arXiv:2611.29559v1 [cs.FL] 22 Nov 2026

Neary 的 PCP(5) 构造的详细验证

Vesa Halava 数学与统计学系 图尔库大学，芬兰 电子邮箱：
vesa.halava@utu.fi

2026年11月24日

摘要

Neary 证明，对于包含五对字的实例，Post 对应问题仍然不可判定。本文对他的构造进行了详细验证。我们首先验证归约中所用的受限二进制标签系统停机问题的不可判定性。随后，利用 Rote 提出的较长闭合对，证明归约到五对问题的两个方向。特别地，我们证明，所构造的每个 PCP 实例的解都对应于模拟的标签系统的一次停机计算。

1 引言

Post 对应问题的一个实例 [5]由有限个字对组成（u_i，v_i），这些字来自同一个字母表。解是一个非空序列 i_1，……，i_t，允许重复，使得

$$u_{i_1} \cdots u_{i_t} = v_{i_1} \cdots v_{i_t}.$$

我们用 PCP(k) 表示限制为恰好包含 k 对的实例。本文讨论 PCP(5) 的不可判定性，即使在二元字母表 {0，1}上也是如此。

Matiyasevich 和 Sénizergues 证明了一个三产生式半图厄系统的不可判定性，从而得到了一个 PCP(7) 构造 [2]。他们使用了 Claus 将 n产生式半图厄系统的字问题归约到 PCP($n+4$) 的方法 [1]。Neary 使用二进制标签系统证明了 PCP(5) 的不可判定性 [3]。本文旨在详细验证 Neary的构造。

Neary的构造使用了类似于 Matiyasevich 和 Sénizergues 所采用的编码原理。被模拟系统的程序被编码在单个产生式中。在 Neary 的二进制标签系统中，规则 $c \to Ub$ 的右侧 Ub 编码了整个循环标签程序，而另一条规则为 $b \to b$。读取 Ub 所在的轨道会选择相应的指令。

多位读者询问了 Neary 证明中的逆向蕴含；由于已发表的论证十分简短，因此很难据此还原。第2 节验证了

归约中使用的受限标签系统构造。我们指出循环模拟中的停机指令，说明二进制标签系统如何检测到它，并验证使 b 指定计算中的唯一终止词成立的长度同余。第3 节和4 节使用 Rote 的较长闭合对，验证 Neary 的五对归约的两个方向 [6, 第 2.8 节，脚注 3]。

2 受限二进制标签系统

我们从零开始为字中的位置编号。因此，对于长度为n的字 $v = v_0v_1 \cdots v_{n-1}$， v_i 表示位置 i 上的符号，其中 $0 \leq i < n$。

标签系统是一个三元组 $T = (A, R, \beta)$，其中 A 是有限字母表， $R : A \to A^*$ 为每个字母指定一个产生词，而 $\beta \geq 1$ 是删除数。构型是 A上的一个字，称为数据字。若数据字为 $a_0 \cdots a_{n-1}$且$n \geq \beta$，则一步转移得到

$$a_0 \cdots a_{n-1} \longmapsto a_\beta \cdots a_{n-1}R(a_0).$$

当数据字的长度小于 β时，系统停机。当其字母表包含两个字母时，该系统称为二进制系统。

我们通过先用循环标签系统、再用二进制标签系统模拟图灵机，构造用于邮编归约的那一族系统。循环模拟提供了一条用于标示停机的特殊指令。图灵机 M 是确定性的，并有一条以 $\mathbb{Z}$为索引的纸带。它具有有限状态集、一个初始状态、一个指定的停机状态，以及一个带有空白符号的有限带字母表。对于每个非停机状态和当前扫描符号，转移函数都会指定一个新状态、一个要写入的符号，以及向左或向右移动一个单元格。输入为 y时， y中的字母从单元格0开始依次占据连续单元格，其余单元格均为空白，读写头从单元格0开始，初始状态为初始状态。机器在进入停机状态时停机。

2.1 循环停机信号

顺时针图灵机。 二进制顺时针图灵机是一个元组 $R = (Q_R, \Sigma_R, \delta_R, q_{\text{in}}, q_{\text{h}})$，其中 Q_R 是有限状态集， $\Sigma_R = \{A, B\}$ 是带字母表，而 $q_{\text{in}}, q_{\text{h}} \in Q_R$ 分别是初始状态和停机状态。其转移函数为

$$\delta_R : (Q_R \setminus \{q_{\text{h}}\}) \times \Sigma_R \longrightarrow Q_R \times (\Sigma_R \cup \Sigma_R^2).$$

用（q， au）表示一个构型，其中 au 是按顺时针顺序排列的非空带字，从当前扫描符号 a 开始。如果 δ_R（q， a） = （q'， v），则下一个构型为（q'， uv）。这会将当前扫描单元替换为 v 的一个或两个字母，并按顺时针方向将磁头移至替换部分之后。该机器在 q_{h}中停机。Neary 和 Woods 的构造给出了 M的二进制顺时针模拟器 R_M。对于每个输入 y，令 $\tau_y \in \{A, B\}^+$ 表示该构造为 R_M 给出的初始带字，按顺时针顺序从当前扫描单元开始书写。因此， R_M 当且仅当 M 在第2节和引理 1]中对 y [4, 停机时，才会从（q_{in}， τ_y）停机。

