arXiv:2611.32633v1 [cs.CC] 24 Nov 2026

多项式时间量子计算的空间下界

A. C. Cem Say
cem.say@khas.edu.tr

计算机工程系，*Kadir Has*大学，伊斯坦布尔，土耳其

摘要

我们给出了关于可由多项式时间量子图灵机求解的问题的最小空间需求的新结果。我们为量子计算定义了蒙特卡洛空间复杂度的概念（对于每个输入，机器都可以以很小但非零的概率输出错误结果或超出指定的空间预算），并证明：对于任意可计算的非递减空间界 $s(n) \in \omega(1)$，存在一种语言，可以由使用 $s(n)$ 空间的量子图灵机在多项式时间内识别，但在该范围内不能由任何使用 $O(1)$ 空间的机器识别。我们还展示了多项式时间量子计算的次对数最坏情况空间层次，证明 BQTISP($n^{O(1)}$, $o(s_i(n))$) $\subsetneq$ BQTISP($n^{O(1)}$, $O(s_i(n))$)，其中对于每个$i \geq 0$，$s_i(n) = \left(\log n (\log \log n)^i\right)$。

关键词： 量子图灵机、量子有限自动机、时间复杂度、空间复杂度

1 引言

量子计算机解决问题的能力对其时间和内存预算的依赖性，是一个重要的研究课题。只有当算法对这两类资源的消耗随输入长度增长得足够缓慢时，才会被认为在实际中可行。在时间复杂度方面，在这一背景下，多项式界被普遍认为是合理的。鉴于量子硬件目前的发展阶段，对空间预算提出更严格的要求，即采用对数界甚至亚对数界，似乎更为现实。

尽管对于流模型、查询复杂度以及排序和矩阵乘法等具有较长输出的任务，人们已经得到了大量量子空间下界和时间—空间下界，但对于一般量子图灵机（QTMs）进行语言识别的空间下界——这正是本文关注的内容——却要少得多[2, 3, 6], [19]。此前关于多项式时间量子图灵机在最坏情况下采用小空间界时可解决问题全貌的研究 [1, 11, 12, 13] 得出了以下结果：

- Remscrim [12] 证明了一条重要定理，该定理基于一种广为人知的难度度量，确立了小空间量子算法的运行时间与内存使用量之间的权衡关系；这种难度度量针对的是这些算法要识别的语言[17] 。借助这一工具，他得以证明，没有任何次对数空间的量子图灵机能够识别 L_{pal}，即字母表 $\{0,1\}$ 上的回文串语言，并在 $2^{n^{1-\Omega(1)}}$ 时间内完成。注意， L_{pal} 可以由确定性图灵机在多项式时间内、使用 $O(\log n)$ 空间识别。
- Say [13, 15] 通过展示一种语言，证明了另一个“空间阈值”的存在：当允许空间为 $o(\log \log n)$ 时，不存在多项式时间量子算法可以识别该语言；但在 $O(\log \log n)$ 空间内，可以在多项式时间内确定性地识别它。

- 与它们的经典对应物不同，后者只能在 $o(\log\log n)$ 最坏情况空间 [4], 内以多项式时间识别正则语言，量子图灵机可以识别许多非正则语言，例如

$$L_{eq} = \{\, 0^n 1^n \mid n \geq 0 \,\},$$

以及所有有限生成虚拟阿贝尔群的字问题，均可在多项式期望时间内仅使用 $O(1)$ 空间 [11] 求解。Ambainis 和 Watrous [1], 定义的具有量子态和经典态的双向有限自动机（2QCFA），可视为一种从不使用工作带的量子图灵机，因此适合用于建模此类常空间量子算法。

本文首先考察，为多项式时间量子图灵机提供极小的非恒定内存预算（即增长极其缓慢的空间界，例如 $\omega(1) \cap o(\log\log n)$），是否会使其计算能力超越 2QCFA。为此，我们将采用以下空间复杂度度量[8]；它可能比上述研究所采用的最坏情况使用量定义所施加的度量更宽松。

定义 1。 若存在常数 $\varepsilon \in [0, \frac{1}{2})$，使得一台（概率或量子）图灵机 M 满足以下条件，则称其在蒙特卡罗空间 $s(n)$内识别语言 L

- 对于所有 $w \in L$，$\Pr\{M$ 使用至多 $s(|w|)$ 个工作带单元并接受 $w\} \geq 1-\varepsilon$，且
- 对于所有 $w \notin L$，$\Pr\{M$ 使用至多 $s(|w|)$ 个工作带单元并拒绝 $w\} \geq 1-\varepsilon$。

我们将时间空间复杂度类 BQTIMCSP($t(n)$ $s(n)$) 定义为：由量子图灵机可识别的语言集合，这些量子图灵机同时使用蒙特卡罗空间 $s(n)$ 和期望时间$t(n)$运行。

我们的主要结果是

定理 2。 对于任意可计算、非递减且无界的函数 $s(n) \in \omega(1$),

$$\mathsf{BQTIMCSP}\left(n^{O(1)}, O(1)\right) \subsetneq \mathsf{BQTIMCSP}\left(n^{O(1)}, O(s(n))\right).$$

在一项类似的研究中， [5] 针对经典概率计算，Freivalds 和 Karpinski 证明了一个结果，将使用常数空间的概率图灵机（PTM）所识别的语言类与使用增长极其缓慢的蒙特卡罗空间界的语言类区分开来。我们在定理 2 的证明中采用的技术，在两个重要方面不同于 Freivalds 和 Karpinski 的方法：首先，他们的构造产生了运行时间呈指数级的机器，而我们关注多项式时间计算。其次， [5] 利用小空间 PTM 无法识别 L_{pal} 这一点来建立这种区分。即使放宽时间要求，我们也必须为这种区分确定另一个“困难问题”，因为 L_{pal} 已知 [1] 可由指数时间 2QCFA 识别。相反，我们定义并使用一个新颖问题：通过结合 Karpinski 和 Verbeek [8] 以及 Remscrim 的工作中的不同思路，该问题可在多项式时间内使用极小（但非常数）的量子蒙特卡罗空间求解。 [12]。

我们还考察了 $\Omega(\log\log n) \cap o(\log\ n)$ 中的中间范围空间界，并证明存在一个问题的无限层次结构：这些问题均可在多项式时间和小空间内求解，且该层次结构的每一层都对应于上述意义下的不同空间阈值。按照标准的同时期望时间与最坏情况空间复杂度类的术语，我们证明了以下结果，确立了无穷多个新的空间阈值的存在性（除上述提到的对数、双对数和常数界之外）；超过这些阈值后，量子计算机可以在多项式时间内求解更多问题：

定理 3。对于 $i \geq 0$,

$$\mathsf{BQTISP}\left(n^{O(1)}, o\left(\frac{\log n}{(\log\log n)^i}\right)\right) \subsetneq \mathsf{BQTISP}\left(n^{O(1)}, O\left(\frac{\log n}{(\log\log n)^i}\right)\right).$$

本文其余部分的结构如下：第2 节介绍必要的背景知识。定理2 的证明见第3节。第4 节给出定理3的证明。第5 节为结论。

2 定义和背景

我们假设读者熟悉确定性图灵机（DTM）。为研究空间复杂度，我们采用一种多带变体，其中包括一条只读输入带和一条单独的可读写工作带。两条带分别由专用的双向磁头访问。转移函数受到限制，使得输入头绝不会试图移出包含字符串$\triangleright w \triangleleft$的带区域，其中 $\triangleright$ 和 $\triangleleft$ 是特殊的端标记符号，而 w 是输入字符串。在计算开始时，输入头位于左端标记 $\triangleright$上，工作头位于工作带最左侧的单元格上；工作带沿另一方向无限延伸。工作带的所有单元格最初均为空白。我们还假设工作头绝不会试图向左移出其初始位置。我们不计入存储输入、记录内部状态以及跟踪输入头位置所需的“记忆”，而仅使用工作带来衡量空间消耗，具体说明如下。

量子图灵机 [20] 可以看作一台确定性图灵机，并为其增设一个有限量子寄存器（此外还有由经典内部状态集表示的经典寄存器）、一条量子工作带（每个带格存放一个量子比特），以及一个固定大小的测量寄存器，用于存储对量子部分最近一次测量的结果。量子工作带由一个双向读写头访问，并假设该读写头绝不会向左移动到其初始位置（即最左侧带格）的左边。[1]

