arXiv:2303.09338v1 [cs.DM] 16 Mar 2023

均匀随机生成与拒绝法（I）：采用二项式上界

Laurent Alonso*

2023年3月17日

摘要。我们介绍三种简单算法，用于均匀生成“斐波那契词”（即由斐波那契数计数的某些词）、大小为 n 的 Schr¨oder 树，以及大小为 n 且最终高度为 h 的 Motzkin 左因子。这些算法在单位成本 RAM 模型[1]下的平均复杂度为 $O(n)$，且仅使用小整数（小于 n^2 的整数）。

关键词：生成、均匀、拒绝

1 引言

最早用于均匀生成各种数据结构的通用方法之一是递归法[NW78][FZC94]。该方法通过计算不同大小结构的数量，并利用这些数量递归地生成一个元素。其缺点在于系数往往非常大： $\approx C^n$对于某些常数而言 C（因此需要链接一个能够计算超大数字的库），而且该方法非常慢。Denise 等人[DZ99] 指出，通过执行惰性浮点运算可以改进该方法，从而得到更高效的算法[2]，但实现起来仍然非常困难。

对于某些问题：“斐波那契词”、施罗德树[PPPR01], 莫茨金词[Alo94],[BBJ17], [GBN10] . . . 拒绝法能给出高效算法；一般而言，这些方法基于巧妙的一一对应关系，而且易于实现[3]。

*INRIA Nancy-Grand Est，615, rue du Jardin Botanique, 54600 Vandoeuvre-l`es-Nancy, France。电子邮件： Laurent.Alonso@inria.fr[1]由于本文只介绍使用小整数的算法，我们将采用单位成本 RAM 模型来描述复杂度；将该复杂度乘以因子 $log\, n$，即可得到对数成本 RAM 模型下的复杂度。[2]本文介绍了适用于某些上下文无关文法的算法，其平均复杂度为 $O(n^{1+\epsilon})$。[3]事实上，生成随机 Motzkin 词可以得到一种简单的算法；但对于其他问题，这些方法以较低的概率需要计算精度极高的浮点数，.....

本文尝试扩展 [Alo94]提出的方法：该方法首先使用拒绝法，以适当的概率选取水平步数 m ，然后生成一个含有 m 个水平步的随机 Motzkin 词（这是一个简单问题）。在 [GBN10], 中，Gouyou-Beauchamps等人证明，该方法可以扩展为生成带颜色的一元二叉树，其平均复杂度为线性；也可以扩展为生成部分单射，其平均复杂度为 $O(n^{5/4}$。他们的方法包括以概率 m 选取一个数 $\frac{F_n(m)}{\sum_i F_n(i)}$ 通过找到一个分布 $\overline{B}_n(m)$，使得：

- 很容易以概率 $\frac{\overline{B}_n(m)}{\sum_i \overline{B}_n(i)}$生成一个数 m，

- 对于所有 m，我们都有 $F_n(m) \leq \overline{B}_n(m)$，并且有一种高效算法，能够以概率 $\frac{F_n(m)}{\overline{B}_n(m)}$接受一个 m 的值。

这种方法相当通用，所提出的算法易于实现且巧妙；然而，要找到满足这些性质的分布 $\overline{B}_n(m)$ 并不简单，即使他们建议对 $\overline{B}_n(m)$ 使用乘以常数的二项分布。

本文将说明，选择一种与最终分布接近的二项分布（仅作少许调整），至少对于某些基本结构而言，可以简单地得到一种高效算法。更具体地说，我们将证明，可以高效地随机、均匀地生成大小为 n 的“斐波那契词”（最简单的问题）、大小为 n 的 Schr¨oder 树（使用几乎相同的算法），以及大小为 n 、最终高度为 h 的 Motzkin 左因子。与 [GBN10], 一样，这些算法只使用小整数（即小于 n^2 的整数），平均复杂度为 $O(n)$，而且实现起来非常简单。另一个有趣的性质是，这些算法中几乎所有算法平均只执行 $O(\sqrt{n})$ 次涉及大于 n 的数字的操作；只有在 $h \geq n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$ 不同的情况下，生成大小为 n 、最终高度为 h 的 Motzkin 左因子例外，因为它平均执行 $O(n)$ 次操作，且这些操作涉及的数字都小于 $3n + 7$。

在未来的一篇论文中[Alo23], ，我们将提出另一种方法，通过不同的方式生成 m 的初始值：只需按照基本均匀分布（即以 $\frac{1}{n}$ 中的概率）生成一个整数 m 。我们将证明，至少对于简单问题而言，这样做可以得到非常高效的算法。虽然我们可能会选择在简单问题中使用第二种方法，但我们认为这两篇论文很有意义，因为：

- 它们共同表明， m 的初始值可以通过不同类型的方法来选择，

- 这些解具有不同的性质：在本文中，我们提出猜测 m 的第一个取值，使其分布接近最终分布，这样拒绝次数较少，但主循环中的操作较多。新论文将提出相反的方法，因此拒绝率会非常高，而循环所耗费的时间则很少。我们怀疑，在某些情况下，

例如，如果我们想要找到一种算法，来确定 $3-ary$ 树中二叉节点和三叉节点的数量，那么高拒绝率是不可取的，

- 实际上，可以结合这两篇文章的思路： [Alo23] ，按照本文提出的方法生成一个初始 m 值，其分布接近最终分布。

在第2节中，我们介绍了记号、一个抽象算法，以及用于生成初始 m 值的各种生成函数。接着，在第3–5节中，我们将介绍如何具体实现这一抽象算法，以生成“斐波那契词”、大小为 n 的 Schr¨oder 树，以及大小为 n 且最终高度为 h 的 Motzkin 左因子，并证明这些算法以适当的概率选取一个值m ，且平均复杂度为 $O(n)$。

最后，稍作讨论之后，我们在附录中给出了许多引理的证明。这些证明只用到初等数学（而且确实有必要给出）；由于写出详细证明可能会太冗长乏味，我们尽量只提供足够的细节，使读者能够毫不费力地核查.....

2 记法与抽象算法

2.1 记法

以下将使用：

- $[P]$ 肯尼斯·艾弗森的约定[GKP88]（第 24 页）： $[P]=1$ 若 P 为真，且 $[P]=0$ 若 P 为假，
- $random(k)$ 以概率 $\frac{1}{k}$[4]返回一个介于 0 和 $k-1$ 之间的整数的函数，以及：
- $F(m)$ 将统计大小为 m的元素的最终数量，
- $M=$使$_m$ $F(m)$取最大值的 m 中的最小值，
- $\underline{B}(m)$是对应于选择 m的初始生成器实例数量，
- B 对 $\underline{B}$ 稍作修改的函数，例如施加 $B(M-1)=B(M)$，.....，
- $\overline{B}(m)=B(m)\frac{F}{F(M)}B(M)$： 满足$\overline{B}(M)=F(M)$的归一化函数，
- $\underline{G}(m)=\frac{F(m)\underline{B}(m)}{F(m)B(m)}$， $G(\underline{m})=\frac{F(m)B(m)}{F(m)B(m)}$，$\underline{\overline{G}}(m)=\frac{F(m)B(m)}{F(m)B(m)}$。

[4]为简化某些记号，当 p、q、r、s 是四个整数时，我们有时记作$random(p/q)\geq r/s$，而不写作 $random(p\,s)\geq q\,r$，.....

最后，我们将指出：

- $\mathcal{R}F(m) = F(m+1)/F(m)$ 在其有定义时，表示 F 的两个连续项之间的比值，

- 若 $\mathcal{R}F(m) = \mathrm{n}umerator(m)/\mathrm{d}enominator(m)$ 是 m 中两个多项式的比值，
  $f(x) = \mathrm{n}umerator(x) - \mathrm{d}enominator(x)$，

我们也将以同样的方式定义 $\mathcal{R}\underline{B}(m), \ldots, \overline{g}(x)$。

2.2 一个抽象算法

采用这些记号，我们将使用以下过程以概率 $\frac{F(m)}{\sum_m F(m)}$ 抽取一个数 m：

算法 1 以概率选择 m $\frac{F(m)}{\sum_m F(m)}$：

1: 函数 ACCEPT_M(M,m) ▷ 以概率 $\overline{G}(m) =$ 接受 m $\frac{F(m)}{\overline{B}(m)}$
2: 如果 $m < M$ 则
3: 对所有 $i \in [m, M-1]$ 执行 ▷ 以概率 $\frac{\mathcal{R}\overline{B}(i)}{\mathcal{R}F(i)}$
4: 如果 $random(\mathcal{R}F(i)) \geq \mathcal{R}\overline{B}(i)$，则返回 FALSE。结束 if
5: 结束 for
6: **else**
7: 对所有 $i \in [M, m-1]$ 执行 ▷ 以概率接受 $\frac{\mathcal{R}F(i)}{\mathcal{R}\overline{B}(i)}$
8: 如果 $random(\mathcal{R}\overline{B}(i)) \geq \mathcal{R}F(i)$，则返回 FALSE 结束判断
9: 结束循环
10: 结束 if
11: 返回 TRUE
12: 结束函数

13: $M \leftarrow m$ 的最大化 $F(m)$的第一个取值
14: 当 true 时执行
15: 以概率选择 m $\frac{B(m)}{\sum_m B(m)} = \frac{\overline{B}(m)}{\sum_m \overline{B}(m)}$
16: 如果 ACCEPT_M(M, m)，则返回 m 结束如果
17: 结束循环

第 1 至 12 行对应主函数；当 $m < M$ 时，该函数以概率 $\frac{F(m)\overline{B}(m)}{\overline{B}}$ 接受值 m。它基于以下关系：

$$\begin{aligned}\frac{F(m)}{\overline{B}(m)} &= \frac{F(m)\overline{B}(m+1)}{\overline{B}(m)F(m+1)}\frac{F(m+1)\overline{B}(m+2)}{\overline{B}(m+1)F(m+2)}\cdots\frac{F(M-1)\overline{B}(M)}{\overline{B}(M-1)F(M)}\frac{F(M)}{\overline{B}(M)} \\ &= \frac{\mathcal{R}\overline{B}(m)}{\mathcal{R}F(m)}\frac{\mathcal{R}\overline{B}(m+1)}{\mathcal{R}F(m+1)}\cdots\frac{\mathcal{R}\overline{B}(M-1)}{\mathcal{R}F(M-1)}\end{aligned}$$