循环标签系统。循环标签程序是一个由二进制字 α_0、.....、 α_{p-1}组成的非空列表。构型是一个有序对（m， w），其中 $0 \le m < p$ 是当前指令索引，$w \in \{0, 1\}^*$ 是队列，队首位于左侧。令 $m' = (m+1) \bmod p$。对于每个二进制字 v，步关系为

$$(m, 0v) \longrightarrow (m', v),$$
$$(m, 1v) \longrightarrow (m', v\alpha_m).$$

因此，每一步都会读取并删除最左端的比特。如果该比特为 1，则使用步骤前的索引，在右端追加 α_m 。队列为空时无法执行步骤。指令h 恰好在从 $(, 1v)$ 出发的步骤中触发。

引理 1。 给定一台普通单带图灵机 M 和一个输入字 y，可以有效地构造一个循环标签程序 $\alpha_0, \ldots, \alpha_{p-1}$、一个索引为零的非空初始队列 w 以及一个指定索引 $0 < h < p$，使其具有以下性质。

1. 指令 h触发，当且仅当 M 在 y上停机。如果 M 不停止，则循环计算有无限多个步骤。*2.* 如果在 h首次触发之前队列为 $1r$，那么在该次触发之后的接下来 $|r|$ 步中（这几步会消耗 r的比特）， h 不会触发。

证明。 首先按上述方式构造 R_M 和初始带字 τ_y。该循环标签程序模拟这台顺时针机器。我们采用 [4, 第 3.1 节引理 2]的证明中的构造。令 $N = |Q_R|$， $z = 30N+61$，并令 $p = 2z$。对状态编号，使得 $q_{\mathrm{h}} = q_N$。

（初始）构型按以下方式编码到标签系统中。普通状态和带码字以及计数字为

$$S_i = 0^{30i+20}1\,0^{p-30i-21},$$
$$T_A = 01\,0^{p-2}, \qquad T_B = 001\,0^{p-3},$$
$$\mu = 1\,0^{z-1}.$$

通过串接带码字，将 T 扩展为字。对于非空带字 τ，令 $K(\tau)$ 为不小于 $|\tau|$ 的最小 2 的幂。对于 $q_i = q_{\mathrm{in}}$，初始循环构型为

$$(0, w), \qquad w = S_i T(\tau_y)\mu^{K(\tau_y)}.$$

这就是 Neary 和 Woods 的方程（3）。

由 [4] 我们得到以下细节：对于每个 $0 \le n < p$，使用以下规则。在保留索引 $h = 30N + 20$ 处设置 $\alpha_h = S_N$，如同 Neary 和 Woods 的停机情形。对于所有其他索引，使用其表 1.1、1.2、2.1、3.1 和 3.2 中的赋值，并将其中的状态和转移替换为 R_M的状态和转移。每一行都指定附加串的索引以及要分配给该索引的二进制字。在表 3.1 中，为所示状态和带符号选择具有 δ_R 所指定写入长度的转移行。将空字分配给每个尚未分配的索引。这些规则有效地给出了整个程序。

Neary 和 Woods 分三个阶段模拟顺时针转移。前两个阶段使用计数器标记并隔离扫描到的带符号。第三个阶段使用当前状态和被隔离的符号来选择 R_M的转移，写入替换符号和新状态，并恢复普通的带码字。如果纸带长度超过当前计数器值，表 3.2 会将该值加倍。每次模拟的转移都需要有限个循环标记步骤，并将指令索引恢复为零。

例如，从 $(0,\ S_iT(au)\mu^{K(au)})$ 开始，一次转移 $\delta_R(q_i,\ a) = (q_k,\ v)$ 给出边界构型

$$(0, T(v)S_kT(u)\mu^{K(uv)}).$$

从 S_k 开始按循环顺序读取带码字，得到带字 uv，起点是新扫描的单元格。

在模拟的停机情形 [4, 第 3.1 节，引理 2],的证明末尾， $q_N = q_{\mathrm{h}}$ 的普通码字在索引零处输入。其唯一的 1 因此触发指令 h，因为

$$S_N = 0^h 1\, 0^{p-h-1} = \alpha_h.$$

为说明 h不可能更早触发，考虑模拟一个非停机转移期间读取 1 的循环步骤。对于每个编码对象，Neary 和 Woods 的表 1.1–3.2 同时给出了对象进入时的标记索引，以及读取对象的 1 时所处的附加项索引。因此只需检查附加项索引列。在下面的依赖状态的索引 z中，唯一与 20 模 30 同余的索引具有形式 $30i + 20$。当普通状态字 S_i 在索引零处输入时，就会出现这些索引。下面的固定索引 z 至多为 $42 < h$，而所有至少为 z 的索引都超过 h。特别地，当表 3.2 在计数器加倍期间复制新的状态字 S_k 时，所选索引为 $30k + 60$。因此， h 恰好在模拟到达停机状态时触发。如果 R_M 永远运行，其连续的转移就会产生无限多个循环标记步骤。

