arXiv:2304.07227v1 [math.LO] 14 Apr 2023

# 关于实数表示及此类表示之间转换的计算复杂性

Amir M. Ben-Amram* LarsKristiansen†
Jakob Grue Simonsen‡

2023年4月17日

摘要

我们研究将一种实数表示转换为另一种表示的计算复杂性。典型的表示包括柯西序列、十进制展开、戴德金分割和连分数。

## 1 引言

### 1.1 研究动机。

在计算环境中，实数可以用柯西序列、十进制展开、戴德金分割、连分数以及其他许多表示方法来表示（本文将探讨其中相当多种）。我们的目标是分析将一种表示转换为另一种表示的计算复杂性。假设我们可以访问实数 $\alpha$ 的戴德金分割，那么计算出 $\alpha$ 的柯西序列有多难？计算出 $\alpha$ 的连分数有多难？或者，假设我们可以访问 $\alpha$ 的连分数，那么计算出 $\alpha$ 的十进制展开又有多难？是否存在高效算法？能否在多项式时间内完成？指数时间呢？

这些问题问得十分自然，但也很天真，而且按上述方式提出这些问题，几乎毫无意义。我们将尽力以数学上令人满意的方式提出这类问题，

*基里亚特奥诺，以色列

†数学系，奥斯陆大学，挪威；以及信息学系，奥斯陆大学，挪威

‡哥本哈根大学计算机科学系（DIKU），丹麦

在可能的情况下，我们还将推导出转换计算复杂性相当紧确的上界。我们涵盖了大多数经典且广为人知的表示，也会考察一些较新的表示。

很可能存在一种算法，可以将一种表示转换为另一种表示，但仍然无法推导出该算法所需计算资源的任何上界。在这种情况下，我们会直观地解释其中的原因（大多省略完整证明，因为相关结果可见参考文献 [23] [24] [16]）。

## 1.2 什么是表示?

形式上，无理数的一种表示是一类函数。该类中的每个函数都表示一个特定的无理数，而每个无理数都由该类中的某个函数表示。所有无理数的戴德金分割构成一种对我们而言的典范表示：无理数 $\alpha$ 的戴德金分割是函数 $\alpha$ : $\mathbb{Q} \longrightarrow \{0,1\}$ ，其中

$$\alpha(q) = \begin{cases} 0 & \text{if } q < \alpha \\ 1 & \text{if } q > \alpha. \end{cases}$$

该类中的每个无理数都有唯一的表示，因此我们可以将无理数 $\alpha$ 与其戴德金分割 $\alpha$ : $\mathbb{Q} \longrightarrow \{0,1\}$等同起来。

我们将利用戴德金分割的唯一性来定义一般意义上的表示。我们把表示中的函数 $R$称为 $R$-表示。当 $f$ 是 $\alpha$的一个 $R$-表示时，我们要求能够计算出 $\alpha$ 在 $f$中的戴德金分割；也就是说，我们要求存在一台预言机图灵机 $M$ ，使得

$$\alpha(q) = \Phi_M^f(q)$$

其中 $\Phi_M^f$ 是 $M$ 以 $f$为预言机时计算出的函数。我们还要求至少有一个 $\alpha$ 的 $R$-表示 $f$ 能够由 $\alpha$的戴德金分割计算出来；也就是说，我们要求存在一台预言机图灵机 $N$ ，使得

$$f(x) = \Phi_N^\alpha(x)$$

其中 $\Phi_N^\alpha$ 是 $N$ 以 $\alpha$的戴德金分割为预言机时计算出的函数。现在我们可以给出正式定义了。

定义 1.1。一类函数 $R$ 称为（无理数的）表示，如果存在图灵机 $M$ 和 $N$ ，使得

1. 对于每个无理数 $\alpha$ ，都存在 $f \in R$ ，使得

$$\alpha = \Phi_M^f \text{ and } f = \Phi_N^\alpha$$

2. 对于每个 $g \in R$，都存在一个无理数 $\alpha$，使得

$$\alpha = \Phi_M^g = \Phi_M^f \text{ where } f = \Phi_N^\alpha$$

当 $\alpha = \Phi_M^g$ 时，我们称 $g$ 表示 $\alpha$，并称 $g$ 是 $\alpha$ 的一个 $R$-表示。

如果对于任意表示 $\alpha$ 的 $f \in R_1$，都存在表示 $\alpha$ 的 $g \in R_2$，使得 $g = \Phi_M^f$，那么我们称预言机图灵机 $M$ 将 $R_1$-表示转换为 $R_2$-表示。

□

让我们研究几个例子，看看这个定义如何发挥作用。我们将 $\alpha$ 的柯西序列定义为一个函数 $C: \mathbb{N}^+ \longrightarrow \mathbb{Q}$，它具有如下性质

$$|C(n) - \alpha| < n^{-1} .$$

设 $\mathcal{C}$ 为所有无理数的所有柯西序列构成的类。现在我们将论证，根据上述定义，$\mathcal{C}$ 是一种表示。

首先我们注意到，如果能访问 $\alpha$ 的戴德金分割，就可以计算出 $\alpha$ 的柯西序列 $C$。我们可以利用戴德金分割找到一个整数$a$，使得 $a < \alpha < a + 1$。此后，我们可以利用戴德金分割以及以下等式

$$C(1) = a + \frac{1}{2} \quad \text{and} \quad C(i+1) = \begin{cases} C(i) - 2^{-i-1} & \text{if } C(i) > \alpha \\ C(i) + 2^{-i-1} & \text{if } C(i) < \alpha \end{cases}$$

对任意 $n$计算 $C(n)$。这是计算戴德金分割中的一个柯西序列的一种可能方法。当然还存在其他方法。其他算法可能生成收敛速度比上述算法计算出的序列更快或更慢的柯西序列。无论如何，都会存在一台预言机图灵机 $N$，使得 $f = \Phi_N^\alpha$，其中 $f$ 是 $\alpha$的某个柯西序列。

接下来我们注意到，可以通过 $\alpha$的任意柯西序列计算无理数 $\alpha$ 的戴德金分割。为了计算 $\alpha(q)$，我们搜索满足 $|C(n) - q| > n^{-1}$的最小$n$。由于 $q$ 是有理数而 $\alpha$是无理数，这个搜索会终止。如果 $q < C(n)$，那么 $\alpha(q) = 0$（我们有 $q < \alpha$）；否则，我们有 $q > C(n)$，于是 $\alpha(q) = 1$（我们有$q > \alpha$）。因此，存在一台预言机图灵机 $M$，使得只要 $f$ 是 $\alpha$的一个柯西序列，就有 $\alpha = \Phi_M^f$。

这表明 $\mathcal{C}$，也就是说，所有无理数的所有柯西序列组成的类，是一种表示。我们有图灵机 $M$ 和 $N$，使得对于任意无理数 $\alpha$，都存在一个柯西序列 $C$，满足 $\alpha = \Phi_M^C$ 和$C = \Phi_N^\alpha$，因此定义 1.1 的第 (1) 条得以满足。此外，第 (2) 条也得以满足，因为对于 $\alpha$的任意柯西序列 $g$，我们都有 $\alpha = \Phi_M^g = \Phi_M^C$。

设 $a_0, a_1, a_2, \ldots$ 是一个整数无限序列，其中 $a_1, a_2, a_3 \ldots$ 为正数。连分数 $[a_0; a_1, a_2, \ldots]$ 定义为

$$[\, a_0; a_1, a_2, a_3 \ldots \,] \;=\; a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ldots}}}$$

众所周知，任何无理数都可以唯一地写成一个无限连分数；此外，每个无限连分数都等于一个无理数（有理数有有限连分数）。无限连分数与无理数之间存在一一对应关系。

令 $\mathcal{F}$ 为所有满足每个 $[a_0$ 的无限连分数的类； $a_1, a_2, a_3 \ldots]$ 被视为一个函数 $f$，其中 $f(n) = a_n$。那么， $\mathcal{F}$ 将是符合定义 1.1 的表示。若我们能够访问 $\alpha$ 的戴德金分割，就可以计算出 $\alpha$ 的连分数；若能够访问 $\alpha$ 的连分数，就可以计算出 $\alpha$ 的戴德金分割。因此，存在图灵机 $M$ 和 $N$，使得

$$\alpha = \Phi_M^f \text{ and } f = \Phi_N^\alpha$$

只要 $f$ 是无理数 $\alpha$ 的连分数即可。这表明定义 1.1 的条件 (1) 得到了满足。由于连分数的唯一性，条件 (2) 显然也得到满足；也就是说，无限连分数与无理数的戴德金分割之间存在一一对应关系。

我们不会正式定义所有实数的表示，而且定义 1.1 限定于无理数是至关重要的。若将有理数也纳入其中，该定义便无法实现其目的，因为我们并非总能将一种标准表示一致地转换为另一种，即便所处理的是可计算性质的表示。例如，上述将柯西序列 $C$ 转换为戴德金分割的算法，会搜索一个数 $n$，使得 $|C(n) - q| > n^{-1}$。当 $C$ 是有理数 $q$ 的柯西序列时，这项搜索不会终止。因此，如果 $\alpha$ 可能是有理数，该算法就无法给出图灵机 $M$，使得每当 $C$ 是 $\alpha$ 的柯西序列时，$\Phi_M^C$ 便是 $\alpha$ 的戴德金分割；此外，可以证明不存在这样的图灵机 $M$（参见 Mostowski [28]）。我们无法将实数的柯西序列一致地转换为戴德金分割，但可以将无理数的柯西序列一致地转换为戴德金分割。

定义 1.1 的目的是刻画我们直觉上认为可计算的实数表示；或许有些自相矛盾的是，我们恰恰是通过将定义限制在无理数上来做到这一点的。读者应当注意，在可计算分析中，实数和无理数表示的标准概念往往比我们的概念更一般，参见 Weihrauch [33]。例如，一个包含所有小于 $\alpha$ 的有理数、且不包含其他数的有理数序列 $q_0, q_1, q_2, \ldots$，按照我们的定义并不构成一种表示。我们不能用这样的序列来计算 $\alpha$ 的戴德金分割（即使 $\alpha$ 是无理数）。在可计算分析文献中，这类序列通常会被视为一种表示。

## 1.3 表示上的序关系。

我们已经看到，对于任意一个 $\alpha$ 的柯西序列，都有一种算法可以计算无理数 $\alpha$ 的戴德金分割。该算法会搜索满足特定条件的最小自然数 $n$。很容易看出，这种无界搜索是必要的。如果不允许进行无界搜索，我们就无法将柯西序列转换为戴德金分割。同样，如果不允许进行无界搜索，我们也无法将十进制展开转换为三进制展开。假设预言机告诉我们，某个无理数的十进制展开以 0.66666 开头。这还不足以让我们判断三进制展开是以 0.1 还是 0.2 开头。因此，为了确定应该选择这两个选项中的哪一个，我们必须向预言机询问十进制展开的下一位数字；但当然，下一位数字也可能是 6，再下一位也可能是 6。预言机可能会持续任意长的时间告诉我们下一位数字是 6。由于该数是无理数，预言机最终会给出一个数字，使我们能够判断三进制展开是以 0.1 还是 0.2 开头；但要得到这个数字，我们需要进行无界搜索。

不进行无界搜索的算法（转换、计算等）通常称为次递归算法（转换、计算等）；一般而言，次递归意味着不存在无界搜索。这种术语用法也许并不标准，但会非常方便。

现在我们将定义表示之间的序关系 $\preceq_S$。直观地说，关系 $R_1 \preceq_S R_2$ 表示表示 $R_2$ 比表示 $R_1$包含更多信息。如果 $R_1 \preceq_S R_2$ 成立，那么一台能够访问 $R_2$ 表示的 $\alpha$ 的预言机的图灵机，就可以通过次递归方式（是的，这意味着不进行无界搜索）计算出 $\alpha$ 的 $R_1$ 表示。因此，如果关系 $R_1 \preceq_S R_2$ 不成立，那么讨论将 $R_2$ 表示转换为 $R_1$ 表示的计算复杂性就没什么意义，因为这种转换需要无界搜索，因而执行转换的图灵机的运行时间将没有上界。另一方面，如果该关系成立，那么分析这种转换的计算复杂性就有意义。到目前为止，我们只是说明了引入这一关系的意图；下文将给出正式定义。不过，首先我们需要一个辅助定义。我们将用时间有界计算来表述定义，因此需要说明哪些函数可以作为时间界。我们采用时间可构造函数，定义如下。这是复杂性理论中的标准选择。

定义 1.2. 函数 $t\colon \mathbb{N} \longrightarrow \mathbb{N}$ 是时间界，当且仅当 (i) $n \leq t(n)$，(ii) $t$ 是递增的，且 (iii) $t$ 是时间可构造的：存在一台单带图灵机，在输入 $1^n$ 时，能在 $\Theta(t(n))$ 步内计算出 $t(n)$。 □

定义中的条款 (iii) 是必要的，因为有些函数的计算复杂性与其取值大小不成比例。我们不将这类函数用作时间界，从而避免证明中的某些陷阱。

我们允许用作时间界的函数类包含算法分析中熟悉的所有函数，例如多项式（具有正系数）、指数函数、塔函数等。

定义 1.3. 设 $t$ 为一个时间界， $R$ 为一种表示。那么，$O(t)_R$ 表示区间 (0, 1) 中所有无理数 $\alpha$ 的类，其中至少有一种 $R$- $\alpha$ 表示可由一台运行时间为$O(t(n))$的图灵机计算得到（其中 $n$ 是输入的长度）。

设 $R_1$ 和 $R_2$ 为两种表示。若对于任意时间界 $t$，都存在一个时间界 $s$ 使得下式成立，则关系 $R_1 \preceq_S R_2$ 成立：

$$O(t)_{R_2} \subseteq O(s)_{R_1} .$$

如果关系 $R_1 \preceq_S R_2$ 成立，我们就称表示 $R_1$ 在表示 $R_2$中是次递归的。

□

现在我们将研究几个例子，并讨论上述定义如何发挥作用。回想一下， $\mathcal{C}$ 表示通过柯西序列进行的表示（见第 3 页）。令 $\mathcal{D}$ 表示通过戴德金分割进行的表示。结果表明，我们有 $\mathcal{C} \preceq_S \mathcal{D}$ 和 $\mathcal{D} \not\preceq_S \mathcal{C}$。下面来看看原因。

直观地说，我们有 $\mathcal{C} \preceq_S \mathcal{D}$ ，因为区间 (0,1) 内无理数 $\alpha$的柯西序列可以在 $\alpha$的戴德金分割中通过子递归方式计算出来。无需进行无界搜索。我们只需令 $C(1)$ 等于 $1/2$，然后利用戴德金分割以及例如以下等式

$$C(i+1) = \begin{cases} C(i) - 2^{-i-1} & \text{if } C(i) > \alpha \\ (C(i) + 2^{-i-1} & \text{if } C(i) < \alpha \end{cases}$$

来计算 $C(n)$。于是 $C: \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 将是 $\alpha$的柯西序列。形式上，我们有 $\mathcal{C} \preceq_S \mathcal{D}$ ，因为对于每个时间界 $t$ ，都存在一个时间界 $s$ ，使得 $O(t)_{\mathcal{D}} \subseteq O(s)_{\mathcal{C}}$。利用上述等式计算 $C(n)$的图灵机需要计算戴德金分割 $O(n)$ 次，也就是 $O(2^{\|n\|})$ 次，其中 $\|n\|$是输入的长度。此外，假设数字以二进制形式表示，该图灵机只需对大小为 $O(2^{\|n\|})$的输入计算戴德金分割。因此，如果 $\alpha$ 的戴德金分割可在时间 $O(t(m))$ 内计算，其中 $m$ 是输入的长度，$t$ 是时间界，那么对于某个常数 $k$，图灵机就能在时间 $O(nt(k2^{\|n\|}))$ 内计算出 $\alpha$ 的柯西序列。因此，令 $s$ 为时间界$s(\|n\|) = nk't2^{\|n\|}$，取适当的 $k' > k$ ，我们有 $O(t)_{\mathcal{D}} \subseteq O(s)_{\mathcal{C}}$。

请注意，集合 $O(t)_{\mathcal{D}}$ 只包含区间 (0, 1) 内的无理数。这一点很重要。当我们为这个区间内的一个无理数计算柯西序列 $C$ 时，只需令 $C(1) = 1/2$。一般来说，我们无法在 $\beta$ 的戴德金分割中，以次递归方式计算无理数 $\beta$ 的柯西序列。为了设定 $C(1)$ 的值，我们需要一个有理数 $a$ ，使得 $a < \beta < a + 1$；若不诉诸无界搜索，我们便无法找到这样的 $a$ 。因此，如果我们不将 $O(t)_{\mathcal{D}}$ 限定为区间 (0, 1) 内的无理数，

关系 $\mathcal{C} \preceq_S \mathcal{D}$ 就不会成立。但我们希望它成立。我们关注的是无理数小数部分的表示。将整数部分抽象掉是很自然的。

现在，让我们讨论为什么我们有 $\mathcal{D} \not\preceq_S \mathcal{C}$。如上所述，我们有 $\mathcal{D} \not\preceq_S \mathcal{C}$，因为在计算柯西序列中的戴德金分割时，我们无法避免无界搜索；但形式证明是什么样的呢？一般来说，证明关系 $\preceq_S$ 不成立，要比证明它成立困难得多。为了证明 $R_1 \preceq_S R_2$，我们只需提出一个次递归算法，将一个 $R_2$-表示转换为一个 $R_1$-表示。为了证明 $R_1 \not\preceq_S R_2$，按照我们的定义，我们必须证明存在一个时间界 $t$，使得对于任意时间界$s$，都有 $O(t)_{R_2} \not\subseteq O(s)_{R_1}$。这可能并不容易，甚至可能需要复杂的对角化论证。在某些情况下，我们可以采用增长论证：也就是说，对于某个时间界 $t$，我们或许能够证明，对于任意时间界 $s$，都存在 $\alpha_s \in O(t)_{R_2}$，使得任何计算 $R_1$-表示的图灵机$\alpha_s$ 都必须给出非常长的输出，也就是说，输出的长度不属于 $O(s)$的量级（因此该图灵机无法在 $O(s)$时间内运行）。增长论证往往比对角化论证更容易，或者至少不那么繁琐；我们将在第8节中给出一个相当详细的、基于增长论证的证明。但我们无法通过增长论证证明 $\mathcal{D} \not\preceq_S \mathcal{C}$，因为计算戴德金分割的图灵机给出的输出长度为 1。让我们看看如何通过对角化论证证明 $\mathcal{D} \not\preceq_S \mathcal{C}$。

令 $s$ 为任意一个时间界。我们将通过标准的对角化技术，构造一个区间 $(0, 1)$ 内无理数 $\alpha$ 的柯西序列，使得 $\alpha$ 与每个 $\beta \in O(s)_{\mathcal{D}}$都不同，因此我们有$\alpha \notin O(s)_{\mathcal{D}}$。我们的构造可以由图灵机完成，也就是说，$\alpha$ 的柯西序列可以由图灵机计算。该图灵机将在时间 $O(t)$ 内运行，其中 $t$是某个时间界，因此我们有$\alpha \in O(t)_{\mathcal{C}}$。此外，结果将表明，$t$ 不依赖于 $s$。因此，存在一个时间界 $t$，并且对于每个时间界 $s$，都存在 $\alpha$ 使得$\alpha \in O(t)_{\mathcal{C}}$ 且 $\alpha \notin O(s)_{\mathcal{D}}$。因此，存在 $t$ 使得对于每个$s$，都有 $O(t)_{\mathcal{C}} \not\subseteq O(s)_{\mathcal{D}}$，从而根据定义 1.3，我们可以得出结论：$\mathcal{D} \not\preceq_S \mathcal{C}$。

现在，我们将给出一个算法，用于计算柯西序列 $C$，使得 $\lim_n C(n) \notin O(s)_{\mathcal{D}}$，其中 $s$ 是任意的时间界。我们需要一个标准的图灵机枚举 $\{e\}_{e\in\mathbb{N}}$，并用 $\{e\}(x)$ 表示第 $e$ 台图灵机在输入 $x$ 上的执行。此外，我们需要一个递增的时间界函数 $S$，它最终会超过任何阶为 $O(s)$ 的时间界；也就是说，对于任意阶为 $O(s)$ 的 $s_0$，当 $m$ 足够大时，都有 $s_0(m) < S(m)$。这样的 $S$ 总是存在，而且还可以选取得满足对所有 $n$，都有 $S(n) \geq 2n$。注意，根据时间界函数的定义，我们有一台图灵机，给定 $n$，可以在至多 $a_S \cdot S(n)$ 步内计算出 $S(n)$，其中 $a_S$ 是某个常数。我们定义序列 $d_0, d_1, d_2, \ldots$ 如下：$d_0 = 2$，并且 $d_{i+1} = S(d_i)$。最后，我们需要一个标准的可计算双射 $\langle\cdot,\cdot\rangle : \mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N}$。对于任意 $i \in \mathbb{N}$，我们的算法都需要计算出唯一的 $j, e \in \mathbb{N}$，使得 $\langle j, e\rangle = i$。

算法设置 $C(1) := 1/2$。如果 $n > 1$，算法会检查是否存在

满足 $n = d_{3i+2}$ 的 $i$。如果不存在这样的 $i$，算法就将 $C(n) := C(n-1)$；如果存在这样的 $i$，算法会找出唯一的 $j, e$，使得 $\langle j, e\rangle = i$，并设置

- $C(n) := C(n-1)$ 若 $\{e\}(C(n-1))$ 在 $n$ 步内不会终止
- $C(n) := C(n-1) - 2^{-n}$ 若 $\{e\}(C(n-1))$ 在 $n$ 步内终止且输出 0
- $C(n) := C(n-1) + 2^{-n}$ 若 $\{e\}(C(n-1))$ 在 $n$ 步内终止且输出非 0 的值。

显然，该算法确实计算出一个区间 (0, 1) 内某个实数的柯西序列；而且也很容易看出，如果我们用二进制形式表示数字，并以合理的方式对有理数编码，那么输出 $C(n)$ 的长度将受一个阶为 $O(n)$ 的函数所界定。不失一般性，我们可以假设 $S$ 最终超过任何阶为 $O(n)$ 的函数（只需选取一个增长足够快的 $S$ 即可）。因此，对于所有充分大的 $n$，有

$$\|C(n)\| < S(n)\ . \tag{1.1}$$

现在，令 $\alpha = \lim_n C(n)$，并令 $\beta \in O(s)_{\mathcal{D}}$。我们将证明 $\alpha \neq \beta$。

$\beta$ 的戴德金分割可以由一台运行时间为 $O(s)$ 的图灵机 $\{e\}$计算。因此，当 $x$ 足够大时，$S(\|x\|)$ 将是计算 $\{e\}(x)$ 所需步数的一个上界。取一个足够大的 $j$，并令$i = \langle j, e\rangle$（通过选取足够大的 $j$，我们可以让 $d_{3i}$ 任意大）。我们的算法经过设计，使得我们有

$$C(d_{3i}) = C(d_{3i+2} - 1)\ . \tag{1.2}$$

因此，根据 (1.2) 和 (1.1)，计算 $\{e\}(C(d_{3i+2}-1))$ 所需的步数有如下上界：

$$S(|C(d_{3i+2}-1)|) = S(|C(d_{3i})|) < S(S(d_{3i})) = d_{3i+2}\ .$$

假设计算 $\{e\}(C(d_{3i+2}-1))$ 的输出为 0。那么我们有

$$C(d_{3i+2}) = C(d_{3i+2}-1) + 2^{-d_{3i+2}}$$

但是，由于 $\{e\}$ 计算的是 $\beta$ 的戴德金分割，我们有 $\beta > C(d_{3i+2}-1)$。由此可得

$$\alpha = \lim_n C(n) < C(d_{3i+2}-1) - 2^{-d_{3i+2}} + \sum_{n > d_{3i+2}} 2^{-n} < C(d_{3i+2}-1) < \beta\ .$$

假设计算 $\{e\}(C(d_{3i+2}-1))$ 的输出不为 0。那么我们有

$$C(d_{3i+2}) = C(d_{3i+2}-1) + 2^{-d_{3i+2}}$$

但由于 $\{e\}$ 计算的是 $\beta$ 的戴德金分割，因此我们有 $\beta < C(d_{3i+2}-1)$。由此可得

$$\alpha = \lim_n C(n) > C(d_{3i+2}-1) + 2^{-d_{3i+2}} - \sum_{n>d_{3i+2}} 2^{-d_n} > C(d_{3i+2}-1) > \beta\,.$$

这证明了 $\alpha \neq \beta$。

同样的论证可以用来证明极限 $\alpha$ 是无理数。在所枚举的图灵机中，有一台机器 $e$ 能够计算任意给定有理数 $q$的戴德金分割；在合理的有理数编码下，这项计算可以在线性时间内完成，因此我们可以假设 $e$的运行时间为 $O(s)$；由此得出 $\alpha \neq q$。

我们已经证明，我们的算法计算出了一个无理数的柯西序列，而且该无理数不属于类 $O(s)_{\mathcal{D}}$。现在我们来分析执行该算法的图灵机 $M$ 的复杂度：

- $M$ 的输入是一个自然数 $n$ （我们将估计 $M$的运行时间关于 $n$的上界）。
- 首先， $M$ 将计算 $C(n-1)$。
- 然后， $M$ 将检查是否存在 $i$ 使得 $n = d_{3i+2}$。回想一下， $d_0 = 2$且 $d_{i+1} = S(d_i)$，其中 $S$ 是一个时间界函数。可以在 $O(n)$时间内检查是否存在这样的 $i$。简言之， $M$会依次计算 $d_0$、$d_1$、$d_2\ldots$，直到遇到满足 $n = d_j$的 $j$，或满足 $n < d_j$的 $j$，或遇到计算 $d_j$ 所需时间超过 $a_S n$ 的 $j$。在所有情况下，最后一个 $d_j$ 的计算要么已完成，要么会在经过 $O(n)$步后停止。序列中此前各项的计算也需要 $O(n)$，因为该序列至少按几何级数式地增长，因此其计算时间也如此增长。
- 如果 $M$ 找到满足 $n = d_{3i+2}$的 $i$，那么 $M$ 将计算 $j$、$e$，使得$\langle j, e\rangle = i$，检查计算 $\{e\}(C(n-1))$是否在 $n$步内终止，最后计算输出。所有这些都可以在多带图灵机上用 $O(n^2)$的时间完成。

这些考量表明， $M$ 在时间 $O(n^2)$内运行，其中 $n \in \mathbb{N}$ 是输入，因此也在时间 $O(2^{2\|n\|})$内运行，其中 $\|n\|$ 是输入的长度（时间复杂度总是表示为输入位长的函数）。由此我们可以得出， $M$ 为某个无理数计算柯西序列，该无理数属于类 $O(2^{2\|n\|})_{\mathcal{C}}$。我们还知道该无理数不属于类 $O(s)_{\mathcal{D}}$，并回忆一下， $s$是任意选取的时间界。因此，对于任意时间界 $s$，我们都有 $O(2^{2\|n\|})_{\mathcal{C}} \not\subseteq O(s)_{\mathcal{D}}$。这证明了 $\mathcal{D} \not\preceq_S \mathcal{C}$。

我们关于 $\mathcal{D} \not\preceq_S \mathcal{C}$ 的证明旨在说明我们的定义如何发挥作用。在本文中，我们通常不会正式证明一种表示相对于另一种表示不是次递归的，但为了帮助读者，我们会在一定程度上对为何某些表示之间不可能存在次递归转换提供非正式的解释和直观的论证。

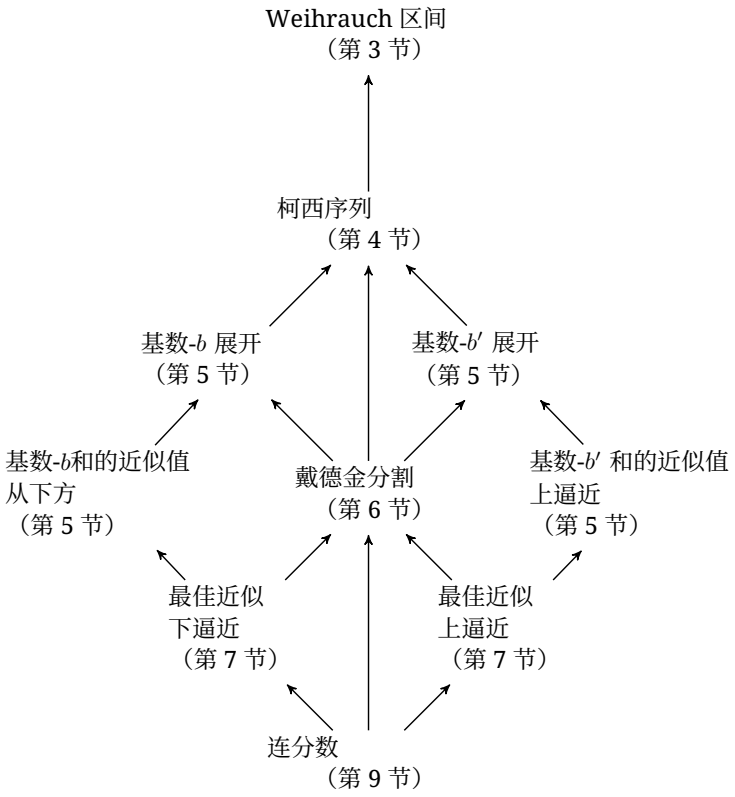


图 1：表示的子递归度（等价类）概览。

定义 1.4。设 $R_1$ 和 $R_2$ 为表示。当 $R_1 \preceq_S R_2$ 和 $R_2 \preceq_S R_1$时，关系 $R_1 \equiv_S R_2$成立。如果关系 $R_1 \equiv_S R_2$ 成立，我们就称表示 $R_1$ 与表示$R_2$子递归等价。