同理，当 $m > M$ 时：

$$\frac{F(m)}{\overline{B}(m)} = \frac{\mathcal{R}F(M)}{\mathcal{R}\overline{B}(M)} \frac{\mathcal{R}F(M+1)}{\mathcal{R}\overline{B}(M+1)} \cdots \frac{\mathcal{R}F(m-1)}{\mathcal{R}\overline{B}(m-1)},$$

最后到 $\frac{F(M)}{\overline{B}(M)} = 1$。因此，它假设当 $i < M$ 时， $\mathcal{R}F(i) \geq \mathcal{R}\overline{B}(i)$；当 $i \geq M$ 时， $\mathcal{R}F(i) \leq \mathcal{R}\overline{B}(i)$。

第 13 至 17 行是程序的主要部分：首先，我们计算 M 的值（第 13 行），然后通过反复抽取一个 m 的值来选择一个 m 的值，其概率为 $\frac{B(m)}{\sum_m B(m)} = \frac{\overline{B}(m)}{\sum_m \overline{B}(m)}$，并以概率 $\frac{F(m)\overline{B}(m)}{\sum_m F(m)\sum_m \overline{B}(m)}$ 接受一个值 m；这个概率必须足够大，才能得到一个高效算法。此外，具体来说，我们还有：

引理 1 记 C_1 为以概率 $\frac{\overline{B}(m)}{\sum_m \overline{B}(m)} = \frac{B(m)}{\sum_m B(m)}$， $C_2 = \frac{\sum_m |M-m|\overline{B}(m)}{\sum_m \overline{B}(m)} = \frac{\sum_m |M-m|B(m)}{\sum_m B(m)}$ 为 B 的平均绝对差。假设第 4 行和第 8 行的测试可以在复杂度 $O(1)$ 下完成，则该算法的平均复杂度为 $O\left((1 + C_1 + C_2)\frac{\sum_m F(m)}{\sum_m \overline{B}(m)}\right)$。此外，如果算法得到的值为 m，平均复杂度[5]为 $O\left(|M-m| + (1 + C_1 + C_2)\frac{\sum_m F(m)}{\sum_m \overline{B}(m)} \frac{\sum_m B(m)}{B(M)}\right)$.

在特定情况下，当 $C_1 = O(n)$、 $C_2 = O(\sqrt{n})$、$\sum_m F(m)\, F(M) = \Omega(\sqrt{n})$，$\sum_m B(m)\, B(M) = O(\sqrt{n})$ 时，我们将得到平均复杂度为 $O(n)$；当算法以值 m 结束时，平均复杂度也为 $O(n)$。

证明。我们首先研究循环内部执行的操作次数。根据定义，在第 15 行需要执行 C_1 次操作；而在第 16 行， ACCEPT_M 以数字 m（以概率 $\frac{B(m)}{\sum_m B(m)}$ 选取）调用： $O(1) + C_2$ 次操作。这对应于主循环单次执行的平均复杂度 $O(1)+C_1+C_2$。

现在我们知道，它接受值 m 的概率为 $C_3 = \frac{\sum_m F(m)}{\sum_m B(m)}$。因此，循环的平均次数为

$$1+(1-C_3)+(1-C_3)^2+\ldots = \frac{1}{1-(1-C_3)} = \frac{\sum_m \overline{B}(m)}{\sum_m F(m)} = \frac{F(M)}{\sum_m F(m)} \frac{\sum_m B(m)}{B(M)}.$$

最后需要指出的是，当算法返回值 m 时，我们在循环的最后一次执行中使用了 $O(C_1 + 1 + |m - M|)$ 次操作。 □

[5]这一复杂度概念很重要，因为对于某些算法，其最大值可能与平均复杂度大相径庭.....

2.3 一些生成函数

我们介绍本文将用于生成初始值 m 的三个生成函数。

‘基本生成器’这是最简单的生成器：要以概率 $\frac{1}{k+1}$ 生成一个介于 $[0,k]$ 之间的整数，我们使用 $random(k+1)$。因此，当 $0 \le m \le k$ 时，我们有 $\underline{B}^k(m) = B^k(m) = 1$（当 $0 \le m < k$ 时，则有 $\mathcal{R}\underline{B}^k(m) = \mathcal{R}B^k(m) = 1$）。它的复杂度非常低：$O(1)$。当 h 接近 n 时，我们将用它来生成最终高度为 h 的 Motzkin 左因子。

‘二项式生成器’我们选择 $\underline{B}_M(m) = \binom{2M}{m}$，当 $m \neq M$ 时选择 $B_M(m) = \underline{B}_M(m)$，而 $B_M(M) = B_M(M+1)$。这些选择为我们提供了一种高效算法，用于抽取 m 的第一个值：随机抽取 $2M$ 个 0 或 1，令 m 统计 0 的数量；最后，如果我们找到 $m = M$，则以概率 $\frac{M}{M+1}$ 接受这一选择。我们还得到：

$$\begin{aligned}
\mathcal{R}\underline{B}_M(m) &= \frac{2M-m}{m+1} \\
\mathcal{R}B_M(m) &= \mathcal{R}\underline{B}_M(m) \text{ if } m < M-1 \text{ or } m > M \\
\mathcal{R}B_M(M-1) &= \mathcal{R}B_M(M) = 1
\end{aligned}$$

该生成器以接近二项分布的概率选取一个介于 0 和 $2M$ 之间的整数 m。其平均复杂度为 $O(M)$。我们需要以下引理来关联 $B_M(M)$ 和 $\sum_m B_M(m)$。

引理 2 当 $M \ge 1$ 时，我们有：$\sum |M-m| B_M(m) \sum B_M(m) = \Theta(\sqrt{M})$，且 $\sum B_M(m) B_M(M) = \Theta(\sqrt{M})$。

证明。$\underline{B}_M$ 是经典二项分布，其 $p = q = \frac{1}{2}$ 乘以 2^{2M}，因此我们有：$\sum |M-m| \underline{B}_M(m) \sum \underline{B}_M(m) = \Theta(\sqrt{M})$，再利用斯特林公式 $\sum \underline{B}_M(m) \underline{B}_M(M) = \Theta(\sqrt{M})$。

注意到当 $m \neq M$ 时，$\underline{B}_M(m) = B_M(m)$，且 $B_M(M) = \frac{m}{m+1}\underline{B}_M(M)$（因此 $B_M(M) = \Theta(m)$ $B_M(M)$），且 $\sum B_M(\Theta(m))$ $\sum \underline{B}_M(\ )$），即可得出结论。

该算法可以编写为：

函数 BIN(M) ▷ choose m with probability $\frac{B_M(m)}{\sum_m B_M(m)}$

当 true 时执行 $m \leftarrow 0$ 对于所有 $i \in [1, 2M]$，

执行：如果 $random(k+1) = 0$，则

$m \leftarrow m+1$

结束 for；如果 $m = M$ 且 $random(M+1) = 0$，则继续；结束 if；返回 m；结束 while

e.....函数

在可能的情况下，我们会使用“二项式生成器”，但请注意，它的局限在于只能抽取 0 到 $2M$ 之间的数字（有时这还不够）。

“扩展二项式生成器”前一个生成器对应于等概率的二项分布：$p = q = \frac{1}{2}$；这里我们将使用一种等价的通用分布，其参数为 $p = \frac{1}{k+1}$ 和 $q = 1 - p$；定义 $\underline{B}_M(m)$ 如下：

$$\underline{B}_M^{k,\alpha}(m) = k^{(k+1)M+\alpha-m} \begin{pmatrix} (k+1)M+\alpha \\ m \end{pmatrix},$$

我们得到 $\mathcal{R}\underline{B}^{k,\ \alpha}_{\ M}(m) = \frac{M+\alpha-m}{(k+1)k^{\ m+1}}()$。如果 $0 \le \alpha < k$，该函数会在 $m = M$ 时达到最大值，它可以抽取 0 到 $(k+1)M+\alpha$ 之间的数字，而且实现起来很简单：只需抽取 $(k+1)M+\alpha$ 个介于 0 和 k 之间的随机整数，并统计 0 的个数。

现在我们可以将 $B(m)$ 定义为：

- $B_M^{k,\alpha}(m) = \underline{B}_M^{k,\alpha}(m)$ when $m \neq M$,
- $B_M^{k,\alpha}(M) = max(\underline{B}_M^{k,\alpha}(M-1), \underline{B}_M^{k,\alpha}(M+1))$.

