arXiv:2303.02855v1 [math.CO] 6 Mar 2023

Friedman 的“长有限序列”：忙海狸竞赛的终结

Michael Vielhaber 不来梅哈芬应用科学大学，FB2，An der Karlstadt 8，D–27568 不来梅哈芬vielhaber@gmail.com

M´onica del Pilar CanalesChac´on
MATEMATICVM，瓦尔迪维亚，智利
@matematicvm
monicadelpilar@gmail.com

Sergio Jara Ceballos 工程学院，智利南方大学，瓦尔迪维亚，智利serjara.ing.mat@gmail.com

摘要

Harvey Friedman 对一个“难以想象的巨大”数量 n(3) 作出了相对简短的描述，超过例如 这些数值

$$A(7,184) < A(7198,158386) < n(3)$$

阿克曼函数的——但有限。

我们在一族图灵机上实现 Friedman 的组合问题，该问题涉及三字母字母表上词的子词。这些图灵机从空白纸带开始运行（超过） n(3) 步，然后停机。示例包括一台 (44,8)（符号数、状态数）机器，以及一台 (276,2) 和一台 (2,1840) 机器。

总计，至多有 37022 个非平凡的 (n,m) 对，其忙海狸数值 BB(n, m) $< A$(7198, 158386)。

我们给出算法，将任意 $(|Q|,|E|)$ 图灵机映射为另一台图灵机，并且可以自由选择 $|Q'| \geq 2$ 或 $|E'| \geq 2$ （空初始纸带的情形 $|Q'| = 2$ 比较棘手）。

考虑到 n(3) 的规模，以及这些图灵机并非候选者，而是保证会停机，Friedman 的组合问题为忙海狸竞赛中可能实现的成就提供了一个确定的上界。我们还讨论 n(4) $> A^{(A(187196))}$(1)。

关键词：忙海狸、长有限序列、图灵机。

引言

Harvey Friedman 描述了 [4] 这样一个问题：对于任意大小为 $k \in \mathbb{N}$ 的字母表B，确定最长词 $s \in B^*$ 的长度，使其满足性质

(∗)⇏ $1 \leq i < j \leq n/2$: $s^{(i)} := s_i s_{i+1} \dots s_{2i}$ is a subword of $s^{(j)} := s_j s_{j+1} \dots s_2$

得到满足。

对于 $k = 1$ 也就是说， $B = \{1\}$，111 满足 (∗)，因为 $s_2 s_3 s_4$ 甚至尚未定义；但 B^4 中唯一的词 $s = 1111$ 已经违反了 $i = 1, j = 2$ 的 (∗)。因此， $n(1) = 3$。

类似地，对于 $k = 2$ 和 $B = \{1, 2\}$，我们找到长度为 11 的词 $s = 11222111111$，它满足 (∗)；但所有长度为 12 的 2^{12} 词（以及更长的所有词）都违反了 (∗)（见 [4, 第 3 页及后续页]）：由此得到 $n(1) = 3, n(2) = 11$。

此后 $n(3)$ $n(4)$.....又是怎样发展的呢?

出乎意料!

Friedman 在第 [4, 4] 节首先证明了 $n(3) > A(7,184)$ 的下界，并在第 [4, 6] 节介绍了 Dougherty 得出的结果，甚至

$$n(3) > A(7198, 158386).$$

目前这个下界 $A(7198, 158386)$（远远！）大于例如$A(3, 158386) = 2^{2^{\cdot^{\cdot^{2}}}}$，其中 158386 个 2 以指数方式层层叠加。

不过，我们确实知道 $n(3)$ 的一位：最后一位。因为所有 $n(k)$ 都是奇数。

我们描述了这样一些图灵机：对于所有非平凡的（状态数、符号数）组合 $(n, m) \in \mathbb{N}^2$ （只有 37022 个例外），它们从空白纸带开始运行，保证经过超过 $n(3)$ 步后才停机。因此，忙海狸竞赛实际上是有限的</style。

第 1 节介绍 Ackermann、图灵、Rad´o 和 Friedman 的相关工作。第 2 节介绍算法，以及一种包含 44 个状态的 8 符号实现。

在第 3 节和第 4 节中，我们展示了如何将任意图灵机的符号数或状态数降至 2。第 5 节简要讨论了在 4 个符号上情形 $n(4)$，而在第 6 节中，我们得到了主要结果：

除至多 37022 个非平凡的（n， m）对之外，其余各对的 `BB(n,m)` 值都高于 $n(3)$，且至多有 51671 `BB(n,m)` 个低于 $n(4)$。

1 四位数学家及其关键成果

1.1 Wilhelm Ackermann，1926年：“阿克曼函数”

此处使用的（修改后的）阿克曼函数 [1] 在 [4] 中定义为 $A(1,c) = 2c$， $A(f,1) = 2$，并递归定义为 $A(f,c) = A(f-1, A(f,c-1))$，这等价于对 $A(f-1,\cdot)$ 进行 c 重嵌套。

参数 f 可以看作定义了一个函数族，而计数器 c 则是指向这个函数族的指针。每个序列 $A(f,\mathbb{N})$都是前一个序列 $A(f-1,\mathbb{N})$ 的子序列。增加 f 正是数值急剧增长的原因。

若想了解大数的精彩概览，甚至包括超出 $|\mathbb{N}| = \aleph_0$ 的大数，请参见 https://sites.google.com/site/largenumbers/home。

很容易得到 $A(1,c) = 2 \cdot c, A(2,c) = 2^c, A(3,c) = 2^{2^{\cdot^{\cdot^{\cdot^{2^2}}}}}$，其中有 c个 2 彼此堆叠。此外，$A(f,1) = 2, A(f,2) = 4, \forall f.$

$$A(4,3) = A(3,A(3,A(3,1))) = A(3,A(3,2)) = A(3,4) = 2^{2^{2^2}} = 65536.$$

$A(4,4) = A(3,A(4,3)) = A(3,65536) = 2^{2^{\cdot^{\cdot^{\cdot^{2^2}}}}}$，$65536-1$ 个指数。

因此，$A(4,5) = A(3,A(4,4))$ 将是一个由 2 组成的幂塔，其高度由一个由 65536 个 2 组成的幂塔来描述，“如此等等”。

$A(5,3) = A(4,A(5,2)) = A(4,4)$，如上所述，而 $A(5,4) = A(4,A(5,3)) = A(4,A(4,4))$ 是对 $A(3,\cdot)$ 进行 $A(4,4)$ 重迭代求值——我们不知道它的值，甚至没有办法想象这个数量。

Friedman [4, 第 106] 页称 $A(5,5)$ 是一个“难以想象的巨大数量”。我们无话可说。

1.2 艾伦·M·图灵，1936：“图灵机”

我们采用图灵的发明 [11]，并作如下修改：– 不设输出纸带或 F（图形）单元格 – 使用双向无限纸带，无需也不提供（左）端标记

存在一个有限状态集合 Q，其中 $n := |Q|$；对于 Rad´o 的原始问题，带字母表固定为 $E = \{0,1\}$；也可以采用更大的、但仍有限的字母表 E，其中 $m := |E|$。

相关的三个函数是Q^+: $Q \times E \to Q$，用于确定下一状态；E^+: $Q \times E \to E$，用于确定要写入纸带的新符号；以及D^+: $Q \times E \to \{R, L\}$，用于确定纸带读写头的移动方向。

1.3 拉多·蒂博尔，1962年："忙海狸"

首先介绍一些术语：

1. 忙海狸是任何 图灵机：从空白纸带开始运行，最终会停机。它停机时的状态，与其是否为忙海狸无关（参见 Rad´o [9], Michel [7, 第 4], 页，或格林 [6]）。广义形式允许使用比拉多原始版本 $B = \{0, 1\}$ 更多的符号集（带字母表）。