The relation $R_1 \prec_S R_2$ holds when $R_1 \preceq_S R_2$ and $R_2 \not\preceq_S R_1$. □

等价关系 $\equiv_S$ 在这些表示上诱导出一个度结构。图 1 中的有向图概述了若干自然度（等价类）之间的关系。节点表示表示的度，每个度都以该度中一种最广为人知的表示标记。对于两个表示 $R_1$ 和 $R_2$，从标记为 $R_1$ 的节点到标记为 $R_2$ 的节点存在一条有向路径，当且仅当 $R_2 \prec_S R_1$。因此，如果存在一条从 $R_1$ 到 $R_2$的有向路径，我们就能将一个 $R_1$表示次递归地转换为一个 $R_2$表示；如果不存在从 $R_1$ 到 $R_2$的有向路径，我们就无法将一个 $R_1$表示次递归地转换为一个

$R_2$表示。不幸的是，我们无法准确描绘基数为$b$ 的展开的度与从下方和上方逼近的基数为$b$的和近似的度之间复杂的关系（其中 $b = 2$取 3、4、.....），但我们的图像大致展示了这一领域的情况。关于这些度彼此之间的关系，更多内容请见第 5 节。

### 1.4 我们的目标及一些参考文献。

我们提出一种基于图灵机和标准复杂性理论的（度）表示理论（关于实数）。这一理论应被视为对 Kristiansen[23] [24] 和 Georgiev 等人 [16] 所发展的理论的重述与改进；后者基于诚实函数和次递归类。这两种研究实数表示及其间转换的方法本质上是相同的，尽管例如，可约关系 $\preceq_S$ 从未在其他论文中得到正式定义；而且，从 [23] [24] [16] 中证明的结果可以较为直接地推知，图 1 所示的图景是正确的。

在诸如 [23] [24] [16], 以及 Georgiev [15] 和 Kristiansen[25], 的论文中，作者只关注从一种表示转换到另一种表示的次递归转换是否存在。他们从不分析次递归转换的计算复杂性，也不致力于寻找高效的转换。在本文中，我们将关注这些问题；事实上，这些问题将是我们的主要关注点：我们将对用于将一种表示转换为另一种表示的预言机图灵机的运行时间设定严格的上界。我们还将给出所需预言机调用的次数及这些调用规模的上界。

在第 4 节中，我们研究与柯西序列表示子递归等价的表示之间的转换。在第 5 节中，我们研究与基数为 $b$ 的展开表示和基数为 $b$ 的和近似表示子递归等价的表示之间的转换。在第 6 节中，我们讨论与戴德金分割表示子递归等价的表示；随后在第 7 节中，讨论与左右最佳逼近表示子递归等价的表示。最后，在第 9 节中，我们研究与连分数表示子递归等价的表示之间的转换。参见图 1。

## 2 预备知识

### 2.1 预言机图灵机与复杂性理论。

我们假设读者对可计算性和计算复杂性有基本了解（标准教材包括 Sipser [30], 、Du 与 Ko [13] 以及 Arora 与 Barak [1]）。

我们将使用能够访问待转换表示的预言机的图灵机。除非另有说明，否则假定 $\mathbb{N}$ 的元素在输入带、查询带和输出带上均以二进制表示书写，

最低有效位在前。假定整数对$(p,q)$采用交错表示法书写（即先写 $p$ 的二进制表示的第一位，再写 $q$的二进制表示的第一位，依此类推）。注意，整数对$(p,q)$的表示长度为 $O(\log \max\{p,q\})$。假定元素$p/q \in \mathbb{Q}$ 由$(p,q)$的表示来表示。我们用 $\|x\|$表示 $x$ 的二进制表示长度。

函数预言机图灵机广泛应用于实数集上函数的复杂性理论中（例如参见 Ko [20]），接下来的定义是一个标准定义。

定义 2.1. 参数化函数预言机图灵机是一种（多纸带）图灵机 $M=(Q,q_0,F,\Sigma,\Gamma,\delta)$，具有初始状态 $q_0 \in Q$、终止状态 $F \subseteq Q$、输入字母表和纸带字母表 $\Sigma$ 和 $\Gamma$ （其中 $\Sigma \subseteq \Gamma$ 和 $\{\sqcup\} \subseteq \Gamma \setminus \Sigma$），以及部分转移函数 $\delta$，并且 $M$ 具有一条特殊的查询纸带和两个不同的状态 $q_q$、 $q_a \in Q$ （查询状态和应答状态）。

执行前，须为 $M$ 提供一个全函数 $f:(\Gamma \setminus \{\sqcup\})^* \longrightarrow (\Gamma \setminus \{\sqcup\})^*$ （即预言机），然后才能在任何输入上执行。若 $f$ 已固定，我们用 $M^f$ 表示 $M$。我们用 $\Phi_M^f$表示由 $M^f$计算出的函数。

$M^f$ 的转移关系与通常图灵机的定义相同，但查询状态 $q_q$除外：如果 $M$ 进入状态 $q_q$，令 $x$ 为当前位于查询纸带上的字； $M$ 在一步内转移到状态 $q_a$，查询纸带的内容随即变为 $f(x)$。查询纸带读写头重置到起点，而其他读写头保持不动。函数预言机机的时间和空间复杂度按通常图灵机的方式计算，其中从 $q_q$ 到 $q_a$ 的转移耗费 $\|f(x)\|$ 个时间步。查询的输入规模为 $M$ 进入状态 $q_q$时查询纸带上非空符号的数量。

□

在实数计算的其他研究中，表示之间的可约性已有成熟的概念；粗略地说，这一概念要求将表示作为一个无限字符串写在第二型图灵机 [22, 34, 32, 8]的一条输入纸带上。在这种框架下，例如，函数 $f:\ \mathbb{Q} \cap [0,1] \longrightarrow \{0,\ldots,b-1\}$最自然的表达方式是对其定义域施加一种可计算排序（例如，有理数按分母非递减的顺序排列），并按照这一顺序将函数值 $f(q)$ 编码为位串。我们强烈猜测，我们的结果经过适当调整后也适用于第二型框架。

## 2.2 一些记号。

如果 $f:\ \mathbb{N} \longrightarrow \mathbb{N}$ 被一个关于 $n$的、系数为正整数的多项式从上界定，则记作 $f(n)=\mathrm{poly}(n)$；如果 $f$ 被一个关于 $\log n$ 的、系数为正整数的多项式从上界定，则记作 $f(n)=\mathrm{polylog}(n)$。

我们用记号 $f^{(n)}$ 表示函数 $f:\ \mathbb{N} \to \mathbb{N}$ 的第 $n$ 次迭代，即 $f^{(0)}(x)=x$和 $f^{(n+1)}=f \circ f_{(n)}$。注意，上标中的括号

位置将此记号与普通乘方区分开来。这种记号常与 $\lambda$ 记号搭配使用，例如：

$$(\lambda x.g(x))^{(4)}(0) = (\lambda x.g(x))(\lambda x.g(x))^{(3)}(0)$$
$$= g((\lambda x.g(x))^{(3)}(0)) = \cdots = g(g(g(g(0)))) .$$

## 2.3 法里数列与斯特恩-布罗科特树。

法里数列是由 0 和 1 之间的分数组成的严格递增序列。阶数为 $k$ 的法里数列记作 $F_k$，其中包含所有以最简形式表示时分母小于或等于 $k$ 的分数。例如，$F_5$ 是序列

$$0/1\ ,\ 1/5\ ,\ 1/4\ ,\ 1/3\ ,\ 2/5\ ,\ 1/2\ ,\ 3/5\ ,\ 2/3\ ,\ 3/4\ ,\ 4/5\ ,\ 1/1\ .$$

法里数列中两个连续分数组成的有序对称为法里对。设 $(a/b, c/d)$ 是一个法里对。分数 $(a+c)/(b+d)$ 称为 $a/b$ 和 $c/d$ 的中项。下一个定理最初由柯西[10]于 1826年 证明。

定理 2.2。设 $(a/b, c/d)$ 是一个法里对。(i) 有 $cb - ad = 1$（或者等价地，$c/d - a/b = 1/(bd)$）；(ii) 中项 $(a+c)/(b+d)$ 是最简形式，且严格位于 $a/b$ 和 $c/d$ 之间；此外，任何其他严格位于 $a/b$ 和 $c/d$ 之间的分数，其分母都严格大于 $b+d$。

例如，$(1/3, 2/5)$ 是一个法里对，因为 $1/3$ 和 $2/5$ 是序列$F_5$ 中的相邻项（见上文）。 $1/3$和 $2/5$的中项是 $3/8$。因此， $3/8$位于开区间 $(1/3, 2/5)$ 内，而此开区间内除 $3/8$以外的任何分数，其分母都严格大于 8。关于法里对和法里数列的更多内容，请参阅哈代与赖特 [17]。

我们将严格介于 0 和 1 之间的分数排列成一棵二叉搜索树 $\mathcal{T}_{\mathrm{F}}$。

定义 2.3。法雷对树 $\mathcal{T}_{\mathrm{F}}$ 是一棵完全无限二叉树，其中每个节点都关联一个法里对 $(a/b, c/d)$，该法里对根据节点在 $\mathcal{T}_{\mathrm{F}}$ 中的位置 $\sigma \in \{0,1\}^*$ 递归定义如下： $\mathcal{T}_{\mathrm{F}}(\epsilon) = (0/1, 1/1)$，并且若 $\mathcal{T}_{\mathrm{F}}(\sigma) = (a/b, c/d)$，则 $\mathcal{T}_{\mathrm{F}}(\sigma 0) = (a/b, (a+c)/(b+d))$，且 $\mathcal{T}_{\mathrm{F}}(\sigma 1) = ((a+c)/(b+d)\ c/d)$。节点在 $\mathcal{T}_{\mathrm{F}}$ 中的深度即其位置的长度（根节点的深度为 0）。

稍微滥用一下记号，我们不区分数对 $\mathcal{T}_{\mathrm{F}}(\sigma) = (a/b, c/d)$ 和开区间 $(a/b, c/d)$。

（左）斯特恩-布罗科特树[1] $\mathcal{T}_{\mathrm{SB}}$ 是由法雷对树得到的无限二叉树，其中每个法里对（$a/b, c/d$）都替换为其中项（$a+c$）/（$b+d$）。

□

[1]这里说“左”，是因为斯特恩-布罗科特树最初涉及区间 (0, 2)，而我们只关注 (0,1)，它对应于斯特恩-布罗科特树的左子节点。

因此，例如，我们有

$$\mathcal{T}_{\mathrm{F}}(0)=\left(\frac{0}{1},\frac{1}{2}\right),\ \mathcal{T}_{\mathrm{F}}(1)=\left(\frac{1}{2},\frac{1}{1}\right),\ \mathcal{T}_{\mathrm{F}}(10)=\left(\frac{1}{2},\frac{2}{3}\right),\ \mathcal{T}_{\mathrm{F}}(0000)=\left(\frac{0}{1},\frac{1}{5}\right)\ .$$

为了完整起见，我们将斯特恩-布罗科特树纳入定义中，尽管不会直接使用它。

斯特恩-布罗科特树（因而也包括法雷对树）的元素可以高效计算，参见 Bates 等人 [9]；就我们的目的而言，只需用到下面的命题。

命题 2.4. 存在一台图灵机 $M$，使得对于任意 $\sigma\in\{0,1\}^*$,$\Phi_M(\sigma)=\mathcal{T}_{\mathrm{F}}(\sigma)$，且 $M$，其运行时间为 $\mathrm{poly}(1+|\sigma|)$。

本节最后，我们陈述并证明法雷对树的若干性质。

命题 2.5. 若 $(a/b,c/d)$ 是 $\mathcal{T}_{\mathrm{F}}$ 中深度为 $h$ 的一个法里对，则 $a+b+c+d\geq h+3$。

证明. 对于 $h=0$，有 $0+1+1+1=3=h+3$。

令 $h>0$。假设命题对 $h-1$ 成立，并令 $(a/b,c/d)$ 为深度 $h-1$ 处的任意一对。那么，层级 $h$ 处的一对具有以下形式：(i) $(a/b,(a+c)/(b+d))$，或 (ii) $((a+c)/(b+d),c/d)$。在情形 (i) 中，我们有

$$\begin{aligned}a+(a+c)+b+(b+d)\ &=\ (a+b+c+d)+(a+b)\ \geq\ (a+b+c+d)+1\\ &\geq\ (h+2)+1=h+3\ .\end{aligned}$$

通过对称论证可知，命题在情形 (ii) 中也成立。 □

命题 2.6。设 $p/q\in\mathbb{Q}\cap[0,1]$ 是一个最简形式的分数。那么，$p/q$ 是自然数集 $\mathcal{T}_{\mathrm{F}}$ 中深度至多为 $p+q-2$ 的某个法里对中的一个分数。

证明。根据构造，对于任意深度 $n\geq 0$，在 $\mathcal{T}_{\mathrm{F}}$ 中深度为 $n$ 处出现的区间集合 $[a/b,c/d]$ 覆盖单位区间，且任意两个区间至多有一个公共点（该点必为端点）。因此，$p/q$ 在任意深度 $n$ 处都会出现在某个区间 $[a/b,c/d]$ 中，并且根据定理 2.2，我们有$cb-ad=1$。反设 $n>p+q-2$ 且 $p/q\notin\{a/b,c/d\}$。由于 $p/q\notin\{a/b,c/d\}$，我们有 $a/b<p/q<c/d$，因此也有 $1\leq pb-qa$ 和 $1\leq qc-pd$。于是

$$\begin{aligned}a+b+c+d&\leq(a+b)(qc-pd)+(c+d)(pb-qa)\\ &=p(cb-ad)+q(cb-ad)=p+q\ .\end{aligned}$$

这与命题 2.5 矛盾，命题 2.5 表明 $a+b+c+d\geq p+q+1$。因此，$p/q$ 必须作为一个端点出现，而 $p/q$ 首次作为端点出现的层级至多为 $p+q-2$。 □

引理 2.7. 设 $I = (a_n/b_n, c_n/d_n)$ 是自然数集 $\mathcal{T}_{\mathrm{F}}$ 中深度为 $n$ 的一个法里区间；对于 $i = 0, \ldots, n$，设 $(a_i/b_i, c_i/d_i)$ 表示从根节点到 $I$ 的路径上的法里对。那么数值 $a_i, b_i, c_i, d_i$ 均有上界 $(n+1)(a_n + b_n)$。

证明。我们声称，对于所有 $i$, $a_i + b_i + c_i + d_i \leq (i+1)(a_n + b_n)$。对于 $i = 0$，我们有 $a_0 + b_0 + c_0 + d_0 = 1 \cdot (a_0 + b_0)$。假设这对任意的 $i$ 都成立。如果下一个节点是左子节点，那么

$$\begin{aligned} a_{i+1} + b_{i+1} + c_{i+1} + d_{i+1} &= a_i + b_i + (a_i + c_i) + (b_i + d_i) \\ &\leq a_i + b_i + (i+1)(a_n + b_n) = (i+2)(a_n + b_n)\,. \end{aligned}$$

如果下一个节点是右子节点，那么

$$\begin{aligned} a_{i+1} + b_{i+1} + c_{i+1} + d_{i+1} &= (a_i + c_i) + (b_i + d_i) + c_i + d_i \\ &\leq (i+1)(a_n + b_n) + c_i + d_i \leq (i+1)(a_n + b_n) + a_{i+1} + b_{i+1} \\ &\leq (i+2)(a_n + b_n)\,. \end{aligned}$$

□

# 3 Weihrauch 交

定义 3.1。函数 $I : \mathbb{N} \longrightarrow \mathbb{Q} \times \mathbb{Q}$ 是实数 $\alpha$ 的 Weihrauch 交，当且仅当数对 $I(i)$ 的左分量严格小于数对 $I(i)$ 的右分量（对所有 $i \in \mathbb{N}$ 均成立），并且

$$\{\ \alpha\ \} \ = \ \bigcap_{i=0}^{\infty} I_i^O$$

其中 $I_i^O$ 表示由数对 $I(i)$ 给出的开区间。

定理 3.2。任何可计算实数都可以表示为一个多项式时间可计算的 Weihrauch 交。

证明。可计算实数 $\alpha$ 有一个可计算柯西序列 $C : \mathbb{N} \longrightarrow \mathbb{Q}$，满足性质 $|C(n) - \alpha| < 2^{-n}$。设 $M$ 是一个计算 $C$的图灵机。不失一般性，我们可以假设 $\alpha \in (0, 1)$。

按以下算法计算 $I(k)$：找出满足$n \leq k$ 且 $C(n)$可由 $M$ 在 $k$ 步内计算的最大 $n$。令 $I(k) = (C(n) - 2^{-n}, C(n) + 2^{-n})$。若不存在这样的 $n$，则令 $I(k) = (0, 1)$（可以通过适当安排使得 $I_{k+1}^O \subseteq I_k^O$成立）。

□

通过 Weihrauch 交进行的表示是 Weihrauch 的开创性著作 [33], 中的主要表示方式之一，也是本文所讨论的表示方式中特别的一种：存在一个时间界 $t$，使得每个

可计算实数都有一种 Weihrauch 交表示，可由一台运行时间为 $O(t)$ 的图灵机计算得到（根据定理 3.2，对于任何支配所有多项式的 $t$，这肯定成立）。对于本文讨论的其他每一种表示 $R$，对任意时间界 $t$，都存在一个时间界 $s$，使得 $O(t)_R$严格包含于 $O(s)_R$。通过 Weihrauch 交进行的表示，其度将是第 1 节所述度结构中的零度。

通过 Weihrauch 交表示实数，也称为通过嵌套区间表示。为了简化定义，我们没有要求这些区间必须嵌套，但任何 Weihrauch 交都可以轻易转换为嵌套形式。在 [33], 中可以找到若干子递归等价的表示，但从我们的角度来看，它们都非常相似，因此我们不再讨论其中任何一种。

# 4 与柯西序列子递归等价的表示

## 4.1 柯西序列。

定义 4.1。设 $\alpha \in (0,1)$ 是一个无理数。那么，如果 $|\alpha - C(n)| < n^{-1}$，$\mathbb{N}^+ \longrightarrow \mathbb{Q}$ 就是 $\alpha$ 的一个柯西序列。 □

引理 4.2。设 $C\colon \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是无理数$\alpha \in (0,1)$的一个柯西序列。设 $p/q = C(n)$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^C\colon \mathbb{N} \longrightarrow \mathbb{Q} \times \mathbb{Q}$ 是 $\alpha$的 Weihrauch 交
- $M^C$ 在输入 $n$ 时，运行时间为 $\mathrm{polylog}(\max\{p,q\}, n)$，并且恰好进行一次输入规模为 $O(\log n)$的预言机调用。

证明。令

$$I(n) = (C(n) - n^{-1}, C(n) + n^{-1})\,.$$

那么，如果 $C$ 是 $\alpha$的柯西序列，$I$ 就是 $\alpha$ 的 Weihrauch 交。因此，只需对 $C$ 进行一次预言机调用，而且可以通过对 $p$、$q$ 和 $n$（的二进制表示）进行基本算术运算得到 $I(n)$。 □

引理 4.2 表明，柯西序列可以次递归地转换为 Weihrauch 交。是否可能将 Weihrauch 交次递归地转换为柯西序列？为了对这个问题给出否定的回答，我们需要一个想必相当知名的定理。

定理 4.3。对于任意时间界 $t$，都存在一个可计算的无理数 $\alpha$，使得没有任何柯西序列能由运行时间为 $O(t)$ 的图灵机计算出来。

不难看出该定理成立。令 $t$ 为一个增长相当快的时间界，并令 $A$ 为任意一个自然数集，使得可以用图灵机判定某个自然数是否属于 $A$，但无法用运行时间为 $O(t)$ 的图灵机判定（可以通过标准的对角化论证证明这样一个集合存在）。考虑由二进制展开 $0.a_1a_2a_3\ldots$ 给出的无理数 $\alpha$，其中两个数字 $a_{2i-1}a_{2i}$ 在 $i \in A$ 时为 11，否则为 01。显然，$\alpha$ 有一个可计算柯西序列，但没有任何运行时间为 $O(t)$ 的图灵机能计算出这样的柯西序列。如果一个关于 $\alpha$ 的柯西序列能在时间 $O(t)$ 内计算出来，那么图灵机 $M$ 就能在时间 $O(t)$ 内判定 $m$ 是否属于集合 $A$：首先，$M$ 计算 $C(2^{2m})$。根据假设，这可以在时间 $O(t(\|2^{2m}\|) = O(t(2m+1))$ 内完成。之后，$M$ 确定 $C(2^{2m})$ 的二进制展开中的数字 $0.\mathrm{D}_1\mathrm{D}_2\ldots$。注意，$C(2^{2m})$ 与 $\alpha$ 足够接近，因此可以确保数字 $\mathrm{D}_{2m-1}$ 与数字 $a_{2m-1}$ 相同，从而，$m \in A$ 当且仅当 $\mathrm{D}_{2m-1} = 1$。因此，$M$ 可以通过计算 $\mathrm{D}_{2m-1}$ 来判定 $m$ 是否属于 $A$，而这显然可以在时间 $O(t)$ 内完成。由于没有图灵机能在时间 $O(t)$ 内判定某个自然数是否属于 $A$，我们可以得出结论：该定理成立。

定理 3.2 指出，任何可计算实数都可以用多项式 -
时间可计算的 Weihrauch 交来表示。因此，任何实数都可以用一个在时间 $O(2^n)$ 内可计算的 Weihrauch 交来表示。现在，$2^n$ 是一个固定的时间界；如果可以将 Weihrauch 交次递归地转换为柯西序列，那么任何可计算无理数都可以用一个在时间 $O(t)$ 内可计算的柯西序列来表示，其中时间界 $t$是某个固定值。根据定理 4.3，这样的 $t$ 并不存在，因此我们可以得出结论：Weihrauch 交不能次递归地转换为柯西序列。

## 4.2 定义。

下一个定义给出了一些与柯西序列表示次递归等价的表示。

定义 4.4。设 $\alpha \in (0,1)$ 是一个无理数。

1. $C:\ \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是 $\alpha$ 的严格递增柯西序列，当且仅当 (i) $C$ 是一个 $\alpha$ 的柯西序列以及 (ii) $C(n) < C(\{v5)$。
2. 设 $b \geq 2$ 是一个自然数。那么，$A:\ \mathbb{N}^+ \longrightarrow \mathbb{Z}$ 是一个收敛的基数-$b$ 当且仅当 $A(n)b^{-n}$ 是 $\alpha$ 中的柯西序列时，$\alpha$ 是一个序列。
3. $D:\ \mathbb{Z} \times \mathbb{N}^+ \longrightarrow \{0,1\}$ 是 $\alpha$ 的模糊（戴德金）分割，如果

$$D(p,q) = 0 \ \Rightarrow\ \alpha < \frac{p+1}{q} \qquad \text{and} \qquad D(p,q) = 1 \ \Rightarrow\ \frac{p-1}{q} < \alpha\,.$$

4. $S:$ 当且仅当 $\mathbb{N}^+ \longrightarrow \{-1,0,1\}$ 是 $\alpha$ 的有符号数字展开

$$\alpha = \sum_{i=1}^{\infty} S(i)2^{-i}\,.$$

□

Friedman 和 Ko [21] 以及专著 Ko [20]中都使用了收敛的二进制序列。有符号数字展开似乎也广为人知。Weihrauch 的著作 [33] 讨论了这种表示，近年发表的若干论文中也提到了它，例如 Berger 等人 [5] 和 Bauer 等人 [3]。本文首次讨论了严格递增柯西序列和模糊戴德金分割这两种表示。

## 4.3 从柯西序列到模糊切口。

设 $C$ 是 $\alpha$的柯西序列。我们定义映射 $D\colon \mathbb{Z}\times\mathbb{N}^+ \longrightarrow \{0,1\}$，如下：

$$D(p,q) \;=\; \begin{cases} 0 & \text{if } C(q) \le pq^{-1} \\ 1 & \text{if } C(q) > pq^{-1} \end{cases}$$

引理 4.5. $D$ 是 $\alpha$的模糊切口。

证明。首先我们证明

$$D(p,q) = 0 \;\Rightarrow\; \alpha < \frac{p+1}{q}\,. \tag{4.1}$$

假设 $D(\,p,q\,) = 0$。如果 $\alpha < C(q)$，那么显然有

$$\alpha \;<\; C(q) \;\le\; \frac{p}{q} \;<\; \frac{p+1}{q}\,.$$

因此，当 $\alpha < C(q)$ 时，(4.1) 成立。现在，假设 $\alpha > C(q)$。根据柯西序列的定义，我们有 $\alpha - C(q) \;< q^{-1}$。因此 $\alpha < C(q) + q^{-1}$。此外，根据 $D$的定义，我们有

$$\alpha \;<\; C(q) + \frac{1}{q} \;\le\; \frac{p}{q} + \frac{1}{q} \;=\; \frac{p+1}{q}\,.$$

这就证明了 (4.1)。我们还需要证明

$$D(p,q) \;= 1 \;\Rightarrow\; \frac{p-1}{q} < \alpha\,. \tag{4.2}$$

(4.2) 的证明与 (4.1) 的证明对称。由 (4.1) 和 (4.2) 可得该引理。

□

引理 4.6. 设 $C\colon \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是无理数 $\alpha \in (0,1)$ 的一个柯西序列。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^C : \mathbb{Z}\times\mathbb{N}^+ \longrightarrow \{0,1\}$ *is a fuzzy Dedekind cut for* $\alpha$

- $M^C$ 在输入 $(p,q)$ 上的运行时间为 $\mathrm{poly}(\max\ \|p\|, \|q\|, \|C(q)\|)$，并且只进行一次预言机调用，其输入规模至多为 $O(\log q)$。

证明。以 $q$ （其二进制表示）为输入，对 $C$ 调用一次，便可得到 $C(q)$（其二进制表示）；根据引理 4.5，将 $C(q)$ 与 $pq^{-1}$ 比较一次，便可得到 $D(p,q)$。有理数 $pq^{-1}$的二进制表示可在关于 $p$ 和$q$ 的表示大小的多项式时间内计算得出（即在 $\mathrm{polylog}(\max\{p,q\})$ 时间内），而对 $C(q)$ 和$pq^{-1}$ 的最终比较，可在关于 $\max\{\|C(q)\|, \|pq^{-1}\|\}$ 的多项式时间内完成。 □

## 4.4 模糊切口到带符号数字展开。

设 $\alpha \in (0,1)$，且 $D$ 是 $\alpha$的一个模糊切口。我们将用 $D$ 定义一个有符号数字展开 $S$。

对于任意映射 $S\colon \mathbb{N}^+ \longrightarrow \{-1,0,1\}$，令 $s_n = \sum_{i=1}^{n} S(i)2^{-i}$。此外，令 $a_n$ 为满足 $a_n 2^{-n} = s_n$的唯一整数，并令 $M(r_1,r_2)$表示有理数 $r_1$ 和 $r_2$之间的中点，即 $M(r_1,r_2) = r_1 + (r_2 - r_1)/2$。注意到

$$M\left(s_n - \frac{1}{2^{n+1}}\,,\, s_n\right) = \frac{4a_n - 1}{2^{n+2}} \quad \text{and} \quad M\left(s_n\,,\, s_n + \frac{1}{2^{n+1}}\right) = \frac{4a_n + 1}{2^{n+2}}$$

此外，请注意 $s_n = 2a_n/2^{n+1}$。

我们将有符号数字展开 $S$ 定义为 $S(1) = 1$ 并且

$$S(n+1) \;=\; \begin{cases} -1 & \text{if } D(2a_n, 2^{n+1}) = 0 \text{ and } D(4a_n - 1, 2^{n+2}) = 0 \text{ (Case 1)} \\ 0 & \text{if } D(2a_n, 2^{n+1}) = 0 \text{ and } D(4a_n - 1, 2^{n+2}) = 1 \text{ (Case 2)} \\ 1 & \text{if } (D(2a_n, 2^{n+1}) = 1 \text{ and } D(4a_n + 1, 2^{n+2}) = 1 \text{ (Case 3)} \\ 0 & \text{if } D(2a_n, 2^{n+1}) = 1 \text{ and } D(4a_n + 1, 2^{n+2}) = 0 \text{ (Case 4)} \end{cases}$$

