \tau(\delta(q_{x'}, y\mathtt{0}))$，所以两个状态 $q = q_x$和 $q' = q_{x'}$不等价。

情况 2： 存在 $k \geq 1$ 使得 $c + 1 = F_{k+1}$ 并且 $c + 2 = F_{k+1} + 1$。在这种情况下，根据注 2，点 $\{-(c+1)\varphi\}$ 成为阶段 k的划分端点，而点 $\{-(c+2)\varphi\}$ 成为阶段 $k+1$的划分端点。与情况 1 一样，令x_1，x_2 为两个临界串，此时 $|x_2| = |x_1| + 1$。

现在，由于 $|x_1|$、$|x_2|$ 具有不同的奇偶性，根据式（4），向 0 后附加 x_1意味着当且仅当向 0 后附加 x_2 意味着进入相应的右子区间时，才会进入相应的左子区间。因此，附加 0 在两种情况下要么都表示输出 0，要么都表示输出 1。于是，我们可以合并最后两个中间状态q_{x_1}和 q_{x_2}，因为它们是等价的。因此，状态总数仍为 $2(|(c)_F| + 1) + 1 - g(c)$，正如所需。

证明在此情况下不存在其他两个等价状态的方法，与情况 1 中的证明类似。唯一的区别在于，最小性证明中情况 b 的论证在这里不起作用。然而，情况 b 在此处永远不会发生。事实上，可以检验，如果 $c = F_{k+1} - 1$，那么临界串必定具有 $\mathtt{0}^*$ 和 $(\varepsilon|\mathtt{0})(\mathtt{10})^*$ 的形状，因此，特别地，x_1、x_2 中倒数第二个数字是不同的。

总而言之，我们已经证明，生成 $(f(i+c))_{i\geq 0}$ 的最小低位优先确定性有限自动机输出机中的状态数确实为 $2(|(c)_F| + 1) + 1 - g(c)$。由于函数 $c \mapsto |(c)_F|$ 和 $c \mapsto g(c)$

都是斐波那契正则的，因此将 c 映射到 $(f(i+c))_{i\geq 0}$ 的状态复杂度的函数也是斐波那契正则的；关于正则序列的更多细节，[18], 例如可参见此处。

例4。作为另一个例子，设 $c = 12$；我们有 $f(10) = f(11) = 0$。由于 $(-13)_F = (-1)0000$, $m_1 = 0$且 $(-14)_F = (-1)001010$, $m_2 = 7$。因此，在这种情况下，我们有两条分别由 0000 和 01010 给出的路径，并且状态 q_{0000} 和 q_{01010} 等价。由于 $f(c+m_1) = f(c+m_2) = 1$，这就得到了图5所示的最小自动机。