2. 拉多提出的忙海狸竞赛要求给出图灵机配置的 QED$^+$ 值，并提供一个声称的停机时间 T。如果该配置在空白纸带上运行恰好 T 步后停机，则该参赛项有效。

3. 在其类别 $(n, m) = (|Q|, |E|)$ 中，当前的忙海狸冠军是停机时间 T 最大的停机图灵机——截至目前为止。

4. 忙海狸函数 BB(n) = BB(n,2) 或 BB(n,m) 是 (n,m) 类中 BB 冠军机的运行时间 T——前提是通过某种方式（例如穷举搜索）证明该类中不存在其他可能在空白纸带上停机、且运行超过 T 步的机器。否则，当前冠军机的运行时间 T 只是 BB(n,m)的下界。

已知结果见图1，参见 Michel [7]。数字为精确值，阿克曼函数的值为下界（其中$A(2, 2^{16}) = A(3,4)$）。

7	7						
6	6	A(3,15)					
5	5	A(2,25)					
4	4	107	A(2,$2^{\lceil 15.5 \rceil}$)				
3	3	21	A(2,56)	A(2,$2^{\lceil 15.4 \rceil}$)			
2	2	4	38	A(2,21)	A(2,2^{11})	A(2,2^{15})	
1	1	1	1	1	1	1	1
n/m	1	2	3	4	5	6	7

图1：已知值或 BB(n,m)的下界

1.4 Harvey Friedman，2001年："长有限序列"

本文将证明，非平凡的（$|Q|, |E| \geq 2$）忙海狸竞赛至多有 37022 场，因为所有其他数对都会导致一个实现 Friedman 的组合问题"长有限序列"的停机配置，其步数下界为$A(7198,158386) < \mathtt{BB}(\mathtt{n},\mathtt{m})$。

Friedman 研究字母表为 k 的有限词，特别是 $k = 3$ 和 $B = \{1, 2, 3\}$。他的关键定义（$*$）如下：

> 当子词集合 $s^{(i)} = s_i s_{i-1+1} \dots s_{2i}$、$1 \leq i \leq n/2$，即 $s_1 s_2$、$s_2 s_3 s_4$、……、$(s_i \dots s_{2i})$ ……、$(s_{n/2} \cdots s_n)$，不包含两个满足 $i < j$ 的词，且 $s^{(i)}$ 是 $s^{(j)}$ 的子词时，词 $s = s_1 s_2 \dots s_n$（来自 B^n，长度为 n）满足性质（$*$）。
>
> 如果存在指标 $1 \leq \iota_1 < \iota_2 < \cdots < \iota_m \leq n$，使得 $a_k = b_{\iota_k}$，则称词 $a_1 \dots a_m$ 是 $b_1 \dots b_n$ 的子词。

对于每个 $k \in \mathbb{N}$，令 $n(k)$ 表示满足 $(*)$ 的 $\{1, \dots, k\}^*$ 中最长词的长度。容易验证 $n(1) = 3$。

Friedman 证明，$12221111111 \in \{1, 2\}^*$ 满足 $(*)$，但不存在更长的双字母词，因此 $n(2) = 11$。

顺便说一句，所有 $n(k)$ 都是奇数。最后一个奇数索引字母不会生成新的子词，因此 $(*)$ 的状态不会改变。

在 $n(1) = 3$ 和 $n(2) = 11$ 之后，我们迎来了一个大幅跃升：

定理 1. (Friedman[4,定理 4.7])

$$n(3) > A(7, 184).$$

证明。 参见 [4]中的定理 4.7。 □

本文将实现对 $B = \{1, 2, 3\}$ 上不满足 $(*)$ 的第一个词的搜索；这一搜索将需要超过 $n(3)$ 步——远远超过"难以想象的巨大数量" $A(5,5)$——然后才会停机。

Dougherty 甚至得出了以下极其巨大的上界：

定理 2. (Friedman[4,定理 6.9])

$$n(3) > A(7198, 158386).$$

证明。 见 [4]中的定理 6.9。 □

Friedman 进一步猜想 [5, p. 7] 上界

$$n(3) < A(A(5, 5), A(5, 5)).$$

2 算法与一个 8 符号实现

我们将从符号集 $E = \{Y, X, 1, 2, 3, -, \$, +\}$ 开始，其中 Y是空白符号。

纸带的活动部分分为 3 个片段：

I 两个一元计数器 $0 \le i \le imax = N/2$ 和 $0 \le l \le lmax \le N/2$。

II 形式为 $s_1 s_2 \dots s_N \in \{1, 2, 3\}^N$或某些符号被替换为对应的撇号符号的当前字 $s \in B^N$，其中 $- \equiv 1'$，$\$ \equiv 2', + \equiv 3'$。

III $N/2$ 个 s的副本，以 '+' 分隔，随后将它们修剪为$s^{(1)} =$(**$s_1 s_2$**) . . . , $s^{(i)} =$(**$s_i..s_{2i}$**) . . . , $s^{(N/2)} =$(**$s_{N/2}..s_N$**)。

区段边界由标记符 Y,Y,X,Y 给出，也就是说，整条纸带的结构如下

$$^{\omega}\mathrm{Y}\cdots\mathrm{I}\cdots\mathrm{Y}\cdots\mathrm{II}\cdots\mathrm{X}\cdots\mathrm{III}\cdots\mathrm{Y}^{\omega}$$

该标记序列 YYXY 的优点在于——将纸带的空白符号定义为 Y——两端都处于双向无限纸带的 $^{\omega}$ Y...Y$^{\omega}$ 中，因此在扩展片段时会自动保持正确。

该算法由 9 行组成，如图 2 所示。

Q 中的状态命名为"ql-c"，其中 $l \in \{1, \dots, 9\}$ 指算法的行，而 $c \in \mathbb{N}_0$ 只是该行内的一个计数器。

在第 8–12 页，我们描述实现该算法的 44 个状态。对于每个程序行，我们指出所处理的片段（I、II 或 III），并且

1. 将 s 从 II 复制到 III，共 N/2 次，每次之间用 $+$分隔。 2. 在 III 中，切除初始三角形 $\varepsilon, s_1, s_1 s_2, \dots, s_1 \cdots s_{i-1}, \dots, s_1 \cdots s_{N/2-1}$。 3. 在 III 中，切除双重尾随三角形 $s_3 \cdots s_N, s_5 \cdots s_N, \dots, s_{N-1} s_N, \varepsilon$，将 $s_1 s_2, s_2 s_3 s_4, \dots, s^{(i)}, \dots, s^{(N/2)}$ 留在纸带上作为 III。 4. 移除（$i - 1$）模式 $s^{(1)}, \dots, s^{(i-1)}$。 5. 检查 $s^{(i)}, \dots, s^{(N/2)}$是否存在子词匹配，性质（$*$）。 6. 将片段 III 清空为 X$\varepsilon$Y$^{\omega}$。如果在第 5 步匹配，则转到 7，否则转到 8。 7. $s++$；如果 $s =$ (**1**)$^{N+1}$ 停机，否则 $i := 0, l := 0$。转到 1。 8. $i := 0$; $lmax++$; 如果 $lmax < N/2$ 转到 1，否则转到 9。 9. $(N/2)++$; $s :=$ (**1**)^N; 转到 1。

图 2：算法：长有限序列

定义：

符号 – $			+	1	2	3
l	–	0	1	–	0	1
i	0	0	0	1	1	1

示例：

符号 – –	1	2	2	3	3	3	3			
l	–	–	–	0	0	1	1	1	1	
i	0	0	1	1	1	1	1	1	1	

图 3：计数器 l 和 i 及其编码

.