由此可得：

- 当 $m < M-1$ 或 $m \ge M+1$时，$\mathcal{R}B^{k,\ \alpha}_{\ M}(m) = \mathcal{R}\underline{B}^{k,\ \alpha}_{\ M}(m)$,
- 当 $B^{k,\ \alpha}_{\ M}(M) = B^{k,\ \alpha}_{\ M}(M+1)$

$$\mathcal{R}B_M^{k,\alpha}(M-1) = \frac{(kM+\alpha+1)(kM+\alpha)}{k^2M(M+1)}, \mathcal{R}B_M^{k,\alpha}(M) = 1,$$

- 且当 $B^{k,\ \alpha}_{\ M}(M) = B^{k,\ \alpha}_{\ M}(M-1)$

$$\mathcal{R}B_M^{k,\alpha}(M-1) = 1, \mathcal{R}B_M^{k,\alpha}(M) = \frac{(kM+\alpha+1)(kM+\alpha)}{k^2M(M+1)}.$$

我们将在附录 A.1 中证明以下引理：

引理 3 当 $0 \le \alpha \le k-1$, $k \ge 1$ 和 $M \ge 1$ 时，我们有：

- $\underline{B}_M^{k,\alpha}(m) \ge \underline{B}_M^{k,\alpha}(m+1)$ *iff* $m \ge M$,
- $\frac{\sum_m B_M^{k,\alpha}(m)}{B_M^{k,\alpha}(M)} = \Theta(\sqrt{M})$,

$\sum_{m:|m-M|B^{k},_{M}{}^{\alpha}(m)} \sum_{m} B^{k},_{M}{}^{\alpha}(m) = O(\sqrt{M})$。当 $0 \le \alpha \le k-1$ 时，该算法可编写为：函数 EXTENDED_BIN$(M, k, \alpha, which)$ 要求： $M > 0$, k, $0 \le \alpha \le k-1$当真时执行$m \leftarrow 0$对于所有 $i \in [1,(k+1)M+\alpha]$，执行 如果 $random(k+1) = 0$ 则 $m \leftarrow m+1$ 结束如果 结束循环 如果 $m \neq M$ 则返回 m 结束如果 如果 $which =$ 'M+1' AND $random(k(M+1)) \ge k\,M+\alpha$ 则继续 ▷ 拒绝 m否则如果 $which =$ 'M-1' AND $random(k\,M+\alpha+1) \ge k\,M$ 则继续 ▷ 拒绝 m结束如果 返回 m结束循环 结束函数

它以概率 $B^{k,}{}_{M}{}^{\alpha}(m) \sum_{i=0}^{(k+1)M+\alpha} B^{k,}{}_{M}{}^{\alpha}(i)$ 生成整数 m，其平均复杂度为 $O(k(M+1)+\alpha)$。

3 斐波那契词的生成

斐波那契数的历史非常悠久：它们对应于在线整数数列百科全书中的数列 A000045[OEI22]，并由方程 $F_n = F_{n-1} + F_{n-2}$ 定义，其中 $F_0 = F_1 = 1$；它们与黄金数 $\varphi = \frac{1+\sqrt{5}}{2}$ 相关，满足众所周知的关系式 $F_n = \frac{\varphi^{n+1} - (1-\varphi)^{-(n+1)}}{\sqrt{5}} = \left\lceil \frac{\varphi^{n+1}}{\sqrt{5}} \right\rfloor$）（使用最近舍入函数）。

首先请注意，我们可以通过递归地选择字母 a（概率为 $\frac{F_{n-1}}{F_n} \approx \varphi$）和字母 b（概率为 $\frac{F_{n-2}}{F_n} \approx \varphi^2$）来得到一个简单的算法，但这个算法效率不高。因此，我们更倾向于使用拒绝算法：只需通过选择 a（概率为 α）和 b（概率为 α^2）来构造一个字母序列，其中 $\alpha + \alpha^2 = 1$ $(\alpha = \frac{-1+\sqrt{5}}{2})$；当得到一个 $\mathcal{F}_n$ 的词或一个 $\mathcal{F}_{n+1}$ 的词时（即失败），就停止。若出现后一种情况，我们便从头开始新一轮生成。可以验证，在一次尝试中，我们以概率 α^n 生成每个 $\mathcal{F}_n$ 的词；以概率 α^{n+1} 生成每个 $\mathcal{F}_{n-1}$ 后接一个字母 b 的词；因此，失败的概率为 $\frac{\alpha^{n-1}F_{n-1}\alpha^2}{\alpha^n F_n + \alpha^{n+1}F_{n-1}} \approx \frac{\alpha}{\varphi+\alpha} = 1 - \frac{\sqrt{5}+1}{2\sqrt{5}}$。这证明平均尝试次数为 $O\left(\frac{2\sqrt{5}}{\sqrt{5}+1}\right) = O(1)$，平均复杂度为 $O(n)$。

然而，该算法的主要步骤是在区间 $[0,1]$ 中生成一个实数 x，并将其与 α 进行比较。因此，即使我们开始生成

x 的最高有效位，也需要检查位数是否足够，以确定 $x < \alpha$ 或 $x \geq \alpha_\circ$ 如果不够，我们就需要继续生成后续位.....

这里我们给出一种算法，用于生成包含字母 a 和 b 的随机词，使其满足 $n = |a|+2|b|$，但只使用小整数（小于 n^2）。我们有 $F_n(m) = \begin{pmatrix} n-m \\ m \end{pmatrix}$，其中 m 统计字母 b的数量，因此可得： $\mathcal{R}F(m) = (n-2m)(n-1-2m)\,(m+1)(n-m)$，$f(x) = (n-2x)(n-\overline{1-2x})-(x+1)(n-x)$，以及以下引理，它归纳了$F_n(m)$的一些必要性质，证明见附录 A.2：

引理 4 当 $n > 0$时，

- 存在唯一的值 $\tilde{m}$ 属于 $[0 \ldots \lfloor n/2 \rfloor]$，使得 $f(x) > 0$，且$x \in [0, \lfloor n/2 \rfloor]$ 当且仅当 $x < \tilde{m}$,
- $\tilde{m} = \frac{5n-3-\sqrt{5n^2+10n+9}}{10}$，$M = \lceil \tilde{m} \rceil \approx \frac{5-\sqrt{5}}{10} n$,
- $\frac{F_n}{F_n}(M) = \Omega(\sqrt{n})$。

$\tilde{m}$ 的存在为我们提供了一种计算 M的线性方法（算法 1，第 13 行）：我们可以让 m 从 0 递增到 $\lfloor n/2 \rfloor$，并在 $\mathcal{R}F_n(m) \leq 1$时立即停止（即 $f(m) \leq 0$）；或者，我们也可以直接计算 $\lceil \tilde{m} \rceil$。

我们选择“二项式生成器”来获得 m的第一个值（算法 1，第 15 行）。因此，我们得到：

$$\begin{aligned}
\mathcal{R}\overline{B}_M(m) &= \mathcal{R}B_M(m) \\
\mathcal{R}\underline{G}_M(m) &= \frac{(n-2m)(n-1-2n)}{(2M-m)(n-m)} \\
\mathcal{R}\overline{G}_M(m) &= \frac{(n-2m)(n-1-2n)}{(2M-m+[m=M]-[m=M-1])(n-m)}.
\end{aligned}$$

现在需要以下引理（证明见附录 A.2）：

引理 5 我们有：

- $\mathcal{R}\overline{B}_M(m) < \mathcal{R}F_n(m)$ *if and only if* $m < M$,

为了证明我们以 $\mathcal{R}\overline{G}_M(m) = \mathcal{R}F_n(m)\ \mathcal{R}\overline{B}_M(m)$ 的概率正确接受值 m，并补全算~~法 1 的~~第 4 行和第 8 行。

现在，根据引理 1，$C_1 = O(n)$、$C2 = O(\sqrt{n})$、.....，我们得到：

命题 1 该算法以 $F_{n(m)}\ F_n$ 的概率选取 m，平均复杂度为 ~~$O(n)$，~~ 且仅使用 $2log_2(n)$ 位的数字。

备注：我们使用位数为 $2log_2(n)$ 的数字：

计算 M 的最大值 F_n^m 需要• n 次。但我们也可以先计算一个粗略近似值 $m' = \left\lceil \frac{5-\sqrt{5}}{10} n \right\rceil$，然后在 m' 附近搜索，使用 $O(1)$ 次这样的大数运算来找到值 M，

• $|m - M|$ 次检查值 m 是否被接受，其概率为 $F_n^m / \overline{B}_M(m)$。因此，我们可以预期 $|m - M|$ 为 $O(\sqrt{M}) = O(\sqrt{n})$。

这表明，使用经过优化、可处理 $2\log_2(n)$ 位数字的库与否，对该算法的平均复杂度影响不大。

最终的算法代码可以写成算法 2 中的形式。

算法 2 使用 $|a| + 2|b| = n$ 绘制一个随机斐波那契词:

```
函数 ACCEPT_M(n,M,m)          ▷ 以概率 G(m) 接受 m = F_n(m) / B_M(m)
    若 m < M 则
        对所有 i ∈ [m, M − 1] 执行
            v ← random((n − 2i)(n − 2i − 1))
            若 v ≥ (n − i)(2M − i − [i = M − 1]) 则返回 FALSE 结束条件判断
        结束循环
    else
        对所有 i ∈ [M, m − 1] 执行
            v ← random((n − i)(2M − i + [i = M]))
            如果 v ≥ (n − 2i)(n − 2i − 1)，则返回 FALSE，结束 if
        结束 for
    结束 if
    返回 TRUE
结束函数

要求: n > 0
M ← m 中使 F_n(m)达到最大值的第一个值
当真时执行                                                    ▷ 选择 m
    m ← BIN(M)
    如果 ACCEPT_M(n, M, m) 则 break 结束 if
结束 while
                ▷ 绘制一个由 n − 2m 个字母组成的随机词 a 和 m 个字母 b。
b ← m, a ← n − 2m
while a + b > 0 执行
    如果 random(a + b) < a 则
        添加一个字母 a, a ← a − 1,
    else
        添加一个字母 b, b ← b − 1。
    结束 if
结束 while
```

---

4 施罗德数

Schr¨oder 数字由递推关系 $S_0 = 1$, $S_1 = 2$, $S_n = 3S_{n-1} + \sum_{k=1}^{n-2} S_k S_{n-k-1}$定义。 $S_n \approx \frac{\sqrt{2}+1}{2^{3/4}\sqrt{\pi}} (3+2\sqrt{2})^n \sqrt{n}$ （参见~~[Au19]~~）也是从 (0, 0) 到 (2n,0) 且不低于 x轴的路径 x(1, 1), $\bar{x}$(1, −1), z(0, 2) 的数量。