在首次表示停机状态的转移边界处，队列包含单个状态字 S_N。 S_N 之前的任何带字都会在索引 0 处读取，并在进入 S_N 之前复制到队尾。当其唯一的 1 触发指令 h 时，未读后缀 r 因此由 S_N 的 $p-h-1$ 个尾随零以及其后的完整带字和计数字组成。越过尾随零后，索引为 0。每个带字的长度为 $p = 2z$ ，其 1 位于偏移量 1 或 2 处。每个计数字为 10^{z-1} ，长度为 z。因此，对象边界会在索引 0 或 z 处被进入，而这些对象中的 1 仅出现在 {0、1、2、 z、 $z+1$、 $z+2$} 所示的索引处，这些索引均不同于 h。 S_N 的新副本会附加在 r 后面，因此在接下来的 $|r|$ 步中不会读取该副本的任何比特。这证明了第二个性质。

□

2.2 受限停机问题

以下构造使用了 [3] 中引理 9 的轨道方法，以及 Neary 的定理 11 中使用的长度同余。

引理 2. 给定一台图灵机 M 和输入 y，可以有效地构造一个删除数 $\beta \geq 2$ 以及一个产生式为 $R(b) = b$, $R(c) = Ub$ 的二进制标签系统，

其中 $U = u_0 \cdots u_\ell$。在指定输入上

$$w_0 = u_{\beta-1} u_\beta \cdots u_\ell b$$

当且仅当 M 在 y上停机时，该系统才会停机。对于所构造的 β，可以为每个充分大的 $s \equiv 1$ （模 $\beta - 1$）指定$|Ub| = s\beta$。

证明。 我们采用 Neary 方法，将循环标签程序存储在一个产生词的各条轨道中。我们为每个普通输出设置相同的初始填充。这样，即使循环附加串为空，也能使用单一的轨道公式。

取引理 1. 中的循环程序 $\alpha_0, \ldots, \alpha_{p-1}$、初始串 w和停机指令 h。

令 $\beta = 10p$，并将待构造的产生词记作 $P = Ub$，其长度为$s\beta$。对于 $0 \le j < \beta$，将其第 j 条轨道定义为

$$T_j = P_j P_{j+\beta} \cdots P_{j+(s-1)\beta}.$$

这些轨道将 P 的各个位置划分为互不相交的部分。因此，为每个 T_j 指定内容，就能确定 P。将产生式扩展为态射 φ，令 $\varphi(b) = b$， $\varphi(c) = P$。如果一次遍历首次读取某个 P 副本中的位置 j，那么它会读取 T_j，并附加 $\varphi(T_j)$。更一般地，将块中首次读取的位置称为该块的偏移量。离开一个长度为 d 、从偏移量 z 进入的块时，下一块中的偏移量为 $z - d \bmod \beta$，因为连续读取的位置相差 β。下文中的偏移量均为最小非负剩余。

令 $G = P$，并定义另外三个块及其短编码如下

	short encoding	block
padding	$e = b^4 c b^6$	$E = b^4 P b^6$
0	$x_0 = b^6 c b^4$	$X_0 = b^6 P b^4$
1	$x_1 = b^8 c b^2$	$X_1 = b^8 P b^2$.

对于比特字 v，用 $x(v)$ 和 $X(v)$ 表示相应的串接。因此$\varphi(e) = E$ 以及 $\varphi(x(v)) = X(v)$。我们使用 G 和完整组 E^p 作为填充。表示程序索引 m 的偏移量为

$$z_m = (1 - 10m) \bmod \beta.$$

读取 G会保留偏移量。 E， X_0， X_1的长度均为 $s\beta + 10$，因此需要 z_m才能到达z_{m+1} 模 p。特别地， E^p 会保留输入索引。

现在我们规定普通块使用的轨道。取 $a \in \{4, 6, 8\}$，并在偏移量 z_m处输入 $b^a P b^{10-a}$。在 P内部，轨道为 T_j，其中 $j = (z_m - a) \bmod \beta$。对于$m = 0$，偏移量为 1，因此遍历会读取一个前导 b，后接 T_j。对于 $m > 0$，偏移量至少为 11，因此它恰好读取 T_j。事实上，当 $z_m = 1$时，该块包含 $s+1$ 个采样位置；当 $z_m \ge 11$时，则包含 s 个。在这两种情况下， P内部的 s 个采样点都会涵盖除可能的前导 b之外的所有读取。