起始位置为 @I，I 的左侧端到 III@，III 的右侧端。

每个条目在第一行列出来自 $E = \{Y, X, -, \$, +, 1, 2, 3\}$ 的相关符号；星号“*”代表此前尚未提及的所有符号。像 $*\backslash 3$、X 这样的条目代表所有其他符号，其中 3 和 X 实际上并未使用（可想作 Q^+(q1-1,3)=Q^+(q1-1,X) = 错误）。省略的符号的处理方式与 $*\backslash 3$、X 中的 3、X 一样：它们不会与该状态同时出现。每个条目的第二行给出新符号 E^+ （“=”表示符号不变）以及磁带头的移动方向 D^+，向左（L）或向右（R）。第三行是下一状态 Q^+。

我们从第 3 行的 $i = l = 0$、I = II = III = ε 开始。

图灵机从状态 q1-4 开始，将一个 Y 初始化为 X，然后依次进入 q3-1、q4-1、q4-2、q5-0、q5-1、q6-2、q8-0、q8-1、q8-2、q9-1（纸带为空，除一个 X 外全为 Y）；在第 9 行，我们开始将支撑集从空集扩展到区段 I 中长度为 $N/2 = 1$ 的范围，并扩展到区段 II 中长度为 $s = -- \equiv 1'1'$ 的范围。

状态 q7-1 是结束状态，带有 Y 进入停机。

状态 q9-3 和 q9-5 分别合并到 q3-3 和 q1-9，以节省状态数。在 q9-3 和 q9-5 中，我们只处理符号“–”和“Y”，而这些符号在（原来的）q3-3 和 q1-9 中并不存在。

1、2、3 代表其自身；同样，–、$、+ 在区段 II 中分别代表 1’、2’、3’。

区段 I 中的两个计数器 l、i 按照图 3 进行编码。示例值为 $i = 7$，最大值为 $imax = 9$；$l = 4$ 的最大值为 $lmax = 6$ （0 可以改为 1，但 ’–’ 不能改动）。

通过移除标记 X，我们将 $|E|$ 降至 7。这要求进行双重扫描 $\cdots \to Y \to Y$，而不是像 $\ldots \xrightarrow{X} Y$ 或 $\ldots \xrightarrow{Y} X$ 那样越过另一个符号，并会影响状态 q1-C1、q1-C2、q1-C3、q1-6、q1-9、q2-1、q4-1、q4-3、q5-0 和 q9-4，共十个状态。复制这些状态后，实现 $|Q| = 54$、$|E| = 7$，用于 n（3）。

我们在第3节中通过以下参数说明这10种状态：Δ：对于 $X \in E$，$|E| = 2k + 2$；对于 $X \notin E$，$\Delta = 1$。

1. @I.将 s 从II复制到III，重复N/2次，每次之间以 +分隔。

q1-1	–	$,+	1,2	Y	
	1,R q1-2	2,R q1-2	=,R q1-1	=,R q2-1	
q1-2	Y	*\X			
	=,R q1-4	=,R q1-2			
q1-4	–	$	+	1,2,3	X
	1,R q1-C1	2,R q1-C2	3,R q1-C3	=,R q1-4	=,R q1-7
	[Y] X,R q3-1				
q1-C1	Y	*			
	1,L q1-6	=,R q1-C1			
q1-C2	Y	*			
	2,L q1-6	=,R q1-C2			
q1-C3	Y	*			
	3,L q1-6	=,R q1-C3			
q1-6	Y	*			
	=,R q1-4	=,L q1-6			
q1-7	Y	*\$,X			
	+,L q1-9	=,R q1-7			
q1-9	X	[Y,–]	*\$		
	X,L q1-10	X,L q1-10	=,L q1-9		
q1-10	1	2	3	Y	*\$
	–,L q1-10	$,L q1-10	+,L q1-10	=,L q1-11	=,L q1-10
q1-11	Y	*			
	=,R q1-1	=,L q1-11			

我们以一元编码递增 $i \leq N/2$
+,2：已递增
–,1：i++。Y：数量 $i = N/2$结束
找出 I 的右侧
在 i++ 之后

从 h = –,$,+≡1',2',3' 中去掉撇号
并转到相应状态 q1-Ch
X: s_i 的结尾
Y：INIT 的部分，不是 q1-4

找到 III 的右侧并放入符号 1

.....符号 2

... 符号 3

查找 II 的左侧 $= s_1 s_2 \ldots$

查找 III 的右侧并放置
词间空隙 +

查找 II 的右侧
Y,–：从 q9-5 合并

将 1、2、3 标记为 –、$、+≡1'、2'、3'

查找 I 的左侧

2. @II. 从 III 中切去左侧三角形。

q2-1	Y	*		找出 III 的右侧
	=,L q2-2	=,R q2-1		
q2-2	X	+	*\Y	将最后一个词间空隙 + 改为 $
	=,R q3-1	$,R q2-3	=,L q2-2	
q2-3	1,2,3	Y	–	清除一个符号，直到 –
	–,R q2-4	=,L q2-2	=,R q2-3	
q2-4	–	Y	*\+,X	找到右侧下一个未完成三角形
	–,R q2-3	=,L q2-2	=,R q2-4	Y：右侧所有三角形均已完成

3. @III.在 III 中切去双直角三角形。

q3-1	$	Y	*\X	找到第一个词间空隙 $
	+,R q3-5	=,L q4-1	=,R q3-1	从左侧开始。更改为 + 从这里开始处理，直到 III 的右侧
q3-5	Y	–	+	若 $\$\rightarrow +$ 位于 q3-1 中，则为必要条件
	=,L q4-1	=,L q3-5	=,L q3-2	正好位于 Y 标记的左侧
q3-2	1,2,3	*\XY$		跳过中间状态 –，+ 然后将一个符号清除为 –
	–,L q3-3	=,L q3-2		
q3-3	1,2,3	[Y]	[–]	将另一个符号清除为 –
	–,L q3-4	=,R q9-4	=,R q3-3	Y,–：从 q9-3 合并 （+、X、$ 不会出现）
q3-4	–,+	X	*\Y、$	在下一个词之前向左移动
	=,L q3-2	=,R q3-1	=,L q3-4	

q2 与 q3 之间的对称性：$Y \leftrightarrow X$、$R \leftrightarrow L$、q2-2≡q3-1、q2-3≡（q3-2、q3-3）、q2-4≡q3-4。

4. III@. 删除 III 中的（$i-1$）模式。

q4-1	Y	*			查找 I 的右端
	=,L q4-2	=,L q4-1			
q4-2	$	2	1,3,–,+	Y\X	将 I 中的 $l:=1,\ldots,i$（一元编码）递增
	+,R q4-3	3,R q4-3	=,L q4-2	=,R q5-0	If Y, 完成： $l==i==lmax$
q4-3	X	*			查找 III 的左端
	=,R q4-4	=,R q4-3			
q4-4	+	*\X,Y,$			将所有 1、2、3 清除为 –，直到.....
	–,L q4-1	–,R q4-4			+：模式结束

5. III，从 I@ 开始。检查 $s^{(i)} \subset s^{(j)}$、 $j>i$ 的模式。

q5-0	X	*			找到 III 的左端末端
	=,R q5-1	=,R q5-0			
q5-1	1	2	3		取出并删除
	–,R q5-V1	–,R q5-V2	–,R q5-V3		$s^{(i)}$ 中最左边的符号
	–	+	Y		
	=,R q5-1	+,R q6-1	=,L q6-2		+：这个 i已完成 Y：相同， $i==N/2$
q5-**Vh**	–,+	Y	*		跳过其余的 $s^{(i)}$ 或 $s^{(j)}$
	=,R q5-**Kh**	=,L q5-2	=,R q5-**Vh**		h $\in\{1,..,k\}$ 是符号
q5-K1	1	+	Y	*	清除所有 $s^{(j)}$ 中不匹配的项
	$,R q5-V1	=,R q5-K1	=,L q5-2	–,R q5-K1	直到匹配 1，将其设为 $ ，
q5-K2	2	+	Y	*	然后跳到 q5-Vh 中的下一个词 清除其他所有项，直到匹配 2
	$,R q5-V2	=,R q5-K2	=,L q5-2	–,R q5-K2	
q5-K3	3	+	Y	*	清除其他所有项，直到匹配 3
	$,R q5-V3	=,R q5-K3	=,L q5-2	–,R q5-K3	
q5-2	X	*			转到 III 的左端
	=,R q5-1	=,L q5-2			

6. 在 III 中。将 III 移除至 εY^{ω}。检查是否匹配 $s^{(i)} \subset s^{(j)}$。

q6-1	Y	*	查找 III 的右端
	=,L q6-2	=,R q6-1	

q6-2	$	X	*	$: 匹配! X: III == εY^{ω} , 不匹配 *: 用空白填充 (Y), 不匹配
	Y,L q6-3	=,L q8-0	Y,L q6-2	
q6-3	X	*		X: III == εY^{ω}, 匹配! *: 用空白填充 (Y) 与 q6-2 相同, 但匹配!
	=,L q7-1	Y,L q6-3		