定义 4. 一个 量子图灵机（*QTM*）是一个 11 元组 $(Q, \Sigma, K, T, r, q_0, q_{acc}, q_{rej}, \tau_0, \delta_Q, \delta_C)$，其中

- Q 是有限经典状态集，
- Σ是有限输入字母表，不包含 $\triangleright$ 和 $\triangleleft$，
- K 是有限工作字母表，其中包含空白符号，$\sqcup$
- T 是测量寄存器的有限可能取值集，
- r是一个正整数，表示量子寄存器中的量子比特数，
- $q_0 \in Q$ 是初始状态，
- q_{acc}， $q_{rej} \in Q$分别是接受状态和拒绝状态，其中 $q_{acc} \neq q_{rej}$，
- $\tau_0 \in T$ 是测量寄存器的初始值，
- $\delta_Q : (Q \setminus \{ q_{acc}, q_{rej} \}) \times (\Sigma \cup \{ \triangleright, \triangleleft \}) \times K \to \Delta$，其中 Δ 是所有具有由代数数矩阵构成的算子和表示的选择性量子操作的集合，是量子转移函数，并且
- $\delta_C : (Q \setminus \{ q_{acc}, q_{rej} \}) \times (\Sigma \cup \{ \triangleright, \triangleleft \}) \times K \times T \to Q \times K \times \{ -1, 0, 1 \}^3$ 是经典转移函数。

在计算开始时，量子图灵机 M 处于状态 q_0，测量寄存器中包含 τ_0，有限量子寄存器和量子工作带中的所有量子比特均处于零态。 M 的每个计算步骤都由一次量子转移构成，该转移作用于

[1]每个量子图灵机算法都可以通过一种简单的编码转换为满足该条件的等价机器：逻辑量子比特 $|0\rangle$ 和 $|1\rangle$ 分别由 $|00\rangle$ 和 $|01\rangle$ 表示（使用两个带单元），并且计算开始时，会将类似 $|10\rangle$ 的“左端标记”组合写入量子工作带。在这种设置下，检测左端标记只需测量任意此类量子比特对中的第一个物理量子比特，同时不会改变整个量子态，包括逻辑量子比特之间的任何纠缠。

机器的量子部分并更新测量寄存器，以及一次经典转移，该转移根据当前扫描到的输入符号和经典工作带符号，以及测量寄存器中的值，更新经典状态、经典工作带和三个读写头的位置：

- 如果 M 处于某个非停机状态 q，且输入头正在扫描符号 σ ，经典工作带头正在扫描符号 κ，则选择性量子操作 $\delta_Q(q,\ \sigma,\ \kappa)$作用于量子寄存器以及量子工作带头当前扫描的量子比特。[2]该操作[3] 使这些 $r+1$ 量子比特发生演化，并将某个输出 τ 以相应的概率存储在测量寄存器中。

- 随后的经典转移 $\delta_C(q,\ \sigma,\ \kappa,\ \tau) = (q',\ \kappa',\ d_I,\ d_W,\ d_Q)$ 将经典状态更新为 q'，把当前扫描的经典工作带符号改为 κ'，并分别按照增量 d_I、 d_W 和 d_Q所指示的方向移动输入头、经典工作带头和量子工作带头。

M 当进入状态 q_{acc} (q_{rej}) 时停机并接受（拒绝） .

对于任意 $\varepsilon \in [0, \frac{1}{2})$，若量子图灵机 M 以识别语言 L ，其误差界为 ε ，则

- 对于每个输入字符串 $w \in L$， M 接受 w 的概率至少为 $1-\varepsilon$，并且
- 对于每个输入字符串 $w \notin L$， M 接受 w 的概率至多为 ε。

如果这种机器在（最坏情况下）空间 $s(n)$内运行，那么对于任意长度为 n的输入，在整个计算过程中，经典工作带和量子工作带上被扫描的单元数至多为$s(n)$。

在空间 $O(1)$内运行的机器所执行的计算，也可以由一台完全不使用工作带、而是将所有可能的内存内容编码在其经典状态集和有限量子寄存器中的机器来完成（且运行时间不会增加）。这种常数空间模型称为具有量子态和经典态的双向有限自动机（*2QCFA*） [12]。请注意， [1] 中2QCFA模型的原始定义采用了限制更严格的量子转移，只允许酉变换和投影测量。Stinespring扩张定理[7]保证，只要两种表述都限定为在转移矩阵中使用代数数，它们的计算能力便是等价的。

BQTISP($t(n)$ $s(n)$) 表示由量子图灵机以有界错误识别的语言类；这些量子图灵机的最坏情况空间不超过 $s(n)$，期望时间为 $t(n)$。全文中，时间界和空间界均假定为非递减的。

量子图灵机总体上以及 2QCFA 尤其如此，其计算能力取决于允许出现在描述量子转移的矩阵中的数的集合 [16]。定义4中采用的代数数集合通常被视为自然的选择[11, 12, 20]。本文将介绍的具体算法只使用代数转移振幅。另一方面，我们的下界结果将采用允许使用全体复数的量子图灵机和 2QCFA 模型，以证明所得界限对于这些更强大的模型同样成立。在此背景下，$\mathsf{BQTISP}_{\mathbb{C}}(t(n)\ s(n))$ 表示以 $\mathsf{BQTISP}(t(n)\ s(n))$ 为对应定义、但基于无限制振幅量子图灵机的类。类似地，$\mathsf{2QCFA}_{\mathbb{C}}$ 就是一种既可使用超越振幅，也可使用代数振幅的 2QCFA [1]。

定义 5. [4] 对于任意语言 L 和数值 $n \geq 0$，如果 $|w| \leq n$、 $|w'| \leq n$，且存在一个区分字符串 v ，满足 $|wv| \leq n$、 $|w'v| \leq n$和 $wv \in L \iff w'v \notin L$，则称两个字符串 w 和 w' 为 (L,n)- 不相似，记作 $w \not\sim_{L,n} w'$。令 $D_L(n)$ 为满足下述条件的 d 的最大值：存在

[2]在这方面，我们对 QTM 的定义不同于 Watrous 最初的定义 [20], ，后者的量子操作仅取决于经典状态 q。正如 [13]，我们在此使用的变体与 Watrous 模型的计算能力等价，并且可以无缝地专门化为 2QCFA 的形式化定义 [12]。[3]第3.2 节给出的用于证明上界的具体量子算法仅使用幺正变换和投影测量，但形式体系也允许其他非幺正量子操作。这一定义的普遍性确保我们的下界适用于采用任何物理上可实现的量子操作的算法 [12, 20]。

d 个互不相同的字符串，它们两两 $\not\sim_{L,n}$，约定若任意两个字符串都不 (L,n)-不相似，则 $D_L(n)=1$。函数 D_L 称为 L 的非正则性。

以下结果由 Remscrim 得出，它们将非正则性度量 D_L 与识别相应语言 L 的量子图灵机的同步时间—空间复杂度联系起来。[12]

定理 6。假设 $L\in \mathsf{BQTISP}_{\mathbb{C}}(t(n)\ s(n))$，并进一步假设 $s(n)=o(\log\log D_L(n))$。那么存在一个实数 $b_0>0$，使得

$$t(n)\in\Omega\left(2^{-b_0 s(n)}D_L(n)^{2^{-b_0 s(n)}}\right).$$

定理 7。假设 L 在误差界为 ε、期望时间为 $t(n)$ 的情况下，由一个 *2QCFA* 识别；该 *2QCFA* 的经典状态集包含 d 个元素，量子寄存器中包含 r 个量子比特。那么存在一个自然数 n_0，使得

$$t(n)\geq\frac{(1-2\varepsilon)^2}{16\sqrt{2}\,(2^r)^4d^2}\,D_L(n)^{\frac{1}{(2^r)^4d^2}}$$

对于所有 $n\geq n_0$。

3 非常小的蒙特卡罗空间对多项式时间量子图灵机很有用

本节给出定理 2 的证明，现将其重述如下。

定理 2。对于任意可计算的、非递减且无界的函数 $s(n)\in\omega(1)$，

$$\mathsf{BQTIMCSP}\left(n^{O(1)},O(1)\right)\subsetneq\mathsf{BQTIMCSP}\left(n^{O(1)},O(s(n))\right).$$