当 $m = 0$ 时令 $\delta_m = 1$，否则令其为 0。对于 E 和 X_0，取 $v = \varepsilon$；对于满足 $m \ne h$ 的 X_1，取 $v = \alpha_m$。规定横跨整个块读取的字为

$$D = e^p x(v) c^t, \qquad t = s + \delta_m - 11p - 11|v|.$$

短编码的长度为 11，因此 D 的长度为 $s+\delta_m$。它以 b 开头。对于 $m>0$，令 $T_j=D$；对于 $m=0$，从 D 中移除第一个 b，得到 T_j。无论哪种情况，该块都恰好追加

$$\varphi(D)=E^pX(v)G^t.$$

因此，即使是空的 v 也只会产生完整的填充。对于 G本身，令 $T_{z_m}=c^s$，从而得到输出 G^s。

对于索引为 h 的 X_1，规定$T_{(z_h-8)\bmod\beta}=bc^{s-1}$。由于 $h>0$，整个块读取该轨道并追加 $H=bG^{s-1}$。将所有偶数轨道都设为 b^s。最后令

$$T_{\beta-1}=b^{\beta-2}x(w)c^ke^p,\qquad k=s-(\beta-2)-11|w|-11p.$$

G、E、X_0、X_1 的轨道对 10 取模后的余数分别为 1、7、5、3。在每一类中，p 程序索引给出模 $10p$意义下彼此不同的轨道。输入轨道的余数为 9；所有偶数轨道彼此分离。例外轨道在 h处取代X_1 轨道。用 b^s填充剩余轨道。这些赋值是有效的。取任意足够大的$s\equiv1$（模 $\beta-1$），使得 $s\geq2\beta+3$、$k\geq0$以及每个普通指数 t都是非负的。P 的最后一个字母是 $T_{\beta-1}$的最后一个字母，即 b。因此，$P=Ub$，如所需。

我们现在可以研究所定义的二进制标签系统的计算。删除 P 的前 $\beta-1$ 个字母，恰好得到 w_0。遍历这个后缀会读取 $T_{\beta-1}$，并附加 $b^{\beta-2}X(w)G^kE^p$。该后缀的长度为 $(s-1)\beta+1$，因此最后一次删除会延伸到输出中$\beta-1$ 个位置。它会消耗前缀 $b^{\beta-2}$ 和 $X(w)$ 中的一个字母。因此，第一个编码位在 $1=z_0$ 处进入。

为了证明这一点以及后续所有块遍历的正确性，我们甚至考虑当前块之后没有等待处理的其他块的情况。初始后缀和每个普通块在此处使用的每种进入偏移量下，都至少包含 s 个采样位置。直到最后一个采样点，每次删除都不会超出块的范围，因为前方 β 个位置处还有另一个采样点。到达最后一个采样点时，至少已有 $s-1$ 个字母被附加。这些字母足以提供该次删除所需的至多 $\beta-1$ 个额外字母。这也适用于在 H中遍历的 G 的每个副本。

在使用例外轨道之前，考虑连续遍历单元 X_0，X_1，G，E^p之间的边界，并将每个完整分组 E^p 视为一个单元。此前的删除操作可能已经消耗了首个待处理单元的一个前缀。其进入偏移量记录了该前缀的长度。

模拟不变量是：省略每个 G 和 E^p，并将每个 X_i 替换为 i后，待处理单元构成当前的循环队列。如果当前指令为 m，则边界处的进入偏移量为 z_m。初始化时，队列为 w，偏移量为 $z_0=1$，由此建立该不变量。遍历 G 或 E^p 会保持偏移量不变，并且只追加填充。遍历 X_i 会将偏移量推进至 $z_{(m+1)\bmod p}$，并在填充之外追加 α_m 的编码（若 $i=1$），若 $i=0$则追加空字，前提是 $(i,\ m)\neq(1,\ h)$。由此，通过对已遍历单元进行归纳法即可证明该不变量。

每个待处理的编码位都会在有限次遍历后被到达。它前面只有有限个单元，而这些单元的输出都会追加到它后面。因此，无限循环计算会产生无限多个合法的二进制标记步骤。

当指令 h 首次触发时，例外轨道会追加 H。已经排在 H 前面的编码位，恰好是引理 r 1中后缀的比特。它们的指令索引和操作不受将后方的循环输出替换为$H\mathbf{1}$ 的影响。引理确保这些比特都不会触发 h。在 H前面只有有限个单元，而这些单元的输出都会追加到它后面，因此 H 会在有限次遍历后被到达。此时所有排在 H 后面等待的单元都是普通单元。

H 的进入偏移量为奇数，因为所有普通块的块长度都是偶数，而所有正常偏移量都是奇数。它开头的 b 被跳过，因此其第一个 G 中的偏移量少一，因而为偶数。在 H 中， G 的所有其余副本也都是从偶数偏移量处进入的。它们只读取 b，且只追加 b。H 后面等待处理的普通块长度均为偶数，其内部 P 前的填充长度也为偶数。因此，每个块都保持偶数奇偶性，并且只读取 b，无论是在 P 内还是在其填充部分。经过这次有限遍历后，数据字完全由 b 组成。此后的每一步都会使其缩短 $\beta-1>0$，所以最终停机。这证明了等价性，进而证明了指定族上的停机问题不可判定。