$ 是来自 $s^{(i)}$ 的最后一个符号, 与 $s^{(j)}$ 中的某个符号匹配 ,
这意味着来自 $s^{(i)}$ 的所有先前符号也都已匹配。

---

7. II@. $s++$, 如果 $s = 3^N$ 则停机, 否则 $i := 0$; $j := 0$。

q7-1	–	$	+	Y	$s++$, 1'→2'→3'→(1'+进位) 达到 n_3 后, HALT 才是真正的停机! 将 $ 替换为(–,L,q7-1), 适用于 $\|B\| = 2$
	$,L q7-3	+,L q7-3	–,L q7-1	=,R HALT	

q7-3	Y	*	查找 II 的左侧
	=,L q7-4	=,L q7-3	
q7-4	Y	*\–	i: = 0; j: = 0。转到 I 的左侧
	=,R q1-1	–,L q7-4	

---

8. @III. $i := 0, lmax++$. 如果 $lmax < N/2$ 转到 1, 否则转到 9。

q8-0	Y	*	查找 I 的右侧
	=,L q8-1	=,L q8-0	

q8-1	–, $, +	1	2	3	Y\X	$i := 0$, l 不变
	=,L q8-1	–,L q8-1	$,L q8-1	+,L q8-1	=,R q8-2	

q8-2	–	Y	*	– → $: $lmax$++ （l的支撑集）, 转到 1 Y: 由于 $lmax$==N/2, 转到 9
	$,L q1-11	=,L q9-1	=,R q8-2	

9. I@. ($N/2 == imax$) $++$, sXY→s–XY, $i := 0, l := 0$.

q9-1	Y	*\X
	–,R q3-3	–,L q9-1
[q9-3]	Y	–
	=,R q9-4	=,R q9-3
q9-4	Y	*
	–,R q1-9	–,R q9-4
[q9-5]	Y	
	X,L q1-10	

求出 I 的左侧。扩展 I，从而
自然数集/2，并新增一个 –。 $i := 0, l := 0$
Y：结束，转到 q9-3，合并至 q3-3
求出 I 的右侧
合并至 q3-3

求 III 的右侧，用 XY→ −−扩展
Y：结束，转到 q9-5，并合并到 q1-9
用 q1-9，以 $- \equiv 1'$扩展
将 YY^{ω} 设为 XY^{ω} (III==ε)
被合并到 q1-9

3 将 $|E|$ 变为 $|E'| = 2$

现在我们有了一个 $|Q \times E| = 44 \cdot 8$ 实现（以及一个 $54 \cdot 7$ 实现）　　.

接下来，为了将符号数降至 $|E'| = 2$ 或 3，我们可以将最多 8 个符号映射为比特三元组，并将最多 9 个符号映射为三进制“位”对。

一般情况具有以下相关参数：新的字母表大小为 $b = |E'|$，通常为 2 或 3；用于模拟来自 E的一个原始符号的 b进制数字 l元组的长度或大小为 $l = \lceil \log_b |E| \rceil$； k 来自我们的问题 $n(k)$，通常为 $k = 3$ 或 4。

每个状态最多通过对该 l元组进行 4 次扫描来模拟：

扫描-0：如果一个状态交替地从两侧进入，扫描-0 会使用$l-1$ 个子状态，将读写头从“另一端”移到通常的一端。如果一个状态始终从同一侧进入，则跳过扫描-0。

Sweep-1 沿着 l 元组移动，以获取当前符号 e；各个连续位置上的状态数依次为 1、b、b^2、.....、b^{l-1}。

对于“扫描状态”，我们搜索一个符号，并将所有其他符号合并到通配符情形“*”中，其数量至多为 $2l-1$ 个子状态，因为在每个位置（第一个位置之后），我们只需区分“该扫描符号仍有可能出现”和“出现了其他某个符号”。

Sweep-2 向回移动，将 e 替换为 $E^+(q, e) \neq e$，或者只是到达另一端以离开 l 元组。每种情形使用 $l-1$ 个子状态。

若我们必须将 e 替换为 $E^+(q, e)$，但让其保持在与进入侧相反的一侧，Sweep-3 会再次向前移动，每种情形移动 $l-1$ 个子状态。

状态	情形	$>$ $=$	$<$ $=$	$<$ $\neq$	$>$ $\neq$	状态	情形	$>$ $=$	$<$ $=$	$<$ $\neq$	$>$ $\neq$
Scan$>$	$6 \cdot 2$	12	-	-	-	q4-2	4	1	1	2	-
Scan$<$	(6+k) $\cdot$ 2	6+k	6+k	-	-	q4-4	2	-	-	1	1
q1-1	4	2	-	-	2	q5-1	k+3	1	1	-	1+k
q1-4	3+k	2	-	-	1+k	q5-Vh	3k	2k	k	-	-
q1-Ch	2k	k	-	k	-	q5-Kh	4k	k	k	-	2k
q1-9	2	-	1	1	-	q6-2	3	1	1	1	-
q1-10	2+k	2	-	-	k	q6-3	2	1	-	-	1
q2-2	3	1	1	1	-	q7-1	1+k	1	-	-	k
q2-3	3	1	1	-	1	q7-4	2	-	1	-	1
q2-4	3	1	1	-	1	q8-1	5	1	1	-	3
q3-1	3	1	1	-	1	q8-2	3	1	1	1	-
q3-5	3	-	3	-	-	q9-1	2	-	-	1	1
q3-2	2	1	-	-	1	q9-4	2	-	-	-	2
q3-3	3	2	-	1	-						
q3-4	3	2	1	-	-						

	状态	情形	$>=$	$<=$	$<\neq$	$>\neq$	
$\sum$	35+3k	63+13k	22+4k	15+2k	9+k	17+6k	$\Delta = 0$
$\sum$	7+ k	14+2k	8+k	3	1+k	2	$\Delta = 1$: 额外情况

图4：状态的详情

我们的 $35 + 3k$ 个状态各有 $v = 2$ 到 $v = 6$ 种不同情形。其中 $\#_{Sweep-0} = 3 + \Delta$ 个（q1-9、q3-3、q3-5）是从两侧进入的；q1-9 对应 $\Delta = 1$ 重复出现。

$\#_{Scan>} = 6 + \Delta \cdot 3$ 状态（q1-2、q4-1、q4-3、q5-0、q7-3、q8-0）是扫描状态，它们位于进入侧的相反一侧，且不会改变符号。因此，只需使用具有 (1, 2, 2, 2, . . .)、即 $2l - 1$ 个子状态的扫描-1。

另外，匹配时留在进入端的 $\#_{Scan<} = 6 + k + \Delta(2 + k)$ 个状态（q1-Ch,q1-6,q1-7,q1-11,q2- 1,q5-2,q6-1）是扫描状态，无论是否将 e 改为 E^+ 都如此。这里的 Sweep-1 与之前相同，而匹配情况中的 Sweep-2 有 $l - 1$ 个子状态。

接下来有 $\#_{Scan\neq} = 5 + \Delta$ 个状态（q4-4、q6-3、q7-4、q9-1、q9-4）会在扫描-1中扫描，但之后会改变 e 为 $E^+(q, e) \neq e$。这里，扫描-1有$2l - 1$ 个子状态，而扫描-2和扫描-3则按一般情况计数。

$\|E\| = ...$				片段中的含义			
8+	2	3	4	I	II	III	
Y	111	22	33	标记	标记	标记	也为空白
X	011	21	22	–	标记	标记	
1	100	10	10	$i = 1, l = -$	1	1	
2	101	11	11	$i = 1, l = 0$	2	2	
3	110	12	12	$i = 1, l = 1$	3	3	
–	000	00	00	$i = 0, l = -$	1’	nil	
$	001	01	01	$i = 0, l = 0$	2’	匹配	
+	010	02	02	$i = 0, l = 1$	3’	词尾	
4	–	–	13	–	4	4	
4’	–	–	03	–	4’	—	