Penaud等人[PPPR01] 提出使用拒绝法。与斐波那契数的情形相比，这种方法更复杂，因为它基于大小为 $2n$ 的施罗德路径与大小为 $2n+1$的施罗德路径的一个前缀子集之间的对应关系。大小为 $2n+1$ 的施罗德路径前缀是逐个字母生成的：首先选择 x ，然后以概率$\sqrt{2}-1$选择 x ，以概率 $\sqrt{2}-1$选择 $\bar{x}$，以概率$(\sqrt{2}-1)^2 = 3-2\sqrt{2}$选择 z，.....如果路径经过y轴下方，就拒绝该路径；随后利用匹配关系拒绝该路径，或将其转换为施罗德路径。总之，这种方法简单，但需要花费一些时间才能高精度地计算浮点数。

我们有：

$$F_n = S_n = \sum_{m=0}^{n} \frac{1}{n+m+1} \left(\begin{array}{c} n+m+1 \\ m, m+1, n-m \end{array} \right) = \sum_{m=0}^{n} F_n(m)$$

这个公式对应于这样一种方法：对某个 m，生成一个包含 $m+1$ 个x、 m 个 $\bar{x}$、 $n-m$ 个 z的随机词，然后使用“循环引理”[DZ90]，它给出了这类词与大小为$2n$ 且包含 m 个 x的 Schr¨oder 路径之间的 $n+m+1$-1 对应关系。

现在，我们可以像斐波那契算法那样使用“二项式生成器”，以概率 $\frac{F_{n(m)}}{F_n}$ 生成一个值 m 。当以适当的~~概率选~~取 m 时，最后生成一个包含 $m+1$ 个 x、 m个 $\bar{x}$、 $n-m$ 个 z 的随机词，并使用“循环引理”[DZ90] 得到最终的 Schr¨oder 路径。

因此，我们有：

$$\begin{aligned} \mathcal{R}F_n(m) &= \frac{(n+m+1)(n-m)}{(m+1)(m+2)} \\ \mathcal{R}\underline{G}_M(m) &= \frac{(n+m+1)(n-m)}{(m+2)(2M-m)} \\ \mathcal{R}\overline{G}_M(m) &= \frac{(n+m+1)(n-m)}{(m+2)(2M-m+[m=M]-[m=M-1])} \end{aligned}$$

最后，我们还需要一个在附录A.3中证明的引理：

引理 6 当 $n > 1$ 时，

- 存在唯一的值 $\tilde{m}$，属于 $[0 \ldots n]$，使得 $f(x) > 0$，且当且仅当 $x < \tilde{m}$时，$x \in [0, n]$
- $\tilde{m} = -1 + \frac{\sqrt{2n^2+2n}}{2}$, $M = \lceil \tilde{m} \rceil \approx \frac{n}{\sqrt{2}}$

- $\frac{F_n}{F_n}(M) = \Omega(\sqrt{n})$,
- $\mathcal{R}\overline{B}_M(m) < \mathcal{R}F_n(m)$ 当且仅当 $m < M$,

这意味着，修改后的算法以均匀概率生成每个单词，平均复杂度为 $O(n)$，且只使用具有 $2log_2(n)$ 位的数字[6]。

5 个大小为 n 、最终高度为 h' 的 Motzkin 左因子

已有许多高效算法可用于高效生成固定大小的莫茨金词，其中最简单的算法基于某种拒绝法：Alonso[Alo94], Bacher 等人[BBJ17], Brlek 等人 [BPR06]。类似地，Barcucci 等人[BPS94] 提出了一种基于简单拒绝法的高效算法，用于生成大小为 n 的 Motzkin 左因子。

所有这些算法都会抽取一些 n 小的随机数字（通常是介于 1 和 3 之间的整数），然后通过使用 $\log n$ 位数字进行$O(n)$检验，接受或拒绝这些序列。要将这些方法扩展到生成给定大小为 n 、最终高度为 h' 的随机 Motzkin 左因子似乎很困难，除非：

- Brlek 等人[BPR06] 的方法可以修改为将总体高度大于或等于 h 的 Motzkin 左因子转换为最终高度为 h' 的 Motzkin 左因子。这样就得到了一种有效算法，但只有在 h' 足够小时，其平均复杂度才会保持为 $O(n)$： $h' = O(\sqrt{n})$,

- Alonso[Alo94] 因为本文是对该算法的扩展。

为简化记号，记作 $h = h' + 1$。当 h 足够小时：$h \leq \frac{3n-25}{7}$，我们可以使用“二项式生成器”，得到一个平均复杂度为 $O(n)$ 的有效算法，但当 h 较大时，该算法会失效。因此，我们将选择“扩展二项式生成器”，以获得一个在 $h < n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$ 时仍可运行的算法[7]。最后，为了完整起见，我们将证明，当 $h \geq n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$ 时，只需使用“基本生成器”即可。

事实上，利用“循环引理”[DZ90], ，我们可以得到最终高度为 $h' = h - 1$ 且满足 $0 < h \leq n + 1$ 的 Motzkin 左因子的数量：

$$F_n^h = \frac{h}{n+1} \sum_{m=0}^{\lfloor \frac{n+1-h}{2} \rfloor} \binom{n+1}{m, m+h} = \sum_{m=0}^{\lfloor \frac{n+1-h}{2} \rfloor} F_n^h(m)$$

[6]使用 Maple[Map], ，我们还可以计算主循环中的接受率极限，得到 $\lim_{n=\infty} C_n = 2^{-3/4} = 0.5946\ldots$[7]当然，如果我们只想生成一个莫茨金词（$h = 1$），该算法的效率低于其他算法，因为有时需要使用 $2 \log n$ 位整数（而不是 $\log n$ 位）

其中， $F_n^h(m) = \frac{h}{n+1}\binom{n+1}{m}, m+h$ 表示最终高度为 $h-1$、包含 n 个字母和 m 个字母 $\bar{x}$（即步骤 (1, −1)）的 Motzkin 左因子的数量。因此，在选择概率适当的 m 后，我们可以通过混合 $m+h$ 个字母x（即步骤 (1, 1)）、 m 个字母 $\bar{x}$ 和 $n+1-h-2m$ 个字母 z（即步骤 (1, 0)），并利用“Cycle 引理”得到最终的 Motzkin 左因子。

由此可得：

$$\mathcal{R}F_n^h(m) = \frac{(n+1-h-2m)(n-h-2m)}{(m+1)(m+1+h)},$$

现在，我们首先需要确立关于 F_n^h 的一些结果（该引理的证明见附录 A.4）：

引理 7 当 $n > 1$ 时，

- *there exists a unique value* $\tilde{m}$ *in* $[0 \dots \lfloor \frac{n+1-h}{2} \rfloor]$ *such that* $f(x) > 0$ *with* $x \in [0, \lfloor \frac{n+1-h}{2} \rfloor]$ *if and only if* $x < \tilde{m}$,
- $\tilde{m} = \frac{2(n+1)}{3} - \frac{h}{2} - \frac{\sqrt{4n^2+20n+28-3h^2}}{6} \leq \frac{n-h}{3}$, $M = \lceil \tilde{m} \rceil \approx \tilde{m}$,
- *when* $h < n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$, $\tilde{m} > 1$, $M \geq 2$,

当 $M \geq 2$ 时， $\frac{F_n^h(M)}{F_n^h} = \Omega(\sqrt{M})$。

5.1 当 $h < n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$ 时生成 Motzkin 左因子

现在，我们可以确定 k 和 α 的取值，以便实例化抽象算法（并证明这些取值能构成一个具有线性复杂度的有效算法）。我们选择：

- k 作为使 $(k+1)M \geq (k+1)\tilde{m} \geq \frac{n+1-h}{2}$ 的 $k = \lceil \frac{n+1-h}{2\tilde{m}} \rceil - 1$, 成立的最小整数,
- α 作为 0 和 $k-1$ 之间使 $\underline{g}(x) = 0$ 的根接近 M： $\alpha = max(k-1, \lfloor k - k(M - \tilde{m}) \rfloor)$ 的最小整数。

我们有：

$$\begin{aligned}
\mathcal{R}\underline{G}_M^{k,\alpha}(m) &= \frac{k(n+1-h-2m)(n-h-2m)}{(m+1+h)((k+1)M+\alpha-m)} \\
\mathcal{R}\overline{G}_M^{k,\alpha}(m) &= \mathcal{R}\underline{G}_M^{k,\alpha}(m) \text{ when } m < M-1 \text{ or } m \geq M+1
\end{aligned}$$

并且当 $B^{k,\alpha}_M(M) = B^{k,\alpha}_M(M+1)$

$$\begin{aligned}
\mathcal{R}\overline{G}_M^{k,\alpha}(M-1) &= \frac{k^2(n+3-h-2M)(n+2-h-2M)(M+1)}{(M+h)(kM+\alpha+1)(kM+\alpha)} \\
\mathcal{R}\overline{G}_M^{k,\alpha}(M) &= \mathcal{R}F_n^h(M)
\end{aligned}$$

并且当 $B^{k,\ \alpha}_{\ M}(M) = B^{k,\ \alpha}_{\ M}(M-1)$

$$\begin{aligned} \mathcal{R}\overline{G}^{k,\alpha}_{M}(M-1) &= \mathcal{R}F^{h}_{n}(M-1) \\ \mathcal{R}\overline{G}^{k,\alpha}_{M}(M) &= \frac{k^2(n+1-h-2M)(n-h-2M)M}{(M+1+h)(k\,M+\alpha+1)(k\,M+\alpha)}. \end{aligned}$$

根据命题 1，我们可以证明该算法的平均复杂度关于 n 是线性的。现在只需证明该算法是有效的：$\mathcal{R}G^{k,\ \alpha}_{\ M}(m) > 1$ 当且仅当 $m < M$；这由以下引理证明，该引理的证明见附录 A.4：

引理 8 当 $m \leq M-1$ 时，我们有 $\mathcal{R}G^{k,\ \alpha}_{\ M}(m) > 1$；当 $m \geq M$ 时，我们有 $\mathcal{R}G^{k,\ \alpha}_{\ M}(m) \leq 1$。