3.1 见证语言

我们首先定义将见证定理 2 中分离结果的语言。设 $\alpha:\ \mathbb{N}\to\mathbb{N}$ 为任意无界且非递减的可计算函数。对于每个这样的 W_α，都将定义一个语言 α。每个 W_α 的成员都是形如 $\pi\%\rho$ 的两部分字符串，其中

- 前缀 π 是自然数 k 的表示，格式将在第3.1.1节中描述，而
- 后缀 ρ 是群 $\mathbb{Z}^k$ 的字问题成员的编码，具体内容将在第3.1.2节中描述。

3.1.1 自然数编码的方法

我们回顾一种经典的通用计算模型。三计数器机（*3*计数器机）[8, 10] M由有限状态控制和三个（一元）计数器组成，分别命名为 C_1、C_2 和 C_3。如果 M 有 T个状态，其状态集合为 $Q=\{q_1,q_2,\ldots,q_T\}$，约定 q_1 为初始状态，q_T为停机状态。M 的行为由其转移函数 δ控制。对于 M的每个非停机状态 q_j，δ（q_j）会指定对某个计数器执行以下操作之一，例如 C_i:

- inc(i,q_k)：将 C_i 增加1，并切换到状态 q_k（该状态可能是新状态）。
- dec(i，q_k，q_l)：将 C_i 减 1。如果 C_i 已达到 0，则切换到状态 q_k；否则，切换到状态 q_l。

我们假设初始状态只关联 INC 操作。因此， M 的配置可表示为如下形式的字符串： $0^j1^*2^*3^*\$$，其中 $q_j \in Q$ 是当前状态， i的最长子串长度等于 C_i 中 $i \in \{1,2,3\}$的数值，并且每个配置的末尾都用 $标记。对于输入 n （一个自然数），3计数器机 M 从初始配置 $01^n\$$（即 C_2 和 C_3 均初始化为 0）开始运行，且仅当到达状态 q_T时才停机。我们将 3计数器机在此类停机配置中第一个计数器 C_1 中的数值定义为其输出。按照上述方式看待输入和输出，三计数器机可以计算所有如下形式的部分可计算函数： $f:\ \mathbb{N} \to \mathbb{N}$ [8, 10]。[4]

对于任何在每个输入上都停机的 3计数器机 M ，令 $h_M(n)$ 表示记录 M 在输入 n上的计算历史的字符串，即将描述 M 在此次计算过程中从开始到停机所达到的完整配置序列的字符串连接起来。

给定任意无界且非递减可计算函数 $\alpha:\ \mathbb{N} \to \mathbb{N}$，我们将展示如何构造一台 3计数器机 $M_{\alpha''}$ ，使其计算另一个由 α确定的全函数 α'' 。对于任意自然数 k，随后将使用字符串 $h_{M_{\alpha''}}(k)$ 作为 k 的表示。以下论证改编自 [8]。

给定 α，定义新函数 α' 如下：

$$\alpha'(n) = \begin{cases} 0, & \text{if } \{m \mid \alpha(m) \le n\} = \emptyset, \\ \max\{m \mid \alpha(m) \le n\}, & \text{otherwise.} \end{cases}$$

由于 α 是无界、非递减且可计算的，因此 α' 也是无界、非递减、可计算且为全函数。

对于任意 n， $\alpha''(n)$ 的值是数字 $\alpha'(0)$、 $\alpha'(1)$、 $\alpha'(2)$..... $\alpha'(n)$ 之和。在输入 0 时，3计数器机 $M_{\alpha''}$ 只需计算并输出 $\alpha'(0)$。对于任何更大的输入 n， $M_{\alpha''}$ 通过增加一次迭代来扩展计算 $\alpha''(n-1)$，该迭代计算 $\alpha'(n)$。

我们注意到

$$\alpha'(n) \le \alpha''(n) < |h_{M_{\alpha''}}(n)| \tag{1}$$

对于所有 n，因为字符串 $h_{M_{\alpha''}}(n)$ 显然包含一个真子串 $1^{\alpha''(n)}$ ，该子串表示其最终配置描述中 $M_{\alpha''}$ 的输出。

以下事实将在引理 10 的证明中发挥作用。

引理 8。 对于所有 $k \ge 0$, $k \le \alpha(|h_{M_{\alpha''}}(k)|)$。

证明。 反设对于某个 k，有 $\alpha(|h_{M_{\alpha''}}(k)|) < k$ 。根据 α' 的定义，必有

$$\alpha'(k) \ge |h_{M_{\alpha''}}(k)|,$$

这与（1）矛盾。 □

3.1.2 表示字问题

对于任意来自同一字母表的字符串 w 和符号 σ ，令 $\#(w,\ \sigma)$ 表示 σ 在 w 中出现的次数。令 ϵ 表示空字符串。对于每个 $k \ge 1$，群 $\mathbb{Z}^k$ 是群 $\mathbb{Z}$ （整数，其群运算为加法）的 k 份直积。[5] 考虑字母表 $\Sigma_k = \{a_1, b_1, \ldots, a_k, b_k\}$，其中每个 a_i 对应于第 i 个群中的“+1”，而 b_i 对应于生成元 a_i 的逆元。 Σ_k^* 中的每个字符串 w 都表示一个 k 元整数元组，其中第 i 个整数对于每个 i 都等于 $\#(w,\ a_i) - \#(w,\ b_i)$。我们将把该语言称为

$$W_{\mathbb{Z}^k} = \{\, w \mid w \in \Sigma_k^*, \#(w, a_i) = \#(w, b_i),\ \forall i \,\}$$

[4]Minsky [10] 描述了如何通过相应的 3计数器机 M_C模拟任意图灵机 M_T： 具体做法是将 M_T 的每一步移动替换为 M_C 的一系列动作，以仿真该移动的效果。[5]有关支撑这项工作的群论相关内容（包括字问题和增长率等概念的形式化定义）的详细阐述，请参阅 [12]。

（即所有表示全 0 元组的字符串的集合） $\mathbb{Z}^k$ 的字问题。

为了用单个字母表表示所有这些字问题 k ，我们将使用以下单射编码。

定义9。 对于所有 $k \geq 1$，函数 $\kappa_k : \Sigma_k^* \to \{0,1\}^*$ 定义如下:

- $\kappa_k(\epsilon) = \epsilon$,
- $\kappa_k(a_i) = 0^i 10$,
- $\kappa_k(b_i) = 0^i 11$，以及
- $\kappa_k(\sigma w) = \kappa_k(\sigma)\kappa_k(w)$，其中 $\sigma \in \Sigma_k$ 和 $w \in \Sigma_k^+$

对所有 $i \in \{1, \ldots k\}$。

对于每个 k，的编码字问题定义为

$$W'_{\mathbb{Z}^k} = \{\, \kappa_k(w) \mid w \in W_{\mathbb{Z}^k} \,\}.$$

对于每个 α，我们现在定义语言

$$W_\alpha = \{\pi \% \rho \mid \exists k \geq 1 \; (\pi = h_{M_{\alpha''}}(k) \wedge \rho \in W'_{\mathbb{Z}^k})\}$$

在公共字母表 $\{0, 1, 2, 3, \$, \%\}$上，使得 W_α 中的每个字符串都由一个前缀构成，该前缀编码某个数 k，后接一个后缀，而该后缀属于 $\mathbb{Z}^k$的编码字问题。

3.2 识别 W_α 的量子图灵机

引理 10。 对于任意可计算的、非递减且无界的函数 α,

$$W_\alpha \in \mathsf{BQTIMCSP}\left(n^{O(1)}, O(\log \alpha(n))\right).$$

证明。对于所有 $\varepsilon \in (0, 1/2)$，我们将证明存在一个多项式时间量子图灵机 $M_{\alpha,\varepsilon}$ 使得

- $M_{\alpha,\varepsilon}$ 其量子寄存器中有一个单量子比特，并且从不使用其量子工作带，
- 对于所有 $w \in W_\alpha$，$\Pr\{M_{\alpha,\varepsilon}$ 扫描其经典工作带的 $O(\log \alpha(|w|))$ 个单元并接受$w\} \geq 1 - \varepsilon$，并且
- 对于所有 $w \notin W_\alpha$，$\Pr\{M_{\alpha,\varepsilon}$ 扫描其经典工作带的 $O(\log \alpha(|w|))$ 个单元并拒绝 $w\} \geq 1 - \varepsilon$。