图 5：不同字母表中的符号概览

其余 $\#_{Sweep1} = 18 + 2k + \Delta \cdot 1$ 个状态在扫描-1中需要 $1 + b + \cdots + b^{l-1} =$（$b^l - 1$）/（$b - 1$）个子状态。它们以及 $\#_{Scan\neq}$ 个状态列于表4。

根据这个上界，子状态的总计数量为$[\mathtt{Sweep} - 0]$ $(l-1) \cdot \#_{Sweep-0}+$

$[\mathtt{Sweep} - 1]$ $(2l-1) \cdot (\#_{Scan>} + \#_{Scan\neq}) + (3l-2) \cdot \#_{Scan<}$
$+(1 + b + b^2 + \cdots + b^{l-1}) \cdot \#_{Sweep-1}+$
$[\mathtt{Sweep} - 2/3]$ $(l-1) \cdot (0 \cdot v_{>=} + 1 \cdot (v_{<=} + v_{<\neq}) + 2 \cdot v_{>\neq})$

注意图4中因子 k 的双重用途：在 q1-4、q1-10、15-1 中，我们必须区分 k 个符号对应的 k 种不同情形。在 q1-Ch、q5-Vh、q5-Kh 中，有 k 个不同状态，每个状态对应固定数量的情形。

对于 $b = 2$, $l = 3$ 我们有
$2 \cdot 3 + 5 \cdot (6+6) + 7 \cdot (6+k) + (1+2+4) \cdot (17+2k) + 2 \cdot (2k+15+k+9+2 \cdot (6k+17)) = 343 + 51k$ 子状态。对于 $k = 3$，这就是 496 个子状态，可与“手工布线”版本实际的 276 个状态相比较。

Sweep	$\Delta = 0$, $\|E\| = 2k + 2$	$\Delta = 1$, $\|E\| = 2k + 1$
扫描-0	$3 \cdot (l-1)$	$4 \cdot (l-1)$
扫描-1	$(17 + k) \cdot (2l-1)$	$(23 + 2k) \cdot (2l-1)$
	$+(18 + 2k) \cdot (1 + b + \cdots + b^{l-1})$	$+(19 + 2k) \cdot (1 + b + \cdots + b^{l-1})$
扫描-2/3	$(6 + k + 58 + 15k) \cdot (l-1)$	$(8 + 2k + 66 + 16k) \cdot (l-1)$

图6：每次扫描的子状态数量

对于 $b=3$， $l=2$ 我们有

$1\cdot 3+3\cdot(6+6)+4\cdot(6+k)+(1+3+9)\cdot(17+2k)+1\cdot(2k+15+k+9+2\cdot(6k+17))$

$= 351+45k$ 个子状态，对于 $k=3$ 而言为 486 个，而针对三进制位对 $(l=2, b=3)$ 的“手工连线”版本只需 155 个状态。

对于 $b=2$ 和 $l=4$，我们有

$3\cdot 3+7\cdot(6+6)+10\cdot(6+k)+(1+2+4+8)\cdot(17+2k)+3\cdot(2k+15+k+9+2\cdot(6k+17))= 582+85k$。

取 $k=4$，我们得到一个适用于 n(4) 的 (922, 2) 实现。

定理 3。 (i) 给定任意一个具有 $|Q|=n$， $|E|=m$ 的图灵机，我们可以得到另一个具有 $|E'|=b, l:=\lceil$对数$_b|E|\rceil$ 且状态数至多为 $|Q'|\leq$ 的图灵机

$$n\cdot[(l-1)+(1+b+b^2+\cdots+b^{l-1})+|E|\cdot 2(l-1)]<2(m+1)n\lceil\log_b(m)\rceil.$$

(ii) 按照前文的定义取 n, m, b, l，并令数量 $\#_{Sweep-0}$, $\#_{Scan}$, $\#_{>=}$, $\#_{>\neq}$, $\#_{<=}$, $\#_{<\neq}$ 如文中所示，我们得到更紧的上界$|Q'|\leq \#_{Sweep-0}\cdot(l-1)+\#_{scan}\cdot(2l-1)+(n-\#_{scan})(1+b+\cdots+b^{l-1})+(\#_{<=}+\#_{<\neq}+2\#_{>\neq})\cdot(l-1)$
$< n\left(\frac{b^l-1}{b-1}+l-1\right)+\#_{Sweep-0}\cdot(l-1)-\#_{Scan}\left(\frac{b^l-1}{b-1}-2l+1\right)$。

证明。 (i) 我们使用 $l-1<\log_b(m)$ $1+b+\cdots+b^{l-1}=\frac{b^l-1}{b-1}<2m$。
(ii) 我们使用 $(\#_{<=}+\#_{<\neq}+\mathbf{1}\cdot\#_{>\neq})\leq n$。 □

定理 3 中的上界与映射 $E\to E'$ 无关，这或许能改善数量。

4 将 $|Q|$ 降至 $|Q'|=2$

4.1 $|Q'|=3$

通过模拟将状态数降至 3。

我们只使用三个状态：qX（扩展）、qL（向左移动）和 qR（向右移动），因此 $|Q'|=3$。新符号集为 $E'=Q\times\{X,L,R\}\times E\ni$（$q$、$d$、$e$），我们通过 qL 或 qR 逐位传递新状态，从而依次生成 $e'=$（q、e、d）的 q 部分。

广义方向 d 表示状态是否仍需扩展， $d=X$；或者方向与原始图灵机中的方向相同，即 $d\in\{L,R\}$。 e 部分是原始符号。

	q	e		Q^+	E^+	D^+	
(1)	qX	$[q,d,e]$	$\to$	qd	$[q-1,d,e]$	d	$q\in\mathbb{N}$
(2)	qL	$[q,X,e]$	$\to$	qX	$[q+1,X,e]$	R	
(3)	qR	$[q,X,e]$	$\to$	qX	$[q+1,X,e]$	L	
(4)	qX	$[0,d,e]$	$\to$	qX	$[0,X,e]$	d	$d\in L,\quad R$
(5)	qX	$[q,X,e]$	$\to$	$q\tilde{d}$	$[\tilde{q}-1,^{\tilde{d},\ \tilde{e}]}$	$\tilde{d}$	

图 7: $|Q'|=3$的转移

.

扩展时，当前符号 e 中 e 的部分会被替换为 $e'=Q^+(q,e)$。组装 q 期间，方向为 $d=\mathrm{X}$，随后变为 $d=D^+(q,e)$。

我们首先使用图 7 中的转移 (1)，其中 $d\in\{L,R\}$ 也就是说，对于 $d=L$，我们以 qL 向左移动；对于 $d=R$，我们以 qR 向右移动。然后，我们使用转移 (2) 或 (3) 从相邻单元格返回，此时该单元格的 q 递增。我们重复此过程，直到 $q=0$ 与 qX，然后通过 (4) 释放当前单元格：$d:=X$，从而完成对当前单元格的处理。

通过 (4) 移至相邻单元格，并将其作为新的当前单元格，同时保持状态 qX。在这里，我们通过转移 (5) 扩展符号（q，e，X）：首先，我们使用原始机器的函数计算新值 $\tilde{q}:=Q^+(q,e)$ $\tilde{e}:=E^+(q,e)$，以及 $d:=D^+(q,e)$。中间结果将 为（qX，$[\tilde{q},d,\tilde{e}]$，d）。不过，我们立即纳入第一步移动，使 $\tilde{q}$递减，并得到总值，开始沿新的方向 $\tilde{q}$对$\tilde{q}\in\mathbb{N}$ 进行逐位传输。

两点说明：纸带的空白符号是 $(0,X,-)$，其中 $-$ 是 E中的原始空白符号；原始机器停机时，Q^+=停机，模拟器实际上也会停机。

定理 4。给定任意一台具有 $|Q|=n,|E|=m$ 的图灵机，我们可以得到另一台具有 $|Q'|=3$ 且至多有

$$|E'|\le 3(n+1)\cdot m$$

个状态。

证明。根据图 7，我们有一个可能的 $Q':=\{qX,qL,qR\}$ 和$E':=(Q\cup\{0\})\times\{L,R,X\}\times E$，其大小如给定值。 □

q	e	Q^+	E^+	D^+(1)
$qD_{\overline{q_a}}$	$[q_{a-1}\dots q_1q_0,-,e]$	qX	$[q_aq_{a-1}\dots q_1q_0,-,e]$	$\overline{D}$(2)
qX	$[q_{n-1}q_{n-2}\dots q_a,D,e]$	qD_{q_a}	$[q_{n-1}q_{n-2}\dots q_{a+1},D,e]$	D(3)
qX	$[q_{n-1}q_{n-2}\dots q_1q_1,-,e]$	$qD'_{q'_{n-1}}$	$[q'_{n-2}\dots q'_1q'_0,D',e']$	D'，证毕