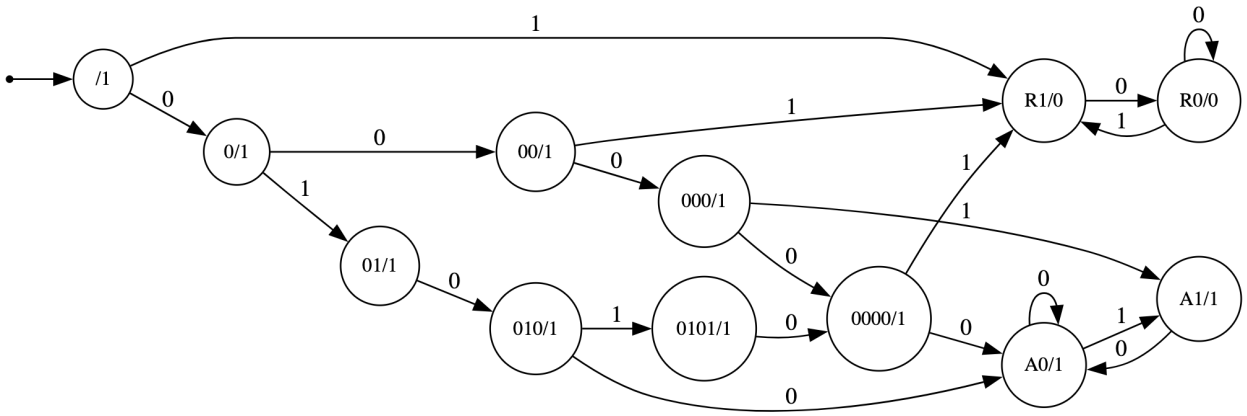

图5。 $(f(i+12))_{i\geq 0}$ 的最低有效位优先 DFAO。

B 最高有效位优先细节

为了证明 引理 5，我们需要一些辅助结果。下面的引理陈述了预期的性质：删除一位和追加一位互为逆操作。

引理 6。 对于所有满足 $n \geq 0$ 或 $n \leq -3$ 的整数 n，我们有

$$(n^{(a)})' = n, \tag{10}$$

其中 $a = 0$ 当 $(n)_F$ 以 0结尾时，而 $a \in \{0, 1\}$ 则不然。此外，对于 $n = -1$，等式对 (10) $a = 0$ 和 $a = 1$均成立；对于 $n = -2$，等式 (10) 仅对 $a = 0$成立。

证明。 如果 $n \geq 0$ 或 $n \leq -3$，那么 $n^{(a)}$ 在（2）或（3）中的表示是有效的表示（限制条件是：当 $a = 0$ $(n)_F$ 以 0 结尾时）。因此，显然，去掉最后一个数字即可逆转该操作。

现在来检查 $n = -1$ 和 $n = -2$。如果附加一个 0，显然仍会得到有效的表示，而去掉这个 0 即可逆转该操作。对于 $n = -1 = -F_2$ 和$a = 1$ 的情况，我们有

$$(-1)^{(1)} = -F_3 + F_2 = -1,$$

并且由于 $(-1)' = -1$，式（10）在这种情况下也成立。对于 $n = -2 = -F_3$ 和$a = 1$ 的情形，我们有 $(-2)^{(1)} = -F_4 + F_2 = -F_3 = -2$，因此 $((-2)^{(1)})' = -1 \neq -2$。

B.1 翻转引理

本节将证明一些关于读取更多输入位时区间如何变化的基本但略显技术性的结果。我们的证明基于 Walnut，这是一个用于证明自动序列性质的基于自动机的证明器，其中包括斐波那契词 f。关于 Walnut 的介绍，请参见 [18]。

引理 7. 设 x、y 为负整数，使得区间 $I = (x\varphi, y\varphi)$ 模 1 不包含点 $\{-\varphi\}$ 和 $\{-2\varphi\}$。设 n 为非负整数。若 $f(n) = 1$ 集合 $a = 0$，否则 $a \in \{0, 1\}$。那么我们有

$$\{n\varphi\} \in I \implies \left\{n^{(a)}\varphi\right\} \in (y^{(a)}\varphi, x^{(a)}\varphi) \bmod 1 =: I_a. \tag{11}$$

此外，若 $x \neq -1$，区间 I_a 不包含 $\{-\varphi\}$、$\{-2\varphi\}$。

证明。首先我们定义以下 Walnut 个命令

```
reg lshift {0,1} {0,1} "([0,0]|[0,1][1,1]*[1,0])*":
# accepts (x,y) iff y is the left-shift of x.

reg rshift {0,1} {0,1} "([0,0]|[1,0][1,1]*[0,1])*(()|[1,0][1,1]*)":
# accepts (x,y) if y is the right-shift of x

def phin "?msd_fib (x=0 & y=0) | Ez $lshift(x-1,z) & y=z+1":
# (x,y) is accepted iff floor(x*phi) = y.

reg isfib msd_fib "0*10*":
# accepts x iff x is a positive Fibonacci number.

def smallest_fib_geq "$isfib(y) & y>=x & Az ($isfib(z) & z>=x) => z>=y":

# accepts (x,y) iff y is the smallest Fibonacci number >= x, for x>=1.# 5 states


def fib_neg_repr "?msd_fib (x=0 & y=0 & z=0) |

(x>=1 & $smallest_fib_geq(x,y) & z+x=y)":

# accepts (x,y,z) iff -y+z is the negative Fibonacci expansion of -x,
where y is a Fibonacci number.# 9 states


def prime_inv_0 "?msd_fib Er,s,t,u $fib_neg_repr(x,r,s) &
$rshift(t,r) & $rshift(u,s) & F[t]=@0 & F[u]=@0 & t=y+u":

# accepts (x,y) if -y is the integer obtained from

appending 0 to the representation of -x# 7 states
```

```
def prime_inv_1 "?msd_fib (x=1 & y=1) | (x=2 & y=2) |
    Er,s,t,u $fib_neg_repr(x,r,s) &
$rshift(t,r) & $rshift(u,s) & F[t]=@0 & F[u]=@1 & t=y+u":
# accepts (x,y) if -y is the integer obtained from
appending 1 to the representation of -x# 9 states
```