我们首先解释$M_{\alpha,\varepsilon}$中大量使用的“量子子程序”背后的思想。考虑 $\mathbb{Z}$的字问题，即语言

$$W_{\mathbb{Z}} = \{\, w \mid \#(w, a) = \#(w, b)\}$$

字母表为 $\{a, b\}$。存在一个多项式时间的 2QCFA 算法 $M_{W_{\mathbb{Z}}}$ [11]，它以负单边有界误差识别$W_{\mathbb{Z}}$，也就是说， $M_{W_{\mathbb{Z}}}$ 以概率 1 接受 $W_{\mathbb{Z}}$ 的成员，并且可以通过选择合适的实例$M_{W_{\mathbb{Z}},\varepsilon}$ （从这类机器的无限集合中选取）进行调节，从而对于 ε的任意所需小正值，都以至少 $1 - \varepsilon$ 的概率拒绝非成员。$M_{W_{\mathbb{Z}}}$ 在其量子寄存器中使用单量子比特。

我们采用一种技术 [13]，通过该技术可以调整 $M_{W_{\mathbb{Z}}}$ 算法 $M_{W_{\mathbb{Z}}}$，从而构造出执行以下基本任务变体的子程序（即构成更大型量子图灵机程序一部分的“子机器”）：比较嵌入输入字符串中的两个易于识别的子序列的长度。在下文将要介绍的$M_{\alpha,\varepsilon}$ 描述中，EqNum这一通用名称将用于指代此类子机器；可以将每个子机器看作在“虚拟带”上运行 $M_{W_{\mathbb{Z}}}$ 的一个实例。每个这样的子机器都将专门用于比较由特定输入带区段中的指定对象组成的两个子序列

的长度。例如，M_α将执行的一项子任务是：比较其输入带某个区段中两个不同子串（如010和011）的出现次数；该区段左侧由符号%界定，右侧由结束标记 $\triangleleft$ 界定。（假设存在形如% $\{0,1\}^+ \triangleleft$ 的区段，并且两个分隔符之间的二进制串可以由M_α，ε 无歧义地解析为若干个010、011以及可能存在的其他二进制子串的串接。正如我们将看到的，预处理阶段保证了对于M_α，ε而言这些条件成立。）为执行此子任务，EqNum从M_α，ε 的输入头位于符号%的位置开始。每当它扫描到表1第一列中的某个子串时（通过执行M_α，ε的一步或多步），EqNum就向 $M_{W_\mathbb{Z}}$ “馈入”该表第二列中对应的符号，模拟 $M_{W_\mathbb{Z}}$ 与该虚拟符号相关联的计算步骤。被模拟的 $M_{W_\mathbb{Z}}$ 步骤的经典转移随后将决定输入头的新方向。接着，读写头可能移动若干个M_α，ε 步骤（可能跳过010和011以外的二进制子串），直到定位到该方向上最近的目标子串。EqNum以这种方式继续执行，直到被模拟的 $M_{W_\mathbb{Z}}$到达其某个接受或拒绝状态。如果 $M_{W_\mathbb{Z}}$ 的模拟执行拒绝，EqNum子机器也将拒绝，并终止M_α，ε的执行。如果被模拟的 $M_{W_\mathbb{Z}}$ 接受，EqNum则简单地将控制权返回给M_α，ε的下一个状态。通过设置底层 $M_{W_\mathbb{Z}}$的误差界，可以调节EqNum未能发现所比较子序列长度不匹配的最大概率。

表1：比较 % 与 $\triangleleft$之间“010”和“011”的出现次数

输入带上 $M_{\alpha,\ \varepsilon}$ 的子串	M_W 在虚拟带上对应的符号 $_{\mathbb{Z}}$
%	$\triangleright$
010	a
011	b
$\triangleleft$	$\triangleleft$

用于获得$M_{\alpha、\ \varepsilon}$ （任意所需的（α、 ε）对）的通用模板见图1。我们从子机器 HistoryCheck （阶段（A））的实现开始讲解该算法。

如果 w 不是形如 $\pi\%\rho$ 的形式，则拒绝，其中 $\pi \in \{0,1,2,3,\$\}^+$ 且 $\rho \in \{0,1\}^*$。（A）运行 HistoryCheck，以确定是否 $\pi \in H_{M_{\alpha''}}$。（B）将数字 $k = |\mathrm{inp}(\pi)|$ 以二进制形式写在经典工作带上。（C）如果 ρ 不是有效的 κ_k 编码，则拒绝。对从 1 到 k 的每个整数 i，执行以下操作：（D）运行 EqNum，比较码字 0^i10 和 0^i11 在 ρ 中的出现次数。若此前未被拒绝，则接受。

图1：量子图灵机 $M_{\alpha,\ \varepsilon}$。 w 表示长度为 n 的输入字符串。

令 M 为一台最终会在每个输入上停机的 3计数器机。定义 H_M 为 M的所有计算历史的集合 M （格式如第 3.1.1 节所述），其中输入均为正数：

$$H_M = \{h_M(n) | n > 0\}$$

我们将证明，HistoryCheck 体现了一种常数空间多项式时间量子算法，它使用量子寄存器中的单量子比特，检查输入字符串的前缀 π 是否属于 3计数器机的历史语言 $M_{\alpha''}$（针对函数 α 而定制，详见第 3.1.1节）。如果它在 π中检测到语法错误，HistoryCheck 便以拒绝的方式终止执行。否则，它会将控制权交还给主机，以继续阶段（B）。

HistoryCheck 首先利用其有限状态控制检查一般配置语法，确认第一个配置为某个 $k \geq 1$ 对应的 $01^k\$$，并确认最后一个配置处于停机状态。然后，它扫描 π，检查每一对相邻配置

$$(0^{i_0}1^{i_1}2^{i_2}3^{i_3}\$, 0^{j_0}1^{j_1}2^{j_2}3^{j_3}\$)$$

对配置描述进行检查，以确定其是否表示机器 $M_{\alpha''}$ 的有效转移。有限状态控制足以追踪程序流程，并检查 i_0 和 j_0 是否正确关联。至于计数器，请注意， $M_{\alpha''}$ 的每次此类转移都恰好要求一个计数器值改变 1，而其余计数器值保持不变。对于每个 $c \in \{1, 2, 3\}$，都可以通过专门用于比较以下符号出现次数的 EqNum“调用”来控制这些条件：左侧配置描述中最右侧的 0 与末尾的 \$ 之间的符号 c，以及右侧配置对应片段中的符号 c。如果 $M_{\alpha''}$ 的转移函数规定所考察的计数器值应增加 1，则由一个子机器处理：每当扫描左侧配置中最左侧的 0 时，它都会向 $M_{W_{\mathbb{Z}}}$ 输入符号 $\triangleright a$ （而不只是 $\triangleright$）。 $M_{\alpha''}$ 的减法操作也以类似方式处理，即为右侧配置的右分隔符引入一个额外的虚拟 b。

所有 EqNum 实例（包括 HistoryCheck 中调用的实例以及后续阶段（D）中的实例）在 $M_{\alpha,\varepsilon}$ 中均设定为以期望的总体误差界 ε 运行。当 $\pi \notin H_{M_{\alpha''}}$ 时，阶段（A）未能拒绝的概率至多为 ε，因为任何通过 HistoryCheck 的此类“非法”计算历史都至少包含一个缺陷，而相应的 EqNum 子机器以至少 $1-\varepsilon$ 的概率检测到该缺陷。

我们定义偏函数 inp : $\{0, 1, 2, 3, \$\}^* \to \{1\}^*$，使其将任何形如 $(0^+1^*2^*3^*\$)^+$ 的字符串 s 映射为 s 中最左侧 \$ 符号左边最长的（可能为空）连续 1 子串。阶段（B）只需沿着 inp(π) 移动输入头，并在经典工作带上以二进制表示法写下该子串的长度 k。

引理 8 以及上述分析表明，阶段（B）在工作带上写入一个大于 $\alpha(|\pi|)$ 的数（由于 $\alpha(n)$ 是非递减的，因此该数不会超过 α）的概率有上界；因此，该数超过空间预算 $O(\log \alpha(n))$的概率也有上界，该上界为 ε。