QED$^+$($q_n-1\dots q_0,e$) =($q'_{n-1}\dots q'_0,e',D'$)，在原始图灵机中

图 8： $|Q|=2b+1$的转移

4.2 $|Q'|=2b+1,\quad b\geq 2$

对于 $|Q'|=3$，我们使用了一个只含一个字母的字母表来表示“信息传输”，即每一步要传递 log(1) $=0$ 个信息比特。

正如 Chaitin [2] 指出的，关于传输长度或传递终点的信息，是必不可少且重要的一部分。在这里，它是唯一的信息。

不过，我们可以使用含有 b 个字母的更大字母表，以及 $Q'=\{qX,qL_0,qL_1$、.....、qL_{b-1}、qR_0、$\dots qR_{b-1}\}$，从而 $|Q'|=2b+1$。

因此，我们的符号集为:

定义 5. 符号集 E' 用于 $|Q'|=2b+1$

设 E':=$\{-,(q'_{l-1}q_{l-2}\dots q_1)_b,\dots(q'_{l-1}q_{l-2})_b,(q'_{l-1})_b\}\}\times E\times\{L,R\}\dot\cup(\{-\}\dot\cup\{($ $q'_{l-1}q_{l-2}\dots q_1q_0)_b\}\dot\cup[b]^{l-1}\dot\cup[b]^{l-2}\dot\cup\dots\dot\cup[b])\times\{-\}\times E$其中 $l:=\lceil\log_b|Q|\rceil,(q'_{l-1}q_{l-2}\dots q_0)$ $_b\leq|Q|$ 且 $[b]:=\{0,1,\dots,b-1\}$。

第一部分中的数量描述了 $|Q|/b,|Q|/b^2,\dots|Q|/b^{l-1}$ 前缀，第二部分中的数量描述了来自$Q\equiv\{1,\dots,|Q|\}$的元素的 $|Q|,b^{l-1},\dots,b$ 后缀。

We have $E'\leq(1+\lceil n/b\rceil+\lceil n/b^2\rceil+\dots+\lceil n/b^{l-1}\rceil)\cdot 2m+nm+\frac{b^l-1}{b-1}\cdot m$ $<\left[n\cdot\frac{b+1}{b-1}+2(l-1)+\frac{b^l-1}{b-1}\right]\cdot m$, using $1+\frac{2}{b-1}\cdot\frac{b^{l-1}-1}{b^{l-1}}<\frac{b+1}{b-1}$.

状态 X 对应于位于当前纸带单元格中，状态$L_i,R_i,0\leq i\leq b-1$ 分别用于左侧或右侧相邻单元格。

设当前状态和符号分别为 qX 和 $[(q_fq_{f-1}\dots q_1q_0)\ L,e],f\geq 1$。然后 $D^+=L$ 取自符号的最后一个分量。我们截去一个 b 进制数字 q_f，它进入下一个状态 $Q^+=qL_{q_f}$，从而得到 E^+ =($q_{f-1}\dots q_1q_0,e,L$)，即图 8 中的转移 (2)。

现在，当前状态为 qL_i，此时已向左移动。该处的符号为 $[(q_{l-1}q_{l-2}\dots q_{f+1})\ -,e]$。于是 $Q^+=qX,D^+=R$，即与 qL_i 相反的方

	位置 q		e	→	Q^+	E^+	D^+	
	102:	qX	[021,−,e_1]	→	qL_1	[20,L, e_2]	L	其中 $(120)_3 = 15 = Q^+$
	101:	qL_1	[−, −, e_3]	→	qX	[1,−,e_3]	R	
	102:	qX	[20,L, e_2]	→	qL_2	[0,L, e_2]	L	
	101:	qL_2	[1,−,e_3]	→	qX	[12,−,e_3]	R	
	102:	qX	[0,L, e_2]	→	qL_0	[−, L, e_2]	L	
	101:	qL_0	[12,−,e_3]	→	qX	[120,−,e_3]	R	$q = 15 = (120)_3$ 有已完全到达
	102:	qX	[−, L, e_2]	→	qX	[−, −, e_2]	L	清理，注释 $Q^+ = X$
(*)	101:	qX	[120,−,e_3]	→	qR_1	[01,R,e_5]	R	$QED^+(15, e_3)$
	102:	qR_1	[−,−,e_2]	→	qX	[1,−,e_2]	L	
	101:	qX	[01,R,e_5]	→	qR_1	[0,R,e_5]	R	
	102:	qR_1	[1,e_2,−]	→	qX	[11,−,e_2]	L	
	101:	qX	[0,R,e_5]	→	qR_0	[−,R,e_5]	R	
	102:	qR_0	[11,−,e_2]	→	qX	[011,−,e_2]	L	
	101:	qX	[−, R, e_5]	→	qX	[−,−,e_5]	R	
(**)	102:	qX	[011,−,e_2]	→	qD_i	[q',D,e_x]	D	$QED^+(4, e_2)$, D $\in \{L, R\}$

图9： $b = 3, |Q| = 7$的示例

向 $q = L_i$。此外，令 $E^+ = [(q_{l-1}q_{l-2}\dots q_{f+1}i)\ -, e]$，将作为索引 i 传递的 b 进制数字追加到当前第一个分量的右侧，在那里变为 q_f，如图 8 中转移 (1) 所示。

示例 （见图9）

我们运行原始机器的两条转移 $QED^+(7, e_1) = (15, e_2, L)$ 和 $QED^+(15, e_3) = (4, e_5, R)$，其中 7、15、$4 \in Q$ 以及 $e_1, e_2, e_3, e_5 \in E$。

我们假设位于例如 位置102，位置101上存有符号 e_3，位置103上存有 e_4，因此模拟器的

其纸带上有 101: (−, −, e_3)、102: (021, −, e_1)、103: (−, −, e_4)，其中 $(021)_3 = 7$ 是当前状态。

在图9的 (∗) 中，我们使用图 8 的转移 (3)，并根据原始转移 $(15, e_3) \to (4, e_5, R)$ 进行展开。

新状态 $(011)_3$ 被拆分为符号中的 (01) 和作为 $q = R_1$ 索引的末尾 1。

在 (∗∗) 中也会发生同样的情况，此时 $QED^+(4, e_2) = (q'|i, e_x, d)$ 定义了新的部分 $(3 \cdot q' + i) \in Q, e_x \in E, d \in \{L, R\}$。例如，对于 $QED^+(4, e_2) = (8, e_6, L)$，其中 $8 = (022)_3$，，我们有 $q' = 02, i = 2, e_x = e_6, D = L$，因此右侧为 $(L_2, [02, L, e_6], L)$。

	q	e	$\to$	Q^+	E^+	D^+
(1)	$X,$	$[0,-,e]$	$\to$	$\overline{X},$	$[1,X_{new},e],$	$\overline{X}$
(2)	$X,$	$[n,X_{new},e]$	$\to$	$\overline{X},$	$[n+1,X_{new},e],$	$\overline{X}$
(3)	$X,$	$[n,\overline{X}_{old},e]$	$\to$	$\overline{X},$	$[n-1,\overline{X}_{old},e],$	$\overline{X}$
(4)	$X,$	$[0,\overline{X}_{old},e]$	$\to$	$X,$	$[0,-,e],$	$\overline{X}$
(5)	$X,$	$[n,\overline{X}_{new},e]$	$\to$	$X',$	$[n'-1,X'_{old},e],$	X' $(*)$

$(*)$ with $(n,e)\mapsto(n',e',X')$ in the original machine

图 10: $X\in Q'=\{L,R\}$, $|Q'|=2$ 的转移

.