请注意， `prime inv 1` 中的特殊情形 $(x, y) = (1, 1)$ 和 $(x, y) = (2, 2)$，体现了这样一个事实：如果我们在 $-1, -2$ 的表示后附加1，就会得到无效表示。不过，这些情形仍由该引理涵盖。

接下来，我们定义以下两个关于小数部分的函数。请注意，此处不采用循环解释。

```
def compare_frac_part "?msd_fib
    (x>y & Et,u,v $phin(x,t) & $phin(y,u) & $phin(x-y,v) & v+u<t) |
(x<y & Et,u,v $phin(x,t) & $phin(y,u) & $phin(y-x,v) & u<t+1+v)":
# accepts (x,y) iff {x*phi} < {y*phi}  for x, y >= 0# 22 states


def compare_frac_part2 "?msd_fib Et,u,v
   $phin(x,t) & $phin(y,u) & $phin(x+y,v) & v>u+t":
# accepts (x,y) iff {-x*phi} < {y*phi} for x, y >= 1
# 26 states

def lies_in "?msd_fib
 ($compare_frac_part(z,x) & $compare_frac_part2(x,y)
            & ~$compare_frac_part2(y,z)) |
 ($compare_frac_part(x,z) &
    ($compare_frac_part2(x,y) | ~$compare_frac_part2(z,y)))":
# checks if  {y*phi} is in the interval ({-x*phi}, {-z*phi})
    for the circular interpretation of intervals
# 97 states

def cw "?msd_fib
    ($compare_frac_part(x,z) & $compare_frac_part(x,y) &
        $compare_frac_part(y,z)) |
    ($compare_frac_part(z,x) & ($compare_frac_part(x,y) |
        $compare_frac_part(y,z)))":
# checks if {y*phi} is in the interval ({x*phi}, {z*phi})
    for circular interpretation of intervals; cw = clockwise
# 44 states
```

既然我们已经有了这些，就可以用一阶逻辑重述定理，并要求 `Walnut` 证明它：

```
eval check_flip_append_0 "?msd_fib An,n0,a,b,a0,b0
(a>=1 & b>=1 & F[n0]=@0 &
$rshift(n0,n) & $prime_inv_0(a,a0) & $prime_inv_0(b,b0) &
($cw(a,1,b)|a=1|b=1) & ($cw(a,2,b)|a=2|b=2)& $lies_in(a,n,b))
=> ($lies_in(b0,n0,a0))":

eval check_flip_append_0_interval "?msd_fib An,n0,a,b,a0,b0
(a>=2 & b>=1 & F[n0]=@0 &
$rshift(n0,n) & $prime_inv_0(a,a0) & $prime_inv_0(b,b0) &
($cw(a,1,b)|a=1|b=1) & ($cw(a,2,b)|a=2|b=2) &$lies_in(a,n,b))
=> (($cw(b0,1,a0)|a0=1|b0=1) & ($cw(b0,2,a0)|b0=2|a0=2) )":

eval check_flip_append_1 "?msd_fib An,n1,a,b,a1,b1
(a>=1 & b>=1 & F[n1]=@1 &
$rshift(n1,n) & $prime_inv_1(a,a1) & $prime_inv_1(b,b1) &
($cw(a,1,b)|a=1|b=1) & ($cw(a,2,b)|a=2|b=2) &$lies_in(a,n,b))
=> ($lies_in(b1,n1,a1))":

eval check_flip_append_1_interval "?msd_fib An,n1,a,b,a1,b1
(a>=2 & b>=1 & F[n1]=@1 &
$rshift(n1,n) & $prime_inv_1(a,a1) & $prime_inv_1(b,b1) &
($cw(a,1,b)|a=1|b=1) & ($cw(a,2,b)|a=2|b=2) &$lies_in(a,n,b))
=> (($cw(b1,1,a1)|a1=1|b1=1) & ($cw(b1,2,a1)|b1=2|a1=2))":
```