此外：

- *when* $B^{k,\alpha}_{M}(M) = B^{k,\alpha}_{M}(M+1)$, $\frac{k(n+3-h-2M)(n+2-h-2M)}{(M+h)(k\,M+\alpha)} > 1$,
- *when* $B^{k,\alpha}_{M}(M) = B^{k,\alpha}_{M}(M-1)$, $\frac{k(n+1-h-2M)(n-h-2M)}{(M+1+h)(k\,M+1+\alpha)} \leq 1$

因此，我们可以更新前一个算法的主循环，以概率 $\frac{F^{h}_{n}(m)}{F^{h}_{n}}$ 生成 m，从而得到一个在 ~~$h \leq$~~ $n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$ 时可用的算法，其平均运行时间为 $O(n)$，且我们只使用位数少于 $2\log_2(n)$ 的数字：见算法 3 和 4。

算法 3 以概率接受 m $\frac{F_n^h(m)}{\overline{B}^{k,\alpha}_M(m)}$ 当 $1 \le \delta \le k-1$:

函数 ACCEPT_M$(M, k, \alpha, which, m)$
如果 $m > (n+1-h)/2$, 则拒绝 m 结束如果
如果 $m < M$ 那么
▷ 以概率 $\mathcal{R}\overline{B}^{h,\alpha}_M(M-1)/\mathcal{R}F_n^h(M-1)$ 接受 m
如果 $which =$ 'M+1' 那么
如果 $random(k(M+1)) \ge k\,M+\alpha+1$ 则拒绝 m 结束 if
$v \leftarrow random(k(n+3-h-2M)(n+2-h-2M))$
如果 $v \ge (M+h)(k\,M+\alpha)$, 则拒绝 m 结束如果
else
$v \leftarrow random((n+3-h-2M)(n+2-h-2M))$
如果 $v \ge M(M+h)$, 则拒绝 m 结束如果
结束如果
对于所有 $i \in [m, M-2]$, 执行
▷ 以概率 $\mathcal{R}\overline{B}^{h,\alpha}_M(i)/\mathcal{R}F_n^h(i)$ 接受 m
$v \leftarrow random(k(n+1-h-2i)(n-h-2i))$
如果 $v \ge (i+1+h)((k+1)M+\alpha-i)$, 则拒绝 m 结束条件判断
结束循环
else
▷ 接受 m , 概率为 $\mathcal{R}\overline{B}^{h,\alpha}_M(M)/\mathcal{R}F_n^h(M)$
若 $which =$ 'M-1' 则
若 $random(k\,M+\alpha) \ge k\,M$ 则拒绝 m 结束若
$v \leftarrow random((M+h+1)(k\,M+\alpha+1))$
若 $v \ge k(n+1-h-2M)(n-h-2M)$ 则拒绝 m 结束若
else
$v \leftarrow random((M+1)(M+1+h))$
若 $v \ge (n+1-h-2M)(n-h-2M)$ 则拒绝 m 结束若
结束条件判断
对所有 $i \in [M+1, m-1]$ 执行
▷ 以概率 $\mathcal{R}F_n^h(i)$ 接受 $m\,/\mathcal{R}\overline{B}^{h,\alpha}_M(i)$
$v \leftarrow random((i+1+h)((k+1)M+\alpha-i))$
如果 $v \ge k(n+1-h-2i)(n-h-2i)$, 则拒绝 m 结束条件判断
结束循环
结束条件判断
接受 m
结束函数

算法 4 绘制大小为 m 且最终高度为 $h-1$ 的 Motzkin 左因子

1: 如果 $h \geq n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$，则失败。结束如果 ▷ h 太大
2: $M \leftarrow m$ 的使 $F_n^h(m)$ 最大化的第一个值
3: $\tilde{m} \leftarrow \frac{4n+4-3h-\sqrt{4n^2+20n+28-3h^2}}{6}$
4: $k \leftarrow \lceil \frac{n+1-h}{2\tilde{m}} \rceil - 1$
5: $\alpha \leftarrow max(k-1, k - \lceil k(M-\tilde{m}) \rceil)$
6: 如果 $(kM+\alpha+1)(kM+\alpha) \geq k^2 M(M+1)$，则
7: $which =$ 'M+1'
8: 否则
9: $which =$ 'M-1'
10: 结束如果
11: 当真时执行
12: ▷ 以概率抽取 m $\frac{B_M^{k,\alpha}(m)}{\sum_{i=0}^{(k+1)M} B^{k,\alpha}_{M}(i)}$
13: $m \leftarrow \text{EXTENDED\_BIN}(M, k, \alpha)$
14: ▷ 以概率 $F_n^h(m)$ 接受 m $B_M^{k,\alpha}(m)$
15: 如果 ACCEPT_M$(M, k, \alpha, which, m)$，则跳出循环。结束条件判断
16: 结束 while 循环
17: ▷ 绘制一个大小为 n、包含 $m+h-1$ 个字母 x 和 m 的 Motzkin 左因子字母 $\bar{x}$

我们可以验证，除了检查 h 是否足够小（第 1 行），以及计算 k 和 α（第 3–5 行）时，我们只使用 $2\log_2(n)$ 位数字。然而，在算法的预计算阶段（第 1–5 行）完全避免使用某些浮点计算似乎更加困难[8]。

5.2 当 $h \geq n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$ 时 Motzkin 左因子的生成

在这种情况下，我们知道 M 等于 0 或 1；我们需要在 $[0, \lfloor \frac{n+1-h}{2} \rfloor]$ 中找到一个值 m，使其具有合适的概率：$F_n^h(m)$ F_n^h。

我们使用带有 $k = \lfloor \frac{n+1-h}{2} \rfloor \leq \frac{1}{4} + \frac{\sqrt{8n+9}}{4} = O(\sqrt{n})$ 的“基本生成器”。我们有 $\mathcal{R}G_M^k(m) = \mathcal{R}F_n^h(m)$，因此 $\mathcal{R}G_M^k(m) > 1$ 当且仅当 $m < M$，所以该算法是有效的。

我们可以应用命题 1，取 $C_1 = O(1)$、$C_2 = O(k)$、$F(M) \sum_m F(m) = O(1)$、$\sum_m B(m) B(M) = k+1$，从而证明主循环的总复杂度平均为 $O(1+k^2) = O(n)$，如我们所愿。

[8] 最棘手的问题是计算 k。事实上，似乎可以（如 $M \geq 2$）用 $k \leftarrow \lceil \frac{n+1-h}{2M-1}() \rceil - 1$ 来替代这项计算，但这意味着必须彻底重写引理 8 的证明。至于 α，只需找到一个 $\alpha\ in [0, k-1]$ 的值，使得 $\max(\mathcal{R}G(M)\ \mathcal{R}G(M+1)) \leq 1 \leq \min(\mathcal{R}G(M-2)\ \mathcal{R}G(M-1))$，这可以通过测试 $O(k)$ 中所有可能的值来完成。

6 讨论

许多近期的生成算法使用拒绝法来高效生成简单结构。它们大多基于巧妙的一一对应关系。这里我们尝试寻找将问题拆分成小部分的方法：找到某个概率足够大的 m，从而得到一个易于解决的简单问题。据我们所知，这种方法已用于生成莫茨金词、部分单射、带颜色的一元二叉树 [GBN10]。

将这种方法以不同方式应用于斐波那契数后，我们得到了一个高效且极易实现的算法。令人惊讶的是，只需稍作修改，就能用它来生成 Schro¨eder 树[9]。为了研究这类算法的局限性，我们尝试生成最终高度为 h 的 Motzkin 左因子：这个问题似乎更复杂，因为字数变化很大，从 $M_n^0 \approx \frac{3^n}{n\sqrt{n}}$ 到 $M_n^n = 1$ [10] 不等；我们需要扩展“二项式生成器”，但最终得到的算法既高效，实现起来也相对简单。

还要注意，正如在 [GBN10]，中一样，我们寻找的是尝试再现目标分布的生成器：取使初始分布和最终分布达到最大值的相同 M，并使“离散度”相近（相差一个常数因子以内）。我们发现，至少对于这三个问题，确实存在一些非常简单的生成器。不过，这样做究竟是好主意，还是会产生误导？这将是未来一篇文章[Alo23]的主题。这里只需指出，本文将使“二项式生成器”和“扩展二项式生成器”的实现更加高效（从而降低文中所述算法的平均复杂度）。

还能说些什么？或许，这篇文章中提出的问题对应的数字，其渐近式为 $l(n)C^n$，其中 C 是一个二次多项式的解；这或许可以解释为什么我们只需使用 $2\,log_2(n)$ 位的数字就能解决这些问题。当 C 是一个三次多项式的解时，.....，这种方法（若适用）可能需要使用$3\,log_2(n)$ 位的数字，.....。

我们能做得更好吗？例如，我们能否找到一种算法，只使用小于 n 的整数来生成一个“大小”为 n 的斐波那契词？

与 Gouyou-Beauchamps 等人进行比较 [GBN10] 中 [GBN10],他们提出了一种通用算法，通过拒绝法抽取一个数 m，这样我们“自动”得到的分布便可用于定义他们的初始分布 $\overline{B}_n(m)$。该分布使我们能够找到高效算法，以概率 $\frac{F_n(m)}{\overline{B}_n(m)}$接受 m。事实上，我们可以将自己的算法视为 [GBN10] 的一种实例化，它采用固定方法，以

[9]事实上，对于许多类似的问题，它似乎能给出一个 $O(n)$ 算法，例如莫茨金词、二叉树森林、三叉树森林等；本文未对这些问题作出说明。它还为部分单射[10]提供了一个$O(n^{5/4})$算法。另请注意，只有当 $h \le n - \frac{1}{2} - \frac{\sqrt{8n+9}}{2}$ 时，‘扩展二项式生成器’才适用，因此我们有 $\ln(M_n^{n-\sqrt{2n}}) \approx \frac{\sqrt{2}}{2}\sqrt{n}\ln n$