定理 6。给定任意一个具有 $|Q|=n,|E|=m$ 的图灵机，存在另一个具有 $|Q'|=2b+1$ 个状态且至多

$$|E'|\le\left(n\cdot\frac{b+1}{b-1}+2(l-1)+\frac{b^l-1}{b-1}\right)\cdot m$$

符号，其中 $l:=\lceil$对数$_b$（n）$\rceil$。

证明。令 $Q'=\{X,L_1,\ldots,L_{b-1},R_1,\ldots,R_{b-1}\}$, E' 如定义 5 所示，并令 QED$^+$ 如图 8 所示。 □

4.3 $|Q'|=2$，一个初始非空白符号

我们使用 $Q'=\{L,R\}$, E':$=\{0,1,2,\ldots,|Q|\}\times\{-,L_{new},L_{old},R_{new},R_{old}\}\times E$，以及图 10 中的转移

这里，状态 L（分别为 R）表示位于 两个活动单元格中的左侧（右侧）。如果我们向右转移（对于 $D^+=R$），则 L 中的状态在 R_{old} 中递减其符号的 q 部分（转移 (3)，其中 $X=L,\overline{X}=R$）；而 R_{new} 中的 R 状态则递增该部分，这是转移 (2)，其中 $X=R,\overline{X}=L$。

我们从状态 L 开始，位于非空白符号 $(1,R_{new},e_0)$ 上，其中 1 表示初始状态，而纸带其余部分均以 $(0,-,e_0)$ 或任何其他初始内容 $(0,-,e_n)$ 填为空白。我们立即根据原始图灵机的 (q_1,e_0) 执行一次 QED$^+$、转移 (5)。

4.4 $|Q'| = 2$，从空白纸带 $^{\omega}0^{\omega}$ 开始：通过“溢出错误”引入第三种选择

显然，当看到一个空白符号 $[0, -, e_0]$ 时（而且 BB 规则要求初始时仅有空白），我们必须区分 3 种情况：

– 我们位于当前位置左侧，激活此单元格，并通过图 10 中类型 (1) 的转移，将 $X = L$ 计数器从空白值 0 增加到 1，然后返回右侧

– 我们位于当前位置右侧，并采取对称的行为——在开始时，我们必须将那个空白转换为 $(q_1, -, 0)$ 以启动计算，因为纸带上还没有状态。

当只有 $|Q'| = 2$ 个状态且初始时只有一个符号（空白）时，我们显然也无法区分三种情形。不过，如果允许有一个非空白纸带单元格，我们就能顺利做到，见图 10。

正如我们所见，三个状态，或两个状态加一个非空白单元格，就足以启动模拟。然而，我们的任务是从空白纸带和 $Q = \{L, R\}$开始。那么实际会发生什么：

$(L, [0, -, e_0]) \to (R, [1, L_{new}, e_0], R)$ $(R, [0, -, e_0]) \to (L, [1, R_{new}, e_0], L)$ $(L, [1, L_{new}, e_0]) \to (R, [2, L_{new}, e_0], R)$ $(R, [1, R_{new}, e_0]) \to (L, [2, R_{new}, e_0], L)$ $(L, [2, L_{new}, e_0]) \to (R, [3, L_{new}, e_0], R)$ $(R, [2, R_{new}, e_0]) \to (L, [3, R_{new}, e_0], L)$ .....以此类推，直到无穷。

实际上，E' 中并不存在符号 [q,...]，因此对于 $q > |Q|$，这就产生了第三种选择：

设 $X := D^+(q_1, 0)$ 是原始图灵机的第一步移动。然后我们从 X 开始，并设定以下等价关系：$X, [|Q|\underline{+}1, X_{new}, 0] :\equiv X, [q_1, \overline{X}_{new}, 0]$ (自举启动) 和 $\overline{X}, [|Q| + 1, \overline{X}_{new}, 0] :\equiv X, [0, -, 0]$ (空白)。

这为任何其他使用空白纸带的图灵机提供了一个 2 状态空初始纸带模拟。

示例（见图 11）

令原始机器从空白纸带上的位置 1 开始，并执行 $(q_1, 0) \to (q_2, 1, R)$ $(q_2, 0) \to (q_3, 2, L)$ $(q_3, 1) \to (q_4, 3, L)$ $(q_4, 0) \to$ (停机)。

我们从一条全为空白的 $(0, -, e_0)$ 纸带开始，处于来自原始机器的状态 $R = D^+(q_1, 0)$。我们假设 $|Q| + 1 = 5$。详情见图 11/12。

定理 7。(i) 一台具有 $|Q| = n$，$|E| = m$ 且使用空初始纸带的图灵机，可以由另一台具有 $n' = 2$、$Q' := \{L, R\}$ 和 $m' = 5m(n + 2)$、$E' := \{0, 1, 2, \ldots, |Q|, |Q| + 1\} \times \{-, L_{new}, L_{old}, R_{new}, R_{old}\} \times E$ 的图灵机模拟。

轨迹。位置。q e Q^+ E^+ D^+(1) 1：
R $[0,-,0]) \to$ L $[1,R_{new},0]$ L(1) 0：L $[0,-,0])$
$\to$ R $[1,L_{new},0]$ R(1) 1：R $[1,R_{new},0])$
$\to$ L $[2,R_{new},0]$ L(1) 0：L $[1,L_{new},0])$
$\to$ R $[2,L_{new},0]$ R.....（直到）.....1：R $[5,R_{new},0])$ (5)
$\equiv$ R $[1,L_{new},0]$ $\to$ L $[2-1,L_{old},1]$ L0：L $[5,L_{new},0])$
(1) $\equiv$ L $[0,-,0]$ $\to$ R $[1,L_{new},0]$ R(3) 1：
R $[1,L_{old},1]) \to$ L $[0,L_{old},1]$ L(2) 0：
L $[1,L_{new},0]$ $\to$ R $[2,L_{new},0]$ R(4) 1：R $[0,L_{old},1]) \to$
右 $[0,-,1]$ L(5) 0：R $[2,L_{new},0]) \to$ R $[3-1,R_{old},2)$ R，
QED^+(1) 1：R $[0,-,1]) \to$ L $[1,R_{new},1]$ L(3) 0：
L $[2,R_{old},2]) \to$ R $[1,R_{old},2]$ R(2) 1：R $[1,R_{new},1])$
$\to$ L $[2,R_{new},1]$ L(3) 0：L $[1,R_{old},2])$
$\to$ R $[0,R_{old},2]$ R(2) 1：R $[2,R_{new},1])$
$\to$ L $[3,R_{new},1]$ L(4) 0：L $[0,R_{old},2]) \to$左
$[0,-,2]$ R(5) 1：L $[3,R_{new},1]) \to$ R $[4-1,R_{old},3)$ R，
QED^+(1) 2：R $[0,-,0]) \to$ L $[1,R_{new},0]$ L(3) 1：
L $[3,R_{old},3]) \to$ R $[2,R_{old},3]$ R

图 11：两个状态，空白纸带（第 1 部分）

(ii) 如果模拟图灵机不要求空初始纸带，我们有$|E'| = 5m(n+1)$，省略符号 $[|Q|+1,...,...]$。

证明。 模拟图灵机如图 10 所示。 □

Tr.	Pos.	q	e		Q^+	E^+	D^+	
(2)	2 :	R	$[1, R_{new}, 0])$	$\rightarrow$	L	$[2, R_{new}, 0]$	L	
(3)	1 :	L	$[2, R_{old}, 3])$	$\rightarrow$	R	$[1, R_{old}, 3]$	R	
(2)	2 :	R	$[2, R_{new}, 0])$	$\rightarrow$	L	$[3, R_{new}, 0]$	L	
(3)	1 :	L	$[1, R_{old}, 3])$	$\rightarrow$	R	$[0, R_{old}, 3]$	R	
(2)	2 :	R	$[3, R_{new}, 0])$	$\rightarrow$	L	$[4, R_{new}, 0]$	L	
(4)	1 :	L	$[0, R_{old}, 3])$	$\rightarrow$	**L**	$[0, -, 0]$	R	
(5)	2 :	L	$[4, R_{new}, 0])$	$\rightarrow$	HALT			$, QED^+$

图12：两个状态、空白纸带（第2部分）

.