这些都返回 `TRUE`。

注记3. 如果 $x = -1$ 满足引理 7 中的条件，那么区间 I_a 确实包含两个点 $\{-\varphi\}$, $\{-2\varphi\}$ 之一。事实上，对于 $a = 0$，注意到 $(-1)^{(0)} = -2$，因此对每个 y， I_0 都具有 $I_0 = (y^{(0)}\varphi, -2\varphi) \bmod 1$ 的形式。一方面，根据引理 7 的第一部分，我们知道对于所有满足 $\{n\varphi\} \in I$ 的 n，都有 $\left\{n^{(0)}\varphi\right\} \in I_0$。另一方面，由于 $n^{(0)}$ 的最低有效数字显然是 0，根据引理 1，我们知道 $\left\{n^{(0)}\varphi\right\} \in (-2\varphi, -\varphi) \bmod 1$。因此，我们有

$$I_0 \cap (-2\varphi, -\varphi) \bmod 1 = (y^{(0)}\varphi, -2\varphi) \bmod 1 \cap (-2\varphi, -\varphi) \bmod 1 \neq \emptyset,$$

这意味着 $\{-\varphi\} \in I_0$。同理，可以看出 $\{-2\varphi\} \in I_1$。

引理 8. 设 x、y 是两个负整数，满足 $(x, y) \neq (-1, -2)$，且 $\{-\varphi\}, \{-2\varphi\} \notin (x\varphi, y\varphi) \bmod 1$。设 r 是一个负整数。那么

$$\{r\varphi\} \in (x\varphi, y\varphi) \bmod 1 \implies \{r'\varphi\} \in (y'\varphi, x'\varphi) \bmod 1.$$

证明。 我们再次使用 Walnut 定理证明器。

```
def negprime "?msd_fib (x=1 & y=1) |
    (x>=2 & Er,s,t,u $ingrid_exp(x,r,s) &
    $rshift(r,t) & $rshift(s,u) & y+u=t)":

eval check_flip_drop "?msd_fib Aa,b,r,t,u,v
(a>=1 & r>=1 & b>=1 & (~((a=1&b=2)|(a=2&b=1))) &
$cw(b,r,a) & ($cw(a,1,b)|a=1|b=1) & ($cw(a,2,b)|a=2|b=2) &
$negprime(a,t) & $negprime(b,u) & $negprime(r,v))=> $cw(t,v,u)":
```

并且 Walnut 返回 TRUE。

B.2 引理 5 的证明引理 5

现在我们可以证明引理 5。

证明（引理 5）。设 $r = -F_R + \sum_{i=2}^{R-3} b_i F_i$ 为 $r \le -2$的斐波那契表示，并对 $0 \le k \le R-2$定义 $r_k = -F_{R-k} + \sum_{i=2}^{R-3-k} b_{i+k} F_i$。回忆一下， $\mathcal{P}$（r）是由满足 $0 \le k \le R-2$的点 r_k 所诱导的划分，并且 $r_{R-2} = -1$ 和 $r_{R-3} = -2$。因此，特别地，没有任何区间 $I \in \mathcal{P}$（r）会在其内部包含 $\{-\varphi\}$ 或 $\{-2\varphi\}$ 。我们来证明 $\mathcal{P}$（r）与最高有效位表示相容，也就是说，对于每个 $I \in \mathcal{P}$（r），都存在I_0、 $I_1 \in \mathcal{P}$（r），使得对所有非负整数 n，我们都有

$$\{n\varphi\} \in I \implies \left\{n^{(0)}\varphi\right\} \in I_0 \text{ and } \left\{n^{(1)}\varphi\right\} \in I_1,$$