□

引理 3. 从 w_0 可达的每个数据字都非空，以 b 结尾，且其长度模 $\beta-1$ 与 1 同余。当且仅当计算到达单字母字 b 时，它才会停机。

证明。 初始长度为 $(s-1)\beta+1$。一个 b 步骤会使长度变化 $1-\beta$，而一个 c 步骤会使长度变化 $(s-1)\beta$。这两种变化都保持同余关系。初始串和两个产生式都以 b 结尾，因此每个可达字都非空，并以 b 结尾。因此，长度小于 β 的字只能是长度为 **1** 的字。

□

3 PCP(5)实例

PCP(5) 模拟仅将 U 作为由 $\{b,c\}$ 中符号构成的字来使用。它不会对存储在该字中的循环程序进行解码。定义态射

$$\langle\cdot\rangle : \{b,c\}^* \longrightarrow \{0,1\}^*$$

by

$$\langle b\rangle = 10^\beta 1, \qquad \langle c\rangle = 1,$$

并通过串接将其扩展：

$$\langle x_1 \cdots x_m\rangle = \langle x_1\rangle \cdots \langle x_m\rangle, \qquad \langle\varepsilon\rangle = \varepsilon.$$

特别地，$\langle U\rangle$ 是通过对 U 中的每个字母进行编码得到的。PCP 对利用这种编码重现标签步骤，而不依赖于对 U 中各个字母的解释。反向证明表明，每个 PCP 解都会导出一个停机的标签计算。

符号的块是仅由该符号重复组成的非空字。若字中的某个块无法向任一侧再延伸一个相同符号的出现，则称其为极大的。

对于闭合对 F，我们采用 Rote 更长的 1 序列块 [6, 第 2.8 节，脚注 3]。两个产生词中 c 序列块的长度都至多为 $|Ub|-1$，因为产生词分别是 b 和 Ub，且都以 b 结尾。在每个可达数据字中，这一上界也成立：初始串是 Ub 的后缀，删除前缀不会使 c 序列块变长，而且任何剩余后缀都以 b 结尾。因此，没有 c 序列块能够跨越边界，延伸到追加的产生词中。

Set

$$B = \langle b \rangle = 10^{\beta}1, \qquad V = \langle U \rangle, \qquad L = |Ub| + 99.$$

我们恰好构造五个二进制对，表示为（上串，下部词）：

pair	upper	lower
A	1	$1V10$
B_+	B	110
B_-	B	0
C_-	1	0
F	$10^{\beta}1^{L}$	1^{L}.

(1)

前四个对属于 Neary。由 Rote [6], 建议的较长闭合对 F替代了他的（$10^{\beta}1111$，**1111**），且没有增加对数。引理 7 的证明只使用不等式 $L > |Ub| + 1$。要看出原始指数 4 的问题，可以考虑一个以 $bccb$ 开头的字 w。它的悬垂部分 $Z(w)$ 以 $10^{\beta}1^4$ 开头，因此原始闭合对的上串与此前缀匹配。对 A初始化一个 c规则，之后表示该规则； B_+ 表示一个 b规则； B_-， C_- 删除字母。

对于 $w = xb$，其中 $x \in \{b,\ c\}^*$，定义

$$Z(w) = \langle x \rangle 10^{\beta}. \tag{2}$$

特别地， $Z(b) = 10^{\beta}$。最后的 b 由截断的 B表示。设 r 和 v是与 PCP 解的某个前缀对应的上串和下串的拼接。如果$v = rZ(w)$，则下串拼接中未匹配的后缀 $Z(w)$ 称为下部余串。它表示标签数据字 w。

从 A开始，留下下部余串 $V10$。对于 $j = 0, \ldots, \beta-2$，如果满足 B_- if$u_j = b$，则使用 C_-；如果满足 $u_j = c$，则使用 C_-。它们的上串消耗了 U 的前 $\beta-1$ 个编码字母，而它们的下部词将末端 **10** 延伸至 10^{β}。所得的下部余串为

$$\langle u_{\beta-1} \cdots u_{\ell} \rangle 10^{\beta} = Z(w_0),$$

其中 $w_0 = u_{\beta-1} \ldots u_{\ell} b$。

在一个非停机字处，根据其首字母是 A 还是 B_+，使用 c 或 b；然后使用 $\beta-1$、C_- 或 B_-删除后续字母。引理 6 验证了精确的字恒等式。在停机字 b处， $Z(b) = 10^{\beta}$；使用 F 可使两个拼接相等，因为$10^{\beta}1^{L} = 10^{\beta}1^{L}$。因此，每个停机计算都会产生一个邮编解。这一正向模拟由 Neary [3, 在定理 11]的证明中给出；每个模拟标签步骤对应的字恒等式将在下文引理6中验证。这里我们使用 Rote 的较长闭合对 [6]。