概率 $F_{n(m)}\,\overline{B}_{n(m)}$接受某个值。固定这种方法后，我们便能通过选择一个生成器，并检查（对于某些 n 值）$\mathcal{R}\overline{B}_n(m) < \mathcal{R}F_n(m)$ 当且仅当 $m < M$，快速判断能否为某些结构找到高效算法。当然，这也限制了 $\overline{B}_n(m)$ 的潜在选择；但至少对于我们考察过的结构，我们能够迅速得到一种高效算法。

还需注意，我们也为此付出了代价：尽管复杂度相近，但证明更加复杂；而且，在使用 $h = 0$ 生成 Motzkin 左因子时，用于接受 m 的算法也不如 [Alo94] 中提出的算法优雅；如果我们尝试用“二项式生成器”来实例化我们的方法以生成部分单射，并将其与 [GBN10][11] 进行比较，情况也会如此。

最后请注意，用于猜测 m 初始值的分布略有不同，

- 我们选择标准二项分布，但稍微调整一下，将 $\overline{B}_n(M)$ 的值减小（以强制使 $\overline{B}_n(M) = \overline{B}_n(M \pm 1)$），
- 在 [GBN10]，中，他们使用标准二项分布，这也解释了为什么他们需要选择一个有时大于 $F_n(M)$ 的 $\overline{B}_n(M) =$ 值。

参考文献

[Alo94] Laurent Alonso。《莫茨金词的均匀生成》。《理论计算机科学》，134(2):529–536，1994。[Alo23] Laurent Alonso。《均匀随机生成与拒绝法（ii）：使用平凡上界》。2023。[Au19] Yu Hin Au。《加权 Schröder 数的一些性质及其组合学意义》。arXiv：组合学，2019。[BBJ17] Axel Bacher、Olivier Bodini 和 Alice Jacquot。通过全纯方程高效随机抽样二叉树和一元-二元树。《理论计算机科学》，695:42–53，2017。[BPR06] Srecko Brlek、Elisa Pergola 和 Olivier Roques。广义 Motzkin 路径的非均匀随机生成。《Acta Informatica》，42:603–616，2006。[BPS94] Elena Barcucci、Renzo Pinzani 和 Renzo Sprugnoli。有向动物的随机生成。《理论计算机科学》，127:333–350，1994。[DZ90] Nachum Dershowitz 和 Shmuel Zaks。循环引理及其若干应用。《European Journal of Combinatorics》，11(1):35–40，1990。

[11]对于这个问题，最好使用带有 $k = \lceil\sqrt{n}\rceil + 1$ 的“扩展二项式生成器”来选择 $n - m$，这样可以得到一个有效的 $O(n)$ 复杂度算法，并且该算法在 $n \geq 2$ 时有效。

[DZ99] Alain Denise 和 Paul Zimmermann。使用浮点运算均匀随机生成可分解结构。《理论计算机科学》，218(2):233–248，1999。[FZC94] Philippe Flajolet、Paul Zimmermann 和 Bernard Van Cutsem。带标号组合结构随机生成的演算。《理论计算机科学》，132:1–35，1994。[GBN10] Dominique Gouyou-Beauchamps 和 Cyril Nicaud。使用二项式近似进行随机生成。《Discrete Mathematics & Theoretical Computer Science》，第359–372页，2010。[GKP88] Ronald L. Graham、Donald Ervin Knuth 和 Oren Patashnik。《具体数学》，1988。[Map] Maplesoft，Waterloo Maple Inc. 的一个部门。Maple。[NW78] Albert Nijenhuis 和 Herbert S. Wilf。《组合算法》（第二版），1978。[OEI22] OEIS Foundation Inc.《在线整数数列百科全书》，2022。电子版发布于 http://oeis.org。[PPPR01] Jean-Guy Penaud、Elisa Pergola、Renzo Pinzani 和 Olivier Roques。Schröder 路径与随机层级。《理论计算机科学》，255(1):345–361，2001。

A 附录：证明

A.1 附录：“扩展二项式生成器”的证明 引理 3 的证明。我们有：

$$\frac{\underline{B}_M^{k,\alpha}(m+1)}{\underline{B}_M^{k,\alpha}(m)} = \frac{(k+1)M+\alpha-m}{k(m+1)} \leq 1$$

当且仅当 $(k+1)M+\alpha-m \leq k(m+1)$ 当且仅当 $M+\frac{\alpha-k}{k+1} \leq m$ 当且仅当 $M \leq m$。

我们先证明这些关系对 $\underline{B}_M$ 成立。注意：

$$l(\alpha) = \frac{\underline{B}_M^{k,\alpha}(M)}{\sum_m \underline{B}_M^{k,\alpha}(m)} = \frac{((k+1)M+\alpha)!k^{k\,M+\alpha}}{M!(k\,M+\alpha)!(k+1)^{(k+1)M+\alpha}}.$$

利用斯特林公式，我们得到：

$$l(0) = \frac{((k+1)M)!k^{k\,M}}{M!(k\,M)!(k+1)^{(k+1)M}} \approx \frac{\sqrt{2(k+1)}}{2\sqrt{\pi\,k\,M}}$$

更精确地，利用 $\sqrt{2\pi\,n}\left(\frac{n}{e}\right)^n \leq n! \leq \sqrt{2\pi\,n}\left(\frac{n}{e}\right)^n \exp\left(\frac{1}{12n}\right)$：

$$\begin{aligned} 0.84 &\leq exp\left(-\frac{1}{6}\right) \leq exp\left(-\frac{1}{12\,M}-\frac{1}{12\,k\,M}\right) \leq \frac{2\sqrt{\pi\,k\,M}l(0)}{\sqrt{2(k+1)}} \\ &\leq exp\left(\frac{1}{12(k+1)M}\right) \leq exp\left(\frac{1}{24}\right) \leq 1.05. \end{aligned}$$

我们还有：

$$\frac{l(\alpha)}{l(\alpha-1)} = \frac{k((k+1)M+\alpha)}{(k+1)(k\,M+\alpha)} = 1-\frac{\alpha}{(k+1)(k\,M+\alpha)}$$

因此我们得到：$l(\alpha) \leq l(0)$ 和

$$\frac{l(\alpha)}{l(0)} \geq 1-\sum_{i=1}^{\alpha}\frac{i}{(k+1)(k\,M+i)} \geq 1-\frac{\alpha(\alpha+1)}{2(k+1)(k\,M+1)} \geq 1-\frac{k}{2(k\,M+1)} > \frac{1}{2}.$$

最后请注意，$n=(k+1)M+\alpha$，乘以一个常数因子 $(k+1)^n$，$\underline{B}_M$ 对应于参数为 $p=\frac{1}{k+1}$、$q=1-p$ 的二项分布。因此，$\underline{B}_M$ 的方差为 n、p、$q = O\left(\frac{(M+1)(k+1)k}{k+1}()^2\right) = O(M+1)$。利用詹森不等式，我们得到：$\sum_m |m-M|\underline{B}_{,M}^{k\ \alpha}(m) = O(\sqrt{M})\sum_m \underline{B}_M^{k,\alpha}(m)$

最后指出，对于 B_M，当 $B_M(M) = \frac{kM}{kM+\alpha+1}\underline{B}_M(M)$ 或 $B_M(M) = \frac{kM+\alpha}{k\,M+1}()$ $\underline{B}_M(M)$ 时，这些关系仍然成立，因此 $B_M(M) = \underline{B}_M(M)(1-O(1/M))$。

□

A.2 附录：斐波那契的证明

引理 4 的证明。基本计算得出 $f'(x) = -5n + 10x + 3$, $f(0) = n^2 - 2n \geq 0$，若 $n > 1$，则 $f(\lfloor n/2 \rfloor) = -\lceil n/2 \rceil (\lfloor n/2 \rfloor + 1) < 0$，若 $n \geq 1$。这意味着存在 $x \in [0, \lfloor n/2 \rfloor]$ 的唯一取值： $\tilde{m}$，使得 $f(x) > 0$ 当且仅当 $x < \tilde{m}$。

利用 $\mathcal{R}F_n(x) > 1$ 当且仅当 $f(x) > 0$ 且 $\mathcal{R}F_n(m) = F_n(m+1)/F_n(m)$，我们得到 $M = \lceil \tilde{m} \rceil$。在 $[0, \lfloor n/2 \rfloor]$ 中求解 $f(\tilde{m}) = 0$，得到：

$$\tilde{m} = \frac{5n - 3 - \sqrt{5n^2 + 10n + 9}}{10} = \frac{5 - \sqrt{5}}{10} n - \frac{3 + \sqrt{5}}{10} + O(1/n) \approx \frac{5 - \sqrt{5}}{10} n.$$

假设 $\epsilon \geq 0$ 和 $n > 2$，我们有：

$$\begin{aligned} \mathcal{R}F_n(\tilde{m} + \epsilon) &= \frac{(n - 2\tilde{m})(n - 1 - 2\tilde{m})}{(\tilde{m} + 1)(n - \tilde{m})} \frac{(1 - 2\frac{\epsilon}{n - 2\tilde{m}})(1 - 2\frac{\epsilon}{n - 1 - 2\tilde{m}})}{(1 + \frac{\epsilon}{\tilde{m}+1})(1 - \frac{\epsilon}{n - \tilde{m}})} \\ &\geq 1(1 - 2\frac{\epsilon}{n - 2\tilde{m}} - 2\frac{\epsilon}{n - 1 - 2\tilde{m}} - \frac{\epsilon}{\tilde{m} + 1} + 0) \geq 1 - \frac{5\epsilon}{\tilde{m}} \end{aligned}$$