其中，如果 $f(n) = 1$，我们忽略 $n^{(1)}$ 和 I_1。注意， $\mathcal{P}(r)$ 中的所有区间都是半开区间，且所有端点均形如 $\{m\varphi\}$，其中 m 为负整数。由于对于每个非负整数 n，点 $\{n\varphi\}$ 不可能与带有负 m 的端点 $\{m\varphi\}$ 重合，为简便起见，我们将 $\mathcal{P}(r)$ 中的所有区间写成开区间。

首先考虑 $r = -2$ 的情形。于是，划分 $\mathcal{P}(r) = \mathcal{P}(2)$ 由两个区间 $(-2\varphi, -\varphi) \bmod 1$ 和 $(-\varphi, -2\varphi) \bmod 1$ 组成。令 I_0 为第一个区间，令 I_1 为第二个区间。于是，根据 引理 1，对于所有非负整数，我们有 $\left\{n^{(0)}\varphi\right\} \in I_0$，以及 $\left\{n^{(1)}\varphi\right\} \in I_1$（其中，如果 n ($n^{(1)}$ 和 I_1 为 $f(n) = 1$，则忽略它们）。

现在假设 $-r \le -3$。特别地，由于 $r_{R-3} = -3$，我们有 $\{-3\varphi\}$ 是划分中的一个端点。令 $I \in \mathcal{P}(r)$，其中

$$I = (r_k\varphi, r_\ell\varphi) \bmod 1,$$

其中 $0 \le k, \ell \le R-2$ 和 $k \ne \ell$。

我们区分以下三种情况：即 $r_k \le -3$、 $r_k = -1$ 或 $r_k = -2$。事实上，在每种情况下，所需集合 I_0, I_1 的定义都会略有不同。

情况 1： $r_k \le -3$。根据引理 7 可得

$$\{n\varphi\} \in I \implies \left\{n^{(a)}\varphi\right\} \in \left((r_\ell)^{(a)}\varphi, (r_k)^{(a)}\varphi\right) \bmod 1 =: I_a$$

当 $a \in \{0,1\}$（且当 $f(n)$ 时 $a=0$ $=1$）时。此外，区间 I_a 不包含这两个点 $\{-\varphi\}$，$\{-2\varphi\}$。我们的目标是证明，I_a 完全包含在 $\mathcal{P}(r)$ 中的某个区间内。我们根据 r_ℓ 的值区分两种子情况。

情形 1a： $r_\ell = -1$ 或 $r_\ell \le -3$。为得到矛盾，假设 I_a 不包含于来自 $\mathcal{P}(r)$ 的某个区间中。这意味着存在一个点 $\{r_j\varphi\}$，对于内部中的某个 $0 \le j \le R-2$ I_a，即，

$$\{r_j\varphi\} \in ((r_\ell)^{(a)}\varphi, (r_k)^{(a)}\varphi) \bmod 1. \tag{12}$$

由于根据假设 r_k，$r_\ell \neq -2$，现在可以将 引理 8 和 引理 6 应用于（12），从而得到

$$\left\{r'_j\varphi\right\} \in \left(((r_k)^{(a)})'\varphi, ((r_\ell)^{(a)})'\varphi\right) \bmod 1 = (r_k\varphi, r_\ell\varphi) \bmod 1 = I. \tag{13}$$

如果 $j < R-2$，根据定义有 $r'_j = r_{j+1}$。如果 $j = R-2$，则有 $r'_j =$ $(-1)'=-1=r_j$。无论哪种情况，式（13）都与以下事实矛盾：$I \in \mathcal{P}$（r）。

情形 1b：$r_\ell = -2$。在这种情况下，$I = (r_k\varphi, -2\varphi)$ 模 $1 \subset (-\varphi, -2\varphi)$ 模 1，因此根据引理 1，对于所有满足 $\{n\varphi\} \in I$ 的 n，都有 $f(n) = 1$。因此，只需考虑 $a=0$。随后证明的其余部分与前一个子情形完全相同，因为对于 $a=0$，我们仍有 $((r_k)^{(a)})' = r_k$ 以及 $((-2)^{(a)})' = -2$。