4 逆向命题：PCP 解可推出停机

在任何邮编解中，每个前缀对应的上串和下串的拼接都是最终公共词的前缀。因此，其中一个必为另一个的前缀。当当前下方拼接等于上方拼接后接上 z，且一对 $(a,\ d)$ 中的 a 以 z为前缀时，新的下部余串为

$$z' = a^{-1}zd. \tag{3}$$

更一般地，如果（a, d）是解中的下一对，那么 a 和 zd 必须可以按前缀关系比较，因为它们都是最终公共词剩余部分的前缀。因此，它们公共范围内的任何不一致都无法由后续配对修正。

我们先确定任意 PCP 解在首次出现 F 之前的形式。

引理 4. 每个解都以 A 开始，并以 F 结束。直到首次出现 F 为止，其形式为

$$AD_{1,1}\cdots D_{1,\beta-1}\ T_2D_{2,1}\cdots D_{2,\beta-1}\ \cdots\ T_kD_{k,1}\cdots D_{k,\beta-1}\ F, \tag{4}$$

其中 $T_i \in \{A, B_+\}$ 且 $D_{i,j} \in \{B_-, C_-\}$。

证明。只有 A 能使等式成立。它的两个字都以 1 开头，而 B_-, C_- 的下部词以 0 开头，B_+, F 的下部词以 11 开头，而它们的上串以 10 开头。每个上串都以 1 结尾，而只有 F 的下部词以 1 结尾。因此，每个完整匹配都以 F 结束。

上串中的每个极大零块长度均为 β。由于每个上串都以 1 开始并以 1 结束，任何上串的串接也具有相同性质。$V = \langle U\rangle$ 中的每个极大零块长度均为 β。每个 A or B_+ 的下部词都以 10 结束，从而开始一个长度为 1 的零块。每个删除对贡献一个零，而在这个边界处，任何产生式或闭合对都不能贡献零。因此，在每次出现 A 或 B_+ 之后、下次出现 A、B_+ 或 F 之前，必须恰好跟随 $\beta-1$ 个删除对。从初始的 A 开始应用这一结论，直到首次出现 F。解可能在首次出现 F 之后继续；此处只考虑截至该次出现为止的前缀。

□

下一个引理说明，初始的一组配对会产生指定标签输入的编码。

引理 5。 假设某个 *PCP* 解的形式为 (4)。经过其初始序列

$$AD_{1,1}\cdots D_{1,\beta-1},$$

下部余串为 $Z(w_0)$。此外，对于 $j = 0, \ldots, \beta-2$，二元组 $D_{1,j+1}$ 由u_j唯一确定：

$$D_{1,j+1} = \begin{cases} B_-, & u_j = b, \\ C_-, & u_j = c. \end{cases}$$

证明。紧接着 A，下部余串为 $\langle U\rangle 10$，其中 $U = u_0 \cdots u_\ell$。根据（4），接下来的 $\beta - 1$ 对为 $D_{1,1}, \ldots, D_{1,\beta-1}$，每一对都属于 $\{B_-, C_-\}$。

我们依次证明，对于 $j = 0, \ldots, \beta - 2$，$D_{1,j+1}$ 由 u_j 决定。按照规定应用前 j 个删除对后，下部余串为

$$\langle u_j \cdots u_\ell \rangle 10^{j+1}.$$

如果 $u_j = b$，B_- 的上串会消耗 $\langle b\rangle = 10^\beta 1$。若改选 C_-，则只会消耗其中开头的 1，留下一个前导零。由于每个上串都以 1 开头，这种选择与 PCP 解不相容。

如果 $u_j = c$，悬垂部分以 11 开头：第一个 1 编码 u_j，第二个 1 则是下一个编码或终止符 10^{j+1} 的开头。因此，无法使用上串以 10 开头的 B_-，而必须选择 C_-。

无论哪种情况，所选的配对都会消耗 $\langle u_j\rangle$，并在下方拼接中追加一个零。因此，在 $AD_{1,1} \cdots D_{1,\beta-1}$ 之后，下部余串为

$$\langle u_{\beta-1} \cdots u_\ell \rangle 10^\beta = Z(w_0),$$

正如所要求的。 □

下一个引理表明，每个后续组 $T_i D_{i,\,1} \cdots D_{i,\,\beta-1}$ 恰好表示一个标签步骤。

引理 6。 假设一个邮编解具有如下形式 (4)，并令 $2 \le i \le k$。假设在 T_i 出现之前的紧邻时刻，下部余串为 $Z(w)$，其中

$$w = x_0 x_1 \cdots x_{m-1}$$

是一个可达的标签数据字，且 $m \ge \beta$。那么序列中的各对

$$T_i D_{i,1} \cdots D_{i,\beta-1}$$

由 w 的前 β 个字母唯一确定。具体而言，

$$T_i = \begin{cases} B_+, & x_0 = b, \\ A, & x_0 = c, \end{cases} \qquad D_{i,j} = \begin{cases} B_-, & x_j = b, \\ C_-, & x_j = c, \end{cases} \quad (1 \le j \le \beta - 1).$$