在阶段（C）中，$M_{\alpha,\,\varepsilon}$会检查 ρ 是否具有 $\kappa_k(\sigma_1)\kappa_k(\sigma_2)\cdots\kappa_k(\sigma_m)$ 的形式，其中所有 σ_i 都属于 Σ_k。通过反复计数至多达到 k，即可对 ρ 进行一次扫描来完成此操作。阶段（D）也使用这样的计数器，在循环的第 i 次迭代时，忽略 ρ 中除 0^i10 和 0^i11 之外的所有码字；并如上文在解释 EqNum.[6] 时所述，对于 ρ 中的每个 0^i10 （分别为 0^i11），向算法 $M_{W_{\mathbb{Z}}}$ 提供一个虚拟的 a （分别为 b）。

每个阶段使用的空间都不超过 $O(\log k)$ 位。总运行时间有多项式上界，因为即使在低概率情况下，各个阶段处理的是较大的错误 k 值（该值不可能超过输入长度 n），它们仍都能在期望多项式时间内运行。$M_{\alpha,\,\varepsilon}$ 会在连续调用 EqNum 时重置并重复使用其单量子比特。

如果输入字符串属于 W_α， $M_{\alpha,\,\varepsilon}$ 就以概率 1 接受它。字符串 $w \notin W_\alpha$ 被接受的概率最高不超过 ε；当 w 只有一个缺陷，且对应的 EqNum 子程序未能检测到该缺陷时，就会达到这一上限。□

3.3 不存在能识别 W 的常数空间多项式时间机器 α

本节最后将证明以下引理。

引理 11。 对于任意可计算、非递减且无界的函数 α,

$$W_\alpha \notin \mathsf{BQTIMCSP}\left(n^{O(1)}, O(1)\right).$$

我们首先指出，对于常数界，最坏情况空间复杂度类与蒙特卡洛空间复杂度类相同：

[6] No由于阶段（C）已经完成，因此在此处 ρ 无疑可以被“标记化”为码字 d.

引理 12. BQTIMCSP $\left(n^{O(1)}, O(1)\right) = \text{BQTISP}\left(n^{O(1)}, O(1)\right)$.

证明。对于任意一对复杂度函数 t 和 s，BQTISP $(t(n)\ s(n)) \subseteq$ BQTIMCSP $(t(n)\ s(n))$，因为任何由某台在最坏情况空间 $s(n)$ 内运行的量子图灵机所识别的语言，都已经满足关于属于相应蒙特卡洛空间复杂度类的定义 1。反之，假设某台多项式时间量子图灵机 M 在蒙特卡洛空间 $O(1)$ 中识别语言 L。这意味着存在一个自然数 s_0 和一个实数 $\varepsilon < \frac{1}{2}$，使得对于每个输入 w，M 在其工作带上扫描的单元格数至多为 s_0，并以至少 $1-\varepsilon$ 的概率终止，同时给出关于 w 是否属于 L 的正确接受/拒绝响应。我们构造一个 2QCFA M_0，在“截断的”工作带上模拟 M 的行为：M_0 使用其经典态模拟 M 的经典工作带最左侧的 s_0 个单元格，并使用其有限量子寄存器模拟 M 的量子工作带中最左侧的 s_0 个量子比特。每当 M 试图访问任意工作带上的第（s_0+1）个单元格时，M_0 就拒绝。显然，M_0 以误差界 ε 识别语言 L，且其运行时间不会超过 M 的运行时间，因此可以得出 BQTIMCSP $\left(n^{O(1)}, O(1)\right) \subseteq$ BQTISP $\left(n^{O(1)}, O(1)\right)$。 □

为证明 引理 11，因此只需证明不存在任何多项式时间 2QCFA 能
识别 W 对任何满足所述条件的 α 均成立。

定义元素 g of $\mathbb{Z}^k$ 的长度为 Σ_k^* 中表示 k-元组 g的最短字符串的长度。$\mathbb{Z}^k$ 中长度至多为 n 的元素数量由函数$\beta_{\mathbb{Z}^k}(n)$表示，称为 $\mathbb{Z}^k$的增长率。

对于一对非递减函数 f_1、$f_2 : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$，若存在正实数 c_1、c_2，使得对于任意非负实数 r，都有 $f_1(r) \leq c_1 f_2(c_1 r + c_2) + c_2$，则记作 $f_1 \prec f_2$；若 $f_1 \prec f_2$ 和 $f_2 \prec f_1$二者同时成立，则记作 $f_1 \sim f_2$。

事实 13。 [9] 对于所有 $k \geq 1$，$\beta_{\mathbb{Z}^k}(n) \sim n^k$。

引理 26 阐明了 [12] 任意有限生成群的增长率上界G 可用其字问题的非正则性 G来表示。将该引理直接应用于群 $\mathbb{Z}^k$ 可得到以下结果，这一结果将对我们证明引理 15时很有用。

事实 14。 对于任意 $k \geq 1$、$n \geq 0$、$\beta_{\mathbb{Z}^k}(n) \leq D_{W_{\mathbb{Z}^k}}(2n)$。

引理 15。 设 α 是一个可计算的、非递减且无界的函数。那么不存在能够识别语言 W_α 的多项式时间 *2QCFA*$_{\mathbb{C}}$。

证明。设 W_α 由一个 2QCFA$_{\mathbb{C}}$ M_α^* 以误差界 $\varepsilon < \frac{1}{2}$ 在期望时间 $O(n^c)$ 内识别，该 2QCFA 的量子寄存器中有 r 个量子比特，其中某个 $c, r \in \mathbb{N}$。令 Q 为 M_α^*的经典状态集。不失一般性地，假设 M_α^* 在计算开始后不会再次进入初始经典状态 q_0。我们将展示如何将 M_α^* 用作模板，构造一个能够识别 $W'_{\mathbb{Z}^k}$ 的 2QCFA$_{\mathbb{C}}$ M_k，其中 $k \geq 1$ 可取任意期望值。重要的是，我们将证明，对于 k 的不同取值，所有这些 M_k 机器都具有相同的经典状态集、相同的量子寄存器大小，以及其期望运行时间界中相同的多项式指数 c（这些量均与 k 无关），就像 M_α^* 一样。最后，我们将证明这与关于群 $\mathbb{Z}^k$ 增长率的一个众所周知的事实相矛盾，从而完成证明。

对于任意 $k \geq 1$, M_k 都可通过修改 M_α^* 得到，使得 M_k 对任意输入 $w \in \{0,1\}^*$ 的输出都与 M_α^* 在输入字符串 $h_{M_{\alpha''}}(k)\%w$ 上的输出相同。当其输入头正在扫描除左端标记 $\triangleright$ 之外的任何符号时，M_k 只需执行由 M_α^* 的程序所规定的计算步骤。当扫描 $\triangleright$ 时，M_k 执行一个步骤，以“概括”M_α^* 在输入带内容的前缀 $p_k = \triangleright h_{M_{\alpha''}}(k)\%$ 上的行为，具体如下所述。

按照 [12], , 我们定义空间

$$H_\partial = \mathbb{C}^Q \otimes \mathbb{C}^{2^r}.$$

我们将一对（q, A）称为边界状态，其中 $q \in Q$，而 A 是定义在 $\mathbb{C}^{2^r}$上的密度算子，并将其与定义在 H_∂上的密度算子 $|q\rangle\langle q| \otimes A$ 等同。边界状态表示 q的“组合内部

状态”，由其经典状态和量子寄存器的状态组成，它们各自的维度均与 k无关。

考虑以下两种情况之一： M_α^* 在 p_k中的初始计算，从其左端标记开始；或者对 p_k进行后续遍历，从读写头向左越过 p_k与 w 之间的“边界”后，位于符号 % 处开始。这样的遍历可以通过接受、拒绝或向右离开 p_k 并处于某个经典状态而结束。对于在此类遍历开始时以正概率出现的每个边界状态（q， A），这些结果的总概率为 1，因为 M_α^* 具有有限期望运行时间。现在，我们说明如何将每次此类遍历替换为 M_k的一次转移。

第 3.2 节中定义 [12, 之后的第二条备注里的固定前缀遍历摘要构造 12] ，针对每个经典状态 q, 给出了原始量子寄存器上的一个选择性量子操作[7] ，用于概括从状态 p_k 开始对 q的遍历。其经典结果记录接受、拒绝，或以指定经典状态退出。该操作定义于所有寄存器状态，并且对每个可达边界状态都能复现该遍历。