情况 2： $r_k = -1$。于是 $I = (-\varphi,\ r_\ell\varphi)$ 模 $1 \subseteq (-\varphi,\ -2\varphi)$ 模 1，因此与情形 1b 中一样，我们可以令 $a=0$。和之前一样，引理 7 表明，对于 n 满足 $\{n\varphi\} \in I$ 的情形，我们有

$$\left\{n^{(0)}\varphi\right\} \in ((r_\ell)^{(0)}\varphi, (-1)^{(0)}\varphi) \bmod 1 = ((r_\ell)^{(0)}\varphi, -2\varphi) \bmod 1.$$

鉴于引理 1，由于 $n^{(0)}$ 的最低有效位当然是 0，我们实际上必有

$$\left\{n^{(0)}\varphi\right\} \in \left((r_\ell)^{(0)}\varphi, -\varphi\right) \bmod 1 =: I_0.$$

现在，与情况 1 中一样，我们检查是否有其他端点位于 I_0中。反设如此，则有 $\{r_j\varphi\} \in ((r_\ell)^{(0)}\varphi, -\varphi)$ 模 1，其中 $0 \le j \le R-2$。注意，$(r_\ell)^{(0)} \neq -2$，因为$r_\ell \neq -1$。因此，我们可以应用引理 8，得到

$$\left\{r'_j\varphi\right\} \in (-\varphi, r_\ell\varphi) \bmod 1 = I,$$

我们便可如情形 1a 中一样完成证明。

情形 3： $r_k = -2$，因此 $I =$ $(-2\varphi,\ r_\ell\varphi)$ 模 1。注意，在此情况下 $r_\ell = -1$ 是不可能的，因为 $r \le -3$ 且 $\{-3\varphi\} \in$ $(-2\varphi,\ -\varphi)$ 模 1，所以这最后一个区间不在 $\mathcal{P}$（r）中。因此，我们可以假设 $r_\ell \le -3$。对于 $a=0$，引理 7 再次表明，对于 n 满足 $\{n\varphi\} \in I$ 时，我们有

$$((r_\ell)^{(0)}\varphi, (-2)^{(0)}\varphi) \bmod 1 = ((r_\ell)^{(0)}\varphi, -3\varphi) \bmod 1 := I_0,$$

且区间 I_0 不包含 $\{-\varphi\}$，$\{-2\varphi\}$。我们可以像情况 1a 中一样证明 $I_0 \in \mathcal{P}$（r）（因为对于 $a=0$，我们仍有 $((-2)^{(a)})' = -2$）。

最后，对于 $a=1$，引理 7 表明，对于 n 满足 $\{n\varphi\} \in I$ 的情形，我们有

$$\left\{n^{(1)}\varphi\right\} ((r_\ell)^{(1)}\varphi, -2\varphi) \bmod 1 := I_1,$$

且区间 I_1 不包含 $\{-\varphi\},\{-2\varphi\}$。我们检查得知，没有其他端点位于 I_1 中。反设不然，则对于某个 $0 \le j \le R-2$，有 $\{r_j\varphi\} \in I_1$。于是，根据引理 8（注意 $(r_\ell)^{(1)} \neq -1$）和引理 6（注意 $r_\ell \neq -2$），我们得到 $\{r'_j\varphi\} \in (-\varphi, r_\ell\varphi) \bmod 1$（因为 $(-2)' = -1$）。由于 $\{r_j\varphi\} \in I_1 \subset (-\varphi, -2\varphi) \bmod 1$，$r_j$ 的最低有效位是 1。但这样一来，第二低有效位就必须是 0。换言之，r'_j 的最低有效位是 0，因此实际上

$$\{r'_j\varphi\} \in (-2\varphi, r_\ell\varphi) \bmod 1 = I,$$

并且我们可以像情形 1a 中那样完成证明。