情况 1 中的第一个合取项并非必需，因为它可由第二个合取项推出。情况 3 中的第一个合取项也是如此。保留这些多余的合取项，是为了便于看出这四种情况互斥且穷尽。

引理 4.7。

$$\alpha \;=\; \sum_{i=1}^{\infty} S(i)2^{-i}\,.$$

证明。我们将证明

$$s_n - 2^{-n} < \alpha < s_n + 2^{-n} \tag{4.3}$$

对 $n$ 使用归纳法。该引理可直接由 (4.3) 得出。显然，当 $n = 1$ 时，(4.3) 成立（因为我们假设了 $\alpha \in (0,1)$）。

根据归纳假设，假设 (4.3) 成立。我们需要证明

$$\alpha \in (s_{n+1} - 2^{-(n+1)}\,,\, s_{n+1} + 2^{-(n+1)}) \tag{4.4}$$

（情况 1。）在这种情况下，我们有 $s_{n+1} = s_n - 2^{-(n+1)}$。因此，为了证明 (4.4)，我们需要证明 $\alpha \in (s_n - 2^{-n}, s_n)$。根据归纳假设 (4.3)，我们有 $s_n - 2^{-n} < \alpha$。此外，由于 $D(4a_n - 1, 2^{n+2}) = 0$，我们有

$$\alpha \;<\; \frac{4a_n - 1 + 1}{2^{n+2}} \;=\; \frac{a_n}{2^n} \;=\; s_n\;.$$

这就证明了 (4.4) 在（情况 1）下成立。

（情况 2。）在这种情况下，我们有 $s_{n+1} = s_n$。因此，为了证明 (4.4)，我们需要证明 $\alpha \in (s_n - 2^{-(n+1)}, s_n + 2^{-(n+1)})$。由于 $D(2a_n, 2^{n+1}) = 0$，我们有

$$\alpha \;<\; \frac{2a_n + 1}{2^{n+1}} \;=\; \frac{a_n}{2^n} + \frac{1}{2^{n+1}} \;=\; s_n + \frac{1}{2^{n+1}}\;.$$

由于 $D(4a_n - 1, 2^{n+2}) = 1$，我们有

$$\alpha \;>\; \frac{4a_n - 1 - 1}{2^{n+2}} \;=\; \frac{a_n}{2^n} - 2^{-(n+1)} \;=\; s_n - 2^{-(n+1)}\;.$$

这证明了 (情况 2) 中的 (4.4) 成立。

(情况 3) 与 (情况 1) 对称，而 (情形 4) 与 (情形 2) 对称。 □

引理 4.8. 设 $D\colon \mathbb{Z} \times \mathbb{N}^+ \longrightarrow \{0,1\}$ 是无理数 $\alpha \in (0, 1)$ 的一个模糊切口。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^D\colon \mathbb{N}^+ \longrightarrow \{-1, 0, 1\}$ 是 $\alpha$ 的一个有符号数字展开
- $M^D$ 在输入 $n$ 时的运行时间为 poly($n$)，并且使用 3($n-1$) 次预言机调用，其输入规模至多为 $O(n)$。

证明。根据引理 4.7，$S(1) = 1$，而对 $n > 1$ 计算 $S(n)$，可以通过计算满足 $a_{n-1}s^{-n-1} = s_n = \sum_{i=1}^{n-1} S(i)2^{-i}$ 的整数 $a_{n-1}$ 来完成，随后进行预言机调用以返回以下值：

$$D(2a_{n-1}, 2^n)\;,\; D(4a_{n-1} - 1, 2^{n+1}) \text{ and } D(4a_{n-1} + 1, 2^{n+1})\;. \tag{4.5}$$

注意到，有理数 $2^{-1}$、. . .、$2^{-(n-1)}$ 的二进制表示长度为 $O(n)$，且每个表示都可以在 poly($n$) 时间内计算得到。因此，如果已知 $S(1)$ . . .、$S(n-1)$，则可以使用标准算术运算在 poly($n$) 时间内计算出 $s_{n-1}$。此外，注意到 $s_{n-1}$ 的二进制表示（因而 $a_{n-1}$ 的二进制表示）长度为 $O(n)$。这意味着：(i) (4.5) 中预言机调用的规模至多为 $O(n)$；(ii) $a_{n-1}$、$2a_{n-1}$ 和 $4a_{n-1}$ 都可以在 poly($n$) 时间内计算得到。

使用显然的递归算法计算 $S(n)$，需要计算 $S(1)$ . . .,$S(n-1)$，因此耗时为 $O(n$ poly($n$)) =poly($n$)，总共进行 3($n-1$) 次预言机调用，每次调用的规模为 $O(n)$。 □

## 4.5 带符号数字展开到柯西序列。

设 $S$ 是 $\alpha \in (0,1)$ 的有符号数字展开。那么我们有 $\left|\alpha - \sum_{i=1}^{n} S(i)2^{-i}\right| < 2^{-n}$。令

$$C(n) \;=\; \sum_{i=1}^{\lceil \log_2 n \rceil} S(i)2^{-i}$$

并且我们有

$$|\alpha - C(n)| \;=\; \left|\alpha - \sum_{i=1}^{\lceil \log_2 n \rceil} S(i)2^{-i}\right| \;<\; 2^{-\lceil \log_2 n \rceil} \;\leq\; n^{-1}\,.$$

因此， $C$ 是关于 $\alpha$的柯西序列。

引理 4.9. 设 $S:\ \mathbb{N}^+ \longrightarrow \{-1,0,1\}$ 是无理数 $\alpha \in (0,1)$的有符号数字展开。存在一个参数化函数预言机图灵机 $M$ ，使得

- $\Phi_M^S:\ \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是关于 $\alpha$的柯西序列
- $M^S$ 在输入 $n$ 时以 polylog($n$) 时间运行，并使用 $\lceil \log_2 n \rceil$ 次预言机调用，输入规模至多为 $O(\log n)$。

证明。该结果几乎可以直接从引理前面的文字得出。注意，计算 $C(n) = \sum_{i=1}^{\lceil \log_2 n \rceil} S(i)2^{-i}$ 可通过对有理数进行 poly($\log n$) = polylog($n$) 次运算来完成，这些有理数的表示长度至多为 $O(\log n)$，因此总耗时为 polylog($n$)。 □

## 4.6 从柯西序列到严格递增柯西序列。

设 $C$ 是某个实数 $\alpha$的柯西序列。因此，对于所有 $n$，我们有

$$\alpha \in (\ C(2^n) - 2^{-n}\,,\, C(2^n) + 2^{-n}\ ) \tag{4.6}$$

我们用 $C(n)_\ell$ 表示式 (4.6) 中区间的左端点，即 $C(n)_\ell = C(2^{-n}) - 2^{-n}$，并定义 $C$ 如下:

$$\widehat{C}(n) \;=\; C(n+2)_\ell - 2^{-(n+1)} \;=\; C(2^{n+2}) - 2^{-(n+2)} - 2^{-(n+1)}\,.$$

引理 4.10. 若 $C$ 是 $\alpha$的柯西序列，则 $C$ 是 $\alpha$的严格递增柯西序列。

证明。设 $C$ 是 $\alpha$的一个柯西序列。我们将证明

$$\left|\alpha - \widehat{C}(n)\right| \;<\; 2^{-n} \tag{4.7}$$

and

$$\widehat{C}(n) \;<\; \widehat{C}(n+1)\,. \tag{4.8}$$

因此引理成立。

首先我们注意到 $|\alpha - C(n+2)_\ell| < 2^{-(n+1)}$，因此有

$$\left|\alpha - \widehat{C}(n)\right| \;=\; \left|\alpha - (\; C(n+2)_\ell - 2^{-(n+1)}\;)\right| \;=$$
$$|\,\alpha - C(n+2)_\ell\,| + 2^{-(n+1)} \;<\; 2^{-(n+1)} + 2^{-(n+1)} \;=\; 2^{-n}\,.$$

$\widehat{\ }$第一个等式根据 $C$ 的定义成立，第二个等式成立是因为 $C(n+2)_\ell$ 小于 $\alpha$。这证明了 (4.7) 成立。

接下来我们注意到

$$\left|C(2^n) - C(2^{n+1})\right| \;<\; 2^{-n} + 2^{-(n+1)}\,. \tag{4.9}$$

对所有 $n$ 都成立。现在，反设 (4.8) 不$\widehat{\ }$ $\widehat{\ }$成立，也就是说，假设存在 $m$ 使得 $C(m) \geq C(m+1)$。那么有

$$\widehat{C}(m) - \widehat{C}(m+1) \geq 0\,.$$

$\widehat{\ }$根据 $C$ 的定义，有

$$C(m+2)_\ell - 2^{-(m+1)} - (\; C(m+3)_\ell - 2^{-(m+2)}\;) \geq 0\,.$$

因此

$$C(m+2)_\ell - C(m+3)_\ell \geq 2^{-(m+2)}\,.$$

根据 $C(\cdot)_\ell$ 的定义，我们有

$$C(2^{m+2}) - 2^{-(m+2)} - (\; C(2^{m+3}) - 2^{-(m+3)}\;) \geq 2^{-(m+2)}\,.$$

因此，我们有

$$C(2^{m+2}) - C(2^{m+3}) \geq 2^{-(m+2)} + 2^{-(m+2)} - 2^{-(m+3)} = 2^{-(m+2)} - 2^{-(m+3)}\,.$$

这与 (4.9) 矛盾，因此我们证明了 (4.8) 成立。 □

引理 4.11. 设 $C\colon \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是一个收敛于无理数 $\alpha \in (0,1)$ 的柯西序列。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^C\colon \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是 $\alpha$ 的严格递增柯西序列
- $M^C$ 在输入 $n$ 上的运行时间为 $\mathrm{poly}(\max\{\|C(2^{n+2})\|, n\})$，并且只进行一次输入规模为 $O(n)$ 的预言机调用。

证明。给定输入 $n, M$， $2^{n+2}$ 构造数 $2^{n+2}$ （可用 $O(n)$ 位表示），并执行预言机调用 $C(2^{n+2})$，然后计算并输出有理数 $C(2^{n+2}) - 2^{-(n+2)} - 2^{-(n+1)} = \widehat{C}(n)$。该计算涉及对可用 $\max\{\log \|C(2^{n+2})\|, O(n)\}$ 位表示的数进行基本算术运算，因此总耗时为 $\mathrm{poly}(\max\{\|C(2^{n+2})\|, n\})$。根据引理 4.10， $\widehat{C}$ 是 $\alpha$ 的严格递增柯西序列。

□

## 4.7 从收敛的基数-$b$ 序列到柯西序列。

设 $A$ 是 $\alpha$ 的一个收敛的基数为 $b$ 的序列，并设 $C(n) = A(n)b^{-n}$。现在，根据我们的定义， $C$ 是 $\alpha$ 的柯西序列。因此，下一个引理的证明是显而易见的。

引理 4.12。设 $b \geq 2$，并设 $A\colon \mathbb{N}^+ \longrightarrow \mathbb{Z}$ 是一个收敛的基数-$b$ 序列，对应于一个无理数 $\alpha \in (0,1)$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^A\colon \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是 $\alpha$的柯西序列
- $M^A$ 在输入 $n$ 上运行的时间为 $\mathrm{poly}(\max\{\log A(n), n\})$，并进行一次输入规模为 $O(\log n)$ 的预言机调用。

证明。在输入 $n \in \mathbb{N}^+$时， $M$ 执行预言机调用 $A(n)$（其中 $n$ 可用 $O(\log n)$ 位表示），然后计算有理数 $A(n)b^{-n}$。众所周知，计算 $b^{-n}$ 需要进行 $O(\log n)$ 次乘法（假设 $b^{-1}$已硬编码）。所有涉及的数占用 $O(\max\{\log A(n), n\})$ 位。因此， $M$ 的运行时间至多为 $\mathrm{poly}(\max\{\log A(n), n\})$。 □

## 4.8 从柯西序列到收敛的基数-$b$ 序列。

设 $C$ 是 $\alpha$ 的一个柯西序列。我们将说明如何计算 $A(n)$，其中 $A$ 是 $\alpha$ 的一个收敛基数-$b$ 序列。

首先定义序列 $X_0$、$X_1$、$X_2$、.....以及 $Y_0$、$Y_1$、$Y_2$、.....。设 $p \in \mathbb{Z}$ 和 $q \in \mathbb{N}^+$ 任意。令 $X_0 = p$ 除以 $q$，令 $Y_0 = p$ 对 $q$ 取模，并令

$$X_{n+1} = X_n \times b + ((Y_n \times b) \operatorname{div} q) \quad \text{and} \quad Y_{n+1} = (Y_n \times b) \bmod q$$

其中运算符

- $x$ 除以 $y$（整数除法）
- $x$ 对 $y$ 取模（整数除法的余数）

具有该性质

$$(x \operatorname{div} y) \times y + (x \bmod y) = x\,. \tag{4.10}$$

引理 4.13. 对任意 $n$，我们有 (i) $Y_n < q$、(ii) $pq^{-1} = X_n b^{-n} + Y_n b^{-n} q^{-1}$ 和 (iii) $X_n b^{-n} \leq pq^{-1} < (X_n + 1)b^{-n}$。

证明。显然 (i) 成立，而 (iii) 可直接由 (i) 和 (ii) 推出。我们对 $n$ 使用归纳法证明 (ii)。显然，当 $n = 0$ 时，(ii) 成立。

此外，我们有

$$\begin{aligned} X_{n+1}b^{-(n+1)} + Y_{n+1}b^{-(n+1)}q^{-1} = \\ (X_n b + ((Y_n b) \operatorname{div} q))b^{-(n+1)} + ((Y_n b) \bmod q)b^{-(n+1)}q^{-1} = \\ [X_n bq + ((Y_n b) \operatorname{div} q)q + ((Y_n b) \bmod q)]b^{-(n+1)}q^{-1} = \\ [X_n bq + Y_n b)]b^{-(n+1)}q^{-1} = X_n b^{-n} + Y_n b^{-n}q^{-1} = pq^{-1} \end{aligned}$$

其中，第一个等式由 $X_{n+1}$ 和 $Y_{n+1}$ 的定义得出，第三个等式由 (4.10) 得出，最后一个等式则由归纳假设得出。

□

现在我们可以通过以下步骤计算 $A(n)$：

- 令 $p/q = C(2n)$，其中 $p \in \mathbb{Z}$ 且 $q \in \mathbb{N}^+$
- 令 $k = \lceil \log_b 2n \rceil$
- 计算 $X_k$
- 输出 $A(n) := X_k$。

由引理 4.13 (iii)，我们有

$$A(n)b^{-k} \le C(2n) < (A(n)+1)b^{-k}$$

因此

$$\left|A(n)b^{-k} - C(2n)\right| < b^{-k} = b^{-\lceil \log_b 2n \rceil} \le (2n)^{-1} .$$

此外，由于 $C$ 是关于 $\alpha$ 的柯西序列，我们有 $|\alpha - C(2n)| < (2n)^{-1}$，因此也有 $\left|A(n)b^{-k} - \alpha\right| < n^{-1}$。这证明 $A$ 是关于 $\alpha$ 的收敛基数-$b$序列。

引理 4.14. 设 $C : \mathbb{N}^+ \longrightarrow \mathbb{Q}$ 是关于无理数 $\alpha \in (0,1)$ 的柯西序列，并设 $b \ge 2$。存在一个参数化函数预言机图灵机$M$，使得

- $\Phi_M^C : \mathbb{N}^+ \longrightarrow \mathbb{Z}$ 是 $\alpha$ 的收敛基数-$b$ 展开
- $M^C$ 在输入 $n$ 上的运行时间为 $\text{poly}((\log n) + \|C(2n)\|)$，并进行一次输入规模为 $O(\log n)$ 的预言机调用。

证明。上述算法使用 $C(2n)$，因此预言机调用的输入规模为 $O(\log n)$。此外，该算法至多使用 $\text{polylog}(n)$ 时间来计算$k = \lceil \log_b 2n \rceil$。最后的 for 循环包含 $k$ 次迭代，每次迭代涉及 5 次算术运算；通过直接归纳法可知，这些运算的参数均为非负整数，且其规模至多为

$$\|C(2n)\| + k(\|b\| + 1) \;=\; O((\log n) + (\log \|C(2n)\|)$$

位。由于每次算术运算都能在表示规模的多项式时间内完成，因此总耗时为

$$O(k\,\mathrm{poly}((\log n) + \|C(2n)\|)) \;=\; \mathrm{poly}((\log n) + \|C(2n)\|)\;.$$

□

### 4.9 总结。

回顾一下，$O(t)_R$ 表示区间 (0,1) 中所有无理数的类，其中至少有一种 $R$-表示可由运行时间为 $O(t(n))$ 的图灵机计算得出（$n$ 是输入长度，见第 6 页的定义 1.3）。结合本节关于不同表示之间转换复杂度的结果，我们得到以下定理。

定理 4.15。考虑以下表示：(1) 柯西序列，(2) 递增柯西序列，(3) 模糊分割，(4) 带符号数字展开，以及 (5) 收敛的基数-$b$ 序列；令 $R_1$ 和 $R_2$ 为这五种表示中的任意两种。对于任意时间界 $t$，我们有

$$O(t)_{R_2} \;\subseteq\; O(\mathrm{poly}(t(2^n)))_{R_1}\;.$$

## 5 基数-$b$ 展开与和近似

### 5.1 基数-$b$ 展开。

实数的基数-$b$ 展开表示，或者更准确地说，基数-10 展开表示，是众所周知的。这就是日常生活中实数的标准表示。我们将只关注 0 和 1 之间的实数。

定义 5.1. 基数是一个严格大于 1 的自然数，而基数-$b$数字是集合 $\{0, 1, \ldots, b-1\}$ 中的自然数。

设 $b$ 是一个基数，$\mathtt{D}_1$、.....、$\mathtt{D}_n$ 是基数-$b$ 数字。我们用 $(0.\mathtt{D}_1\mathtt{D}_2\ldots\mathtt{D}_n)_b$ 表示有理数 $0 + \sum_{i=1}^{n} \mathtt{D}_i b^{-i}$。

设 $\mathtt{D}_1, \mathtt{D}_2$、..... 是一个由基数-$b$ 数字组成的无限序列。若满足以下条件，我们称 $(0.\mathtt{D}_1\mathtt{D}_2\ldots)_b$ 为实数 $\alpha$ 的基数-$b$ 展开：

$$(0.\mathtt{D}_1\mathtt{D}_2\ldots\mathtt{D}_n)_b \;\leq\; \alpha \;<\; (0.\mathtt{D}_1\mathtt{D}_2\ldots\mathtt{D}_n)_b + b^{-n}$$

对所有 $n \geq 1$。令 $E_b^\alpha$: $\mathbb{N}^+ \longrightarrow \{0, .., b-1\}$ 为函数，它给出 $i$ 个数字在 $\alpha$ 的基数为 $b$ 的展开中的值。更准确地说，令 $E_b^\alpha(i) = \mathtt{D}_i$，其中 $(0.\mathtt{D}_1\mathtt{D}_2\ldots)_b$ 是 $\alpha$ 的基数为 $b$ 的展开。我们称 $E_b^\alpha$ 是 $\alpha$ 的基数为 $b$ 的展开。

□

容易看出，基数为 $b$ 的展开可以次递归转换为柯西序列：若 $(0.\mathtt{D}_1\mathtt{D}_2\ldots)_b$ 是 $\alpha$ 的基数为 $b$ 的展开，则

$$(0.\mathtt{D}_1)_b\ ,\ (0.\mathtt{D}_1\mathtt{D}_2)_b\ ,\ (0.\mathtt{D}_1\mathtt{D}_2\mathtt{D}_3)_b\ ,\ \ldots$$

将是 $\alpha$ 的柯西序列的前几个元素，因此，如果我们能获取基数为 $b$ 的展开，就不需要通过无界搜索来计算柯西序列。

引理 5.2. 设 $E_b^\alpha$: $\mathbb{N}\longrightarrow\{0,..,b-1\}$ 是无理数 $\alpha\in(0,1)$ 的基数为 $b$ 的展开。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^{E_b^\alpha}:\mathbb{N}\longrightarrow\mathbb{Q}$ 是 $\alpha$ 的柯西序列
- $M^{E_b^\alpha}$ 在输入 $n$ 上运行的时间为 poly($n$)，并使用 $n$ 次预言机调用，输入规模至多为 $\log n$。

证明. 在输入 $n$ 时， $M$ 对 $E_b^\alpha$ 进行 $n$ 次预言机查询，每次查询的规模至多为 $\log n$，以获得 $\alpha$ 的基数为 $b$ 的展开的前 $n$ 个数字 $\mathtt{D}_1$，...，$\mathtt{D}_n$。每个数字需要 $\log b$ 空间，因此计算有理数 $p/q=\sum_{i=1}^n \mathtt{D}_i b^{-i}$ 可以在时间 poly($n$) 内完成。

□

事实证明，我们无法用次递归方法将柯西序列转换为基数为$b$ 的表示。一般而言，我们也无法用次递归方法将基数为$b$的展开转换为基数为$a$ 的展开。让我们回顾 Kristiansen[24] 中的一个定义。

定义5.3。我们用 prim($b$) 表示基数$b$的素因子集合，也就是说，prim($b$) $=\{p\mid p$ 是一个素数，且 $p|b\}$。

设 $a$ 和 $b$ 为基数，且 prim($a$) ⊆prim($b$)。现在我们将定义从 $a$ 到 $b$的进制转换因子。设 $b=p_1^{k_1}p_2^{k_2}\ldots p_n^{k_n}$为 $b$的素因数分解，其中 $p_i$ 是素数，且 $k_i\in\mathbb{N}^+$（对于 $i=1,\ldots,n$）。那么， $a$可以写成$a=p_1^{j_1}p_2^{j_2}\ldots p_n^{j_n}$的形式，其中 $j_i\in\mathbb{N}$（对于 $i=1,\ldots,n$）。从 $a$ 到 $b$ 的进制转换因子是满足下式的自然数 $k$

$$k\ \ =\ \ \max\{\,\lceil j_i/k_i\rceil\ \mid\ 1\le i\le n\,\}\ .$$

□

注意，从 $a$ 到 $b$ 的进制转换因子有定义，当且仅当 prim($a$) ⊆ prim($b$)（当 prim($a$) ⊈ prim($b$) 时，该定义没有意义）。假设从 $a$ 到 $b$ 的进制转换因子存在时，默认 prim($a$) ⊆ prim($b$)。

定理 5.4（基数转换定理）。设 $k$ 是从 $a$ 到 $b$的进制转换因子，且 $(0.D_1D_2\ldots)_a$ 和 $(0.\dot{D}_1\dot{D}_2\ldots)_b$ 分别是实数 $\alpha$ 的基 -$a$ 展开和基 -$b$ 展开。那么，对所有 $n \in \mathbb{N}$，有

$$\begin{aligned}(0.D_1\ldots D_n)_a \;&\leq\; (0.\dot{D}_1\ldots\dot{D}_{kn})_b \;\leq\; \alpha \\ &<\; (0.\dot{D}_1\ldots\dot{D}_{kn})_b \;+\; b^{-kn} \;\leq\; (0.D_1\ldots D_n)_a \;+\; a^{-n}\,. \quad \text{(I)}\end{aligned}$$

此外，对于所有 $n$、$\ell \in \mathbb{N}$，我们有

$$(0.\dot{D}_1\ldots\dot{D}_{kn})_b \;<\; (0.\dot{D}_1\ldots\dot{D}_{\ell})_b \;\Rightarrow\; (0.D_1\ldots D_n)_a \;<\; (0.D_1\ldots D_{\ell m})_a \qquad \text{(II)}$$

其中 $m = \lceil \log_a b \rceil$。

底数转换定理的证明见 [24]。假设从基数 $a$ 到基数 $b$ 的进制转换因子 $k$ 存在。那么，根据该定理的条款 (I)，$n$ 个小数位 $0.\mathrm{D}_1\ldots\mathrm{D}_n$ 的基数为 $a$ 的展开将由. $\alpha$ 的基数为 $b$ 的展开中的 $kn$ 个小数位 $0.\mathrm{D}_1\ldots\mathrm{D}_{kn}$ 确定，因此，我们可以用次递归方式将基数为 $b$ 的展开转换为基数为 $a$ 的展开。

引理 5.5。假设从基数 $a$ 到基数 $b$ 的进制转换因子 $k$ 存在，并设 $E_b^\alpha$: $\mathbb{N}^+ \longrightarrow \{0,\ldots,b-1\}$ 是无理数$\alpha \in (0, 1)$的基数为 $b$ 的展开。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^{E_b^\alpha}$：$\mathbb{N}^+ \longrightarrow \{0,\ldots,a-1\}$ 是 $\alpha$ 的基数-$a$ 展开
- $M^{E_b^\alpha}$ 在输入 $n$ 时，运行时间为 poly($n$)，并使用 $kn$ 次预言机调用，每次调用的输入规模至多为 $O(\log n)$。

证明。注意，$\alpha$ 的 $b$ 进制展开的前 $kn$ 个数字可以通过 $kn$ 次预言机调用计算得到，每次调用的规模至多为 $\log kn = O(\log\ n)$。将这 $kn$ 个数字转换为形如 $q \cdot b^{-kn}$ 的数，可以在 poly($kn$) $=$ poly($n$) 的时间内完成，因此计算 $p = \lfloor a^n \cdot q/b^{kn} \rfloor$ 可以在 poly($kn$) $=$ poly($n$) 的时间内完成。根据底数转换定理的条款 (I)，$(0.\mathrm{D}_1\ldots\mathrm{D}_n)_a = p/a^n$，并且由于每个基数为 $a$ 的数字 $\mathrm{D}_1$，...，$\mathrm{D}_n$ 都属于 $\{0,\ldots,a-1\}$，且 $p/a^n = \sum_{i=1}\mathrm{D}_i a^{-i}$，一个简单的贪心算法可以在 $p$ 已经计算出来后，以 poly($n$) 的时间按递增顺序计算 $(0.\mathrm{D}_1\ldots\mathrm{D}_n)_a$ 的数字 $\mathrm{D}_1$，...，$\mathrm{D}_n$。□

如果从 $a$ 到 $b$ 的进制转换因子不存在，那么即使假设 $\alpha$ 是无理数，我们也无法在 $E_b^\alpha$ 中以次递归方式计算 $E_a^\alpha$。这一点在 [24], 中得到了形式化证明，但直观上不难理解原因：考虑一个非常接近有理数$(0.1)_{10}$的无理数 $\alpha$。我们有 $(0.1)_{10} = (0.0(0011)^*)_2$, ，假设 $\alpha$ 的二进制展开以

$$\alpha \;=\; 0.00011001100110011001100110011001100110011001100110011\ldots$$

根据上面显示的二进制展开的数字，我们无法判断十进制展开的第一位小数是0还是1。我们需要更多的二进制展开数字，才能确定十进制展开的第一位数字。现在，$\alpha$ 不可能等于$(0.1)_{10}$，因为我们已假设 $\alpha$ 是无理数。因此，迟早会在二进制展开中找到一个能够确定十进制展开第一位数字的数字，但找到该数字需要无界搜索。

一般而言，如果从 $a$ 到 $b$ 的进制转换因子不存在，也就是说，如果 $\mathrm{prim}(a) \not\subseteq \mathrm{prim}(b)$，我们就无法在 $E_b^\alpha$ 中子递归计算 $E_a^\alpha$。由此可知，对于 $\alpha$ 的柯西序列，我们无法子递归计算 $E_b^\alpha$（对任意基数 $b$ 均如此）。为导出矛盾，假设我们能够做到，也就是说，假设我们能在任意一个表示 $\alpha$ 的柯西序列中子递归计算 $E_b^\alpha$。选取一个基数 $b_0$，使得 $\mathrm{prim}(b) \not\subseteq \mathrm{prim}(b_0)$。根据引理 5.2，我们可以在 $E_{b_0}^\alpha$ 中子递归计算一个表示 $\alpha$ 的柯西序列 $C$。根据我们的假设，我们可以在 $C$ 中子递归计算 $E_b^\alpha$。因此，我们可以在 $E_{b_0}^\alpha$ 中子递归计算 $E_b^\alpha$，但这是不可能的，因为从 $b$ 到 $b_0$ 的进制转换因子不存在。

## 5.2 基数-$b$ 求和近似。

基数-$b$ 的和近似（从下方和上方）由 Kris- tiansen 在 [23] 中引入，并在 [24] 中作了进一步研究；随后又与 Georgievand Stephan 一起在 [16] 中进行了研究。

定义 5.6。设 $(0.\mathtt{D}_1\mathtt{D}_2\ldots)_b$ 是无理数 $\alpha \in (0,1)$ 的基数-$b$ 展开（因此，我们有 $E_b^\alpha(n) = \mathtt{D}_n$）。