我们的 2QCFA$_\mathbb{C}$ 定义允许在 M_k 扫描▷时用单步执行此操作。随后，其经典转移会接受、拒绝，或转到所指示的经典状态并向右移动。初始遍历也以同样的方式概括，其中 q_0 将其与后续遍历区分开来。因此， M_k 使用相同的经典状态集 Q ，以及与 M_α^*相同的 r量子比特寄存器。[8]

由于 M_k 在模拟 M_α^* 于p_k上执行的多步遍历时，每次只花费一步，因此它对任意特定输入 w的运行时间不会超过 M_α^*对输入 $h_{M_{\alpha''}}(k)$ %w的运行时间。注意，提供给 M_k 的每个输入字符串都比 M_α^*在相应运行过程中处理的输入短 $|p_k|-1$ 个符号；根据我们的初始假设，该运行耗时为 $O(n^c)$。因此， M_k的运行时间作为其输入长度 n 的函数，属于 $O((|p_k|-1+n)^c)$，所以必定存在数值c_0, n_k ，使得对于长度大于 n_k的输入， M_k 的期望运行时间至多为 $c_0 n^c$ ，其中 n_k 是一个可能取决于 k的常数。这证明了我们的主张： M_k 的运行时间也属于 $O(n^c)$。

现在，我们代入 ε、运行时间 $O(n^c)$、量子比特数 r 和经典状态数d ，这些量子机器见定理 7 ，从而得出存在一个自然数 n_k' ，其取决于 k ，使得

$$D_{W'_{\mathbb{Z}^k}}(n) \leq c_1 n^{c(2^r)^4 d^2} \tag{2}$$

对所有 $n \geq n_k'$均成立，其中 c_1 是取决于 M_α^*的另一个常数。

我们将使用（2）来界定 $W_{\mathbb{Z}^k}$ 的非正则性。我们断言

$$D_{W_{\mathbb{Z}^k}}(n) \leq D_{W'_{\mathbb{Z}^k}}((k+2)n)$$

对所有 $k \geq 1$ 和所有 $n \geq 0$均成立。对于所有满足 $D_{W_{\mathbb{Z}^k}}(n) = 1$的 n ，这一点显然成立。否则，令x_1、...、...、$x_t \in \Sigma_k^*$为两两（$W_{\mathbb{Z}^k}$， n）-相异的，其中 $t = D_{W_{\mathbb{Z}^k}}(n)$。对于任意长度至多为 n的 $x \in \Sigma_k^*$，定义

$$x' = \kappa_k(x).$$

由于 Σ_k 中符号的每个码字长度都至多为 $k+2$,

$$|x_i'| \leq (k+2)|x_i| \leq (k+2)n.$$

固定 $i \neq j$。由于 x_i和 x_j是（$W_{\mathbb{Z}^k}$， n）-不相似的，因此存在 $v \in \Sigma_k^*$使得

$$|x_i v| \leq n, \qquad |x_j v| \leq n,$$

[7]该操作由作用于组合经典与量子状态空间的非截断转移算子 [12], 得到：固定进入的经典状态，并将离开的经典状态视为经典结果。[8]即使我们采用限制更多的 2QCFA$_\mathbb{C}$ 定义， [1, 11] 该定义仅允许在量子转移中使用幺正变换和投影测量，但 Stinespring扩张定理（第 2节）保证选择性量子操作仍可由 2QCFA$_\mathbb{C}$ 构件实现，且其时间、比特开销和量子比特开销均为与 k无关的常数。

and

$$x_i v \in W_{\mathbb{Z}^k} \iff x_j v \notin W_{\mathbb{Z}^k}.$$

Then

$$|x_i' v'| = |\kappa_k(x_i v)| \leq (k+2)n, \qquad |x_j' v'| = |\kappa_k(x_j v)| \leq (k+2)n,$$

and

$$x_i' v' \in W'_{\mathbb{Z}^k} \iff x_i v \in W_{\mathbb{Z}^k}, \qquad x_j' v' \in W'_{\mathbb{Z}^k} \iff x_j v \in W_{\mathbb{Z}^k}.$$

因此， x_i'和 x_j'是（$W'_{\mathbb{Z}^k}$，（$k+2$）n）-不相似的。因此

$$D_{W_{\mathbb{Z}^k}}(n) \leq D_{W'_{\mathbb{Z}^k}}((k+2)n).$$

将这一事实与（2）结合，并令 $u = c\,(2^r)^{\,4} d^2$ 可得

$$D_{W_{\mathbb{Z}^k}}(n) \leq c_2 n^u \tag{3}$$

当 n足够大时，其中 c_2 是一个依赖于 k的常数。

事实 14 和不等式（3）表明，对于任意 $k \geq 1$ 以及所有充分大的 n,

$$\beta_{\mathbb{Z}^k}(n) \leq c_k n^u,$$

其中常数 c_k 依赖于 k，但 u 不依赖于此。对于所有 $k > u$，这与事实 13相矛盾。事实13证明 $\beta_{\mathbb{Z}^k}$（n） $\sim n^k$，这表明我们最初关于多项式时间 2QCFA$_{\mathbb{C}}$能够识别 W_α 的假设是错误的。 □

如上所述，引理11 可由引理 15 和 12推出。

对于任意可计算、非递减且无界的函数 $s(n) \in \omega(1)$，令 $\alpha(n) = \max\{2, \lfloor s(n) \rfloor\}$。引理10 和11 构成了定理2的证明。

4 多项式时间量子图灵机的次对数空间层次

本节给出定理3的证明，现将其重述如下。

定理3。对于每个 $i \geq 0$,

$$\mathsf{BQTISP}\left(n^{O(1)}, o\left(\frac{\log n}{(\log\log n)^i}\right)\right) \subsetneq \mathsf{BQTISP}\left(n^{O(1)}, O\left(\frac{\log n}{(\log\log n)^i}\right)\right).$$

我们将定义一个语言族，用以证明层级中相邻层级之间的分离结果。

对于任意整数 $l \geq 0$，令 bin(l) 表示以二进制表示法表示 l 的最短字符串。该语言

$$L_S = \{\mathrm{bin}(0)\$\mathrm{bin}(1)\$\mathrm{bin}(2)\$\cdots\$\mathrm{bin}(k) \mid k > 0\}$$

属于确定性复杂度类 SPACE(log log n) [18]。不难证明， L_S 中任意 n 符号成员最右侧“段”（即最右侧 $ 后的后缀 bin(k_n)）的长度与 n 的关系为

$$\frac{1}{2}\log n \leq |\mathrm{bin}(k_n)| \leq \log n.^9 \tag{4}$$

多项式时间确定性图灵机可以通过逐位比较连续片段来识别 L_S，并在检测到不一致时停机，或者在所有比较均成功时停机。由于

[9]对数以 2 为底。

每次比较都涉及一个计数器，其值最多为 $|\mathrm{bin}(k_n)|$，因此空间复杂度为$O(\log\log n)$。在下文将要给出的引理 16 的证明中，该算法将由一台确定性图灵机用于将一些变量设为其初始值，同时在工作带上使用不超过 $O(\log\log n)$ 位。

下面将描述的语言格式基于 [14] 中的一个类似构造。考虑字母表 $\Sigma = \{0,1,\$\} \times \{0,1,\#\}$。使用 Σ 书写的字符串应被解释为包含两条轨道，其中符号 (σ_u, σ_l) 表示上轨道（分别是下轨道）在该位置包含符号σ_u （分别是 σ_l）。

对于每个非负整数 i，我们将语言 L_i 定义为满足以下条件的所有字符串 $w \in \Sigma^*$ 的集合：

- w 的上轨道是 L_S 的某个成员 u_n，且以后缀 $\$\mathrm{bin}(k_n)$ 结尾 .
- w 的下轨道是集合中的一个成员

$$\left\{ p\#^{|u_n|-j} \;\middle|\; p \in L_{pal} \cap \{0,1\}^j,\ j = 2^{\left\lceil \frac{|\mathrm{bin}(k_n)|}{|\mathrm{bin}(|\mathrm{bin}(k_n)|)|^i} \right\rceil} \right\}. \tag{5}$$

定理 3的证明由以下两个引理完成。

引理16. 对所有 $i \geq 0$，$L_i \in \mathsf{BQTISP}\left(n^{O(1)}, O\left(\frac{\log n}{(\log\log n)^i}\right)\right)$。