注意到 $\tilde{m} \leq \frac{n-1}{3}$（因此 $n - 1 - 2\tilde{m} \geq \tilde{m}$）。因此，当 $i \geq 0$ 时，我们得到 $\mathcal{R}F_n(M + i) \geq 1 - \frac{5}{\tilde{m}} i$；而当 $i < \frac{\tilde{m}}{5}$ 时，$\frac{F_n(M+i)}{F_n(M)} \geq \prod_{j=0}^{i-1} (1 - \frac{5}{(j+1)\tilde{m}}) \geq 1 - \frac{5i(i+1)}{2\tilde{m}}$。这意味着，当 $0 \leq i \leq \frac{\sqrt{\tilde{m}-1}}{\sqrt{5}}$ 时，$F_n(M + i) \geq \frac{1}{2} F(M)$，因此 $F_n = \Omega(\sqrt{M}) F_n(M) = \Omega(\sqrt{n}) F_n(M)$。 □

引理 5 的证明。我们有 $\mathcal{R}\overline{B}_M(M - 1) = \mathcal{R}\overline{B}_M(M) = 1$，并利用前一个引理，得到 $\mathcal{R}F_n(M - 1) > 1, \mathcal{R}F_n(M) \leq 1$。这验证了 $m = M - 1$ 和 $m = M$ 时的不等式。

对于 m 的其他取值，我们有 $\mathcal{R}\overline{B}_M(m) < \mathcal{R}F_n(m)$ 当且仅当 $\underline{g}(m) = (n-2m)(n-2m-1) - (2M-m)(n-m) > 0$。当 $n > 2$ 时，我们有 $\underline{g}(0) = n(n-2M-1) > 0$, $\underline{g}(n/2-1) = -(2M+1-n/2)(n/2-1) < 0$， $\underline{g}'(x) = 6x - 3n + 2M + 2$。这意味着，当 $x \in [0, n/2-1]$ 时，存在一个值 $\tilde{x}$，使得 $\underline{g}(x) \geq 0$ 当且仅当 $x \leq \tilde{x}$。

因此，只需证明 $\underline{g}(M - 2) > 0$ 和 $\underline{g}(M + 1) \leq 0$。
利用 $M = \lceil \tilde{m} \rceil$，我们得到（因为 $\frac{\delta g}{(M-2)\,\delta M} < 0$）：

$$\begin{aligned} \underline{g}(M - 2) &= 5M^2 - (5n + 14)M + n^2 + 5n + 8 \\ &\geq 5(\tilde{m} + 1)^2 - (5n + 14)(\tilde{m} + 1) + n^2 + 5n + 8 \\ &= \frac{11}{10} - \frac{3n}{2} + \frac{7\sqrt{5n^2 + 10n + 9}}{10} > 0. \end{aligned}$$

同理，我们得到（当 $n \geq 6$ 时，$\frac{\delta g}{(M+1)\,\delta M} < 0$）：

$$\begin{aligned} \underline{g}(M + 1) &= 5M^2 - (5n - 10)M + n^2 - 4n + 5 \\ &\leq 5\tilde{m}^2 - (5n - 10)\tilde{m} + n^2 - 4n + 5 \\ &= \frac{29}{10} + \frac{3n}{2} - \frac{7\sqrt{5n^2 + 10n + 9}}{10}, \end{aligned}$$

当 $n \geq 20$ 时，它小于或等于 0。

最后，我们可以计算 $n < 20$ 时 $\underline{g}(M+1)$ 的值，得到：

n	0	1	2	3	4	5	6	7	8	9
M	0	0	0	1	1	1	2	2	2	2
$\underline{g}(M+1)$	5	2	1	2	0	0	-3	-4	-3	0

n	10	11	12	13	14	15	16	17	18	19
M	3	3	3	4	4	4	4	5	5	5
$\underline{g}(M+1)$	-10	-8	-4	-18	-15	-10	-3	-24	-18	-10

由此结束证明：注意到当 $n \leq 3$ 时，不存在含有 $M+2$ 个字母 $|b|$ 的词。

□

A.3 附录：Schr¨oder 的证明

引理 6 的证明。事实上，$\mathcal{R}F_n(x) > 1$ 当且仅当 $f(x) = (n-x)(n+1+x) - (x+1)(x+2) > 0$。基本计算得出 $f'(x) = -4x-4$，$f(0) = n(n+1) - 2 \geq 0$ 当 $n \geq 1$，$f(n) = -(n+1)(n+2) < 0$。这说明存在唯一的 $x \in [0,n]$ 值：$\tilde{m}$，使得 $f(x) > 0$ 当且仅当 $x < \tilde{m}$。

关系 $\mathcal{R}F_n(x) > 1$ 当且仅当 $f(x) > 0$，且 $\mathcal{R}F_n(m) = F_n(m+1)/F_n(m)$，由此得到 $M = \lceil \tilde{m} \rceil$。通过在 $[0,n]$ 中求解 $f(\tilde{m}) = 0$，我们得到：

$$\tilde{m} = -1 + \frac{\sqrt{2n^2+2n}}{2} = \frac{n}{\sqrt{2}} - \frac{4-\sqrt{2}}{4} + O(1/n) \approx \frac{n}{\sqrt{2}}.$$

我们还可以注意到 $\tilde{m} < \frac{n}{\sqrt{2}}$（因此 $n - \tilde{m} \geq (\sqrt{2}-1)\tilde{m}$）。

为了证明 $F_n = \Omega(\sqrt{n})F_n(M)$，我们可以仿照引理 4 的证明。事实上，假设 $\epsilon \geq 0$，则有：

$$\mathcal{R}F_n(\tilde{m}+\epsilon) = \frac{(1-\frac{\epsilon}{n-\tilde{m}})(1-\frac{\epsilon}{n+1-\tilde{m}})}{(1+\frac{\epsilon}{\tilde{m}+1})(1+\frac{\epsilon}{\tilde{m}+2})} \geq 1 - \frac{7\epsilon}{\tilde{m}}.$$

因此，当 $0 \leq i < \frac{\tilde{m}}{7}$，$\frac{F_n(M+i)}{F_n(M)} \geq 1 - \frac{7i}{(i+1)2\tilde{m}}$ 时，我们得到 $F_n = \Omega(\sqrt{n})F_n(M)$。

我们有 $\mathcal{R}\overline{B}_M(M-1) = \mathcal{R}\overline{B}_M(M) = 1$，以及 $\mathcal{R}F_n(M-1) > 1, \mathcal{R}F_n(M) \leq 1$。这验证了当 $m = M-1$ 和 $m = M$ 时的最后一个不等式。

对于 m 的其他取值，当且仅当 $\underline{g}(m) = (n-m)(n+1+m) - (2M-m)(m+2) > 0$ 时，$\mathcal{R}\overline{B}_M(m) < \mathcal{R}F_n(m)$。我们有 $\underline{g}(0) = n^2+n-4M > 0$，$\underline{g}(n) = -(2M-n)(n+2) < 0$（当 $n > 3$ 时），$\underline{g}'(x) = 1-2M$。这意味着，当 $x \in [0,n]$ 时，存在一个值 $\tilde{x}$，使得 $\underline{g}(x) \geq 0$ 当且仅当 $x \leq \tilde{x}$。因此，只需证明 $\underline{g}(M-2) > 0$ 且 $\underline{g}(M+1) \leq 0$。

利用 $M = \lceil \tilde{m} \rceil$，我们得到

$$\begin{aligned} \underline{g}(M-2) &= -2M^2+n^2+M+n-2 > -2(\tilde{m}+1)^2+n^2+(\tilde{m}+1)+n-2 \\ &= \frac{\sqrt{2n^2+2n}-4}{2} \end{aligned}$$

当 $n \geq 3$ 时，该值大于 0。

同理，

$$\underline{g}(M+1) = -2M^2+n^2-5M+n+1 \leq -2\tilde{m}^2+n^2-5\tilde{m}+n+1 = \frac{8-\sqrt{2n^2+2n}}{2}$$

当 $n \geq 6$ 时，它小于或等于 0。

最后，我们可以计算当 $n < 6$ 时 $\underline{g}(M-2)$ 和 $\underline{g}(M+1)$ 的值，结果为：

n	0	1	2	3	4	5
M	-1	0	1	2	3	3
$\underline{g}(M-2)$	-5	0	3	4	3	13
$\underline{g}(M+1)$	4	3	0	-5	-12	-2

□

A.4 附录：Motzkin 左因子的证明

引理 7 的证明。事实上，$\mathcal{R}F_n^h(x) > 1$ 当且仅当 $f(x) = (n-h-2x)(n+1-h-2x) - (x+1)(x+1+h) > 0$。基本计算得出 $f'(x) = -4n+6x+3h-4$，$f(0) = (n-h)^2 + n - 2h - 1 > 0$ 当 $n \geq 1$，$f(\lfloor \frac{n+1-h}{2} \rfloor) = -(\lfloor \frac{n+1-h}{2} \rfloor + 1)(\lfloor \frac{n+1-h}{2} \rfloor + 1 + h) < 0$。这意味着存在唯一的 $x \in [0, \lfloor \frac{n+1-h}{2} \rfloor]$ 值：$\tilde{m}$，使得 $f(x) > 0$ 当且仅当 $x < \tilde{m}$。

关系 $\mathcal{R}F_n^h(x) > 1$ 当且仅当 $f(x) > 0$ 且 $\mathcal{R}F_n^h(m) = F_n^h(m+1)/F_n^h(m)$ 时，得出 $M = \lceil \tilde{m} \rceil$。在 $[0, n]$ 中求解 $f(\tilde{m}) = 0$，得到：

$$\tilde{m} = \frac{2(n+1)}{3} - \frac{h}{2} - \frac{\sqrt{4n^2+20n+28-3h^2}}{6} \leq \frac{n-h}{3} \leq \frac{n+1-h}{3},$$

因此 $\tilde{m} = 1$ 对应于 $h = n - \frac{1}{2} + \frac{\sqrt{8n+9}}{2}$，这意味着当$n - \frac{1}{2} + \frac{\sqrt{8n+9}}{2}$ 时 $M \geq 2$，因为 $\frac{\delta \tilde{m}}{\delta h} < 0$。