$\alpha$ 的基数-$b$ 条件和近似是函数 $\hat{A}_b^\alpha(0) = 0$，且 $\hat{A}_b^\alpha(n+1) = E_b^\alpha(m)b^{-m}$，其中 $m$ 是满足下列条件的最小 $m$：

$$\sum_{i=0}^{n} \hat{A}_b^\alpha(i) \;<\; (0.\mathtt{D}_1\ldots\mathtt{D}_m)_b$$

也就是说，$\hat{A}_b^\alpha(n)$ 是 $\alpha$ 的基数-$b$展开中第 $n$ 个非零数字所表示的值。

令 $\overline{\mathtt{D}}$ 表示基数-$b$数字 $\mathtt{D}$ 的补数字，即令 $\mathtt{D} = (b-1) - \mathtt{D}$（注意，对于任意基数 $b$ 以及任意基数-$b$ 展开 $(0.\mathtt{D}_1\mathtt{D}_2\ldots)_b$，我们都有 $(0.\mathtt{D}_1\mathtt{D}_2\mathtt{D}_3\ldots)_b + (0.\overline{\mathtt{D}}_1\overline{\mathtt{D}}_2\mathtt{D}_3\ldots)_b = 1$）。

上文中对 $\alpha$ 的基数-$b$ 和近似是函数 $\check{A}_b^\alpha(0) = 0$ 且 $\check{A}_b^\alpha(n+1) = E_b^\alpha(m)b^{-m}$ 定义，其中 $\check{m}$ 是满足条件的最小 $m$

$$1 - \sum_{i=0}^{n} \check{A}_b^\alpha(n) \;>\; 1 - (0.\overline{\mathtt{D}}_1\ldots\overline{\mathtt{D}}_m)_b\,.$$

□

ˆˇ 如果 $\alpha$ 是有理数，则函数 $A_b^\alpha$ 和 $A_b^\alpha$ 没有定义。使用该记法时，默认 $\alpha$ 是无理数。很容易证明

$$\sum_{i=0}^{\infty} E_b^\alpha(i) b^{-i} \;=\; \sum_{i=0}^{\infty} \hat{A}_b^\alpha(i) \;=\; 1 \;-\; \sum_{i=0}^{\infty} \check{A}_b^\alpha(i)\,.$$

详细证明见 [24]。

ˆˇ 我们无法在 $A_b^\alpha$ 中子递归计算 $A_b^\alpha$，同样也无法在 ˆ $A_b^\alpha$ 中子递归计算 ˇ $A_b^\alpha$（对于任意基数 $b$）。下面的增长论证（见第 7 页）说明了为什么我们无法在 ˇ $A_2^\alpha$ 中子递归计算 ˆ $A_2^\alpha$：设 $t$ 为任意时间界，并令 $f$ 为一个单调严格递增函数，其增长速度快于任何可在时间 $O(t)$ 内计算的函数；此外，设 $f$ 的图像可在多项式时间内计算，也就是说，关系 $f(x) = y$ 可在关于自然数 $x$ 和 $y$ 的位数为多项式的时间内判定。显然，这样的 $f$ 存在，而且不失一般性，我们可以假设 $2^x \le f(x)$。考虑由 ˆ$A_2^\beta(n) = 2^{-f(n)}$ 给出的无理数 $\beta$。我们有

$$\beta = (0.0\ldots010\ldots010\ldots010\ldots010\ldots)_2$$

其中，零的序列 0 . . . 0 越来越长。现在

$$\beta = (0.0\ldots010\ldots010\ldots010\ldots)_2 = 1 - (0.1\ldots101\ldots101\ldots101\ldots)_2$$

因此 ˇ$A_2^\beta(n) = 2^{-g(n)}$ 其中 $g:\ \mathbb{N} \longrightarrow \mathbb{N}$ 是一个增长缓慢的函数（我们有 $g(x) \le 2x$），事实上，$g(x)$ 可在自然数 $x$ 的多项式时间内计算，因为 $f$ 的图像可在多项式时间内计算。由于我们可以在多项式时间内计算 $g$，因此也可以在多项式时间内计算 ˇ$A_2^\beta$。显然，我们无法在时间 $O(t)$ 内计算 ˆ$A_2^\beta$；若能做到，我们也就能在时间 $O(t)$ 内计算 $f$，这与我们的假设相矛盾。

因此我们得出结论：对于任意时间界 $t$，都存在一个无理数 $\beta$，使得 ˇ $A_2^\beta$ 可在多项式时间内计算，而 ˆ$A_2^\beta$ 无法在时间 $O(t)$ 内计算。这表明我们无法在 ˇ $A_2^\alpha$ 中次递归计算 ˆ $A_2^\alpha$。这一论证很容易推广到任意基数 $b$，因此，我们无法在 $A_b^\alpha$ 中次递归计算 ˆˇ $A_b^\alpha$。对称的论证将表明，我们无法在 ˆ $A_b^\alpha$ 中次递归计算 ˇ $A_b^\alpha$。详细证明见 [24]。

在 [24] 中还证明了，从 $a$ 到 $b$ 的进制转换因子存在，当且仅当我们可以在 $A_b^\alpha$ 中子递归计算 $A_a^\alpha$；也当且仅当我们可以在 $A_b^\alpha$ 中子递归计算 $A_a^\alpha$。我们已经说明了，当从 $a$ 到 $b$ 的进制转换因子不存在时，为什么无法在 $E_b^\alpha$ 中子递归计算 $E_a^\alpha$（见第 27 页）。同样的论证也应能直观地解释，为什么当从 $a$ 到 $b$ 的进制转换因子不存在时，我们无法在 $A_b^\alpha$ 中子递归计算 $A_a^\alpha$，也无法在 $A_b^\alpha$ 中子递归计算 $A_a^\alpha$。在下一小节中，我们将分析在所需的进制转换因子存在时，在 $A_b^\alpha$ 中计算 $A_a^\alpha$ 的复杂度。

## 5.3 从基数 $b$ 的和近似到基数 $a$ 的和近似。

假设从基数 $a$ 到基数 $b$ 的进制转换因子 $k$ 存在。现在我们将给出并解释一种算法：把 $\alpha$（$\alpha$ 的基数为 $b$ 的和近似下界）作为预言机，用它来计算 $A_a^\alpha$（$\alpha$ 的基数为 $a$ 的和近似下界）。当然，利用对称算法，可以根据 $\alpha$ 的基数为 $b$ 的和近似上界，计算出 $\alpha$ 的基数为 $a$ 的和近似上界。

假设 $\hat{}\ A_a^\alpha(0), \ldots, \hat{}A_a^\alpha(n)$ 的值已经计算出来。然后，算法通过执行以下指令来计算 $\hat{}\ A_a^\alpha(n+1)$ 的值：

- 步骤 1：计算有理数 $\sum_{i=0}^{n} \hat{}A_a^\alpha(i)$。该数的形式为 $(0.\mathtt{D}_1 \ldots \mathtt{D}_p)_a$，其中 $p$。计算 $p$。$\hat{}\,\hat{}$

- 步骤 2：向预言机 $A_b^\alpha$ 查询 $A_b^\alpha(kp+1)$ 的值（其中 $k$ 是进制转换因子）。预言机将给出一个形式为 $\mathtt{D}b^{-\ell}$ 的有理数，其中 $\ell \geq kp+1$，且 D 是一个非零的基数 $b$ 数字。计算 $\ell$。

- 步骤 3：使用预言机 $A_b^{\hat{\alpha}}$ 计算有理数

$$R = \sum_{i=0}^{k\ell m} \hat{A}_b^\alpha(i)$$

其中 $k$ 是进制转换因子，$m = \lceil \log_a b \rceil$，而 $\ell$ 是在步骤 2 中计算出的值。

- 步骤 4：计算最小的 $i$ 使得在以 $p+i$ 为基数的$a$ 展开中，第 $R$ 个数字非零（其中 $p$ 是在步骤 1 中计算出的值，而 $R$ 是在步骤 3 中计算出的值）。输出 $\mathtt{D}a^{-(p+i)}$ 其中D 是以 $p+i$ 为基数的$a$ 展开中第 $R$个数字$A_a^\alpha(n+1)$，即有 $\hat{} = \mathtt{D}a^{-(p+i)}$

现在我们来说明该算法给出的输出是正确的。设 $(0.\mathtt{D}_1\mathtt{D}_2\ldots)_a$ 和 $(0.\dot{\mathtt{D}}_1\dot{\mathtt{D}}_2\ldots)_b$ 分别为 $\alpha$ 的以 $a$ 和 $b$ 为基数的展开。在步骤 1 中，该算法计算出 $(0.\mathtt{D}_1 \ldots \mathtt{D}_p)_a$。根据 $A_a^\alpha$ 的定义，输出应为 $\mathtt{D}_{p+i}a^{-(p+i)}$，其中 $i$ 是满足下列条件的最小数

$$(0.\mathtt{D}_1 \ldots \mathtt{D}_p)_a < (0.\mathtt{D}_1 \ldots \mathtt{D}_{p+i})_a \ .$$

在步骤 2 中，算法计算 $\ell$。根据和近似的定义，

$$(0.\dot{\mathtt{D}}_1 \ldots \dot{\mathtt{D}}_{kp})_b < (0.\dot{\mathtt{D}}_1 \ldots \dot{\mathtt{D}}_\ell)_b \ . \tag{5.1}$$

因此，根据底数转换定理的条款 (II)，我们有

$$(0.\mathtt{D}_1 \ldots \mathtt{D}_p)_a < (0.\mathtt{D}_1 \ldots \mathtt{D}_{\ell m})_a \tag{5.2}$$

其中 $m = \lceil \log_a b \rceil$。这表明，基数为 $a$ 的和近似中下一项的分母至多为 $a^{\ell m}$。在步骤 3 中，算法计算$R$。根据 (5.2) 和底数转换定理的条款 (I)，我们有

$$(0.\mathtt{D}_1 \dots \mathtt{D}_p)_a < (0.\mathtt{D}_1 \dots \mathtt{D}_{\ell m})_a \le (0.\dot{\mathtt{D}}_1 \dots \dot{\mathtt{D}}_{k\ell m})_b = R < \alpha \\ < (0.\dot{\mathtt{D}}_1 \dots \dot{\mathtt{D}}_{k\ell m})_b + b^{-k\ell m} \le (0.\mathtt{D}_1 \dots \mathtt{D}_{\ell m})_a + a^{-\ell m} . \quad (5.3)$$

现在，(5.3) 意味着，$\alpha$ 的基数为 $a$ 的展开中前 $\ell m$ 位小数，也就是 $\mathtt{D}_1 \dots \mathtt{D}_{\ell m}$，与有理数 $R$ 的基数为 $a$ 的展开中前 $\ell m$ 位小数相同。因此，该算法在步骤 4 中计算出了正确结果。

引理 5.7。假设从基数 $a$ 到基数 $\hat{}b$ 的进制转换因子 $k$ 存在，并设 $A_b^\alpha \colon \mathbb{N} \longrightarrow \mathbb{Q}$ 是无理数 $\alpha \in (0, 1)$ 的基数为 $b$ 的向下和近似。此外，假设 $\hat{} A_b^\alpha$ 可在时间界 $s$ 内计算，并令 $g(n) = (\lambda x.\|m\| + s(\|k\| + x + 1))^{(n)}(0)$，其中 $m = \lceil \log_a b \rceil$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^{\hat{} A_b^\alpha} \colon \mathbb{N} \longrightarrow \mathbb{Q}$ 是 $\alpha$ 的基数为 $a$ 的向下和近似
- $M^{\hat{} A_b^\alpha}$ 在输入 $n$ 上的执行时间为 $\mathrm{poly}(2^{g(n)} + s(\|k\| + g(n)))$，且至多使用 $k2^{g(n)}$ 次预言机调用，每次调用的输入规模至多为 $\|k\| + g(n)$。

证明。通过对 $n$ 进行归纳，我们证明时间界，同时也给出 $a$ 在 $A_a^\alpha(n)$ 的分母中 $\hat{}$ 指数的一个界。

为了计算 $A_b^\alpha(1)$，我们以 1 作为输入调用预言机，并得到一个位置$\ell$；然后计算 $R$，从而由此得到基数 $a$ 下的第一个非零数字。根据 $\hat{}$ 假设，$A_b^\alpha(1)$ 可在时间 $s(\|1\|)$ 内计算；这意味着 $s(\|1\|)$ 也是此次计算结果位长的一个界，因此也是 $\ell$的位长界（计算结果是某个 $\hat{}c$ 的二进制表示）。机器必须访问 $i = 1, \dots, k\ell m$ 对应的 $A_b^\alpha(i)$ 值，因此使用$k\ell m \le km2^{s(\|1\|)} = k2^{g(1)}$ 次预言机调用。由于结果的大小受其计算时间所界定，我们有最大的预言机回答的大小至多为 $s(\|k\| + g(1))$。不难验证，执行时间由表达式 $k\ell m$ 主导；该表达式等于步骤 3 中求和的长度乘以一次预言机调用所返回的最大数值的大小。我们已将 $k\ell m$ 界定为 $k2^{g(1)}$，而最后一次预言机调用的大小至多为$s(\|k\ell m\|) \le s(\|k\| + g(1))$，这就证明了 $n = 1$ 的执行时间界。$\hat{}$ 我们注意到，$A_a^\alpha(1)$ 分母中的指数至多为 $\ell m$ (5.2)，因此至多为 $2^{g(1)}$。

现在来看归纳步骤。假设 $n > 1$，并归纳地假设位置 $p$ 有界于 $2^{g(n-1)}$。值 $\ell$ 将有界于 $2^{s(\|kp+1\|)}$，而预言机调用的最大输入将为

$$k\ell m \le km2^{s(\|kp+1\|)} \le km2^{s(\|k\|+\|p\|+1)} \le km2^{s(\|k\|+g(n-1)+1)} \le k2^{g(n)} .$$

这也限制了预言机调用的次数，因为我们可以存储先前查询的结果，因此不会在同一个输入上重复查询预言机。预言机调用结果的位长上界为 $s(\|k\| + g(n))$。执行时间是该数量与调用次数之和的多项式，从而得到上界 $\mathrm{poly}(2^{g(n)} + s(\|k\| + g(n)))$，而最后一个基数为 $a$ 的数字的位置有界于 $\ell m \leq 2^{g(n)}$。

□

注意到，该引理意味着：如果$\hat{}\ A_b^\alpha$ 可由一台运行时间为 $O(s)$的图灵机计算，那么 $A_a^\alpha$ 可由一台运行时间为 $O(s')$的图灵机计算，其中时间界 $s'$ 关于时间界 $s$是原始递归的。

## 5.4 从基数-$b$ 和近似到基数-$a$ 展开

这与将基数-$b$ 展开转换为基数-$a$ 展开非常相似。与第5.2节一样，我们论证：如果从 $a$ 到 $b$ 的进制转换因子不存在，那么从基数-$b$ 和近似转换为基数-$a$ 展开就无法通过次递归方式完成。若该转换因子存在，则可以直接给出一个转换算法。

引理 5.8。假设从基数 $a$ 到基数 $b$的进制转换因子 $k$ $\hat{}$存在，并令 $A_b^\alpha \colon \mathbb{N} \longrightarrow \mathbb{Q}$ 为无理数 $\alpha \in (0, 1)$从下方得到的基数-$b$ 和近似。存在一台参数化函数预言机图灵机$M$ ，使得

- $\Phi_M^{\hat{}A_b^\alpha}$： $\mathbb{N} \longrightarrow \{0, \ldots, a-1\}$ 是 $\alpha$的基数为$a$ 的展开
- $M^{\hat{}A_b^\alpha}$ 在输入 $n$ 上运行的时间为 $\mathrm{poly}(\|\hat{}A_b^\alpha(kn)\|)$，并且至多使用 $kn$ 次预言机调用，每次调用的输入规模为 $O(\log n)$。

证明。与引理 5.5 的证明一样，计算 $(0.\mathrm{D}_1\mathrm{D}_2\mathrm{D}_3\cdots)_a$ 的第 $n$ 个数字，需要先获得 $(0.\mathrm{D}_1\mathrm{D}_2\mathrm{D}_3\cdots)_b$ 的前 $kn$ 个数字。根据 $\hat{}A_b^\alpha$ 的定义，如果对于某个 $m \in \mathbb{N}$ 和 $c \in \{0, \ldots, b-1\}$，$\hat{}A_b^\alpha(n) = c \cdot b^{-m}$，那么 $(0.\mathrm{D}_1\mathrm{D}_2\mathrm{D}_3\cdots)_b$ 的第 $n$ 个非零数字就是 $c$。因此，为了找到基数为 $b$ 的展开的前 $kn$ 个数字， $M$只需依次查询 $\hat{}\ A_b^\alpha(1)$、$\hat{}A_b^\alpha(2)$.....，直到找到第一个满足 $A_b^\alpha(i) = c_i b^{-m_i}$ 且 $m_i \geq kn$ 的 $\hat{}\ i$。显然，这是一次有界搜索，因为对于某个 $i \leq kn$，这样的 $m_i$ 必定存在。得到 $(0.\mathrm{D}_1\mathrm{D}_2 \ldots \mathrm{D}_{kn})_b$ 后，我们通过反复乘以 $b$ 并除以$a$，提取出 $(0.\mathrm{D}_1\mathrm{D}_2\mathrm{D}_3 \ldots \mathrm{D}_n)_a$。显然，该过程至多使用 $kn$ 次预言机调用，每次调用的规模至多为 $\log kn$。该过程的时间复杂度由最后一次预言机调用结果规模的多项式所主导，即 $\mathrm{poly}(\|A_b^\alpha(kn)\|)$（注意，这个数的分母至少为 $b^{kn}$）。

□

当然，引理 5.8 对从上方给出的和近似也成立。

## 5.5 格雷码。

Tsuiki [31] 引入了实数的格雷码表示，Berger 等人 [5] 对其进行了进一步研究。该表示与二进制展开表示子递归等价。

定义 5.9。函数 $G:\mathbb{N}\longrightarrow\{0,1\}$ 是无理数 $\alpha$ 的格雷码，当且仅当 $G(i)=0$ 存在偶数 $m$ 使得

$$m2^{-i}-2^{-(i+1)}<\alpha<m2^{-i}+2^{-(i+1)}$$

并且 $G(i)=1$ 当且仅当对某个奇数 $m$同样成立。 □

格雷码通常定义为映射 $G:\mathbb{N}\longrightarrow\{0,1,\perp\}$，其中 $G(i)=\perp$当且仅当 $\alpha$ 具有如下形式：$m2^{-i}-2^{-(i+1)}$，其中 $m$ 和 $i$，因此它是有理数。由于我们只考虑无理数，因此不需要 $\perp$。

引理 5.10（Tsuiki [31]）。设 $E_2^\alpha$ 和 $G^\alpha$分别是无理数 $\alpha\in(0,1)$ 的二进制展开和格雷码。则有

$$E_2^\alpha(1)=G^\alpha(0)$$
$$E_2^\alpha(n+1)=E_2^\alpha(n)\oplus G^\alpha(n)$$

其中 $\oplus$ 表示异或函数。

引理 5.11. 令 $G:\mathbb{N}\longrightarrow\{0,1\}$ 为无理数 $\alpha\in(0,1)$ 的格雷码。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^G:\mathbb{N}^+\longrightarrow\{0,1\}$ 是 $\alpha$ 的二进制展开
- $M^G$ 在输入 $n$ 上的运行时间为 $O(n\log n)$，并使用 $n$ 次预言机调用，每次调用的输入规模为 $O(\log n)$。

证明。由引理 5.10 给出的转换算法可直接得出。

□

引理 5.12. 令 $E:\mathbb{N}^+\longrightarrow\{0,1\}$ 为无理数$\alpha\in(0,1)$ 的二进制展开。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^E:\mathbb{N}\longrightarrow\{0,1\}$ 是 $\alpha$ 的格雷码
- $M^E$ 在输入 $n$ 上运行时间为 $O(\log n)$，并进行 2 次预言机调用，每次的输入规模为 $O(\log n)$。

证明。对引理 5.11 中的方程进行简单变换可得:

$$G^\alpha(0)=E_2^\alpha(1)$$
$$G^\alpha(n+1)=E_2^\alpha(n+1)\oplus E_2^\alpha(n)$$

该算法的复杂度如引理所述。 □

## 5.6 总结。

本节的主要结果涉及基数为 $b$ 的展开表示（格雷码等价于二进制展开），以及基数为 $b$ 的和近似表示。这些表示构成若干簇，其中并非所有表示都允许彼此进行次递归转换。当从 $a$ 到 $b$ 的进制转换因子存在时，基数为 $b$ 的展开可转换为基数为 $a$ 的展开；此时，转换的开销相对于输入的位长呈指数级。当从 $a$ 到 $b$ 的进制转换因子存在时，基数为 $b$ 的下（上）方和近似可转换为基数为 $a$ 的下（上）方和近似；此时，转换开销涉及迭代，这保证了一个可在 $O(s)$ 时间内计算的、基数为 $b$ 的表示，可转化为一个可在 $O(t)$ 时间内计算的、基数为 $a$ 的表示，其中 $t$ 关于 $s$ 是原始递归的。

# 6 与戴德金分割子递归等价的表示

## 6.1 戴德金分割。

定义 6.1。设 $\alpha \in (0,1)$ 是一个无理数。$\alpha$ 的戴德金分割是函数 $D^\alpha$：$\mathbb{Q} \longrightarrow \{0,1\}$，定义为：$D^\alpha(q) = 0$ 当且仅当 $q < \alpha$。 □

用戴德金分割表示无理数，与用基数为 $b$ 的展开表示并非子递归等价；与对任意基数 $b$ 从下方或上方给出基数为 $b$ 的和近似表示，也并非子递归等价。很容易看出，我们可以在 $D^\alpha$ 中子递归计算基数为 $b$ 的展开 $E_b^\alpha$，见下文第 6.3 节；但对于任何固定的基数 $b$，都不可能在 $E_b^\alpha$ 中子递归计算 $D^\alpha$。对此不可能性的直观解释，与我们在第 27 页解释为何无法在 $E_2^\alpha$ 中子递归计算 $E_{10}^\alpha$ 非常相似：设 $\alpha$ 是一个无理数，其二进制展开以

$$\alpha \;=\; 0.000110011001100110011001100110011001100110011001100110011\ldots\,.$$

周期 0011 可以任意多次重复，因此，我们需要无界搜索来确定 $\alpha$ 是高于还是低于 $10^{-1}$，也就是说，我们需要无界搜索来计算 $D^\alpha(10^{-1})$。这个简单的例子也有助于直观地解释，为什么我们无法从上方或下方子递归计算 $\alpha$ 的基数-$b$ 和近似中的 $D^\alpha$，也就是说，$A_b^\alpha$ 或 $A_b^\alpha$。我们同样无法在 $D^\alpha$ 中子递归计算 $A_b^\alpha$ 或 $A_b^\alpha$，但要直观地解释其原因并不那么简单；有兴趣的读者可参阅 Kristiansen [24] 的第 7 节和第 8 节，了解有关戴德金分割与基数-$b$ 和近似之间关系的更多内容。

## 6.2 定义。

下面的定义给出了一些与戴德金分割表示子递归等价的表示。

定义 6.2。令 $\alpha \in (0,1)$ 为一个无理数。

1. $\alpha$ 的贝蒂序列是函数 $B^\alpha\colon\ \mathbb{N}^+ \longrightarrow \mathbb{N}$，定义为
$$\frac{B^\alpha(n)}{n} \;<\; \alpha \;<\; \frac{B^\alpha(n)+1}{n}\,.$$
2. $\alpha$ 的一般进制展开是函数
$$E^\alpha : (\mathbb{N} \setminus \{0,1\}) \times \mathbb{N}^+ \longrightarrow \mathbb{N}$$
其中 $E^\alpha(b,n) = E_b^\alpha(n)$（回忆一下，$E_b^\alpha$ 是 $\alpha$ 的基数为 $b$ 的展开，见定义 5.6）。

3. $\alpha$ 的赫尔维茨特征是函数 $H^\alpha : \mathbb{N} \longrightarrow \{0,1\}^*$，满足 $H^\alpha(0)\ H^\alpha(1)\ H^\alpha(2)\ldots$ 是法雷对树[2] $\mathcal{T}_F$中的一条路径；此外，对所有 $n \in \mathbb{N}$，都有 $\alpha \in \mathcal{T}_F(H^\alpha(n))$。

□

用戴德金分割 [6, 12], 、贝蒂序列 [14] 和赫尔维茨特征 [18] 表示实数，在19世纪或更早时就已为人所知。将赫尔维茨特征用于表示数，而不是作为学习其他内容的垫脚石，则是晚近得多的发明，参见 Lehman [27]。此外，如今所称的贝蒂序列，伯恩哈德·波尔查诺早先就已使用 [7], ，因此这种实数表示也可以称为波尔查诺测度。一般进制展开这种表示方法或许此前尚未得到研究，但它非常自然。

本节基于会议论文 Kristiansen & Simonsen [26]。

## 6.3 一般进制展开与戴德金分割之间的转换。

我们将通过逐一计算 $\alpha$ 的基数为 $b$ 的展开中的数字 $\mathtt{D}_1, \mathtt{D}_2, \mathtt{D}_3$，…，来计算 $E^\alpha(b,n)$。当我们确定了数字 $\mathtt{D}_1$，…，$\mathtt{D}_n$ 后，就知道

$$(0.\mathtt{D}_1 \ldots \mathtt{D}_n)_b < \alpha < (0.\mathtt{D}_1 \ldots \mathtt{D}_n)_b + b^{-n}$$

然后我们可以将区间分成

$$\left( (0.\mathtt{D}_1 \ldots \mathtt{D}_n)_b\, ,\, (0.\mathtt{D}_1 \ldots \mathtt{D}_n)_b + b^{-n} \right)$$

$b$ 个子区间，每个区间长度为 $b^{-n-1}$，并利用 $\alpha$ 的戴德金分割来确定数字 $D_{n+1}$。

引理 6.3. 设 $D : \mathbb{Q} \longrightarrow \{0,1\}$ 是无理数 $\alpha \in (0,1)$ 的戴德金分割。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^D : (\mathbb{N} \setminus \{0,1\}) \times \mathbb{N}^+ \longrightarrow \mathbb{N}$ 是 $\alpha$ 的一般进制展开
- $M^D$ 在输入 $(b,n)$ 上的运行时间为 $O(b^2\mathrm{poly}(n))$，且至多使用 $n \log b$ 次预言机调用，每次调用的输入规模至多为 $O(n \log b)$。

证明。 $M$ 通过归纳方式构造序列 $E_b^\alpha(1)\ E_b^\alpha(2)\ldots, E_b^\alpha(n)$：对每个 $i \in \{0,\ldots,n-1\}$，始终保持一个有理数端点为 $v_i, w_i \in \mathbb{Q}$ 的开区间 $I_i = (v_i, w_i)$，使得 (i) $\alpha \in I_i$，(ii) $v_i$ 是 $b^{-i}$的倍数，且 (iii)

[2]严格来说，经典的赫尔维茨特征对应于一条经过完整斯特恩-布罗科特树（而非我们在此考虑的法雷对树）的路径，因此，$(0,1)$ 中 $\alpha \in$ 的经典赫尔维茨特征 $H'$ 是由 $H'(0) = 0$ 和 $H'(q) = 0 \cdot H(q-1)$（其中 $q > 0$）定义的函数。这不会实质性地改变我们的结果。

$w_i - v_i = b^{-i}$。初始时，$I_0 = (0, 1)$。对于每个区间 $I_i$，$M$ 将 $I_i = (v_i, w_i)$ 分成 $b$ 个等长区间

$$(v_i, v_i + b^{-(i+1)}), \ldots, (v_i + (b-1)b^{-(i+1)}, v_i + b^{-i}) = (v_i + (b-1)b^{-(i+1)}, w_i) .$$

注意，对于任意区间 $(r_1, r_2)$，如果 $D(r_1) = D(r_2) = 0$，那么 $\alpha > r_2$；如果 $D(r_1) = D(r_2) = 1$，那么 $\alpha < r_1$（而 $D(r_1) = 1 \wedge D(r_2) = 0$ 这种情况不可能发生）。因此，$M$ 可以利用 $D$ 对上述区间集合（的端点）进行二分查找，以找到包含

$$\left( v_i + jb^{-(i+1)}, v_i + (j+1)b^{-(i+1)} \right) \tag{6.1}$$

$\alpha$ 的区间（注意，对于此区间，$D(v_i + jb^{-(i+1)}) = 0$，且 $D(v_i + (j+1)b^{-(i+1)}) = 1$）。接着，我们令 $(v_{i+1}, w_{i+1})$ 等于区间 (6.1)。根据构造，我们有 $E^\alpha(b, i+1) = j$。