证明。 我们描述一台多项式时间确定性图灵机 M_i （图2），它识别L_i。

(A) 若 w 的上轨道上的字符串是 L_S 的成员 u_n，且其末段为$\mathrm{bin}(k_n)$，则将值$|\mathrm{bin}(k_n)|$与$|\mathrm{bin}(|\mathrm{bin}(k_n)|)|$以二进制形式存储在工作带上。否则，拒绝。(B) 计算并将数 $l = \left\lceil \frac{|\mathrm{bin}(k_n)|}{|\mathrm{bin}(|\mathrm{bin}(k_n)|)|^i} \right\rceil$ 以二进制形式写到工作带上。(C) 写入一个1，后接若干个 l 0，以将数 $j = 2^l$ 存储在工作带的指定区域。(D) 拒绝若 w 的下轨道上的字符串不是形如 $p\#^*$的形式，其中 $p \in \{0,1\}^j$。(E) 若 p 是回文串，接受。否则，拒绝。

图2：确定性图灵机 M_i。 w 表示长度为 n 的输入字符串。

在阶段（A）中， M_i 运行上述 [18] 描述的 $O(\log\log n)$ 空间算法，以确定其输入的上轨道是否属于 L_S 的某个成员 u_n。如果检查失败， M_i 将拒绝输入并停机。否则， M_i 使用一段 $O(\log\log n)$ 位的工作带空间，写下字符串 $\mathrm{bin}(k_n)$ 的长度（方程 4）。 M_i 对该表示中的数字进行计数，从而在带上得到数字 $|\mathrm{bin}(|\mathrm{bin}(k_n)|)|$ 更短的二进制表示。在阶段（B）中计算所需的算术运算也可以在 $O(\log\log n)$ 空间内完成。

阶段（C）使用 $\left\lceil \frac{|\mathrm{bin}(k_n)|}{(\log|\mathrm{bin}(k_n)|\rfloor+1)^i} \right\rceil + 1$，即 $O\left(\frac{\log n}{(\log\log n)^i}\right)$ 位，在带上写下数字 2^l。同一段空间还用作计数器，以便在阶段（D）检查 w 的下轨道是否符合要求的形式。如果控制流程到达阶段（E），只需 $O(\log|p|)$ 位（即同样只需 $O\left(\frac{\log n}{(\log\log n)^i}\right)$ 位），即可通过检查 p 是否对所有 $m \in \{1,\ldots,|p|\}$ 都满足其第 m 个符号与第 $(|p|-m+1)$ 个符号相同，来判断它是否为回文串。

由于其空间使用量受对数函数限制，且不会进入无限循环，因此M_i的运行时间受到多项式上界的限制。最后我们注意到

$$\mathsf{TISP}\left(n^{O(1)}, O\left(\frac{\log n}{(\log\log n)^i}\right)\right) \subseteq \mathsf{BQTISP}\left(n^{O(1)}, O\left(\frac{\log n}{(\log\log n)^i}\right)\right)$$

对所有 i 都成立，因为量子图灵机可以精确模拟确定性图灵机。 □

引理 17. 对于所有 $i \geq 0$, $L_i \notin \mathrm{BQTISP}_{\mathbb{C}}\left(n^{O(1)}, o\left(\frac{\text{对数}\, n}{(\text{对数对数}\ n)^{i}}\right)\right)$。

证明。 我们首先给语言 L_i 的非正则性设定界限。令

$$N_k = |\mathrm{bin}(0)\$\mathrm{bin}(1)\$\cdots\$\mathrm{bin}(k)|$$

是 L_S 中第 k 短成员的长度。因此， L_i 可能包含成员字符串的长度恰好是 N_k 这些数，其中 $k > 0$。

考虑这样一个“已填充”的长度 N_k。根据方程 (4),

$$\frac{1}{2}\log N_k \leq |\mathrm{bin}(k)| \leq \log N_k. \tag{6}$$

每个长度为 N_k 的成员 s 的下轨道都包含一个回文串 $p \in \{0,1\}^{j_k}$，后跟 # 符号，其中

$$j_k = 2^{\left\lceil \frac{|\mathrm{bin}(k)|}{|\mathrm{bin}(|\mathrm{bin}(k)|)|^i} \right\rceil},$$

这是一个偶数。令 l_s 表示 s 的长度为 $j_k/2$ 的前缀，也就是说， l_s 的下轨道是构成 s 的 j_k 位回文串的左半部分。对于每个 s，定义 r_s 为满足 $l_s r_s = s$ 的后缀。

令 W_{N_k} 为集合 $\{w \mid w = l_s, s \in L_i \cap \Sigma^{N_k}\}$。

由于 W_{N_k} 中的任意两个不同字符串 l_s、 $l_{s'}$ 都可由字符串 r_s区分，因此有

$$D_{L_i}(N_k) \geq |W_{N_k}| = 2^{j_k/2} = 2^{2^{\left\lceil \frac{|\mathrm{bin}(k)|}{|\mathrm{bin}(|\mathrm{bin}(k)|)|^i} \right\rceil - 1}}. \tag{7}$$

利用方程（6）以及以下事实

$$|\mathrm{bin}(m)| = \lfloor \log m \rfloor + 1$$

对于任意 $m > 0$，我们得到

$$\left\lceil \frac{|\mathrm{bin}(k)|}{|\mathrm{bin}(|\mathrm{bin}(k)|)|^i} \right\rceil \geq \frac{1}{2} \frac{\log N_k}{((\log\log N_k)+1)^i} \frac{(\log\log N_k)^i}{(\log\log N_k)^i},$$

这证明了，对于任意 $d_0 \in (0<\text{style id="2"}>, {}_1{}_2)$,

$$\left\lceil \frac{|\mathrm{bin}(k)|}{|\mathrm{bin}(|\mathrm{bin}(k)|)|^i} \right\rceil \geq d_0 \frac{\log N_k}{(\log\log N_k)^i} \tag{8}$$

对于所有充分大的 k，由于

$$\frac{(\log\log N_k)^i}{((\log\log N_k)+1)^i} \longrightarrow 1 \qquad (k \to \infty)$$

对于固定的 i。

利用不等式（7）和（8）以及以下事实

$$\frac{\log N_k}{(\log\log N_k)^i} \longrightarrow \infty \qquad (k \to \infty),$$

这是容易的以证明存在一个常数 $d_1 \in (0, {}_1{}_2)$，使得对于所有充分大的 k,

$$D_{L_i}(N_k) \geq 2^{2^{d_1 \frac{\log N_k}{(\log\log N_k)^i}}}. \tag{9}$$

现在，我们将这个界从有内容长度推广到所有充分大的长度。函数$D_{L_i}(n)$关于 n 是非递减的，因为一组两两（L_i, m）-不相似的字符串，对于每个 $n \geq m$，也两两（L_i, n）-不相似。令 n 充分大，并选取 k 使得

$$N_k \leq n < N_{k+1}.$$

连续有内容长度之间的间隔为

$$N_{k+1} - N_k = 1 + |\mathrm{bin}(k+1)|.$$

由于 $k < N_k \leq n$，我们有

$$|\mathrm{bin}(k+1)| \in O(\log n),$$

因此

$$n - N_k \in O(\log n).$$

特别地，对于所有充分大的 n,

$$N_k \geq \frac{n}{2}.$$

由此可得

$$\frac{\log N_k}{(\log\log N_k)^i} \geq c_0 \frac{\log n}{(\log\log n)^i}$$

对于某个常数 $c_0 > 0$ 以及所有充分大的 n。结合D_{L_i} 和方程（9），我们得到

$$D_{L_i}(n) \geq D_{L_i}(N_k) \geq 2^{2^{d_1 \frac{\log N_k}{(\log\log N_k)^i}}} \geq 2^{2^{d_2 \frac{\log n}{(\log\log n)^i}}} \tag{10}$$

对于某个常数 $d_2 > 0$ 以及所有充分大的 n。

假设，为了导出矛盾， that

$$L_i \in \mathsf{BQTISP}_{\mathbb{C}}\left(n^{O(1)}, o\left(\frac{\log n}{(\log\log n)^i}\right)\right).$$