5 更大的符号字母表： n（k）

5.1 数量 n（4）

对于一个含有 4 个符号的字母表，Friedman 给出了如下“非凡”的下界：

定理8。（Friedman[5,定理8。4]）

$$n(4) > A^{(A(187196))}(1),$$

where $A(k) = A(k, k)$ *and* $A^{(n)}(1) = A(A^{(n-1)}(1)), A^{(1)} \equiv A$.

证明。该陈述出现在 [5, 第 7] 页，被列为一个定理，但那里没有给出证明。 □

我们有 $A(1) = 2$、$A(2) = 4$、$A(3) = 2^{2^2} = 16$、$A(4) = A^{(3)}(1)$（见下文）.....一直到 $A(187196)$，以生成指数。

然后我们进行 $A(187196)$ 次递归，从 $A^{(1)}(1) = 2$、$A^{(2)}(1) = 4$ 开始，

$$A^{(3)}(1) = 2^{2^{\cdot^{\cdot^{\cdot^{2^2}}}}}, A^{(4)}(1) = A\big(2^{2^{\cdot^{\cdot^{\cdot^{2^2}}}}}, 2^{2^{\cdot^{\cdot^{\cdot^{2^2}}}}}\big),$$

每个都堆叠着 65536 个 2，.....

从而得到 $n(4)$ 的下界。借用弗里德曼的话来说，这个数量确实“是完全另一回事” [5, 第 7] 页。

5.2 用于 $n(k)$ 的图灵机

我们的图灵机会如何变化?

我们还需要两个符号，即第二部分中的 4 和 4'。此外，三个新状态 q1-C4、q5-K4、q5-V4 用于处理这个额外符号。因此， n_3 的 (44, 8) 实现给出了 $n(4)$ 的 (47, 10) 实现；一般而言，存在一个 $(35+3k, 2+2k)$ 实现，用于计算 $n(k)$ $k \geq 3 \in \mathbb{N}$。对于一般的 $k \geq 3, \Delta \in \{0, 1\}$，包括 $X \notin E$, $\Delta = 1$ 这种情形，我们有大小为 $(35 + 3k + \Delta(7 + k)\ 2k + 2 - \Delta)$ 的图灵机。

(47,10) 的实现立即给出了规模为 (2,2450) 和 (3,1440) 的实现，以及规模为 (922,2) 和 (353,3) 的实现；这里我们采用了定理 7、定理 4 和定理 3 中较为粗略的上界。

定理 9。 存在大小如下的图灵机，分别用于计算 $n(3)$ 和 $n(4)$：

n	m	情形 $k=3$		$k=4$
(2,	•)	C1	图灵机(2,1840)	图灵机(2,2450)
(3,	•)	C2	图灵机(3,1080)	图灵机(3,1440)
(9,	•)	C3	图灵机(9,800)	图灵机(9,1030)
(•,	$2k+2$)	A	图灵机(44,8)	图灵机(47,10)
(•,	$2k+1$)	A	图灵机(54,7)	图灵机(58,9)
(•,	4)	–/A	—	图灵机(160,4)
(•,	3)	A/B	TM(155,3)	TM(353,3)
(•,	2)	A/B	TM(276,2)	TM(922,2)

证明。 情形 A 由构造得出。情形 B 应用定理 3，情形 C1 应用定理 7，情形 C2 应用定理 4，情形 C3 应用定理 6。 □

6 个不可行的忙海狸竞赛

每当一场 (n, m) 竞赛超出对 $n(3)$ 或 $n(4)$ 的 (n', m') 实现时，也就是说， $n \geq n', m \geq m'$，就可以放心地将其视为不可行。另一方面，所有 BB(1,m) 和 BB(n,1) 竞赛都很简单。

这样一来，剩下的就是有趣的情形，即既可行又非平凡的情形，如图 13 左栏 $n(3)$ 所列；我们只用定理 9 中的情形进行插值。右栏 $n(4)$ 则已超出可行范围。

总而言之，除 37022 个非平凡情形 (n, m) 外，其余都会导出 BB(n,m) $> A$(7198,158386)，而且除 51671 个情形外，其余甚至满足 BB(n,m) $> A^{(A(178195))}(1)$。

结论

我们已经证明，至多有 51671 BB(n,m) 个值低于 $A^{(A(1178195))}$(1)。这为忙碌海狸问题的难度提供了第三种描述：

哥德巴赫 CODEGOLFADDICT [3] 给出了一种 (27, 2) 实现，用于搜索哥德巴赫猜想的反例，即每个偶数都是两个素数之和。 BB(27,2) 的值因此取决于数学家在（数论，而非 理论计算机科学）领域解决哥德巴赫猜想的成功与否。

长有限序列 我们的贡献表明， BB(44,8) 和BB(276,2) 在计算上是不可行的，超出了 A(7198, 158386) 的范围。

ZFC Yedidia 和 Aaronson [12] 给出了一台 (7910,2) 图灵机，用于检查 ZFC 的一致性——这可以追溯到 Friedman 的另一项工作。因此，重新解决 BB(7910,2) 超出了 ZFC 的范围。Stefan O'Rear 有一个 (748,2) 实现 [8]。

请注意，Friedman 的问题给出了一个确定的下界，而另外两个问题可以暂且搁置，视为可能永远不停机、因而不会进入 BB 竞赛的问题；前提是你相信 Christian Goldbach、Ernst Zermelo 和 Adolf Fraenkel。

此外，我们还描述了可针对任意给定图灵机生成图灵机的算法，其状态数或符号数均可自由选择，最低可降至 2。

(C++ implementations available upon request from the first author)

n	m	数量	n	m	数量
2	2..1839	1838	2	2..2449	2448
3..8	2..1079	6468	3..8	2..1439	8628
9..43	2..799	27930	9..43	2..1029	39064
44..53	2..7	60	47..57	2..7	66
54..154	2..6	505	58..159	2..6	510
155..275	2	221	160..352	2..3	386
			353..921	2	569
总计：		37022	总计：		51671

Fi图 13：有趣的 BB 情形，由 n(3)（左）和 n(4)（右）给出的上界

参考文献

[1] Ackermann, Wilhelm，实数的希尔伯特式构造，《数学年刊》**99**，第118–133页，1928年。[2] Chaitin, Gregory，算法信息论，CUP，1987年。[3] CODEGOLFADDICT（一个27状态的哥德巴赫检验器）https://gist.github.com/anonymous/a64213f391339236c2fe31f8749a0df6[4] Friedman, Harvey，“长有限序列”，《组合学理论杂志》A辑，**95**(1)，102-144，2001年。[5] Friedman, Harvey，“巨大整数讲义”，2001年。https://bpb-us-w2.wpmucdn.com/u.osu.edu/dist/1/1952/files/2014/01/EnormousInt112201-167h1l6.pdf[6] Green, Milton W.，“二进制图灵机的拉多σ函数下界”，《第5届IEEE年度开关电路理论与逻辑设计研讨会论文集》，91-94，1964年。[7] Michel, Pascal，忙海狸竞赛：历史综述，https://arxiv.org/abs/0906.3749[8] 奥里尔, Stefan（NQL 中的 ZFC 检验器）https://github.com/sorear/metamath-turing-machines[9] Rad´o, T.，“论不可计算函数”，《贝尔系统技术杂志》，**41**(3)，877-884，1962年。[10] SKELET（G·乔治耶夫）（除43个遗留项外均可解决）BB(5,2) https://skelet.ludost.net/bb/index.html[11] Turing, Alan M.，论可计算数及其在判定问题中的应用”，《伦敦数学学会会刊》**s2-42**(1)，230-265，1937年。[12] Yedidia, Adam；Scott Aaronson，一个行为独立于集合论的相对较小的图灵机，《复杂系统》，**25**(4)，2016年。https://content.wolfram.com/uploads/sites/13/2018/02/25-4-4.pdf，预印本：https://arxiv.org/abs/1605.04343