此序列之后，下部余串为 $Z(w')$，其中

$$w' = x_\beta \cdots x_{m-1} R(x_0)$$

是 w 的标签后继。当 $m = \beta$ 时，后缀 $x_\beta \cdots x_{m-1}$ 为空。

证明。由于 w 是可达的，其最后一个字母是 b。因此 $x_{m-1} = b$，并且

$$Z(w) = \langle x_0 \cdots x_{m-2} \rangle 10^\beta.$$

根据（4），T_i 要么是 A，要么是 B_+。

如果 $x_0 = c$，下部余串以 **11** 开头：第一个 **1** 表示 x_0，第二个 **1** 则开始表示下一个字母。 B_+ 的上串以 **10** 开头，因此 $T_i = A$ 是必然的。如果 $x_0 = b$，下部余串以 $B = 10^\beta 1$ 开头。选择 A 只会消耗它的第一个 **1**，留下一个前导零。由于每个上串都以 **1** 开头，这将与 PCP 解不相容。因此， $T_i = B_+$ 是必然的。

为了将两种情况一并处理，令

$$y = \begin{cases} U, & x_0 = c, \\ \varepsilon, & x_0 = b. \end{cases}$$

于是 $R(x_0) = yb$。在任一种情况下， T_i 的上串都是$\langle x_0 \rangle$，下部词则是 $1\langle y \rangle 10$。因此，在 T_i 之后，下部悬垂部分为

$$\begin{aligned} &\langle x_1 \cdots x_{m-2} \rangle 10^\beta 1 \langle y \rangle 10 \\ &\quad = \langle x_1 \cdots x_{m-1} \rangle \langle y \rangle 10. \end{aligned}$$

这里，追加的下部词中的第一个 **1** 补全了旧的最后一个 b 的截断表示 10^β，使其成为完整编码 B。

删除对 $D_{i,1}, \ldots, D_{i,\beta-1}$ 现在由与引理 5 的证明中相同的论证所强制确定：对于 $1 \le j \le \beta - 1$，必须使用 B_-，若 $x_j = b$；若 C_-，则使用 $x_j = c$。事实上，这些符号各自在悬垂部分中都有其完整编码，包括当 $m = \beta$ 时的最后一个 b。

每个删除对都会消耗$\langle x_j \rangle$ 并追加一个零。因此，经过全部 $\beta-1$ 个删除对后，下部悬垂部分为

$$\langle x_\beta \cdots x_{m-1} \rangle \langle y \rangle 10^\beta = Z(x_\beta \cdots x_{m-1} y b) = Z(w').$$

此处$x_\beta \cdots x_{m-1}$ 在 $m = \beta$时理解为空。因此，该序列在所有情况下都恰好表示一个标签步骤。

相同的恒等式还表明，从 $Z(w)$开始，总能附加规定的配对，得到 $Z(w')$，而无需假设存在完整的 PCP 解。 □

在下面的引理中，我们证明只有当可达的标签词为 b时， F才会出现。

引理 7. 假设一个 *PCP* 解具有如下形式 (4)。设在其首次出现 F之前，下部余串为 $Z(w)$，对应某个可达的标签词 w。那么 $w = b$，且这次出现的 F 使上串和下串的拼接相等。反过来，若下部余串为 $Z(b)$，应用 F 即可得到相等。

证明。 回忆一下， F的上串和下部词分别为 $10^\beta 1^L$ 和 1^L，其中$L = |Ub| + 99$。

如果 w 以 c开头，那么 $Z(w)$以 **11** 开头，因为 w 以 b结尾。这与 F的上串以 **10** 开头相矛盾。

假设 w 以 b 和 $|w| > 1$开头。由于 w 以 b结尾，令 $w = bc^r bt$ 其中某个 $r \ge 0$。第 3 节中的游程界给出 $r \le |Ub| - 1$。在开头的 10^β 中

$Z(w)$ 中，恰好有 $r+2$ 个连续的 1：一个位于第一个 b-编码的末尾，r 用于编码中间的 c，另一个位于下一个 b-编码的开头。下一个符号是 0，即使第二个 b 是 w 的最后一个字母也是如此。由于

$$r+2 \le |Ub| + 1 < L,$$

这个 0 已经与 F的上串 $10^\beta 1^L$ 不一致了。不一致之处位于现有的悬垂部分内。

唯一剩下的可能性是 $w = b$。在这种情况下，$Z(w) = 10^\beta$，并且

$$Z(b)\,1^L = 10^\beta 1^L,$$

这恰好是 F的上串。因此，应用 F 会使两个串接相等。

□

现在我们可以为 Neary 的结果给出完整证明了。

定理 1. 在二元字母表上，*PCP(5)* 是不可判定的。

证明。 考虑引理 2中描述的二进制标签系统受限族，其停机问题是不可判定的。给定该族中的一个系统，其删除数为 β，产生式为

$$b \longmapsto b, \qquad c \longmapsto Ub,$$

以及指定的初始串 w_0，构成式（1）中的五个配对。它们的所有字串都是二进制的，并且可由 β 和 U有效计算得到。我们证明，所得的邮编实例有解，当且仅当标签系统从 w_0开始停机。