显然，在每一步 $i$ 中，对 $D$ 至多进行 $\log b$ 次预言机调用；构造每个 $b$ 区间并写入查询纸带所需的时间，均为涉及数字的二进制表示长度的多项式时间，因此耗时 $O(\text{polylog}(b^i)) = O(\text{poly}(i)\text{polylog}(b))$。因此，生成 $E(b, n)$ 所需的总时间至多为 $O(bn\text{poly}(n)\text{polylog}(b)) = O(b^2\text{poly}(n))$，且对 $D$ 的查询次数至多为 $n\log b$。在每次预言机调用中，涉及的有理数都是区间的端点；这些端点是 $b$ 的负次幂之和，且所有幂的指数至多为 $n$。因此，所有预言机调用均可用至多 $O(n\log b)$ 位的有理数表示。

□

我们将一般进制展开转换为戴德金分割的算法基于以下观察：对于任意 $n, m \in \mathbb{N}$，只要 $0 < n/m < 1$，就有 $D^\alpha(n/m) = 0$ 当且仅当 $n/m < \alpha$，当且仅当 $n \le E^\alpha(m, 1)$。

引理 6.4. 设 $E : (\mathbb{N}\setminus\{0,1\}) \times \mathbb{N}^+ \longrightarrow \mathbb{N}$ 是无理数 $\alpha \in (0, 1)$ 的一般进制展开。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^E : \mathbb{Q} \longrightarrow \{0, 1\}$ 是 $\alpha$ 的戴德金分割
- $M^E$ 在输入 $n/m$ 上的运行时间为 $O(\log(\max\{n, m\}))$，且恰好进行一次预言机调用，其输入规模至多为 $O(\log(m))$。

证明。输入 $n/m \in \mathbb{Q}$、$M$ 后，首先检查 $m = 1$；若 $n \le 0$，则输出 0；若 $n \ge 1$，则输出 1。否则，$m > 1$，并且 $M$ 计算 $E(m, 1)$；根据定义，这是 $\{0, \ldots, m-1\}$ 中的一个元素。此后，若 $n \le E(m, 1)$，$M$ 就输出 0；否则输出 1。$M$ 需要将数对 $(m, 1)$ 的（表示）写入预言机带，并执行一次对大小至多为 $\max\{n, m\}$ 的数的比较，因此 $M$ 进行比较所用的时间为 $O(\log \max\{n, m\})$。$M$ 只需使用一次预言机调用，将数对 $(m, 1)$ 传给 $E$；该数对的表示至多使用 $O(\log m)$ 位。 □

## 6.4 贝蒂序列与戴德金分割之间的转换。

很容易看出如何将戴德金分割 $D^\alpha$ 转换为贝蒂序列 $B^\alpha$ ，因为 $B^\alpha(n)$ 的值是满足 $m/n < \alpha < (m+1)/n$ 的自然数 $m$ 。我们可以使用 $D^\alpha$ 搜索这个唯一的 $m$。

引理 6.5. 设 $D : \mathbb{Q} \longrightarrow \{0,1\}$ 是无理数 $\alpha \in (0,1)$ 的戴德金分割。存在一个参数化函数预言机图灵机 $M$ ，使得

- $\Phi_M^D : \mathbb{N}^+ \longrightarrow \mathbb{N}$ 是 $\alpha$ 的贝蒂序列
- $M^D$ 在输入 $n$ 上的运行时间为 $O(\text{polylog}(n))$，且至多使用 $\lceil \log\ n \rceil$ 次预言机调用，每次调用的输入规模至多为 $O(\log n)$。

证明。对于输入 $n$， $M$ 找到满足 $D(i/n) = 1$ 的最小 $i \in \{1,\ldots,n\}$。由于 $D(i/n) = 1$ 且 $j > i$ 蕴含 $D(j/n) = 1$，可以通过二分查找找到最小的 $i$ ，每一步都将搜索范围减半。具体做法是维护两个取值范围为 $\{1,\ldots,n\}$ 的整数 $l$ 和 $u$，最多需要 $\log\ n$ 次减半步骤。每一步减半时， $M$ 找到 $l$ 和 $u$ 之间的中点 $m$ ，将其二进制表示写入查询纸带，查询 $D$，并记录答案。然后， $l$ 和 $u$ 使用整数的基本二进制算术运算进行更新，这些整数的表示至多需要 $O(\log n)$ 位；若 $D(m/n) = 1$，则 $u := m$；若 $D(m/n) = 0$，则 $l := m$。显然，每一步中的算术和更新操作都能在关于整数表示规模的多项式时间内完成，因此运行时间为 $\text{polylog}(n)$。由于 $(i-1)/n < \alpha < i/n$，有 $B(n) = i - 1$，因此 $M^D$ 返回 $i - 1$。

□

为了验证将贝蒂序列 $B^\alpha$ 转换为戴德金分割 $D^\alpha$ 的算法是正确的，注意到对于任意 $m, n \in \mathbb{N}$，有 $D^\alpha(n/m) = 0$ 当且仅当 $n/m < \alpha$当且仅当 $n \leq B^\alpha(m)$,

引理 6.6. 设 $B : \mathbb{N}^+ \longrightarrow \mathbb{N}$ 是某个无理数 $\alpha \in (0,1)$ 的贝蒂序列。存在一个参数化函数预言机图灵机 $M$ ，使得

- $\Phi_M^B : \mathbb{Q} \longrightarrow \{0,1\}$ 是 $\alpha$的戴德金分割
- $M^B$ 在输入 $n/m$ 上运行的时间为 $O(\log(\max\{n,m\}))$，并且恰好进行一次输入规模为 $O(\log m)$ 的预言机调用。

证明。输入 $n/m$, $M$ 后，调用预言机 $B(m)$，得到一个整数 $B(m)$（其中 $B(m) \in \{0,1\ldots,m-1\}$）。如果 $n \leq B(m)$，则 $M$输出 0；否则， $M$ 输出 1。比较 $n \leq B(m)$ 可以利用 $n$ 和 $B(m)$ 的二进制表示逐位进行，这显然需要与 $\log(\max\{n, B(m)\}) \leq \log \max\{n,m\}$ 成线性关系的时间。在预言机带上写入 $m$ 显然也需要与 $\log m$ 成线性关系的时间。

□

## 6.5 赫尔维茨特征与戴德金分割之间的转换。

首先讨论如何使用赫尔维茨特征 $H^\alpha$ 作为预言机来计算戴德金分割 $D^\alpha$ 。

假设 $0 < n/m < 1$ ，其中 $n$ 和 $m$ 是互素自然数。令 $(a/b, c/d) = \mathcal{T}_{\mathrm{F}}(H^\alpha(n+m))$。根据命题 2.6，任何既约分数 $n/m$ 都会作为一个法里对中的分数出现在 $\mathcal{T}_{\mathrm{F}}$ 中，且深度至多为 $n+m-2$，因此 (i) $n/m \le a/b$ 和 (ii) $c/d \le n/m$ 中必有且仅有一个成立。由于 $a/b < \alpha < c/d$，我们有 $D^\alpha(n/m) = 0$ 当且仅当 $n/m \le a/b$。

引理 6.7. 设 $H:\ \mathbb{N} \longrightarrow \{0,1\}^*$ 是无理数 $\alpha \in \in(0, 1)$ 的赫尔维茨特征。存在一个参数化函数预言机图灵机 $M$ ，使得

- $\Phi_M^H:\ \mathbb{Q} \longrightarrow \{0,1\}$ 是 $\alpha$ 的戴德金分割

在输入 $n/m$ 时，• $M^H$ 的运行时间为 poly(max$\{n,m\}$)，并且恰好进行一次预言机调用，其输入规模至多为 $O$(log max$\{n,m\}$)。

证明。若 $n/m \le 0$，则 $M$ 输出 0。若 $n/m \ge 1$，则 $M$ 输出 1。令 $0 < n/m < 1$。我们假设 $n/m$ 为最简形式。然后， $M$ 使用 polylog(max$\{n,m\}$) 次运算计算 $n+m$ 的二进制表示，从而计算 $H(n+m)$，接着进行一次预言机调用；注意，预言机 $H(n+m)$ 的结果是一个长度恰为 $n+m \le$ poly(max$\{n,m\}$) 的比特串。$M$ 随后计算 $\mathcal{T}_{\mathrm{F}}(H(n+m))$，以得到法里对 $(a/b, c/d)$，使得$a/b < \alpha < c/d$。若 $n/m \le a/b$，则 $M$ 输出 0；否则， $M$ 输出 1。

根据命题 2.4， $M$ 可以在时间内找到 $(a/b, c/d)$

$$\mathrm{poly}(1 + |H(n+m)|) = \mathrm{poly}(\max\{n,m\})$$

并且可以在时间 $O$(log max$\{a,b,n,m\}$) 内检验 $n/m \le a/b$ 是否成立。通过对深度 $d$ 进行归纳，很容易看出， $\mathcal{T}_{\mathrm{F}}$ 中深度为 $d$ 的法里对里出现的任意分数，其分子或分母都不超过 $2^d$。因此，max$\{a,b,n,m\} \le 2^{n+m}$，检验便可在时间 $O(n+m) = O$(max$\{n,m\}$) 内完成。由此， $M$ 总共需要 poly(max$\{n,m\}$) 的时间。 □

我们将 Dedekind $D^\alpha$ 切割转换为 Hurwitz 特征 $H^\alpha$ 的算法并不令人意外。 $H^\alpha(n)$ 的值是法里对树 $\mathcal{T}_{\mathrm{F}}$ 中一条长度为 $n$ 的路径，路径上的每个区间都包含 $\alpha$。有了对 $D^\alpha$ 的访问权限后，我们就能轻松计算出这样一条路径。

引理 6.8. 令 $D:\ \mathbb{Q} \longrightarrow \{0,1\}$ 为无理数 $\alpha \in (0, 1)$ 的戴德金分割。存在一个参数化函数预言机图灵机 $M$ ，使得

- $\Phi_M^D : \mathbb{N} \longrightarrow \{0,1\}^*$ *is the Hurwitz characteristic of* $\alpha$

- $M^D$ 在输入 $n$ 时，运行时间为 poly($n$)，并且恰好使用 $n$ 次预言机调用，每次调用的输入规模至多为 $O(n)$。

证明。输入 $n$，$M$ 后，构造一条长度为 $n$ 的路径，位于 $\mathcal{T}_{\mathrm{F}}$ 中。$M$ 通过从 $i = 0$ 开始并递增 $i$ 来实现，同时维护当前的法里对 $(a_i/b_i, c_i/d_i)$，使得对于 $i = 0, \ldots, n$，$\alpha \in (a_i/b_i;\ c_i/d_i)$。将 $(a_i/b_i, c_i/d_i)$ 的中项作为中项，会在 $\mathcal{T}_{\mathrm{F}}$ 中产生 $(a_i/b_i, c_i/d_i)$ 的两个子节点 $p_L = (a_i/b_i, (a_i + c_i)/(b_i + d_i))$ 和 $p_R = ((a_i + c_i)/(b_i + d_i)\ c_i/d_i)$。由于 $\alpha$ 是无理数，它必定恰好位于开区间 $(a_i/b_i, (a_i + c_i)/(b_i + d_i))$ 和 $((a_i + c_i)/(b_i + d_i)\ c_i/d_i)$ 之一，因此 $(a_{i+1}/b_{i+1}, c_{i+1}/d_{i+1})$ 必定是 $p_L$ 或 $p_R$。显然，$\alpha \in (a_i/b_i, (a_i + c_i)/(b_i + d_i))$ 当且仅当 $D(a_i + c_i/b_i + d_i) = 1$，也当且仅当 $H(n)$ 的第 $i$ 位为 0。因此，$M$ 从 $(a_0/b_0, c_0/d_0) = (0/1, 1/1)$ 开始，并通过计算中项、在每一步查询 $D$，构造 $i = 1, \ldots, n$ 对应的 $n$ 个区间 $(a_i/b_i, c_i/d_i)$。注意，第 $i$ 步的查询是深度为 $i - 1$ 的某个法里对的中项（二进制表示），因此其分母至多为 $2^i$，其二进制表示至多使用 $O(\log 2^i) = O(i)$ 位。

由于在 $\mathcal{T}_{\mathrm{F}}$ 中深度为 $i$ 时的分子和分母大小至多为 $2^i$（因此可用 $i$ 位表示），在步骤 $i$ 计算中项最多需要 $O(i) = O(n)$的时间，通过两次标准竖式加法即可完成，而步骤 $i$ 恰好包含一次对 $D$的查询。因此，$M$ 构造 $H(n)$所需的总时间至多为 $O(n\text{poly}(n)) = \text{poly}(n)$，共进行 $n$ 次预言机调用，每次调用的规模至多为 $O(\log 2^n) = O(n)$。 □

## 6.6 总结。

现在，我们可以总结关于转换复杂度的结果，这些转换涉及与戴德金分割表示子递归等价的表示。

定理 6.9。考虑以下表示：(1) 戴德金分割，(2) 一般进制展开，(3) 赫尔维茨特征，以及 (4) Beatty 序列，并设$R_1$ 和 $R_2$ 是这四种表示中的任意两种。那么，对于任意时间界 $t$，我们有

$$O(t)_{R_2} \subseteq O(\text{poly}(t(2^n)))_{R_1}\ .$$

# 7 与最佳逼近等价的表示

## 7.1 最佳逼近。

通过左侧（或右侧）最佳逼近表示实数可能并不广为人知，但这是一种直观上

易于理解的自然表示，下面的定义应当无需任何解释。

定义 7.1. 设 $\alpha$ 是区间 (0, 1) 中的一个无理数，并设 $r = a/b$，其中 $a$、$b$ 是互素自然数。

若对于任意满足 $d \leq b$ 的自然数 $c$、$d$，都有 $c/d \leq a/b < \alpha$ 或 $\alpha < c/d$，则有理数 $r$ 是 $\alpha$ 的左最佳逼近。若对于任意满足 $d \leq b$ 的自然数 $c$、$d$，都有 $\alpha < a/b \leq c/d$ 或 $c/d < \alpha$，则有理数 $r$ 是 $\alpha$ 的右最佳逼近数。

$\alpha$ 的左最佳逼近是一个有理数序列 $\{r_i\}_{i\in\mathbb{N}}$，满足

$$0 = r_0 < r_1 < r_2 < \dots$$

并且每个 $r_i$ 都是 $\alpha$ 的左最佳逼近。$\alpha$ 的右最佳逼近是一个有理数序列 $\{r_i\}_{i\in\mathbb{N}}$，满足

$$1 = r_0 > r_1 > r_2 > \dots$$

并且每个 $r_i$ 都是 $\alpha$ 的右最佳逼近。 □

本节将研究若干与由左最佳逼近表示子递归等价的表示，以及若干与由右最佳逼近表示等价的表示。这两个等价类彼此不可比较，也就是说，一个等价类中的表示无法次递归转换为另一个等价类中的表示。我们将在本节末尾解释原因。

定义 7.2. 如果 $\{r_i\}_{i\in\mathbb{N}}$ $\alpha$ 的每个左最佳逼近都出现在序列 $\{r_i\}_{i\in\mathbb{N}}$中，那么左最佳逼近 $\{r_i\}_{i\in\mathbb{N}}$ $\alpha$ 是完全的。如果 $\alpha$ 的每个右最佳逼近都出现在序列中，那么右最佳逼近数 $\{r_i\}_{i\in\mathbb{N}}$ $\alpha$ 是完全的。

□

完全最佳逼近与法雷对树 $\mathcal{T}_\mathrm{F}$中的路径之间存在联系。令 $\alpha$为一个无理数，并令 $\sigma_0\sigma_1\sigma_2\cdots \in\{0,1\}^\omega$ 为 $\mathcal{T}_\mathrm{F}$ 中满足以下条件的唯一一条路径：

$$\alpha \in \mathcal{T}_\mathrm{F}(\sigma_0 \dots \sigma_n) = (a_n/b_n, c_n/d_n) \tag{7.1}$$

对任意 $n \in \mathbb{N}$都成立。那么，分数 $p/q$ 是 $\alpha$ 的左最佳逼近，当且仅当 $p/q$出现在序列 $a_0/b_0, a_1/b_1, a_2/b_2, \dots$ 中。

显然，序列中的每个 $a_i/b_i$ 都是$\alpha$ 的左最佳逼近，因为区间 $(a_i/b_i, c_i/d_i)$ 内的任何分数，其分母都严格大于 $b_i$ （见定理 2.2 ）。为了说明每个 $\alpha$ 的左最佳逼近确实都会出现在该序列中，反设某个左最佳逼近 $p/q$ 并未出现在其中。那么我们有

$$\frac{a_i}{b_i} < \frac{p}{q} < \frac{a_{i+1}}{b_{i+1}} = \frac{a_i + c_i}{b_i + d_i}$$

对于某个 $i$。由于 $p/q$ 是最佳逼近，我们必有 $q < b_{i+1} = b_i + d_i$。但我们还有 $p/q \in (a_i/b_i, c_i/d_i)$，这与定理 2.2 相矛盾；该定理指出，区间 $(a_i/b_i, c_i/d_i)$ 内的任何分数，其分母都大于或等于 $b_i + d_i$。

同理，$p/q$ 是 $\alpha$ 的右最佳逼近数，当且仅当 $p/q$出现在序列 $c_0/d_0, c_1/d_1, c_2/d_2, \ldots$中。因此，我们得到下面的引理。

引理 7.3. 设 $\alpha$ 是一个满足（7.1）的无理数。(i) 序列 $\{a_i/b_i\}_{i\in\mathbb{N}}$ 包含 $\alpha$ 的所有左最佳逼近数，且不包含其他数。(ii) 序列 $\{c_i/d_i\}_{i\in\mathbb{N}}$ 包含 $\alpha$ 的所有右侧最佳逼近数，且不包含其他数。

我们可以利用 $\mathcal{T}_{\mathrm{F}}$，从 $\alpha$的任意一个左最佳逼近 $\{\hat{a}_i/b_i\}_{i\in\mathbb{N}}$ 子递归计算出 $\alpha$ 的完全左最佳逼近 $\{a_i/b_i\}_{i\in\mathbb{N}}$ ^。注意，对于任意 $i \in \mathbb{N}$，都有 $a_i/b_i \leq$ ^ $a_i/b_i < \alpha$ 。因此，至少有 $n$ 个 $\alpha$ 的左最佳逼近数小于或等于$\hat{a}_n/b_n$。于是，我们可以在 $\mathcal{T}_{\mathrm{F}}$ 中构造一条路径 $\sigma_0 \ldots \sigma_m$，使得$\hat{a}_n/b_n$是 $\mathcal{T}_{\mathrm{F}}(\sigma_0 \ldots \sigma_m)$的左端点，从而找到 $a_n/b_n$。根据引理 7.3，所有小于或等于$\hat{a}_n/b_n$的 $\alpha$的左最佳逼近数都会出现在路径 $\sigma_0 \ldots \sigma_m$上。我们已论证至少有 $n$ 个这样的数，因此可以选取第 $n$个。根据引理 2.6，有 $m \leq \hat{a}_n + b_n - 2$，因此不需要无界搜索。对称算法将从任意一个右最佳逼近数计算出完全右最佳逼近。

引理 7.4。设 $L\colon \mathbb{N} \longrightarrow \mathbb{Q}$ 是无理数 $\alpha \in (0,1)$ 的左最佳逼近。假设 $L$ 可由一台运行时间为 $O(s)$ 的图灵机计算。存在一台参数化函数预言机图灵机 $M$，使得

- $\Phi_M^L\colon \mathbb{N} \longrightarrow \mathbb{Q}$ 是 $\alpha$ 的完全左最佳逼近
- $M^L$ 在输入 $n$ 上运行时间为 $\mathrm{poly}(2^{s(\|n\|)})$，并且恰好进行一次输入规模为 $\|n\|$ 的预言机调用。

证明。根据上面给出的算法，注意到我们在法里树中追踪的路径长度至多为 ^ $a_n + b_n - 2$，且 $\|a_n/b_n\| \leq s(\|n\|)$。该算法执行的算术运算所需时间关于操作数的长度为多项式时间。

□

根据上面的讨论，不难看出，由戴德金分割表示相对于左最佳逼近表示以及右最佳逼近表示都是次递归的：若 $\{a_i/b_i\}_{i\in\mathbb{N}}$ 是 $\alpha$ 的左最佳逼近，则有 $b_n > n$，因此 $m/n < \alpha$ 当且仅当 $m/n < a_n/b_n$。若 $\{c_i/d_i\}_{i\in\mathbb{N}}$ 是 $\alpha$ 的右最佳逼近，则有$d_n > n$，因此 $\alpha < m/n$ 当且仅当 $c_n/d_n < m/n$。

引理 7.5。设 $L\colon \mathbb{N} \longrightarrow \mathbb{Q}$ 是一个无理数$\alpha \in (0,1)$的左最佳逼近。假设 $L$ 可由一台 $O(s)$时间图灵机计算。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^L$：$\mathbb{Q} \longrightarrow \{0,1\}$ 是 $\alpha$的戴德金分割
- $M^L$ 在输入 $p/q$ 上的运行时间为 poly($s(\|q\|)$)，并且恰好进行一次输入规模为 $\|q\|$的预言机调用。

证明。我们向预言机询问第 $q$ 个最佳逼近 $a_q/b_q$。于是我们有$p/q \le a_q/b_q < \alpha$ 或 $\alpha < p/q$。因此，只需进行一次预言机调用并比较有理数，就能判定 $p/q < \alpha$ 是否成立。 □

## 7.2 定义。

现在我们将定义并说明几种与左侧或右侧最佳逼近子递归等价的表示。

$\alpha$从下方（上方）的基数-$b$近似，其定义和基数-$b$近似的定义在第3节给出。从下方（上方）的一般和近似涵盖了任意基数 $b$ 下 $\alpha$ 从下方（上方）的基数-$b$ 和近似。下面给出正式定义。

定义 7.6。 $\alpha$ 的从下方的一般和逼近是函数 $\hat{}\hat{}\hat{}G^\alpha : (\mathbb{N} \setminus \{0,1\}) \times \mathbb{N} \longrightarrow \mathbb{Q}$，定义为 $G^\alpha(b,n) = A_b^\alpha(n)$。 $\alpha$ 的上方一般和逼近是函数 $\check{}G^\alpha : (\mathbb{N} \setminus \{0,1\}) \times \mathbb{N} \longrightarrow \mathbb{Q}$，定义为 $\check{}G^\alpha(b,n) = \check{}A_b^\alpha(n)$。 □

我们所说的贝尔序列，是由自然数组成的无限序列。这样的序列 $a_0, a_1, a_2, \ldots$ 表示区间 (0, 1) 内的一个无理数 $\alpha$。我们将区间 (0, 1) 分割成无穷多个端点为有理数的开子区间。具体来说，我们采用如下分割：

$$(\ 0/1\ ,\ 1/2\ )\ (\ 1/2\ ,\ 2/3\ )\ (\ 2/3\ ,\ 3/4\ )\ \ldots\ (\ n/(n+1)\ ,\ (n+1)/(n+2)\ )\ldots\ .$$

序列 $a_0$ 的第一个数告诉我们 $\alpha$ 位于这些区间中的哪一个。因此，如果 $a_0 = 17$，我们就知道 $\alpha$ 位于区间 $(17/18, 18/19)$ 内。接着，我们以类似的方式分割区间 $(17/18, 18/19)$。序列 $a_1$ 的第二个数告诉我们 $\alpha$ 位于这些区间中的哪一个，我们就这样继续下去。

一般而言，为了分割区间 $(q, r)$，我们需要一个严格递增的有理数序列 $s_0, s_1, s_2 \ldots$，使得 $s_0 = q$ 且 $\lim_i s_i = r$。我们将使用分割 $s_i = (a + ic)/(b + id)$，其中 $a, b$ 是满足 $q = a/b$ 的（唯一一对）互素自然数，而 $c, d$ 是满足 $r = c/d$ 的（唯一一对）互素自然数（令 $0 = 0/1$ 和 $1 = 1/1$）。这种特定的分割确保了贝尔序列所诱导的每个区间都能在法雷对树 $\mathcal{T}_{\mathrm{F}}$ 中找到。

我们将上述说明的贝尔序列称为标准贝尔序列。无理数 $\alpha$ 的标准贝尔序列按字典序先于无理数 $\beta$ 的标准贝尔序列，当且仅当 $\alpha < \beta$。我们还将使用我们称为对偶贝尔序列的序列。 $\alpha$ 的对偶序列按字典序先于 $\beta$ 的对偶序列，当且仅当 $\alpha > \beta$。我们通过

使用递减的有理数序列来划分区间，从而得到对偶序列，例如，区间 (0,1) 将被划分为以下区间

$$( 1/1 , 1/2 ) ( 1/2 , 1/3 ) ( 1/3 , 1/4 ) \ldots ( 1/n , 1/(n+1) ) \ldots \quad .$$

定义 7.7. 设 $f\colon \mathbb{N} \longrightarrow \mathbb{N}$ 是任意函数，并设 $n \in \mathbb{N}$。我们定义区间 $I_f^n$ 为 $I_f^0 = (0/1, 1/1)$，且

$$I_f^{n+1} \quad = \quad \left( \frac{a + f(n)c}{b + f(n)d} , \frac{a + f(n)c + c}{b + f(n)d + d} \right)$$

若 $I_f^n = (a/b, c/d)$。我们定义区间 $J_f^n$ 为 $J_f^0 = (0/1, 1/1)$，且

$$J_f^{n+1} \quad = \quad \left( \frac{a + f(n)a + c}{b + f(n)b + d} , \frac{f(n)a + c}{f(n)b + d} \right)$$

若 $J_f^n = (a/b, c/d)$。函数 $B\colon \mathbb{N} \longrightarrow \mathbb{N}$ 是无理数 $\alpha \in (0,1)$ 的标准贝尔表示，当且仅当对每个 $n$ 都有 $\alpha \in I_B^n$。函数 $A\colon \mathbb{N} \longrightarrow \mathbb{N}$ 是无理数 $\alpha \in (0,1)$ 的对偶贝尔表示，当且仅当对每个 $n$ 都有 $\alpha \in J_A^n$。

□

单位分数，即分子为 1 的分数，在古埃及已有研究，参见例如 [2], ; 它们也称为埃及分数。在文献中，埃及分数展开可以指将一个数表示为若干分子为 1 的分数之和的任意表示。我们在下文给出的定义确保区间 (0,1) 中的每个无理数都有唯一的埃及分数展开，参见 Cohen [11]。

定义 7.8. 函数 $E^\alpha\colon \mathbb{N}^+ \longrightarrow \mathbb{N}$ 是 $\alpha$ 的埃及分数展开，当且仅当

$$\alpha = \sum_{i=1}^{\infty} \left( \prod_{j=1}^{i} E(j) \right)^{-1}$$

并且 $E(i) \leq E(i+1)$（对所有 $i \in \mathbb{N}^+$）。 □

例如，

$$\sqrt{2} - 1 = \frac{1}{3} + \frac{1}{3 \cdot 5} + \frac{1}{3 \cdot 5 \cdot 5} + \frac{1}{3 \cdot 5 \cdot 5 \cdot 16} + \frac{1}{3 \cdot 5 \cdot 5 \cdot 16 \cdot 18} + \ldots$$

而且这是 $\sqrt{2} - 1$ 的唯一埃及分数展开。另一种基于埃及分数的无理数表示与左最佳逼近有关，参见 Beck 等人 [4]。事实上，根据定理 2.2 以及左最佳逼近与法雷对树之间的关系（引理 7.3），很容易推导出，完全左最佳逼近中相邻分数之差是一个单位分数。因此，$\alpha$ 的完全左最佳逼近会导出一个单位分数级数，其和为 $\alpha$。

定义 7.9。若 $\sum_{i=1}^{\infty} q_i = \alpha$，且序列 $(\sum_{i=1}^{j} q_i)_j$ 是 $\alpha$ 的完全左最佳逼近，则正单位分数序列 $\{q_i\}$ 是 $\alpha \in (0,1)$ 的法里-埃及展开。

□

例如

$$\sqrt{2} - 1 = \frac{0}{1} + \frac{1}{3} + \frac{1}{15} + \frac{1}{85} + \frac{1}{493} + \dots$$

是法里-埃及展开，与完全左最佳逼近 $0/1,\ 1/3,\ 2/5,\ 7/17\dots$ 相关联。

法里-埃及展开表示与完全左最佳逼近表示密切相关，而且很容易将法里-埃及展开转换为完全左最佳逼近，反之亦然。

一般和逼近（从上方和下方）由 Kristiansen [23] 引入，并由 Georgiev 等人 [16] 进一步研究。[16]，研究了左右最佳逼近，Kristiansen [25] 研究了标准与对偶贝尔序列。本文首次研究了埃及分数展开表示的计算复杂性。