现在假设 $\epsilon \geq 0$，我们有：

$$\mathcal{R}F_n^h(\tilde{m}+\epsilon) = \frac{\left(1-\frac{2\epsilon}{n-h-2\tilde{m}}\right)\left(1-\frac{2\epsilon}{n+1-h-2\tilde{m}}\right)}{\left(1+\frac{\epsilon}{\tilde{m}+1}\right)\left(1+\frac{\epsilon}{\tilde{m}+h+1}\right)} \geq 1 - \frac{6\epsilon}{\tilde{m}}$$

因此当 $0 \leq i < \frac{\tilde{m}}{6}$ 时，我们得到 $\frac{F_n^h(M+i)}{F_n^h(M)} \geq 1 - \frac{3i}{(i+1)\tilde{m}}$，由此可得，当 $M \geq 2$ 时，$F_n^h = F_n^h(M)\Omega(\sqrt{\tilde{m}}) = F_n^h(M)\Omega(\sqrt{M})$。□

当 $m \leq M-2$ 或 $m \geq M+1$ 时引理 8 的证明。我们有 $\underline{g}(-h-1) = k(n+3+h)(n+2+h) > 0$ 并且

$$\underline{g}\left(\frac{n-h}{2}\right) = -\frac{n+h+2}{2}\left((k+1)M + \alpha - \frac{n-h}{2}\right) < 0$$

作为 $(k+1)M \geq (k+1)\tilde{m} \geq \frac{n+1-h}{2}$。

我们有:

$$\underline{g}'(x) = (8k+2)x - (4n+2-4h+M)k - M - \alpha + h + 1,$$

因此， $\underline{g}$ 在$[-h-1, \frac{n-h}{2}[$, 中有唯一的根 $\tilde{x}$ ，并且当 $x < \tilde{x}$ 时， $\underline{g}(x) > 0$; 当 $x > \tilde{x}$时， $\underline{g}(x) < 0$ 。

我们有 $\alpha = max(k-1, \lfloor k - k(M-\tilde{m})\rfloor)$，因此:

$$(k+1)\,\tilde{m}+k-1 \leq M+k(\,\tilde{m}+1)-1 \leq (k+1)M+\alpha \leq M+k(\,\tilde{m}+1) < (k+1)\,\tilde{m}+k+1.$$

But:

$$\begin{aligned} \underline{g}(\tilde{m}) &= k\,f(\tilde{m}) - (\tilde{m}+1+h)((k+1)M+\alpha-\tilde{m}) + k(\tilde{m}+1+h)(\tilde{m}+1) \\ &= (\tilde{m}+1+h)((k+1)\tilde{m}+k-((k+1)M+\alpha)) \end{aligned}$$

and

$$g_1(x) = \underline{g}(x+1) - \underline{g}(x) = -((k+1)M+\alpha) - 2(2n-2h-4x-1) + h + 2x + 2.$$

因此，当 $h < n-1$ 时（这始终成立），如 $4h-4n+7\tilde{m}+1 < 0$所示:

$$\begin{aligned} g(\tilde{m}+1) &= g(\tilde{m}) + g_1(\tilde{m}) \leq (4h-4n+7\tilde{m}+1)k + 2h + 2\tilde{m} + 4 \\ &\leq \frac{-10\tilde{m}^2 + (15n-11h+13)\tilde{m} - (4n-4h-1)(n-h+1) + 4\,f(\tilde{m})}{2\tilde{m}} \\ &= \frac{2\tilde{m}^2 + (h-n-3)\tilde{m} - 5h + n - 3}{2\tilde{m}} \end{aligned}$$

当 $n \geq 9$时，它为负，因为 $\tilde{m} \leq \frac{n-h}{3} \leq \frac{n-h+3+\sqrt{(n-h)^2+34h-2n+33}}{4}$。最后，当 $n \leq 8$时，我们验证对于 h的所有可能取值， $2\tilde{m}^2 + (h-n-3)\,\tilde{m} - 5h + n - 3 < 0$。

同理，当 $4n-4h-7\tilde{m}+7 > 0$ 时，我们有:

$$\begin{aligned} g(\tilde{m}-1) &= g(\tilde{m}) - g_1(\tilde{m}-1) \geq (4n-4h-7\tilde{m}+7)k - 2h - 2\tilde{m} \\ &\geq \frac{-4\tilde{m}^2 - (7n-3h+7)\tilde{m} + (4n-4h+7)(n+1-h) - 4\,f(\tilde{m})}{2\tilde{m}} \\ &= \frac{-16\tilde{m}^2 + 9(n-h+1)\tilde{m} - 3h + 7n + 11}{2\tilde{m}} \end{aligned}$$

这为正，因为 $\tilde{m} \leq \frac{n-h}{3} \leq \frac{9n-9h+9+\sqrt{81(n-h)^2-354h+610n+785}}{32}$ 。

因此我们得到 $\underline{g}(M-2) \geq \underline{g}(\,\tilde{m}-1) > 0$ 以及 $\underline{g}(M+1) \leq \underline{g}(\tilde{m}+1) < 0$。这等价于 $\mathcal{R}G^{k,}{}_{M}{}^{\alpha}(m) > 1$ 当 $m \leq M-2$ 且 $\mathcal{R}G^{k,}{}_{M}{}^{\alpha}(m) < 1$ 当 $m \geq M+1$。 □

$m = M$或 $m = M+1$ 时引理 8 的证明。首先注意，当 $B^{k,}{}_{M}{}^{\alpha}(M) = B^{k,}{}_{M}{}^{\alpha}(M-1)$, $\mathcal{R}G^{k,}{}_{M}{}^{\alpha}(M-1) = \mathcal{R}F_n^h(M-1) > 1$; 而当$B^{k,}{}_{M}{}^{\alpha}(M) = B^{k,}{}_{M}{}^{\alpha}(M+1)$, $\mathcal{R}G^{k,}{}_{M}{}^{\alpha}(M) = \mathcal{R}F_n^h(M) \leq 1$。

我们还得到如下结果，因为 $0 \leq \alpha \leq k-1$:

$$\frac{k^2(n+3-h-2M)(n+2-h-2M)(M+1)}{(M+h)(k\,M+\alpha+1)(k\,M+\alpha)} \geq \frac{k(n+3-h-2M)(n+2-h-2M)}{(M+h)(k\,M+\alpha)},$$

and:

$$\frac{k^2(n+1-h-2M)(n-h-2M)M}{(M+h+1)(k\,M+\alpha+1)(k\,M+\alpha)} \leq \frac{k(n+1-h-2M)(n-h-2M)}{(M+h+1)(k\,M+1+\alpha)},$$

当 $k=1$ 和 $\alpha=0$时，这些不等式会变为等式。此外，当 $k=1$时，我们有 $\alpha=0$和 $B^{k,\ {}^{\alpha}}_{M}(M) = B^{k,\ {}^{\alpha}}_{M}(M-1) = B^{k,\ {}^{\alpha}}_{M}(M+1)$；因此可以直接得出结论。

现在注意到，$\alpha = max(k-1, \lfloor k-k(M-\tilde{m})\rfloor$意味着 $\tilde{m} + \frac{k-1-\alpha}{k} \leq M \leq \tilde{m} + \frac{k-\alpha}{k}$。

当 $B^{k,\ {}^{\alpha}}_{M}(M) = B^{k,\ {}^{\alpha}}_{M}(M+1)$（这等价于 $(kM+\alpha+1)(kM+\alpha) \geq k^2M(M+1)$，并且意味着 $\alpha > 0$）时，我们有：

$$\mathcal{R}\overline{G}^{k,\alpha}_{M}(M-1) \geq \frac{k(n+3-h-2M)(n+2-h-2M)}{(M+h)(k\,M+\alpha)}$$

若 $g_1(M) = k(n+3-h-2M)(n+2-h-2M) - (M+h)(k\,M+\alpha) > 0$, 则其大于 1。我们有 $g_1'(x) = (3h-4n+6x-10)k-\alpha < 0$ 当 $x < \frac{n+1-h}{2} + {}^{k(}{}_{n+7)_{6k}}{}^{+\alpha}$ 时。因此：

$$\begin{aligned} g_1(M) &\geq g_1\left(m + \frac{k-\alpha}{k}\right) \\ &= \alpha\frac{k(7\sqrt{-3h^2+4n^2+20n+28}-3h-4n-10)+24\alpha}{6k} \\ &\geq \alpha\frac{7\sqrt{-3h^2+4n^2+20n+28}-3h-4n-10}{6} > 0 \end{aligned}$$

当 $h \leq n+1 < \frac{-2n-5+7\sqrt{16n^2+80n+113}}{26}$ 时。

当 $B^{k,\ {}^{\alpha}}_{M}(M) = B^{k,\ {}^{\alpha}}_{M}(M-1)$（这意味着 $\alpha < k-1$）时，我们有：

$$\mathcal{R}\overline{G}^{k,\alpha}_{M}(M) \leq \frac{k(n+1-h-2M)(n-h-2M)}{(M+h+1)(k\,M+1+\alpha)}$$

若 $g_1(M) = k(n+1-h-2M)(n-h-2M) - (M+1+h)(k\,M+\alpha+1) \leq 0$, 则该值小于或等于 1。我们有 $g_1'(x) = (3h-4n+6x-3)k-\alpha-1 < 0$, 若 $x < \frac{n+1-h}{2} + \frac{k\,n+\alpha+1}{6k}$。因此：

$$\begin{aligned} g_1(M) &\leq g_1\left(m + \frac{k-\alpha-1}{k}\right) \\ &= -(k-\alpha-1)\frac{k(7\sqrt{-3h^2+4n^2+20n+28}-3h-4n-34)+24+24\alpha}{6k} \\ &\leq -(k-\alpha-1)\frac{7\sqrt{-3h^2+4n^2+20n+28}-3h-4n-34}{6} < 0 \end{aligned}$$

当 $h \leq n+1 < \frac{-2n-17+7\sqrt{16n^2+64n+25}}{26}$ 且 $n > 0$ 时。 □