那么，存在一台有界错误量子图灵机识别 L_i，其期望时间为 $t(n) \in n^{O(1)}$，最坏情况空间为

$$s(n) < a_0 \frac{\log n}{(\log\log n)^i} \tag{11}$$

对于任意实数 $a_0 > 0$，对于所有充分大的 n。

根据方程（10），

$$\log\log D_{L_i}(n) \geq d_2 \frac{\log n}{(\log\log n)^i}$$

对于所有充分大的 n。因此

$$s(n) \in o(\log\log D_{L_i}(n)),$$

也就是说， $s(n)$ 足够小，因此可以应用定理 6 。因此，存在某个常数 $b_0 > 0$,

$$t(n) \in \Omega\left(2^{-b_0 s(n)} D_{L_i}(n)^{2^{-b_0 s(n)}}\right).$$

利用方程（10），我们得到

$$t(n) \in \Omega\left(2^{2^{F(n)} - b_0 s(n)}\right),$$

其中我们使用缩写

$$F(n) = d_2 \frac{\log n}{(\log \log n)^i} - b_0 s(n).$$

我们将证明，在给定空间界下， $t(n)$ 的这个下界是超多项式的，从而得到矛盾。为此，只需证明指数

$$2^{F(n)} - b_0 s(n)$$

属于 $\omega(\log n)$。利用方程（11），对于任意正的 a_0 以及所有充分大的 n，可得

$$F(n) > d_2 \frac{\log n}{(\log \log n)^i} - b_0 a_0 \frac{\log n}{(\log \log n)^i}.$$

取 a_0的一个合适值，可得

$$F(n) > \frac{d_2}{2} \frac{\log n}{(\log \log n)^i},$$

因此我们有

$$\frac{2^{F(n)}}{\log n} > \frac{2^{\frac{d_2}{2} \frac{\log n}{(\log \log n)^i}}}{\log n}.$$

对右侧取对数，得到

$$\frac{d_2}{2} \frac{\log n}{(\log \log n)^i} - \log \log n,$$

由于 i 是固定的，它趋于无穷大。因此

$$2^{F(n)} \in \omega(\log n).$$

另一方面，

$$b_0 s(n) \in o(\log n).$$

因此

$$2^{F(n)} - b_0 s(n) \in \omega(\log n).$$

通过证明 $t(n)$ 必须属于 $n^{\omega(1)}$ 得到了矛盾，我们得出结论：

$$L_i \notin \mathsf{BQTISP}_{\mathbb{C}} \left(n^{O(1)}, o\left(\frac{\log n}{(\log \log n)^i} \right) \right).$$

□

5 结论性评论

最后，我们提出一些开放问题。

- 定理 2 能否从蒙特卡罗空间加强为最坏情况空间？具体而言，对于每个可计算的、非递减的 $s(n) \in \omega(1)$，

$$\mathsf{BQTISP}(n^{O(1)}, O(1)) \subsetneq \mathsf{BQTISP}(n^{O(1)}, O(s(n)))?$$

- 定理 3 给出了 $\Theta(\log n/(\log \log n)^i)$ 处的空间阈值，对任意 i。我们认为，在整个范围内，对于更一般的空间函数，也可以得到类似的分离结果

$$\omega(\log \log n) \cap o(\log n).$$

- 我们的技术显然仅适用于次对数空间。如何才能为多项式时间量子计算确立超对数空间阈值?

人工智能的使用

引理15 是经过与 ChatGPT 5.5 多轮交流后形成的一系列思路的成果，证明中最终矛盾的构造则归功于 ChatGPT。定理$\log n\,(\log\log\,n)^i$中使用的特定函数模板由 ChatGPT 提供。本文还受益于 ChatGPT 6 对先前版本的反复审阅。作者对手稿内容承担全部责任。

致谢

作者感谢 Utkan Gezer 为图中的插图准备 LaTeX 基础设施。特别感谢并致以爱意给亲爱的朋友 Gülay Ayyıldız Yiğitcan 和 Kemal Yiğitcan，以及他们的猫 Tosun 和 Sali。在本研究开始时，我们遭遇住房紧急情况，他们接待了我们一家一个月。

参考文献

[1] A. Ambainis 和 J. Watrous. 具有量子态和经典态的双向有限自动机。《理论计算机科学》，287(1):299–311，2002年。[2] S. Arunachalam 和 J. F. Doriguello. 矩阵超收缩性、流式算法与局部可解码码：大字母表情形。《*ACM*计算理论汇刊》，16(4)，2024年。doi: 10.1145/3688824。论文 21。[3] P. Beame、N. Kornerup 和 M. Whitmeyer. 矩阵问题的量子时间—空间权衡。《*SIAM*计算杂志》，55(3):469–519，2026年。doi: 10.1137/24M1710164。[4] C. Dwork 和 L. Stockmeyer. 双向概率有限状态自动机的时间复杂度间隙。《*SIAM*计算杂志》，19(6):1011–1023，1990年。[5] R. Freivalds 和 M. Karpinski. 随机化计算的空间下界。载于《自动机、语言与编程。*ICALP 1994*》，《计算机科学讲义》820。施普林格，1994年。[6] Y. Hamoudi 和 F. Magniez. 查找多个碰撞对的量子时间—空间权衡。载于 M.-H. Hsieh 编，第*16*届量子计算、通信与密码学理论会议（*TQC 2021*），第197卷《莱布尼茨国际信息学会议论文集》（*LIPIcs*），第 1:1–1:21 页，德国达格施图尔，2021年。Schloss Dagstuhl – Leibniz-Zentrum für Informatik。doi: 10.4230/LIPIcs.TQC.2021.1。[7] T. Heinosaari 和 M. Ziman.《量子理论的数学语言》。剑桥大学出版社，2012年。[8] M. Karpinski 和 R. Verbeek. 关于蒙特卡洛空间可构造函数及概率复杂性类的分离结果。《信息与计算》，75:178–189，1987年。[9] C. Löh.《几何群论》。施普林格·查姆，2017年。[10] M. L. Minsky. Post 的“Tag”问题递归不可解及图灵机理论中的其他主题。《数学年刊》，74(3):437–455，1961年。

[11] Z. Remscrim. 单量子比特的力量：双向量子有限自动机与字问题。载于 A. Czumaj、A. Dawar 和 E. Merelli 编，自动机、语言与程序设计第47届国际研讨会（ICALP 2020），第168卷《莱布尼茨国际信息学会议论文集》（LIPIcs），第 139:1–139:18 页，德国达格施图尔，2020年。Schloss Dagstuhl – Leibniz-Zentrum für Informatik。（完整版：https://eccc.weizmann.ac.il/ report/2019/107/）。[12] Z. Remscrim. 双向量子有限自动机和亚对数空间量子图灵机运行时间的下界。载于 J. R. Lee 编，理论计算机科学第12届创新会议（ITCS 2021），第185卷《莱布尼茨国际信息学会议论文集》（LIPIcs），第 39:1–39:20 页，德国达格施图尔，2021年。Schloss Dagstuhl – Leibniz-Zentrum für Informatik。（完整版：https://arxiv.org/abs/2003.09877）。[13] A. C. C. Say. 亚对数空间量子计算的时间层次。《理论计算机科学》，1061:115639，2026年。doi: 10.1016/j.tcs.2025.115639。[14] A. C. C. Say. 面向亚对数空间验证器的简短实用量子证明。载于 E. Formenti 和 L. Manzoni 编，《非常规计算与自然计算》，第 88–99 页，Cham，2026年。Springer Nature Switzerland。ISBN 978-3-032-15641-9。[15] A. C. C. Say. 双对数空间以外的亚指数时间量子优势，2026年。URL https://arxiv.org/abs/2601.16695。[16] A. C. C. Say 和 A. Yakaryılmaz. 魔法硬币对小空间量子机器很有用。《量子信息与计算》，17(11–12):1027–1043，2017年。[17] J. Shallit 和 Y. Breitbart. 自动性 I：一种描述复杂性度量的性质。《计算机与系统科学杂志》，53:10–25，1996年。[18] R. E. Stearns、J. Hartmanis 和 P. M. Lewis. 内存受限计算的层次。载于第6届开关电路理论与逻辑设计年会（SWCT 1965），第 179–190 页，1965年。doi: 10.1109/FOCS.1965.11。[19] D. van Melkebeek 和 T. Watson. 量子计算的时间空间高效模拟。《计算理论》，8(1):1–51，2012年。doi: 10.4086/toc.2012.v008a001。[20] J. Watrous. 关于模拟空间有界量子计算的复杂性。《计算复杂性》，12:48–84，2003年。