## 7.3 一般和逼近与最佳逼近之间的转换。

设 $\{a_i/b_i\}_{i\in\mathbb{N}}$ 是 $\beta$ 的一个左最佳逼近。我们将给出一种算法，使用 $\{a_i/b_i\}_{i\in\mathbb{N}}$ 作为预言机来计算 $\beta$ 的从下方的一般和逼近，即 $\hat{}G^{\beta}$。该算法的第 $i$ 次迭代生成 $\hat{}G^{\beta}(b,i)$。计算出所有 $i < n$ 对应的 $\hat{}G^{\beta}(b,i)$ 后，该算法也将计算出这个和

$$\frac{c}{d} \;=\; \sum_{i=1}^{n-1} \hat{G}^{\beta}(b,i)\ .$$

然后，通过执行以下指令计算 $\hat{}G^{\beta}(b,n)$。

- 步骤 1：向预言机询问 $a_d/b_d$ 的值。令 $c'/d' = a_d/b_d$。注意，$d$ 是 $b$ 的幂；并且我们有 $d' > d$ 和 $c/d < c'/d' < \beta$。
- 步骤 2：计算 $m := \lceil \log_b d'd \rceil$。说明：我们有 $c'/d' - c/d > 0$，因此也有

$$\frac{c'}{d'} - \frac{c}{d} = \frac{c'd - cd'}{d'd} \;\geq\; \frac{1}{d'd}\ .$$

  由此可得 $c/d + 1/b^m \leq c'/d' < \beta$。

- 步骤 3：向预言机询问 $a_{b^m}/b_{b^m}$ 的值，并计算（详见下文）最小的 $k \leq m$ 和基数为 $b$ 的数字 $\mathtt{D} > 0$，使得

$$\frac{c}{d} + \frac{\mathtt{D}}{b^k} \;<\; \beta \;<\; \frac{c}{d} + \frac{\mathtt{D}}{b^k} + \frac{1}{b^k}\ . \tag{7.2}$$

  给出输出 $\mathtt{D}/b^k$。

关于步骤 3 的说明：对于任意左最佳逼近 $a_i/b_i$，我们有 $i < b_i$ 。因此，对于任意分数 $p/b^k$，若 $k \le m$，则有 $p/b^k < \beta$ 当且仅当 $p/b^k \le a_{b^m}/b_{b^m}$；对于这样的 $k$，(7.2) 等价于

$$\frac{c}{d} + \frac{\mathtt{D}}{b^k} \;\le\; \frac{a_{b^m}}{b_{b^m}} \;<\; \frac{c}{d} + \frac{\mathtt{D}}{b^k} + \frac{1}{b^k}\,. \tag{7.3}$$

满足 (7.3) 的最小 $k$ （对于某个 D）就是满足下式的最小 $k$

$$\frac{c}{d} + \frac{1}{b^k} \;\le\; \frac{a_{b^m}}{b_{b^m}}$$

为了求出该 $k$，我们将不等式改写为

$$\frac{1}{b^k} \;\le\; \frac{a_{b^m}}{b_{b^m}} - \frac{c}{d}$$

这又可以改写为

$$b^k \;\ge\; \frac{db_{b^m}}{da_{b^m} - cb_{b^m}}\,.$$

因此，所需的 $k$ 为

$$k = \left\lceil \log_b \frac{d \cdot b_{b^m}}{d \cdot a_{b^m} - c \cdot b_{b^m}} \right\rceil\,.$$

求得 $k$ 后，我们寻找一个使 (7.3) 成立的 D 值，显然

$$\mathtt{D} = \left\lfloor \left( \frac{a_{b^m}}{b_{b^m}} - \frac{c}{d} \right) / b^k \right\rfloor\,.$$

该算法的正确性可直接由关于步骤 1–3 的注释以及从下方的一般和逼近的定义得出。通过对称算法，可以将右最佳逼近转换为从上方的一般和逼近。

引理 7.10. 设 $L : \mathbb{N} \longrightarrow \mathbb{Q}$ 是无理数$\alpha \in (0, 1)$的左最佳逼近。假设 $L$ 可在时间界 $s$内计算，并设 $f(b, n) = (\lambda x.(\|b\| + 2s(x)))^{(n)}(1)$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^L : (\mathbb{N} \setminus \{0, 1\}) \times \mathbb{N} \longrightarrow \mathbb{Q}$ 是 $\alpha$从下方的一般和逼近
- $M^L$ 在输入 $b$上，$n$的运行时间为多项式$(s(f(b, n)))$，且至多使用 $2n$ 次预言机调用，每次调用的输入规模至多为 $f(b, n)$。

证明。当然，这个证明是归纳的。我们声称，$d$、$b^m$的位长有上界 $f(b, n)$。因此，根据假设以及机器的

运行时间限制了其输出大小这一规则，我们还可以得出，分母 $b_d$、$b_{b^m}$ 的位长有上界 $s(f(b,n))$。归纳步骤如下：算法首先设定 $d' = b_d$。我们归纳地假设，每个 $G^\alpha(b,i)$、$i < n$的分母位长均满足上界 $s(f(b,n-1))$。公分母 $d$ 是最后一项的分母，因此其位长有上界 $s(f(b,n-1))$。根据我们对计算 $b_d$所用时间界 $s$ 的假设，可得 $\|d'\| \le s(\|d\|) \le s(f(b,n-1))$。我们如下界定下一个分母 $d_{b^m}$ 的大小。首先，

$$\begin{aligned}\|b^m\| &= \|b^{\lceil log_b(dd')\rceil}\| \le \|bdd'\| \\ &\le \|b\| + \|d\| + \|d'\| \le \|b\| + s(f(b,n-1)) + s(f(b,n-1)) \le \\ &\qquad \|b\| + 2s(f(b,n-1)) = f(b,n)\end{aligned}$$

再根据时间界 $s$ 的假设，我们有

$$\|d_{b^m}\| \le s(\|b^m\|) \le s(f(b,n)) .$$

执行时间关于所处理数字的大小呈多项式关系，因此关于 $s(f(b,n))$ 也呈多项式关系。 □

接下来，我们给出一种算法，用于计算 $\beta$ 的完全左最佳逼近 $\{a_i/b_i\}_{i\in\mathbb{N}}$；该算法将 $\hat{G}^\beta$（$\beta$ 的从下方的一般和逼近）用作预言机。注意，借助预言机，可以轻松计算 $\beta$ 的戴德金分割 $D^\beta$：当且仅当 $\hat{G}^\beta(d,1) \ge c/d$ 时，我们有 $D^\beta(c/d) = 0$。

计算 $a_0/b_0$ 是显而易见的，因为根据约定，我们有 $a_0/b_0 = 0/1$。为了计算 $a_1/b_1$，算法向预言机查询 $\hat{G}^\beta(2,1)$ 的值。对于某个 $M$，我们有 $\hat{G}^\beta(2,1) = 1/M$。随后，算法利用 $\beta$ 的戴德金分割，搜索满足 $1/N < \beta < 1/(N-1)$ 的 $N \le M$，并将 $a_1/b_1$ 设为 $1/N$。（这样的 $N$ 一定存在，因为 $1/M < \beta$。注意，$2/N \ge 1/(N-1) > \beta$，因此算法正确计算出 $a_1/b_1$。）

当 $n \ge 1$ 时，算法通过以下过程计算 $a_{n+1}/b_{n+1}$：

- 设
$$\frac{c'}{d'} = \frac{a_n}{b_n} + \hat{G}^\beta(b_n, 2)$$
其中 $c'$ 和 $d'$ 互素。
- 利用 $\beta$ 的戴德金分割进行搜索，找出 $c$ 和满足 $b_n < d \le d'$ 且 $a_n/b_n < c/d < \beta < (c+1)/d$ 的最小 $d$。设 $a_{n+1}/b_{n+1} = c/d$。

为验证算法正确，注意到 $a_n/b_n = \hat{G}^\beta(b_n,1)$。因此，我们有 $a_n/b_n < c'/d' < \beta < (c'+1)/d'$，除非存在一个分数 $c/d$ 满足 $d < d'$ 且 $a_n/b_n < c/d < \beta < (c+1)/d$，否则 $c'/d'$ 将是一个左最佳逼近。我们要找的是满足条件的最小 $d$，因此得到一个完全最佳逼近。

通过对称算法，可以将从上方的一般和逼近转换为右最佳逼近。

引理 7.11。设 $G\colon (\mathbb{N}\setminus\{0,1\})\times\mathbb{N}\longrightarrow\mathbb{Q}$ 是无理数 $\beta\in(0,1)$ 的从下方的一般和逼近。假设 $G$ 可在时间界 $s$ 内计算，并设 $f(n)=(\lambda x.2s(x))^{(n)}(1)$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^G:\ \mathbb{N}\longrightarrow\mathbb{Q}$ 是 $\beta$的完全左最佳逼近
- $M$ 在输入 $n$ 时，运行时间为 $\mathrm{poly}(2^{f(n)})$，且最多使用 $O(2^{f(n)})$ 次预言机调用，每次输入规模至多为 $f(n)$。

证明。首先，我们说明在归纳步骤中如何实现对 $c,d$ 的搜索。我们检验的条件是 $b_n<d\le d'$ 且 $a_n/b_n\le c/d<\beta<(c+1)/d$。预言机可用于将第二个合取项替换为 $a_n/b_n\le c/d=G(d,1)$。因此，要执行搜索，我们先询问 $G(b_n,2)$，然后从 $b_n$ 开始，依次对 $d$的各个值询问 $G(d,1)$，最多询问到 $G(b_n,2)$。由此可知，计算 $a_{n+1}/b_{n+1}$ 时涉及的最大数值为 $G(b_n,2)$。

现在，考虑计算 $a_0/b_0,\dots,a_n/b_n$的整个过程：在计算 $a_{i+1}/b_{i+1}$ 的过程中，我们只进行一次预言机调用来界定搜索范围，然后从 $b_i+1$ 搜索到 $b_{i+1}$，对每次测试各进行一次预言机调用。容易看出，预言机调用总次数为 $n+b_n$。为了得到关于算法输入的界，我们注意到，归纳步骤中的 $d'$ 至多为 $G(b_{n-1},2)$的分母。根据假设，$G(b_{n-1},2)$占用的位数至多为 $s(2\|b_{n-1}\|)$，因为分数的表示大小至多是其分母大小的两倍。因此

$$b_n<2^{2s(\|b_{n-1}\|)}$$

因此 $\|b_n\|\le 2s(\|b_{n-1}\|)$。这给出了计算中涉及的最大数的位长上界 $f(n)$。该数的值有 $2^{f(n)}$ 作为上界，因此也限制了计算的步数，从而使执行时间不超过一个多项式。

□

## 7.4 贝尔序列与最佳逼近之间的转换。

引理 7.12。我们有

$$I_f^{n+1}=\mathcal{T}_{\mathrm{F}}(1^{f(0)}01^{f(1)}0\dots1^{f(n)}0) \tag{i}$$

*and*

$$J_f^{n+1}=\mathcal{T}_{\mathrm{F}}(0^{f(0)}10^{f(1)}1\dots0^{f(n)}1)\,. \tag{ii}$$

证明。我们证明 (i)。(ii) 的证明是对称的。设 $\sigma=1^{f(0)}01^{f(1)}0\dots1^{f(n-1)}0$。注意，当 $\sigma$ 是空序列时，我们有 $\mathcal{T}_{\mathrm{F}}(\sigma)=(0/1,1/1)=I_f^0$。

假设 $\mathcal{T}_{\mathrm{F}}(\sigma) = I_f^n = (a/b, c/d)$。我们需要证明

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^{f(n)}0) \;=\; I_f^{n+1}\,. \tag{7.4}$$

令 $k = f(n)$。我们对 $k$ 作第二重归纳，以证明 (7.4)。

假设 $k = 0$。由定义 2.3，我们有

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^{f(n)}0) \;=\; \mathcal{T}_{\mathrm{F}}(\sigma 1^{0}0) \;=\; \mathcal{T}_{\mathrm{F}}(\sigma 0) \;=\; (\; a/b\,,\,(a+c)/(b+d)\;)\,.$$

由定义 7.7，我们有

$$I_f^{n+1} \;=\; (\;(a+kc)/(b+kd)\,,\,(a+kc+c)/(b+kd+d)\;) \;=\; (\;a/b\,,\,(a+c)/(b+d)\;)\,.$$

因此，当 $f(n) = 0$ 时，(7.4) 成立。现在，根据归纳假设，假设

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^{k}0) \;=\; \left(\;\frac{a+kc}{b+kd}\,,\,\frac{a+kc+c}{b+kd+d}\;\right)\,. \tag{7.5}$$

注意，(7.5) 的右侧正是用 $k$ 对 $I_f^{n+1}$ 关于 $f(n)$ 的定义。现在，由 (7.5) 和定义 2.3，我们有

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^{k}) \;=\; \left(\;\frac{a+kc}{b+kd}\,,\,\frac{c}{d}\;\right)\,. \tag{7.6}$$

此外，由 (7.6) 和定义 2.3，我们有

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^{k+1}) \;=\; \left(\;\frac{a+kc+c}{b+kd+d}\,,\,\frac{c}{d}\;\right) \;=\; \left(\;\frac{a+(k+1)c}{b+(k+1)d}\,,\,\frac{c}{d}\;\right) \tag{7.7}$$

由 (7.7) 和定义 2.3，我们有

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^{k+1}0) \;=\; \left(\;\frac{a+(k+1)c}{b+(k+1)d}\,,\,\frac{a+(k+1)c+c}{b+(k+1)d+d}\;\right)\,. \tag{7.8}$$

注意，(7.8) 的右侧就是 $I_f^{n+1}$ with $k+1$ for $f(n)$ 的定义。这证明了 (7.4) 成立。 □

由上述引理可知，我们很容易看出如何将标准贝尔序列 $B$ 转换为完全右最佳逼近。我们使用 $B$ 来计算法雷对树中的一个区间 $I$，使得 $I = \mathcal{T}_{\mathrm{F}}(1^{B(0)}01^{B(1)}0\ldots1^{B(n)}0)$。根据引理 7.12 (i)，我们有 $I = I_B^{n+1}$。由此可得，$I$ 的右端点是 $\alpha$ 的完全右最佳逼近中的第 $n$ 个逼近项（见引理 7.3）。

根据引理 7.12 (ii)，我们有一个对称算法，可将对偶贝尔序列转换为左最佳逼近。

引理 7.13。设 $B : \mathbb{N} \longrightarrow \mathbb{N}$ 是无理数$\alpha \in (0,1)$ 的标准贝尔序列。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^B$：$\mathbb{N} \longrightarrow \mathbb{Q}$ 是 $\alpha$ 的完全右最佳逼近
- $M^B$ 在输入 $n \in \mathbb{N}$ 上运行的时间为 poly($n + \sum_{i=0}^{n} B(i)$)，并使用 $n$ 次预言机调用，每次调用的输入规模为 $O(\log n)$

此外，如果 $B$ 可在时间界 $s$ 内计算，则 $\Phi_M^B$ 的运行时间为 poly($n2^{s(\|n\|)}$)。

证明。我们必须计算 $I = \mathcal{T}_{\mathrm{F}}(1^{B(0)}01^{B(1)}0\ldots1^{B(n)}0)$。根据命题 2.4，这需要多项式时间 poly($n+\sum_{i=0}^{n} B(i)$)。按照标准论证，这表明如果 $B$ 可在时间界 $s$ 内计算，则 $\Phi_M^B$ 的运行时间为 poly($n2^{s(\|n\|)}$)。

□

引理 7.12 (i) 还给出了一个将完全右最佳逼近转换为标准贝尔序列的算法：给定 $\alpha$ 的完全右最佳逼近 $\{a_i/b_i\}_{i\in\mathbb{N}}$，我们可以计算一个形如 $1^{k_0}01^{k_1}0\ldots1^{k_n}0$ 的（唯一）字符串，使得区间 $\mathcal{T}_{\mathrm{F}}(1^{k_0}01^{k_1}0\ldots1^{k_i}0)$ 的右端点等于 $\{a_{i+1}/b_{i+1}\}$（对所有 $i \leq n$ 均成立）。根据引理 7.12，我们有 $B(i) = k_i$，其中 $B$ 是 $\alpha$ 的标准贝尔序列。引理 7.12 (ii) 给出了一个将完全左最佳逼近转换为对偶贝尔序列的算法。

引理 7.14. 设 $R$： $\mathbb{N} \longrightarrow \mathbb{Q}$ 是无理数 $\alpha \in (0, 1)$ 的完全右最佳逼近。存在一个参数化函数预言机图灵机$M$，使得

- $\Phi_M^R$： $\mathbb{N} \longrightarrow \mathbb{N}$ 是 $\alpha$的标准贝尔序列
- $M^R$ 在输入 $n$ 上运行，时间为多项式($2^{\|R(n)\|}$)，并使用 $n$ 次预言机调用，每次输入规模为 $O(\log n)$

此外，如果 $R$ 可在时间界 $s$ 内计算，则 $\Phi_M^R$ 的运行时间为 poly($2^{s(\|n\|)}$)。

证明。机器必须在 $\mathcal{T}_{\mathrm{F}}$ 中沿着一条路径，从根节点一直追踪到首次出现 $R(n) = a_n/b_n$ 的位置。根据引理 2.6，这最迟会在深度 $a_n + b_n - 2 < 2^{\|R(n)\|}$ 处发生。每一层所选的分支由相应的最佳逼近决定，因此机器必须计算 $R(0)$ 到 $R(n)$。每一层的工作量相对于深度和所涉及数字的大小都是多项式级的（见引理 2.7），而这些都关于 $a_n + b_n$ 呈多项式级，因此也关于 $2^{\|R(n)\|}$ 呈多项式级。

根据标准论证，这表明如果 $R$ 可在时间界 $s$ 内计算，那么 $\Phi_M^R$ 的运行时间为 poly($2^{s(\|n\|)}$)。 □

## 7.5 从向下的一般和逼近到埃及分数展开。

令 $\hat{G}^{\alpha}:(\mathbb{N}\setminus\{0,1\})\times\mathbb{N}\longrightarrow\mathbb{Q}$ 为 $\alpha\in(0,1)$ 从下方的一般和逼近。我们将说明如何将 $\hat{G}^{\alpha}$ 转换为埃及分数展开$E^{\alpha}$。

该算法递归运行。对于基本情形，假设 $\hat{G}^{\alpha}(2,1)=a/b$（其中 $b$ 是 2 的某个幂）。然后，我们寻找最小的 $2\leq d\leq b$，使得 $\hat{G}^{\alpha}(d,1)$ 的分母等于 $d$。这意味着 $1/d<\alpha<1/(d-1)$，并且我们令 $E^{\alpha}(1)=d$。在一般情形下，算法首先计算所有 $i\leq n$对应的 $E^{\alpha}(i)$，然后按如下方式计算 $E^{\alpha}(n+1)$：

- 步骤 1：计算埃及分数展开中前 $n$ 项的和

$$\frac{c}{d}=\sum_{i=1}^{n}\left(\prod_{j=1}^{i}E^{\alpha}(i)\right)^{-1}.$$

- 步骤 2：令 $c'/d'=\hat{G}^{\alpha}(d,1)+\hat{G}^{\alpha}(d,2)$。现在 $d'$ 是 $d$的某个幂，并且 $c/d<c'/d'<\alpha$。
- 步骤 3：搜索满足 $c/d+(dm)^{-1}<\alpha$ 的最小 $m$。这个搜索ˆ可以通过 $\alpha$ 的戴德金分割来执行，进而如前所示使用 $G^{\alpha}$ 模拟。由于 $E(n)\leq m\leq d'/d$，搜索有界。返回$m$。

除了需要为搜索设定界限外，我们的算法就是自然的贪心算法，众所周知，该算法可以计算埃及分数展开，参见 Cohen [11]。

引理 7.15。设 $G:(\mathbb{N}\setminus\{0,1\})\times\mathbb{N}\longrightarrow\mathbb{Q}$ 是无理数 $\alpha\in(0,1)$ 从下方的一般基数近似。假设 $G$ 在时间界 $s$ 内可计算，并令 $f(n)=(\lambda x.s(2x))^{(n)}(2)$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^G:\ \mathbb{N}^+\longrightarrow\mathbb{N}$ 是 $\alpha$ 的埃及分数展开
- $M^G$ 在输入 $n$ 时，运行时间为 $\mathrm{poly}(2^{f(n)})$，最多使用 $2^{f(n)+1}$ 次预言机调用，每次调用的输入规模至多为 $2f(n)$。

证明。决定该算法复杂度的关键量，是所遇到的最大分母的大小（位长），在上述算法中记为 $d'$。我们按如下方式将 $d'$ 界定为 $f(n)$。在一般情形下，$d'$ 通过调用 $G(d,2)$ 获得，其中 $d$ 来自上一次迭代，因此我们通过归纳法假设 $\|d\|\leq f(n-1)$。输入 $d,2$ 的表示长度是 $d$ 的两倍（假设一对整数通过将两个二进制数交错合并来表示，如第 2 节所述），因此预言机

输入的大小由 $2f(n-1)$ 界定，而预言机输出的大小由 $s(2f(n-1)) = f(n)$ 界定。在基本情形下，我们向预言机查询 $G(2,1)$：结果的大小由 $s(4) = (\lambda x.s(2x))(2)$ 界定。

因此， $2f(n)$限定了整个算法中任意预言机输入的大小。一般情况下，预言机调用的次数不超过 $2 + (E(n) - E(n-1) + 1)$，其中第一次调用用于确定 $d'$，其余调用则用于搜索$E(n)$。很容易得出，总调用次数不超过 $3n + E(n)$；我们将其界定为 $2 \cdot 2^{f(n)}$，因为显然 $E(n)$不超过 $d' < 2^{f(n)}$，而$3n \leq 2^{f(n)}$ 很容易通过归纳法证明。

□

## 7.6 从埃及分数展开到左侧最佳逼近

下一个引理说明了如何通过次递归算法将埃及分数展开表示转换为戴德金分割。这里我们将使用有理数的埃及分数展开。一个有理数可能有两种展开，一种有限，另一种无限（例如， $1/9$ 也可以写成 $1/10 + 1/100 + 1/1000 + \ldots$）。我们给出的算法使用有限展开。在下一个引理的证明中，为了统一记号，我们仍将展开写成无限序列；假设缺失的分数均为零，并将其记作 $1/\infty = 0$（例如，$1/9 + 0 + 0 + \ldots$）。

引理 7.16. 设 $x$、$y$ 是区间 $(0,1)$ 中两个不同的实数（可能是有理数）， $E^x$、$E^y$ 分别是它们的埃及分数展开。那么，当且仅当 $E^x$ 在字典序中先于 $E^y$ 时，$x < y$。

证明。假设对于所有 $i < n$， $E^x(i) = E^y(i)$，且 $n$ 是第一个不同之处的位置。那么就第 $n$ 个元素而言，有以下几种情况：要么其中一个序列终止，此时显然另一个序列所表示的数更大；要么两个序列都继续。在后一种情况下，不失一般性，假设 $E^x(n) < E^y(n)$。记 $d = \prod_{j=1}^{n-1} E^x(j) = \prod_{j=1}^{n-1} E^y(j)$。

Then

$$\begin{aligned} y - x &= \sum_{i=n}^{\infty} \left( \prod_{j=1}^{i} E^y(j) \right)^{-1} - \sum_{i=n}^{\infty} \left( \prod_{j=1}^{i} E^x(j) \right)^{-1} \\ &\geq \frac{1}{dE^y(n)} - \sum_{i=n}^{\infty} \left( \prod_{j=1}^{i} E^x(j) \right)^{-1} \\ &\geq \frac{1}{dE^y(n)} - \frac{1}{d} \sum_{i=n}^{\infty} \left( \prod_{j=n}^{i} E^x(j)) \right)^{-1} \\ &\geq \frac{1}{dE^y(n)} - \frac{1}{d} \sum_{i=n}^{\infty} (E^x(n))^{n-i-1} \geq \frac{1}{dE^y(n)} - \frac{1}{d} \cdot \frac{1}{E^x(n) - 1} \geq 0 \, . \end{aligned}$$

由此可得 $x \leq y$，但已知它们不同，因此 $x < y$。 □

由此可得以下算法，用于计算 $D^{\alpha}(q)$ 在 $E^{\alpha}$ 中的值：计算 $q$ 的有限埃及分数展开，并将其与 $E^{\alpha}$ 按字典序比较。计算 $q$ 的展开时，我们使用 [11], 的算法，下面将对此进行回顾。值得注意的是，从该算法可以很容易看出，有理数 $p/q$ 的展开 $E^{p/q}$ 至多包含 $p$ 项，且每个 $E^{p/q}(i)$ 的大小都有上界 $q$。

展开算法：我们定义辅助函数 $Div(q, p)$，对于正整数 $q, p$，该函数返回一对非负整数 $n \leq q$ 和 $r < p$，使得 $np$ 是满足 $np \geq q$ 且 $r = np - q$ 的 $p$ 的最小倍数。给定一个分数 $p/q$，我们通过设定 $r_0 = p$ 和 $(n_{i+1}, r_{i+1}) = Div(q, r_i)$ 来构造序列 $n_1, n_2, \ldots$ 以及 $r_0, r_1, \ldots$，直到达到 $r_j = 0$。埃及分数展开由 $E(i) = n_i$ 给出，其中 $i = 1, \ldots, j$。

引理 7.17。设 $E : \mathbb{N}^+ \longrightarrow \mathbb{N}$是无理数 $\alpha \in (0,1)$ 的埃及分数展开。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^E : \mathbb{Q} \longrightarrow \{0, 1\}$ 是 $\alpha$ 的戴德金分割
- $M^E$ 在输入 $p/q$ 上运行的时间为 $p \cdot \mathrm{poly}(\|q\|, \|E(p)\|)$，并且至多使用 $p$ 次预言机调用，每次调用的输入规模至多为 $1 + \log p$。

证明。根据 [11], , 计算 $p/q$ 的展开最多需要 $p$ 次迭代，且每次迭代都会对不超过 $q$ 的数执行 $O(1)$ 次算术运算。为此，我们比较展开中的第 $i$ 个数 $E^{p/q}(i)$ 与 $E(i)$。回想一下， $E$ 是一个非递减级数，由此可得所述复杂度界。

□

设 $E^\alpha$ 是 $\alpha$ 的埃及分数展开。我们将给出一个算法，使用 $E^\alpha$ 作为预言机，计算 $\alpha$ 的完全左最佳逼近 $L^\alpha$。该算法以递归方式运行。基本情形是平凡逼近0/1。在一般情形下，对于 $n \geq 1$，算法首先计算 $a_{n-1}/b_{n-1} = L^\alpha(n-1)$。然后，算法按如下方式计算 $L^\alpha(n)$：

- 步骤 1：计算埃及分数展开的 $m$ 项，使其满足最小的 $m$

$$\sum_{i=1}^{m} \left( \prod_{j=1}^{i} E^\alpha(j) \right)^{-1} > \frac{a_{n-1}}{b_{n-1}}$$

  并令 $d' = \prod_{i=1}^{m} E^\alpha(i)$。

- 步骤 2：按照上文所述，使用戴德金分割 $D^\alpha$ 的一种实现，搜索最小的 $d > b_{n-1}$，使得对于某个 $c < d$，分数 $c/d$ 是对 $\alpha$ 比 $a_{n-1}/b_{n-1}$ 更好的左逼近。选取最大的此类 $c$。返回 $c/d$，作为 $L^\alpha(n)$。注意，$d$ 至多为 $d'$。

步骤 1 看起来可能需要无界搜索。但事实上，由于 $a_{n-1}/b_{n-1}$ 的埃及分数展开至多包含 $a_{n-1}$ 项，根据引理 7.16，我们知道 $m \leq a_{n-1} + 1$。

引理 7.18。令 $E : \mathbb{N}^+ \longrightarrow \mathbb{N}$ 为无理数 $\alpha \in (0,1)$ 的埃及分数展开。假设 $E$ 可在时间界 $s$ 内计算，并令 $f(n) = (\lambda x.2^{x \log x + s(\|x\|)})^{(n)}(1)$。存在一个参数化的函数预言机图灵机 $M$，使得

- $\Phi_M^E : \mathbb{N} \longrightarrow \mathbb{Q}$ 是 $\alpha$的完全左最佳逼近
- $M^E$在输入 $n$上运行的时间为多项式$(f(n))$，并且至多使用 $f(n)$次预言机调用，每次调用的输入规模至多为 $\|f(n)\|$。