首先，假设标签计算停机。由引理3可知，其最终字为 b。将该计算写为

$$w_0 \longrightarrow w_1 \longrightarrow \cdots \longrightarrow w_t = b.$$

如第 3 节所示，从 A 开始并附加那里规定的 $\beta - 1$ 删除对后，剩下的下部余串为 $Z(w_0)$。对于每一步 $w_j \longrightarrow w_{j+1}$，附加由 w_j的首个符号确定的产生对，随后附加由其接下来的 $\beta-1$ 个符号确定的 $\beta - 1$ 删除对。引理 6 的证明中的字恒等式表明，这会将下部余串从 $Z(w_j)$变为 $Z(w_{j+1})$。因此，经过全部 t 步后，下部余串为 $Z(b) = 10^\beta$。附加 F 后，上串和下串的拼接相等，因为

$$Z(b)\,1^L = 10^\beta 1^L.$$

因此，这个邮编实例有一个解。

反过来，假设这个邮编实例有一个解。根据引理 4，它直到首次出现 F 为止的前缀形式为（4）。根据引理 5，初始组

$$AD_{1,1} \cdots D_{1,\beta-1}$$

恰好留下下部余串 $Z(w_0)$。

继续处理该因式分解中的其余各组。每当当前悬垂部分为 $Z(w)$，且对应一个可达的非停机词 w时，引理 6便会确定下一个产生对及其 $\beta - 1$ 删除对，并表明所得悬垂部分为 $Z(w')$，其中w' 是 w的真正标签后继。因此，这些组表示从 w_0开始的计算的连续步骤。

以这种方式到达的任何字 b时，标签计算便已停机。否则，由于第一个 F之前只有有限多个组，应用归纳法即可到达紧邻该 F出现位置之前的前缀。其下部余串为 $Z(w)$，其中 w 是一个可达的标签词。引理7 于是迫使 $w = b$成立，因此在这种情况下标签计算也会停机。

因此，我们得到了一个从受限标签系统停机问题到二元字母表上的 PCP(5) 的有效归约。由于前一个问题不可判定，后一个问题也不可判定。

□

5 结论

Neary的构造证明了二元字母表上的 PCP(5) 不可判定。我们的目的是对这一结果进行详细验证，并特别关注逆向推论：每个 PCP 解都会导出模拟的标签系统的一次停机计算。

引理4–7 明确说明了这一推论。零的极大块的长度迫使其具有 (4) 的形式；再与下部余串比较，便可确定初始化以及后续每个标签步骤。最后，只有当所表示的标签词为 b时，才能应用闭合对。

我们使用 Rote 的较长闭合对 [6], 和 $L = |Ub| + 99$，取代 Neary 原始闭合对中连续的四个 1。其所需性质可由可达标签词中 c的块长度上界推出。

受限标签系统族的不可判定性建立在 Neary 的构造之上。邮编验证使用了第 2 节确立的标签系统性质。只有在受限停机问题的证明中才会用到 U 的内部结构。该构造由 Neary 提出，闭合对的修改由 Rote 完成；本文的贡献是详细阐述并验证所得归约的两个方向。

致谢。 感谢 Turlough Neary 多年前在我访问图尔库期间向我讲解他的构造。这次讨论是我对该结果充满信心的重要来源，也促使我撰写本文。

我使用 OpenAI 的 ChatGPT 协助起草文本、修改表述并核查论证细节。我已审阅最终文本，并对其数学内容及任何尚存错误承担全部责任。

参考文献

[1] V. Claus。关于 PCP(k) 及相关问题的一些评论。Bullet. 欧洲理论计算机科学协会，12:54–61，1980。URL: https://page.mi.fu-berlin.de/rote/Kram/Volker_Claus-Some_remarks_on_PCP(k)_and_related_problems-1980-Bull-EATCS-12.pdf.[2] Y. Matiyasevich 和 G. Sénizergues。关于少数规则的半图厄系统的判定问题。Theoret. Comput. Sci.，330(1):145–169，2005。 doi:10.1016/j.tcs.2004.09.016.[3] T. Neary。二进制标签系统的不可判定性以及关于五对字的 Post 对应问题。载于 STACS 2015，LIPIcs 第 30 卷，第 649–661 页，2015 年。doi:10.4230/LIPIcs.STACS.2015.649.[4]Turlough Neary 和 Damien Woods。元胞自动机规则 110 的 P 完全性。技术报告 04/2006，布尔信息学研究中心，科克大学，2006 年。URL: https://dna.hamilton.ie/assets/dw/NearyWoodsBCRI-04-06.pdf.[5] E. L. Post。一个递归不可解问题的变体。Bull. Amer. Math. Soc.，52:264–268，1946。 doi:10.1090/S0002-9904-1946-08555-9.[6] G. Rote。对于某个固定自动机，概率有限自动机空性不可判定。载于 MFCS 2025，LIPIcs 第 345 卷，第 86:1–86:18 页，2025 年。doi:10.4230/LIPIcs.MFCS.2025.86。