证明。我们将看到，在分析该算法时，关键量是$b_n$。我们声称 $b_n \leq f(n)$，这可由以下内容通过归纳法得出：

$$b_n \leq 2^{b_{n-1} \log b_{n-1} + s(\|b_{n-1}\|)} .$$

为证明这一点，请注意，

$$b_n \leq d' \leq \prod_{i=1}^{m} E(i) . \tag{7.9}$$

现在 $m \leq a_{n-1} + 1 \leq b_{n-1}$，正如引理之前所论证的。此外，前 $m-1$ 项 $E(1), \ldots, E(m-1)$ 与 $a_{n-1}/b_{n-1}$ 的埃及分数展开相同，这意味着它们的大小至多为 $b_{n-1}$，因此它们的乘积至多为 $b_{n-1}^{b_{n-1}}$。为了给 $E(m)$ 作上界，我们回想一下，$E(m)$ 可在时间 $s(\|m\|)$ 内计算，因此有界于 $2^{s(\|m\|)} \leq 2^{s(b_{n-1})}$。由 (7.9) 可得

$$b_n \leq 2^{b_{n-1} \log b_{n-1} + s(\|b_{n-1}\|)}$$

这就完成了归纳法。

现在，为了界定预言机输入规模，我们声称，整个算法中传递给预言机的最大值由 $b_n$ 界定；这很容易验证。预言机调用次数的上界也随之得出，因为图灵机可以记录预言机的回答，因此我们可以避免对同一输入重复查询。

为了界定执行时间，我们界定最后一次迭代所需的时间；总时间关于这一数量是多项式的，因为最后一次迭代的上界最大。在最后一次迭代中，我们使用$a_{n-1}/b_{n-1}$ 的埃及分数展开（由前一次迭代得知），并将其与级数 $E(i)$按字典序比较，以确定 $m$；这需要 $O(m) = O(b_{n-1})$次算术运算，且运算所涉及的数均不超过 $b_n$。我们已经证明了 $b_n \leq f(n)$，而计算所需的时间是其位长的多项式倍，再乘以 $O(b_{n-1})$。因此，步骤 1 所需的时间为 $\mathrm{poly}(f(n-1) \log f(n)) = \mathrm{poly}(f(n))$。

步骤 2 的实现如下：对于 $d = b_{n-1}+1, b_{n-1}+2, \ldots$，我们计算$c = \lceil (a_{n-1}d)/b_{n-1} \rceil$，因为这给出了满足 $c/d > a_{n-1}/b_{n-1}$ 的最小分子。我们使用戴德金分割来检验 $c/d < \alpha$。整个过程所需时间为 $\mathrm{poly}(f(n))$，因为 $d \leq f(n)$。找到第一个这样的 $c, d$ 后，我们搜索满足 $c/d$ 仍小于 $\alpha$ 的最大 $c$。这涉及不到 $d$ 次戴德金分割的应用；同样，所需时间仍在 $\mathrm{poly}(f(n))$ 之内。 □

## 7.7 总结。

现在，我们将按照此前总结第 4 节（第 25 页）和第 6 节（第 40 页）时采用的思路，对本节作一番总结。

定理 7.19。考虑以下表示：(1) 右最佳逼近，(2) 完全右最佳逼近，(3) 贝尔序列，以及 (4) 一般的上方和逼近。令 $R_1$ 和 $R_2$ 为这四种表示中的任意两种。那么，对于任意时间界 $t$，都存在一个时间界 $s$，它关于 $t$ 是原始递归的，使得 $O(t)_{R_2} \subseteq O(s)_{R_1}$。

谨向熟悉格热戈尔奇克层级的读者说明：我们的结果比上述定理所断言的略强。很容易验证，只要时间界 $t$ 属于格热戈尔奇克类 $\mathcal{E}_i$，那么对于任意 $i \geq 3$，时间界 $s$都属于格热戈尔奇克类 $\mathcal{E}_{i+1}$。如果省略埃及分数展开表示，下一个定理也同样如此。

定理 7.20。考虑以下表示：(1) 左最佳逼近，(2) 完全左最佳逼近，(3) 对偶贝尔序列，(4) 从下方的一般和逼近，以及 (5) 埃及分数展开；令 $R_1$ 和 $R_2$ 为这五种表示中的任意两种。那么，对于任意时间界 $t$，存在一个关于 $t$ 原始递归的时间界 $s$，使得 $O(t)_{R_2} \subseteq O(s)_{R_1}$。

# 8 略谈表示度

回顾关系 $\preceq_S$ 的定义（定义 1.3），并回顾关系 $\equiv_S$ 和 $\prec_S$ 的定义（定义 1.4）。此外，回顾一下，等价关系 $\equiv_S$ 在这些表示上诱导出一个度结构，其中一个度仅仅是一个 $\equiv_S$-等价类，参见第 10 页。我们将采用度理论的标准术语：如果有 $(R_1 \prec_S R_2)$ $R_1 \preceq_S R_2$，其中 $R_1 \in$a 且 $R_2 \in$b，则称度 a 严格低于度 b；此外，如果有 $(R_1 \prec_S R_2)$ $R_1 \preceq_S R_2$，其中 $R_1 \in$a 且 $R_2 \in$b，则称度 b 严格高于度 a。请注意，图 1 展示的是一个上下颠倒的世界图景，也就是说，如果度 a 低于度 b，那么在图中 a 会被描绘在 b 的上方。

图 1 显示，由左最佳逼近表示的度与由右最佳逼近表示的度不可比较，也就是说，前者既不高于也不低于后者。我们可以通过增长论证（见第 7 页）证明这一点。下面来看看这种论证如何进行。

在第 7 节中，我们看到，对偶贝尔序列表示与由左最佳逼近表示子递归等价，而标准贝尔序列表示与由右最佳逼近表示子递归等价。为了使增长论证更加清晰，我们将用对偶贝尔序列和标准贝尔序列来代替左最佳逼近和右最佳逼近。令 $\mathcal{B}$ 和 $d\mathcal{B}$ 分别表示由标准贝尔序列和对偶贝尔序列给出的表示。我们将论证 $\mathcal{B} \not\preceq_S d\mathcal{B}$ 和 $d\mathcal{B} \not\preceq_S \mathcal{B}$。

令 $s$ 为任意时间界。我们将证明，存在一个区间 (0, 1) 内的无理数 $\alpha$，使得

$$\text{the standard Baire sequence of } \alpha \text{ is not computable in time } O(s) \tag{8.1}$$

但仍然

$$\text{the dual Baire sequence of } \alpha \text{ is computable in time } \mathrm{poly}(2^n). \tag{8.2}$$

其中 $n$ 是输入的长度。

考虑一个增长非常快且图像简单的递增函数 $B : \mathbb{N} \longrightarrow \mathbb{N}$。具体而言，我们假设 $B$的图像，也就是关系 $B(x) = y$，可在 poly(max($x, y$)) 时间内判定，但 $B$ 仍增长得太快，无法在 $O(s)$ 时间内计算。这样的 $B$ 总是存在；为方便起见，我们还将选择 $B$，使得 $2^x \leq B(x)$ 且 $B(x) \leq B(x+1)$。现在，$B$ 是某个无理数 $\alpha$的标准贝尔序列；由于一个无理数只有一个标准贝尔序列，$\alpha$ 的标准贝尔序列无法在 $O(s)$ 时间内计算。因此，(8.1)成立。剩下要证明的是(8.2)成立。

令 $a_n = B(0) + (\sum_{i=1}^{n} B(i) + 1)$。若对某个 $n$，$A(x) = 1$ if $x = a_n$；否则，令 $A(x) = 0$。这定义了一个函数 $A$。我们将证明 $A$ 是 $\alpha$ 的对偶贝尔序列，但首先我们将说明 $A(x)$ 是可计算的，其运行时间为

poly($2^{\|x\|}$): 注意，只要 $x = a_n$ 成立， $n$ 就会小于 $x$。因此，我们可以通过检查来判断是否存在满足 $x = a_n$ 的 $n$

$$(\exists n, y_0, \ldots, y_n < x)[\ B(0) = y_0 \ \wedge \ \ldots \ \wedge \ B(n) = y_n \\ \wedge \ x = y_0 + (\sum_{i=1}^{n} y_i + 1)\ ] \quad (8.3)$$

根据假设，我们可以在 poly(max($z_1, z_2$)) 时间内判定关系 $B(z_1) = z_2$是否成立。因此，我们可以在 poly($x$) 时间内检查 (8.3) 是否成立。这表明 $A(x)$ 可在 poly($x$) 时间内计算，因此也可在 poly($2^{\|x\|}$) 时间内计算，因为当 (8.3) 成立时， $A(x)$ 等于 1；当 (8.3} 不成立时， $A(x)$ 等于 0。

对于任意自然数 $n$，我们定义字符串 $\sigma_n$ and $\tau_n$ 如下

$$\sigma_n = 0^{A(0)}10^{A(1)}1\ldots0^{A(a_n-1)}10^{A(a_n)} \text{ and } \tau_n = 1^{B(0)}01^{B(1)}0\ldots1^{B(n)}0\ .$$

我们对 $n$ 进行归纳法，证明 $\sigma_n = \tau_n$。设 $n = 0$。我们有 $a_0 = B(0)$，因此根据 $A$ 的定义，有

$$\sigma_0 \ = \ 0^{A(0)}10^{A(1)}1\ldots0^{A(a_0-1)}10^{A(a_0)} \ = \ 1^{a_0}0 \ = \ 1^{B(0)}0 \ = \ \tau_0\ .$$

设 $n > 0$。根据 $a_n$ 的定义，我们有 $a_n = a_{n-1} + B(n) + 1$，因此$B(n) = a_n - (a_{n-1} + 1)$。此外，我们有

$$\sigma_n \overset{(1)}{=} \sigma_{n-1}10^{A(a_{n-1}+1)}10^{A(a_{n-1}+2)}\ldots10^{A(a_n-1)}10^{A(a_n)} \overset{(2)}{=} \\ \sigma_{n-1}1^{a_n-(a_{n-1}+1)}0 \overset{(3)}{=} \sigma_{n-1}1^{B(n)}0 \overset{(4)}{=} \tau_{n-1}1^{B(n)}0 \overset{(5)}{=} \tau_n$$

其中，(1) 根据 $\sigma_n$ 的定义成立；(2) 根据 $A$ 的定义成立；(3) 根据 $a_n$ 的定义成立；(4) 根据归纳假设成立；(5) 根据 $\tau_n$ 的定义成立。这证明了对于任意 $n$, $\sigma_n = \tau_n$ 成立；并且根据引理 7.12，我们有

$$J_A^{a_n} \ = \ \mathcal{T}_{\mathrm{F}}(\sigma_n) \ = \ \mathcal{T}_{\mathrm{F}}(\tau_n) \ = \ I_B^n$$

对于任意 $n$。根据标准与对偶贝尔序列的定义（定义 7.7），可知 $A$ 是 $\alpha$ 的对偶贝尔序列。由此完成了对（8.2）的证明。

由（8.1）和（8.2）可得

$$\mathrm{poly}(2^n)_{d\mathcal{B}} \ \not\subseteq \ O(s)_{\mathcal{B}}$$

对于任意时间界 $s$，因此根据我们对序关系 $\preceq_S$ 的定义（定义 1.3），有 $\mathcal{B} \not\preceq_S d\mathcal{B}$。对称的证明可得 $d\mathcal{B} \not\preceq_S \mathcal{B}$。

我们的增长论证表明，由左最佳逼近表示的度（也就是 $d\mathcal{B}$ 的度）与由右最佳逼近表示的度（也就是 $\mathcal{B}$ 的度）不可比较。我们已经看到，如果能够访问 $\alpha \in$ (0,1) 的左最佳逼近或右最佳逼近，就可以子递归计算其戴德金分割

$\alpha$ （引理 7.5）。因此，我们由最佳逼近表示所对应的两个度都高于由戴德金分割表示所对应的度；又由于这两个度不可比较，它们必定严格高于该度。我们无法将由戴德金分割表示子递归地转换为由左最佳逼近表示或由右最佳逼近表示。下一节将看到，这两个度都低于由连分数表示所对应的度；又由于这两个度不可比较，我们可以断定它们都严格低于该度。我们无法将由左最佳逼近表示或由右最佳逼近表示子递归地转换为由连分数表示（但我们将在下一节看到，如果能够同时访问 $\alpha$ 的左最佳逼近和右最佳逼近，我们确实可以子递归计算 $\alpha$ 的连分数）。见图 1。

# 9 等价于连分数的表示

## 9.1 连分数。

我们可以假定读者对连分数已有一定了解，但下面仍将陈述并解释连分数的一些性质。关于连分数的更多内容，请参阅 Khintchine [19] 或 Richards [29]。后者是一篇非常易读的论文，仔细解释了连分数与法里对之间的关系。

设 $a_0, a_1, a_2, \ldots$ 是一个由整数构成的无限序列，其中 $a_1, a_2, a_3 \ldots$ 为正整数。连分数 $[a_0;\ a_1, a_2, \ldots]$ 定义为

$$[\ a_0; a_1, a_2, a_3 \ldots\ ] \ = \ a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \ldots}}}$$

我们将研究 0 与 1 之间的无理数的连分数表示。区间 (0, 1) 中的每个无理数 $\alpha$ 都可以唯一地写成如下形式： $\alpha = [0;\ a_1, a_2, \ldots]$，其中 $a_1, a_2, a_3, \ldots$ 为正整数。因此，下面的定义是有意义的。

定义 9.1. 设 $\alpha$ 是区间 (0, 1) 内的一个无理数，并令 $\alpha = [0;\ a_1、\ a_2、\ \ldots]$。$\alpha$ 的连分数是函数 $C\colon \mathbb{N}^+ \longrightarrow \mathbb{N}$，由 $C(i) = a_i$ 给出。

□

我们对 $n$ 进行归纳，定义 $[a_0;\ a_1,..., a_n]$。若 $n = 0$，令 $[a_0;\ \ ] = a_0$。若 $n > 0$，则令

$$[\ a_0; a_1, \ldots, a_n\ ] \ = \ a_0 \ + \ \frac{1}{[\ a_1; a_2, \ldots, a_n\ ]}\ .$$

有理数 $[a_0;\ a_1, \ldots, a_n]$ 称为无限连分数 $[a_0;\ a_1, a_2, \ldots]$ 的第 $n$ 个渐近分数。

Let

$$\frac{p_0}{q_0} = \frac{0}{1}\,, \quad \frac{p_1}{q_1} = \frac{1}{a_1} \quad \text{and} \quad \frac{p_{k+2}}{q_{k+2}} = \frac{p_k + a_{k+2}p_{k+1}}{q_k + a_{k+2}q_{k+1}}\,. \tag{9.1}$$

众所周知，$p_n/q_n$ 等于 $[0;\, a_1, a_2, a_3, \ldots]$ 的第 $n$ 个渐近分数，即 $p_n/q_n = [0;\, a_1, \ldots, a_n]$。众所周知，

$$\frac{p_0}{q_0} < \frac{p_2}{q_2} \;<\; \frac{p_4}{q_4} \;<\; \ldots \;<\; \alpha \;< \ldots \;<\; \frac{p_5}{q_5} < \;\frac{p_3}{q_3} \;<\; \frac{p_1}{q_1}$$

其中 $\alpha = [0;\; a_1, a_2, \ldots]$，也就是说，偶数项渐近分数从下方趋近于连分数所表示的数，而奇数项渐近分数则从上方趋近于它。每个渐近分数都是一个（左侧或右侧）最佳逼近，但反过来并不成立：最佳逼近不一定是渐近分数。下一个定理将连分数的渐近分数与法雷对树 $\mathcal{T}_{\mathrm{F}}$ 联系起来。

定理 9.2（Hurwitz [18]）。设 $[0;\; a_1, a_2, \ldots]$ 是一个连分数，并令渐近分数 $p_n/q_n$ 由上面的 (9.1) 定义。我们有

$$\mathcal{T}_{\mathrm{F}}(\; 0^{a_1}1^{a_2}\ldots 1^{a_{n-1}}0^{a_n}\;) \;=\; \left(\; \frac{p_{n-1}}{q_{n-1}}, \frac{p_n}{q_n} \;\right) \;=\; [0; a_1, \ldots, a_n]$$

当 $n$ 为奇数时，我们有

$$\mathcal{T}_{\mathrm{F}}(\; 0^{a_1}1^{a_2}\ldots 0^{a_{n-1}}1^{a_n}\;) \;=\; \left(\; \frac{p_n}{q_n}, \frac{p_{n-1}}{q_{n-1}} \;\right) \;=\; [0; a_1, \ldots, a_n]$$

当 $n$ 为偶数时。

## 9.2 定义。

现在我们将定义一些表示，它们最终被证明与连分数表示子递归等价。

定义 9.3。如果函数 $T\colon\; [0,1]\cap\mathbb{Q} \longrightarrow \mathbb{Q}$ 是无理数 $\alpha$ 的迹函数，那么对于每个 $r \in [0,1]\cap\mathbb{Q}$，都有 $|\alpha - r| > |\alpha - T(r)|$ □

如果迹函数 $T$ 使 $r$ 向右（向左）移动，即 $r < T(r)$（$T(r) < r$），我们就称其向右（向左）移动。要认识到迹函数确实定义了唯一的实数，最简单的方法可能是注意到，对于 $\alpha$，迹函数 $T$ 给出了 $\alpha$的戴德金分割：如果 $T$ 向右移动，那么我们知道 $r$ 低于$\alpha$；如果 $T$向左移动，那么我们知道 $r$ 高于 $\alpha$。显然，$T$ 不可能给出任何其他数的戴德金分割，只能给出 $\alpha$的戴德金分割。

直观地说，压缩映射是一个使两个（有理）数彼此靠近的函数。我们将看到，压缩映射也可以用来表示无理数。

定义 9.4。函数 $F$：$[0,1]\cap\mathbb{Q}\longrightarrow\mathbb{Q}$ 是压缩映射，如果满足$F(r)\neq r$且 $|F(r_1)-F(r_2)|<|r_1-r_2|$对任意有理数 $r$、$r_1$、$r_2$，其中 $r_1\neq r_2$。□

引理 9.5. 任意压缩映射都是某个无理数的迹函数。

证明. 设 $F$ 是一个压缩映射。如果 $F$ 将 $r$ 向右（向左）移动，那么 $F$ 也会将任何小于（大于） $r$ 的有理数向右（向左）移动；否则 $F$ 就不是压缩映射。我们定义两个有理数序列 $r_0$、$r_1$、$r_2\ldots$ 和 $s_0$、$s_1$、$s_2\ldots$。设 $r_0=0$ 和 $s_0=1$。如果 $F$ 将 $r_i+s_i$（/2）向右移动，就令 $r_{i+1}=(r_i+s_i)/2$；否则，令 $r_{i+1}=r_i$。如果 $F$ 将 $r_i+s_i$（/2）向左移动，就令 $s_{i+1}=(r_i+s_i)/2$；否则，令 $s_{i+1}=s_i$（定义 9.4 要求压缩映射移动任意有理数）。显然，我们有 $\lim_i r_i=\lim_i s_i$，并且显然，此极限是一个无理数 $\alpha$。容易看出，$F$ 是 $\alpha$ 的迹函数。□

前面的引理表明，接下来的定义是有意义的。

定义 9.6。若 $F$ 是 $\alpha$ 的迹函数，则 contractor $F$ 是无理数 $\alpha$ 的压缩映射。□

迹函数表示法由 Kristiansen [23], 提出，而压缩映射表示法由 Kristiansen [25] 提出。

## 9.3 从完全最佳逼近到连分数。

根据定理 9.2，(0, 1) 中 $\alpha\in$ 的连分数 $[0;x_1,x_2,\ldots]$ 可以视为法雷对树 $\mathcal{T}_{\mathrm{F}}$ 中的无限路径 $0^{x_1}1^{x_2}0^{x_3}1^{x_4}\ldots$。根据引理 7.3，如果我们能够获得 $\alpha$ 的完全左最佳逼近和完全右最佳逼近，就可以构造路径 $0^{x_1}1^{x_2}0^{x_3}1^{x_4}\ldots$。这一思路得出了一个将完全最佳逼近转换为连分数的算法。该算法的伪代码见图2，它将 $\alpha$ 的完全左最佳逼近（记作 $\{a_i/b_i\}_{i\in\mathbb{N}}$）和完全右最佳逼近（记作 $\{c_i/d_i\}_{i\in\mathbb{N}}$）作为预言机。该算法输出 $\alpha$ 的连分数 $[0;x_1,x_2,\ldots]$ 的第 $n$ 个元素 $x_n$（输入为 $n$）。用 (* . . . *) 括起来的注释解释了算法的工作原理，而 $m(a/b,c/d)$ 表示分数 $a/b$ 和 $c/d$ 的中项，即 $m(a/b,c/d)=(a+c)/(b+d)$。

图2中的第一个 while循环统计路径$0^{x_1}1^{x_2}0^{x_3}1^{x_4}\ldots$中出现的连续零的数量，该路径由最佳逼近给出；第二个 while循环则统计连续一的数量。事实证明，实际计数是多余的，这两个循环可以删去：我们可以根据循环开始时已有的值，直接计算 while循环将执行多少次。下面来看第一个 while循环：

$$\texttt{WHILE}\ \ m\left(\frac{a_\ell}{b_\ell},\frac{c_r}{d_r}\right)=\frac{c_{r+1}}{d_{r+1}}\ \ \texttt{DO BEGIN}\ \ r:=r+1;\ x_i:=x_i+1\ \ \texttt{END}$$

$\ell := 0$; $r := 0$; $i := 1$; $x_i := 1$; (* 我们有 $a_\ell/b_\ell = 0/1$ 和 $c_r/d_r = 1/1$ *) 更多: (* $\alpha$ 位于 $(a_\ell/b_\ell, c_r/d_r)$ 内 *) WHILE $m\left(\frac{a_\ell}{b_\ell}, \frac{c_r}{d_r}\right) = \frac{c_{r+1}}{d_{r+1}}$ DO BEGIN (* $\alpha$ 位于 $(a_\ell/b_\ell, c_{r+1}/d_{r+1})$ 内，且路径向左分支 *) $r := r + 1$; $x_i := x_i + 1$ END (* 我们已确定 $x_i$，其中 $i$ 为奇数 *) IF $i = n$ THEN ⟨输出 $x_i$ 并停止⟩ (* 由于循环已终止，$\alpha$ 位于 $(a_{\ell+1}/b_{\ell+1}, c_r/d_r)$ 内 *) $i := i + 1$; $x_i := 1$; $\ell := \ell + 1$; (* $\alpha$ 位于 $(a_\ell/b_\ell, c_r/d_r)$ 内 *) WHILE $m\left(\frac{a_\ell}{b_\ell}, \frac{c_r}{d_r}\right) = \frac{a_{\ell+1}}{b_{\ell+1}}$ DO BEGIN (* $\alpha$ 位于 $(a_{\ell+1}/b_{\ell+1}, c_r/d_r)$ 内，且路径向右分支 *) $\ell := \ell + 1$; $x_i := x_i + 1$ END (* 我们已确定 $x_i$，其中 $i$ 为偶数 *) IF $i = n$ THEN ⟨输出 $x_i$ 并停止⟩ (* 由于循环已终止，$\alpha$ 位于 $(a_\ell/b_\ell, c_{r+1}/d_{r+1})$ 内 *) $i := i + 1$; $x_i := 1$; $r := r + 1$; (* $\alpha$ 位于 $(a_\ell/b_\ell, c_r/d_r)$ 内 *) GOTO 更多

图2：算法

在执行此循环期间，除了 $r$ 和 $x_i$ 之外，没有其他变量会被修改。设

- $S$ 为循环开始执行时 $r$ 的值
- $T$ 为循环终止执行时 $r$ 的值
- $X$ 为循环终止执行时 $x_i$ 的值（注意，执行开始时 $x_i$ 为 1，因此 $X = 1 + T - S$）。

对于 $r = S, \ldots, T-1$，我们有

$$\frac{c_{r+1}}{d_{r+1}} \;=\; m\left(\frac{a_\ell}{b_\ell}, \frac{c_r}{d_r}\right) \;=\; \frac{a_\ell + c_r}{b_\ell + d_r}\,.$$

因此，循环终止时，我们有

$$\frac{c_T}{d_T} \;=\; \frac{a_\ell(X-1) + c_S}{b_\ell(X-1) + d_S}\,. \tag{9.2}$$

循环终止时，我们还有

$$\frac{c_{T+1}}{d_{T+1}} \;\neq\; m\left(\frac{a_\ell}{b_\ell}, \frac{c_T}{d_T}\right)$$

但是，由于中项 $m(a_\ell/b_\ell, c_T/d_T)$ 并不是右侧最佳逼近数完备列表中的下一个分数，因此它将是左最佳逼近数完备列表中的第 $\ell+1$ 个分数。也就是说，我们有：

$$\frac{a_{\ell+1}}{b_{\ell+1}} \;=\; m\left(\frac{a_\ell}{b_\ell}, \frac{c_T}{d_T}\right) \;=\; \frac{a_\ell + c_T}{b_\ell + d_T}\,. \tag{9.3}$$

由 (9.2) 和 (9.3)，我们有

$$\frac{a_{\ell+1}}{b_{\ell+1}} \;=\; \frac{a_\ell + a_\ell(X-1) + c_S}{b_\ell + b_\ell(X-1) + d_S}\,. \tag{9.4}$$

现在，(9.4) 得出等式

$$X = \frac{b_{\ell+1} - d_S}{b_\ell}$$

其中，$d_S$ 是循环开始执行时 $d_r$ 的值，而 $X$ 是循环终止执行时 $x_i$ 的值。因此，上述循环可以替换为

$$x_i := (b_{\ell+1} - d_r)/b_\ell;\; r := r + (x_i - 1)$$

（回想一下，循环开始时 $x_i$ 为 1，因此在循环终止前，$r$ 恰好会递增 $x_i - 1$ 次）。通过对称论证可知，该循环

$$\texttt{WHILE}\;\; m\left(\frac{a_\ell}{b_\ell}, \frac{c_r}{d_r}\right) = \frac{a_{\ell+1}}{b_{\ell+1}}\;\; \texttt{DO BEGIN}\;\; \ell := \ell + 1;\; x_i := x_i + 1\;\; \texttt{END}$$

```
1        ℓ := 0; r := 0; i := 1;
2  more: x_i := (b_{ℓ+1} − d_r)/b_ℓ; r := r + (x_i − 1);
3        IF i = n THEN ⟨give output x_i and halt⟩
4        i := i + 1; ℓ := ℓ + 1;
5        x_i := (d_{r+1} − b_ℓ)/d_r; ℓ := ℓ + (x_i − 1);
6        IF i = n THEN ⟨give output x_i and halt⟩
7        i := i + 1; r := r + 1;
8        GOTO more
```

图 3：算法

可以替换为以下程序

$$x_i := (d_{r+1} - b_\ell)/d_r;\ \ell := \ell + (x_i - 1).$$

因此，我们得到图 3 中的算法。

引理 9.7。设 $L\colon\ \mathbb{N} \longrightarrow \mathbb{Q}$ 和 $R\colon\ \mathbb{N} \longrightarrow \mathbb{Q}$ 分别是无理数 $\alpha \in \in(0,1)$ 的完备左、右最佳逼近。假设 $L$ 和 $R$ 都能在时间界 $s$ 内计算，并且令 $f(n) = \lambda x.(s(1+x))^{(n)}(0)$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi^{L,\ R}_{M}\colon\ \mathbb{N}^+ \longrightarrow \mathbb{N}$ 是 $\alpha$ 的连分数
- $M^{L,\ R}$ 在输入 $n$ 时，运行时间为 $\mathrm{poly}(f(n))$，并且恰好进行 $2n$ 次预言机调用，每次调用的输入规模至多为 $f(n-1)$。

证明。图灵机 $M$ 对图 3 中由 goto 语句实现的循环体进行迭代，次数不超过 $\lceil n/2 \rceil$。为实现第 2 行，它向预言机询问 $a_\ell/b_\ell$ 和 $a_{\ell+1}/b_{\ell+1}$ 的值。随后，$M$ 可以执行赋值操作，因为 $d_r$ 的值已为 $M$ 所知，并将存储在工作带上（除了首次到达第 2 行时，此时按约定有 $r = 0$ 和 $d_r = 1$）。类似地，为实现第 5 行，它向预言机 $R$ 询问 $c_r/d_r$ 和 $c_{r+1}/d_{r+1}$。$b_\ell$ 的值已为 $M$ 所知。因此，每次迭代循环体时，$M$ 会分别咨询每个预言机四次；当 $M$ 输出 $x_n$ 并停机时，恰好已进行了 $2n$ 次预言机调用。

执行算术运算所需的时间是关于所涉及数值的大小（即其表示）的多项式。我们通过归纳法证明整个算法过程中 $\ell, r$ 的位长上界。归纳命题由以下两个陈述合取而成：(1) 每当执行到第 6 行时，我们有

$$\ell \ \le\ d_{r+1} \quad \text{and} \quad \|\ell\| \ \le\ s(1 + f(i-1)) \tag{9.5}$$

并且 (2) 每当执行到第 3 行时，我们有

$$r \;\le\; b_{\ell+1} \quad \text{and} \quad \|r\| \;\le\; s(1+f(i-1))\,. \tag{9.6}$$

请注意，接下来将证明这些界限，从而完成对引理中时间复杂度和预言机输入规模的论证。

考虑第一次执行循环体。此时我们到达第 3 行，且

$$r = (b_1 - d_0)/b_0 - 1 = (b_1 - 1)/1 - 1 = b_1 - 2 \le b_1$$

到达第 6 行时则有 $\ell = (d_1 - b_1)/d_0 = d_1 - b_1 \le d_1$。不难验证，(9.5) 和 (9.6) 均成立。

一般而言，假设我们第 $k$ 次到达第 6 行。现在$i = 2k$。继续执行至第 2 行时，我们将 $i$ 和 $r$ 加 1。第 2 行计算出的 $x_i$ 的值满足

$$x_i \;=\; (b_{\ell+1} - d_r)/b_\ell \;\le\; b_{\ell+1} - b_\ell,$$

其中，最后一个不等式利用了这样一个事实：对于所有满足 $x > y > 0$ 的整数，$(x-1)/y \le x-y$ 均成立。我们将第 2 行之后 $r$ 的新值与上次到达第 3 行时它的值 $r'$ 进行比较。我们有

$$r \;=\; r' + x_i \le b_{\ell'+1} + b_{\ell+1} - b_\ell \;\le\; b_{\ell+1}$$

其中，$\ell'$ 是上次到达第 3 行时 $\ell$ 的值（显然 $\ell' \le \ell - 1$）。

这证明了 $r \le b_{\ell+1}$，因此到达第 3 行时，(9.6) 的第一个合取项成立。为验证 (9.6) 的第二个合取项也成立，注意到 $b_{\ell+1}$ 是通过一次预言机查询得到的，因此我们有 $\|b_{\ell+1}\| \le s(\|\ell+1\|)$。根据归纳假设 (9.5)，到达第 3 行时，我们有 $\|\ell\| \le s(1 + f(i-2))$。因此，由于 $r \le b_{\ell+1}$，我们有

$$\|r\| \;\le\; \|b_{\ell+1}\| \;\le\; s(\|\ell+1\|) \;\le\; s(1+\|\ell\|) \;\le\; s(1+s(1+f(i-2))) \;=\; s(1+f(i-1))$$

（最后一个等式根据 $f$ 的定义成立）。对称论证可证明归纳情形下的 (9.5)。

□

## 9.4 从连分数到完备最佳逼近。

将连分数转换为完备的左（或右）最佳逼近的算法相当简单：令 $[0;\ x_1, x_2, \cdots]$ 是 $\alpha$ 的连分数。根据定理 9.2，$0^{x_1}1^{x_2}0^{x_3}\cdots$ 是法里对树 $\mathcal{T}_{\mathrm{F}}$ 中 $\alpha$ 的唯一一条路径。根据引理 7.3，$\alpha$ 的完备左、右逼近可以依次从路径 $0^{x_1}1^{x_2}0^{x_3}\cdots$ 上的法里对中读出。

每当路径中出现 1（向右分支）时，就会出现一个新的左最佳逼近；每当路径中出现 0（向左分支）时，就会出现一个新的右最佳逼近数。

引理 9.8. 设 $C:\ \mathbb{N}^+ \longrightarrow \mathbb{N}$ 是无理数 $\alpha \in (0,1)$ 的连分数。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^C:\ \mathbb{N} \longrightarrow \mathbb{Q}$ 是 $\alpha$ 的完全左最佳逼近
- $M^C$ 在输入 $n$ 上的运行时间为 $\mathrm{poly}(\sum_{j=1}^{2n} C(j))$，且至多进行 $2n$ 次预言机调用；每次调用的输入规模至多为 $3+\log n$，输出规模至多为 $\max_{j=1}^{2n}\{\log\ C(j)\}$。

证明。设 $\underline{\sigma}$ 表示路径 $0^{C(1)}1^{C(2)}0^{C(3)}\cdots$。要得到完全左最佳逼近的第 $n$ 个元素，只需找到索引 $i_n$，使得 $\underline{\sigma}$ 中第 $n$ 次出现 1 的位置为索引 $i_n$。随后，$M$ 计算位置 $\underline{\sigma}_1\cdots\underline{\sigma}_{i_n}$ 处的法里对，并直接返回该法里对的左分量。

注意，由于 $C(1)\ C(2)\ \ldots \geq 1$，我们有 $i_n \leq \sum_{j=1}^{2n} C(j)$。因此，$M$最多需要进行 $2n$ 次预言机调用，输入规模最多为 $1+\lfloor\log 2n\rfloor \leq 3+\log n$（因为输入需要表示大小至多为 $2n$ 的数），输出规模最多为 $\max_{j=1}^{2n}\{\log C(j)\}$。找到 $i_n$ 后，$M$ 确定位置 $\underline{\sigma}_1\cdots\underline{\sigma}_{i_n}$ 处的法里对。根据命题 2.4，这可以在 $\mathrm{poly}(i_n) \leq \mathrm{poly}(\sum_{j=1}^{2n} C(j))$ 时间内完成。现在很容易看出，计算预言机调用的值、读取结果以及计算相关法里对所需的总时间，上界为 $\mathrm{poly}(\sum_{j=1}^{2n} C(j))$。

□

## 9.5 压缩映射、迹函数与完备最佳逼近之间的转换。

引理 9.9. (i) 设 $\{r_i\}_{i\in\mathbb{N}}$ 是 $\alpha$ 的完全左最佳逼近。对于任意 $i \in \mathbb{N}$，有

$$r_{i+1} - r_i \ > \ r_{i+2} - r_{i+1}\ .$$

(ii) 设 $\{r_i\}_{i\in\mathbb{N}}$ 是 $\alpha$ 的完全右最佳逼近。对于任意 $i \in \mathbb{N}$，有

$$r_i - r_{i+1} \ > \ r_{i+1} - r_{i+2}\ .$$

证明. 我们证明 (ii)；(i) 的证明类似。根据引理 7.3，$r_i$ 将是树 $\mathcal{T}_{\mathrm{F}}$ 中某个区间的右端点（不失一般性，我们可以假设$r_i > 0/1$）。因此，存在 $\sigma \in \{0,1\}^*$ 以及分数 $a/b$ 和 $c/d$，使得$\mathcal{T}_{\mathrm{F}}(\sigma 0) = (a/b, c/d)$，并且 $r_i = c/d$。

根据命题 2.2 和引理 7.3，存在 $k$ 使得

$$\mathcal{T}_{\mathrm{F}}(\sigma 01^k 0) = \left(\ \frac{a+kc}{b+kd}\ ,\ \frac{a+kc+c}{b+kd+d}\ \right) \quad \text{and} \quad r_{i+1} = \frac{a+kc+c}{b+kd+d}\ . \tag{9.7}$$

设 $\mathbf{a} = a + kc$，设 $\mathbf{b} = b + kd$。现在我们可以将 (9.7) 重写为

$$\mathcal{T}_{\mathrm{F}}(\sigma 01^k 0) = \left( \frac{\mathbf{a}}{\mathbf{b}} \, , \, \frac{\mathbf{a}+c}{\mathbf{b}+d} \right) \quad \text{and} \quad r_{i+1} = \frac{\mathbf{a}+c}{\mathbf{b}+d} \, . \tag{9.8}$$

由 (9.8) 和 $\mathcal{T}_{\mathrm{F}}$的定义可得

$$\mathcal{T}_{\mathrm{F}}(\sigma 01^k) = \left( \frac{\mathbf{a}}{\mathbf{b}} \, , \, \frac{c}{d} \right) \quad \text{and} \quad \mathcal{T}_{\mathrm{F}}(\sigma 01^k 1) = \left( \frac{\mathbf{a}+c}{\mathbf{b}+d} \, , \, \frac{c}{d} \right) \, .$$

这表明 $((\mathbf{a} + c)/(\mathbf{b} + d)\ c/d)$ 是 $\mathcal{T}_{\mathrm{F}}$ 中的一个区间。因此，由定理 2.2 可得

$$\frac{c}{d} - \frac{\mathbf{a}+c}{\mathbf{b}+d} = \frac{1}{d(\mathbf{b}+d)} \, . \tag{9.9}$$

由命题 2.2 和命题 2.6，存在 $m$，使得 $r_{i+2}$ 是区间 $\mathcal{T}_{\mathrm{F}}(\sigma 1^k 01^m 0)$ 的右端点。我们可以假设 $m = 0$，因为 $m = 0$ 会使 $r_{i+1}$ 与 $r_{i+2}$ 之间的距离达到最大。因此，由$\mathcal{T}_{\mathrm{F}}$的定义可得

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^k 00) = \left( \frac{\mathbf{a}}{\mathbf{b}} \, , \, \frac{2\mathbf{a}+c}{2\mathbf{b}+d} \right) \quad \text{and} \quad r_{i+2} = \left( \frac{2\mathbf{a}+c}{2\mathbf{b}+d} \right) \, .$$

此外，同样由 $\mathcal{T}_{\mathrm{F}}$ 的定义可得

$$\mathcal{T}_{\mathrm{F}}(\sigma 1^k 0) = \left( \frac{\mathbf{a}}{\mathbf{b}} \, , \, \frac{\mathbf{a}+c}{\mathbf{b}+d} \right) \quad \text{and} \quad \mathcal{T}_{\mathrm{F}}(\sigma 1^k 01) = \left( \frac{2\mathbf{a}+c}{2\mathbf{b}+d} \, , \, \frac{\mathbf{a}+c}{\mathbf{b}+d} \right) \, .$$

这表明 $((2\mathbf{a} + c)/(2\mathbf{b} + d), (\mathbf{a} + c)/(\mathbf{b} + d)$ 是 $\mathcal{T}_{\mathrm{F}}$ 中的一个区间，因此，由定理 2.2 可得

$$\frac{\mathbf{a}+c}{\mathbf{b}+d} - \frac{2\mathbf{a}+c}{2\mathbf{b}+d} = \frac{1}{(\mathbf{b}+d)(2\mathbf{b}+d)} \, . \tag{9.10}$$

现在我们可以用以下内容完成 (ii) 的证明

$$r_i - r_{i+1} = \frac{c}{d} - \frac{\mathbf{a}+c}{\mathbf{b}+d} \overset{(9.9)}{=} \frac{1}{d(\mathbf{b}+d)} > \frac{1}{(\mathbf{b}+d)(2\mathbf{b}+d)} \overset{(9.10)}{=} \frac{\mathbf{a}+c}{\mathbf{b}+d} - \frac{2\mathbf{a}+c}{2\mathbf{b}+d} = r_{i+1} - r_{i+2} \, .$$

□

在引理 9.9 中，必须假设最佳逼近是完备的。一般而言，引理对最佳逼近并不成立：例如，令 $(1000)^{-1} > \alpha > 0$。于是我们有 $1 > 1/2 > 1/3 > 1/99 > \ldots > \alpha$ where $1/2$, $1/3$ and $1/99$ all are $\alpha$ 的右侧最佳逼近数，但以下说法是错误的：

$1/2 - 1/3 > 1/3 - 1/99$。 $\alpha$ 的完全右最佳逼近形式如下:

$$1 \;>\; \frac{1}{2} \;>\; \frac{1}{3} \;>\; \frac{1}{4} \;>\; \ldots \;>\; \frac{1}{i} \;>\; \frac{1}{i+1} \;>\; \ldots \;>\; \frac{1}{1000} \;>\; \ldots \;>\; \alpha\,.$$

设 $\{\hat{r}_i\}_{i\in\mathbb{N}}$ 是 $\alpha$ 的完全左最佳逼近, $\{\check{r}_i\}_{i\in\mathbb{N}}$ 是 $\alpha$ 的完全右最佳逼近。此外, 设

$$\hat{k}_i = (\hat{r}_{i+2} - \hat{r}_{i+1})/(\hat{r}_{i+1} - \hat{r}_i) \quad \text{and} \quad \check{k}_i = (\check{r}_{i+1} - \check{r}_{i+2})/(\check{r}_i - \check{r}_{i+1})\,.$$

根据前面的引理, 对于所有 $i$, 我们有 $\hat{k}_i < 1$ 和 $\check{k}_i < 1$, 因此可以通过下式定义 $\alpha$ 的压缩映射 $F$:

$$F(x) = \begin{cases} \hat{r}_{i+1} + \hat{k}_i(x - \hat{r}_i) & \text{if } \hat{r}_i \le x < \hat{r}_{i+1} \\ \check{r}_{i+1} + \check{k}_i(x - \check{r}_i) & \text{if } \check{r}_{i+1} < x \le \check{r}_i\,. \end{cases} \tag{9.11}$$

为了验证 $F$ 确实是 $\alpha$ 的压缩映射, 我们将证明

$$|F(x) - F(y)| \;<\; |x - y| \tag{9.12}$$

只要 $x \neq y$, 上述结论就成立。证明分为几种情形。不失一般性, 可设 $x < y$。

情形 (i) 存在 $i$、$j$, 使得 $\hat{r}_i \le x < \hat{r}_{i+1}$ 且 $\check{r}_{j+1} < y \le \check{r}_j$。此时 $F$ 将 $x$ 向右移动, 并将 $y$ 向左移动, 因此 (9.12) 成立。

情形 (ii) 存在 $i$, 使得 $\hat{r}_i \le x < y < \hat{r}_{i+1}$。此时有 $|F(x) - F(y)| = k_i|x - y|$, 且由于 $k_i < 1$, (9.12) 成立。

情形 (iii) 存在 $i$、$j$ (其中 $i < j$), 使得 $\hat{r}_i \le x < \hat{r}_{i+1}$ 且 $\hat{r}_j \le y < \hat{r}_{j+1}$。我们写作 $x = \hat{r}_{i+1} - b(\hat{r}_{i+1} - \hat{r}_i)$ 和 $y = \hat{r}_j + a(\hat{r}_{j+1} - \hat{r}_j)$, 其中$0 \le a$、$b \le 1$。于是有

$$y - x = a(\hat{r}_{j+1} - \hat{r}_j) + (\hat{r}_j - \hat{r}_{i+1}) + b(\hat{r}_{i+1} - \hat{r}_i)\,.$$

此外, 我们有

$$\begin{aligned} F(x) = \hat{r}_{i+1} + \hat{k}_i(x - \hat{r}_i) &= \hat{r}_{i+2} - (\hat{r}_{i+2} - \hat{r}_{i+1}) + \hat{k}_i(x - \hat{r}_i) \\ &= \hat{r}_{i+2} - \hat{k}_i(\hat{r}_{i+1} - \hat{r}_i) + \hat{k}_i(x - \hat{r}_i) = \hat{r}_{i+2} - \hat{k}_i(\hat{r}_{i+1} - x) \\ &= \hat{r}_{i+2} - \hat{k}_i b(\hat{r}_{i+1} - \hat{r}_i) \end{aligned}$$

$\hat{r}_{j+1} + k_j a(\hat{r}_{i+1} - \hat{r}_i)$, 因此

$$F(y) - F(x) = \hat{k}_j a(\hat{r}_{j+1} - \hat{r}_j) + (\hat{r}_{j+1} - \hat{r}_{i+2}) + \hat{k}_i b(\hat{r}_{i+1} - \hat{r}_i)\,.$$

现在很容易验证, $y - x > F(y) - F(x)$。

情形 (iv) 存在 $i$ 使得 $\check{r}_{i+1} < x < y \leq \check{r}_i$。此情形与 (ii) 对称 $\check{}$。利用 $k_i < 1$。

情形 (v) 存在 $i$、$j$（其中 $i < j$），使得 $\check{r}_{i+1} < x \leq \check{r}_i$ 且 $\check{r}_{j+1} < y \leq \check{r}_j$。此情形与 (iii) 对称。用引理 9.9 (ii) 代替引理 9.9 (i)。这就完成了 (9.12) 的证明。

引理 9.10。设 $L\colon\ \mathbb{N} \longrightarrow \mathbb{Q}$ 和 $R\colon\ \mathbb{N} \longrightarrow \mathbb{Q}$ 分别是无理数 $\alpha \in (0,1)$ 的完备左、右最佳逼近。设 $L(i) = a_i/b_i$ 和 $R(i) = c_i/d_i$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi^{L,\ R}_M\colon\ [0,1] \cap \mathbb{Q} \longrightarrow \mathbb{Q}$ 是 $\alpha$ 的压缩映射。
- $M^{L,\ R}$ 在输入 $n/m$ 上运行时间为 $O(\log\ m \log^2 \max\{b_m, d_m\})$，并且至多使用 $2 \log m$ 个预言机，每个预言机的输入规模至多为 $\log\ m$。

证明。定义 $M$ 为一台图灵机，在获得对 $L$ 的预言机访问后，$R$ 计算由（9.11）给出的压缩映射 $F$。对于输入 $n/m \in \mathbb{Q}$，由于 $L$ 和 $R$ 给出了最佳逼近，因此存在一个 $i$，使得 $1 < i < m$ 满足

$$L(i) \ =\ \frac{a_i}{b_i} \ \leq\ \frac{n}{m} \ <\ \frac{a_{i+1}}{b_{i+1}} \ =\ L(i+1)$$
$$\text{or} \quad R(i+1) \ =\ \frac{c_{i+1}}{d_{i+1}} \ <\ \frac{n}{m} \ \leq\ \frac{c_i}{d_i} \ =\ R(i)\,.$$

由于 $L$ 严格递增，而 $R$ 严格递减，$M$ 可以通过二分查找找到上述 $i$，最多需要 $\log\ m$ 步（因此对每个预言机最多查询 $\log m$ 次）；每一步都需要进行 4 次有理数比较，这些比较可在 $O(m_0^2)$ 时间内完成，其中 $m_0$ 是有理数的整数部分的二进制表示中最长者的长度（因为 $n/m < a/b$ 当且仅当 $nb < am$，且竖式乘法的时间复杂度是表示长度的二次方）。上述比较中出现的最大整数有如下上界

$$\max\{n, m, a_i, b_i, c_i, d_i\}_{i=1}^{m} \leq \max\{b_m, d_m\}$$

其中，不等式成立是因为所有分数都不超过 1，且 $\{a_i/b_i\}_{i\in\mathbb{N}}$ 和 $\{c_i/d_i\}_{i\in\mathbb{N}}$ 是最佳逼近。因此，使用 $2 \log m$ 次预言机调用（每次调用的规模至多为 $\log\ m$），计算 $i$ 所需的总时间为 $O(\log\ m \log^2 \max\{b_m, d_m\})$。找到 $i$ 后，$M$ 首先通过对原本最多可用 $\log \max\{b_m, d_m\}$ 位表示的数进行常数次加法、减法和乘法，计算 $F(n/m)$ 的一个分数表示（不一定是最简形式），因此至多需要 $O(\log^2 \max\{b_m, d_m\})$ 次运算，并得到一个可用 $O(\log \max\{b_m, d_m\})$ 位表示的数。获取

既约分数可以先计算分子与分母的最大公约数，然后执行所需的 2 次整数除法，总计还需 $O(\log^2 \max\{b_m, d_m\})$ 次运算。

□

在给出将迹函数（从而也包括压缩映射）转换为最佳逼近的算法之前，我们先作几点观察。第一个观察显而易见：如果迹函数将 $\alpha$ 中的一个有理数向右（向左）移动，那么该有理数就小于（大于） $\alpha$。因此，如果我们能够使用 $\alpha$ 的迹函数，就能轻松计算出 $\alpha$的戴德金分割$D^\alpha$ 。下一个观察稍微复杂一些：设 $T_0(r) = (r + T(r))/2$ ，其中 $T$是 $\alpha$的迹函数。那么， $T_0$ 也将是 $\alpha$的迹函数；此外，我们有

$$T_0(r) < \alpha \text{ if } r < \alpha \tag{9.13}$$

and

$$T_0(r) > \alpha \text{ if } r > \alpha. \tag{9.14}$$

为说明 (9.13) 成立，假设 $r < \alpha$和 $T_0(r) = (r+T(r))/2 \geq \alpha$。那么，我们有 $T(r) - \alpha \geq \alpha - r > 0$，因此 $|T(r) - \alpha| = T(r) - \alpha \geq \alpha - r = |\alpha - r|$，这与 $T$是迹函数相矛盾。通过对称论证可知 (9.14) 也成立。

我们将迹函数 $T$转换为 $\alpha$ 的完全左最佳逼近 $\{a_i/b_i\}_{i\in\mathbb{N}}$ 的算法，会用到戴德金分割 $D^\alpha$ 和迹函数 $T_0$。当 $n = 0$时，算法只需令 $a_n/b_n = 0/1$。当 $n > 0$时，算法执行以下步骤。

- 步骤 1：递归计算 $T_0(a_{n-1}/b_{n-1})$的值 $a'/b'$ 。
- 步骤 2：利用 $D^\alpha$，搜索满足以下条件的最小自然数 $b'' \leq b'$ ：存在某个 $a''$，使得 $a_{n-1}/b_{n-1} < a''/b'' < \alpha$。
- 步骤 3：利用 $D^\alpha$，找出满足 $a''/b'' < \alpha$的最大 $a'' < b''$ 。
- 令 $a_n/b_n$ 等于 $a''/b''$。

这样的 $a''$ 和 $b''$ 必定存在，因为如果没有 $b'' < b'$ 满足步骤 2 中的要求，那么 $b'$ 本身就满足该要求。容易看出， $a''/b''$ 将是严格大于 $a_{n-1}/b_{n-1}$的 $\alpha$ 的最小左最佳逼近。

当然，可以通过对称算法将迹函数转换为右最佳逼近。

引理 9.11。设 $T\colon [0,1] \cap \mathbb{Q} \longrightarrow \mathbb{Q}$ 是无理数 $\alpha \in (0, 1)$ 的迹函数。假设 $T(x)$可在时间 $s(\|x\|)$ 内计算，并设 $f(n) = \lambda x.(2s(x))^{(n)}(2)$。存在一个参数化函数预言机图灵机 $M$，使得

- $\Phi_M^T : \mathbb{N} \longrightarrow \mathbb{Q}$ *is the complete left best approximation of* $\alpha$

- $M^T$在输入 $n$ 时运行时间为 $O(2^{f(n)}+ s(f(n)))$，并且至多进行 $2^{f(n)}$ 次预言机调用，每次调用的规模至多为 $f(n)$。

证明。注意，计算 $T_0(r)$ 可以通过一次对 $T(r)$ 的预言机调用，再进行三次算术运算来完成，因此所需时间是关于 $r$ 和 $T(r)$ 的表示规模的多项式；后者不超过 $s(\|r\|)$。我们先对 $b'$ 的规模进行估计：根据假设， $a'/b'$ 可在 $s(\|a_{n-1}/b_{n-1}\|)$ 时间内计算，因此这也构成其规模的上界；根据定义， $b''$ 的规模至多与 $b'$ 相同； $a''$也同理。对 $n$ 作直接归纳可得 $\|a_n/b_n\| \leq f(n)$。

步骤 2 中对 $b''$ 的搜索至多需要 $b'$ 次对 $T$的调用（以计算 $\alpha$ 的戴德金分割），每次调用的参数都是一个有理数，其表示规模至多为 $\|a'/b'\|$。我们对 $b'$ 采用一个粗略的上界，即 $2^{\|a'/b'\|}$，以估算调用次数。

关于执行时间：算术运算、测试等会带来关于预言机调用次数的多项式开销；我们还应考虑到，调用的结果可能比其参数更大，因此需要再应用一次 $s$ 来覆盖这项开销。因此，我们得到表达式 $O(2^{f(n)} + s(f(n)))$。步骤 3 中搜索 $a''$ 的复杂度也恰好以同样的方式界定。这便得出了引理中所述的结论。

□

引理 9.11 的证明中所用的穷举搜索也可以通过斯特恩-布罗科特树来执行，但我们未能为这种方法推导出更好的界。

## 9.6 总结。

现在，我们可以按照此前总结各节相应结果时所采用的方式，总结本节所述各种转换的复杂度方面的结果。

定理 9.12。考虑以下表示：（1）连分数；（2）完全左最佳逼近与完全右最佳逼近；（3）迹函数；（4）压缩映射。设 $R_1$ 和 $R_2$ 是这四种表示中的任意两种。那么，对于任意时间界 $t$，都存在一个时间界 $s$，它是关于 $t$ 的原始递归函数，使得 $O(t)_{R_2} \subseteq O(s)^{R_1}$。

熟悉格热戈尔奇克层级的读者可以轻易验证，对于任意 $i \geq 3$，定理 9.12 中的时间界 $s$ 确实属于格热戈尔奇克类 $\mathcal{E}_{i+1}$，前提是时间界 $t$ 属于格热戈尔奇克类 $\mathcal{E}_i$。

# 参考文献

[1] Sanjeev Arora 和 Boaz Barak。《计算复杂性：一种现代方法》。剑桥大学出版社，2009。

[2] 作者不详 . 莱因德数学纸草书 , 约公元前1500年 .[3] A. Bauer、M. H¨otzel 和 A. Simpson。比较精确实数计算的函数式范式。载于《自动机、语言与编程》（ICALP 2002），《计算机科学讲义》第2380卷，第488–500页。施普林格出版社，2002年。[4]Anatole Beck、Michael N Bleicher 和 Donald W Crowe。《数学漫游：千禧年版》第7章“埃及分数”，第421–434页。CRC出版社，2000。[5] Ulrich Berger、Kenji Miyamoto、Helmut Schwichtenberg 和 Hideki Tsuiki。格雷码计算的逻辑。载于《数学、哲学与计算机科学中的证明概念》，《Ontos数理逻辑》第6卷。德古意特出版社，2016年。[6] Joseph Bertrand。《算术论 〈文 〉》。1849年。[7] B. Bolzano。《数论纯理论》。牛津大学出版社，2004年。收录于由 Steve Russ 编辑并翻译的《Bernard Bolzano数学著作》，第355–428页。[8] Vasco Brattka 和 Peter Hertling。实数表示的拓扑性质。《理论计算机科学》，284(2)：241–257，2002年。[9] Martin Bunder、Bruce Bates 和 Keith Tognetti。如何在斯特恩-布罗科特树中定位项。《欧洲组合学杂志》，31(3)：1020–1033，2010年。[10] Augustin-Louis Cauchy。《数学练习》。Bure Fr`eres，1826年。[11]Robert Cohen。埃及分数展开。《数学杂志》，46(2)：76–80，1973年。[12]Richard Dedekind。《连续性与无理数》。不伦瑞克：维韦格出版社，1872年。[13] Ding-Zhu Du 和 Ker-I Ko。《计算复杂性理论》。威立国际科学出版公司，2000。[14] Beatty 等。问题与解答：待解问题：3173–3180。《美国数学月刊》，33(3)：159，1926年。[15] I. Georgiev。戴德金分割与基数展开中的长零串。载于《CiE 2021：与可计算性相联》，《计算机科学讲义》第12813卷，第248–259页。施普林格出版社，2021年。[16] I. Georgiev、L. Kristiansen 和 F. Stephan。表示复杂度出人意料的可计算无理数。《纯粹与应用逻辑年刊》，172(2)：102893，2021年。

[17] G. H. Hardy 和 E. M. Wright。《数论》。牛津克拉伦登出版社，1975年。[18] A. Hurwitz。关于用有理分数¨近似表示无理数。《数学年刊》，39：279–284，1891年。[19] A. Ya. Khintchine。《连分数》。P. Noordhoff 有限公司，1963年。P. Wynn 译。[20] K. Ko。《实函数复杂性理论》。Birkh¨auser，1991年。[21] K. Ko 和 H. Friedman。实函数的计算复杂性。《理论计算机科学》，20(3)：323–352，1982年。[22] Christoph Kreitz 和 Klaus Weihrauch。表示理论。《理论计算机科学》，38：35–53，1985年。[23]L. Kristiansen。论无理数的次递归可表示性。《可计算性》，6：249–276，2017年。[24] L. Kristiansen。论无理数的次递归可表示性（第二部分）。《可计算性》，8：43–65，2019年。[25] L. Kristiansen。论无理数的次递归表示：收缩映射与贝尔序列。载于《CiE 2021：与可计算性相联》，《计算机科学讲义》第12813卷，第308–317页。施普林格出版社，2021年。[26] L. Kristiansen 和 J. G. Simonsen。论经典实数表示之间转换的复杂度。载于 CiE 2020：《超越可计算性的视野》，《计算机科学讲义》第12089卷，第75–86页。施普林格出版社，2020年。[27] R. S. Lehman。论原始递归实数。《数学基础》，49(2)：105–118，1961年。[28] A. Mostowski。论可计算序列。《数学基础》，44：37–51，1957年。[29] I. Richards。轻松学习连分数。《数学杂志》，54(4)：163–172，1981年。[30] Michael Sipser。《计算理论导论》。PWS Publishing Company，1997年。[31] Hideki Tsuiki。通过格雷码嵌入进行实数计算。《理论计算机科学》，284(2)：467–485，2002年。[32] K. Weihrauch。若干实数表示之间转换器的不连续度。技术报告，Fernuniversit¨at Hagen，1992年。

[33] K. Weihrauch.《可计算分析》。Springer Verlag，2000。[34] Klaus Weihrauch 和 Christoph Kreitz。实数的表示以及实数集的开子集的表示。《纯粹与应用逻辑年刊》，35:247–260，1987。