

Four-Generator Generalized Post Correspondence and Mortality of Five 3×3 Integer Matrices

Anonymous working draft

21 July 2026

Abstract

Neary reduced an undecidable restricted binary-tag halting problem to five-pair Post correspondence, but the published converse contains a defective terminal-pair argument. We prove directly that the four nonterminal pairs satisfy a fixed-right-boundary word equation exactly when the source tag system halts. The proof uses an exhaustive finite-state synchronization argument and a queue-history invariant; it covers arbitrary prefix-compatible deviations rather than only intended simulations. This yields undecidability of binary-target generalized PCP with four source generators. A fresh-marker construction also gives a corrected binary five-pair PCP family whose primitive solutions end in the fifth pair.

Applying the standard ternary matrix representation and one rank-one boundary separator, we obtain undecidability of mortality for at most five 3×3 integer matrices. Four generators are nonsingular and upper triangular; the fifth is nonzero of rational rank one. The resulting mortality frontier is $M_3(5), M_5(4), M_6(3), M_{12}(2)$. The four-tile source equivalence, fresh-marker repair, and exact integer-matrix equivalence have been formalized in Lean 4 without project axioms or admitted statements.

1 Introduction

For a finite set $\mathcal{A}$ of square matrices, mortality asks whether the zero matrix belongs to the multiplicative semigroup $\mathcal{A}^+$. Write $M_d(k)$ for the restriction to sets of at most k rational $d \times d$ matrices. Restricting inputs to integer matrices does not alter the decidability status. Cassaigne, Halava, Harju, and Nicolas [2] established the undecidability frontier

$$M_3(6), \quad M_5(4), \quad M_9(3), \quad M_{15}(2),$$

and recorded $M_3(5)$ as open. Later surveys retained this bound [1, 3].

The apparent route through Neary’s five-pair PCP theorem [6] has a subtle defect. Its claimed exclusion of the terminal pair uses a false assertion about adjacent encoded symbols. Rote diagnosed this and proposed a longer terminal block [8]. That repair controls intended tag configurations, but its printed argument does not classify all malformed prefix-comparable PCP residuals. We avoid both arguments. Equality of the two PCP words itself forces the four ordinary tiles into exact deletion-width blocks. Decoding those blocks yields a global queue equation, and a generic prefix-cancellation lemma turns that equation into a lawful tag execution, stopping at the first short queue. Post-halting or malformed residuals are therefore irrelevant.

Our source result is the following. Here $\text{GPCP}(k)$ counts letters in the source alphabet of the two morphisms, following Nicolas [7].

Theorem 1.1. *GPCP(4) over a binary target alphabet is undecidable. It remains undecidable with empty left boundaries, one empty right boundary, and nonempty morphism images.*

The matrix consequence uses the established fixed-boundary compiler, whose rank-one form also appears in earlier forced-endpoint constructions [4, 2].

Theorem 1.2. *$M_3(5)$ is undecidable. It remains undecidable under the promise that four generators are nonsingular upper-triangular integer matrices and the fifth is a nonzero integer matrix of rank one over $\mathbb{Q}$.*

Neary’s superseded 2013 preprint [5] announced the same numerical statement through an abandoned four-pair PCP claim. Thus Theorem 1.2 is not the first public statement of $M_3(5)$. No earlier valid proof of either theorem was found in our literature search; any priority claim should remain qualified pending author review.

2 Restricted tag systems and four word pairs

Fix an integer $\beta \geq 3$ and a word $q \in \{b, c\}^*$. Consider the deterministic tag system with deletion number β , rules

$$b \mapsto b, \quad c \mapsto qb,$$

and initial queue

$$I = \text{drop}_{\beta-1}(q)b.$$

A step deletes the first β symbols and appends the rule word selected by the first deleted symbol. The system halts when its queue has length less than β .

Put

$$H(b) = 10^\beta 1, \quad H(c) = 1, \quad M = 10^\beta,$$

and extend H morphically to words. Define four pairs, labelled by a rule symbol and an erase symbol:

$$\begin{aligned} R_c &= (1, 1H(q)10), & R_b &= (10^\beta 1, 110), \\ D_c &= (1, 0), & D_b &= (10^\beta 1, 0). \end{aligned}$$

For a label word w , write $U(w)$ and $V(w)$ for the concatenated upper and lower words.

Theorem 2.1 (Four-tile tag compiler). *Suppose $\beta - 1 \mid |q|$ and $|q| \geq \beta - 1$. Then*

$$\exists w \in \{R_c, R_b, D_c, D_b\}^* : U(w)M = V(w) \iff T_{\beta,q} \text{ halts from } I.$$

The proof occupies the remainder of this section. Its soundness direction is the point missing from the earlier terminal-tile arguments.

2.1 Block forcing

A *stroke* is a word $a_0 a_1 \cdots a_{\beta-1} \in \{b, c\}^\beta$. Associate to it the tile block

$$B(a_0 \cdots a_{\beta-1}) = R_{a_0} D_{a_1} \cdots D_{a_{\beta-1}}.$$

For a stroke history $S = (s_1, \dots, s_n)$, let $C(S) = s_1 \cdots s_n$ be the word consumed by the strokes and let $P(S)$ be the concatenation of the tag appendants selected by their first symbols.

Lemma 2.2 (Synchronization). *If $U(w)M = V(w)$, then $w = B(s_0)B(s_1) \cdots B(s_n)$ for a nonempty stroke history.*

Proof. The word $U(w)M$ begins with 1, every positive run of zeros in it has length exactly β , and it ends with a run of β zeros. Scan the equal word $V(w)$ while retaining only the length of the current zero run. From the initial state an erase tile is impossible. Either rule tile moves the run length from 0 to 1. Each erase tile increases it by one; a further rule tile is compatible only after the count reaches β ; and an erase after β is impossible. Acceptance at the terminal zero run therefore forces one rule tile followed by exactly $\beta - 1$ erase tiles, repeatedly. The labels on their upper words give the corresponding strokes. The empty history cannot satisfy the original equality. $\square$

This scan is a two-shape automaton with states “before the first 1” and “ j zeros since the last 1”, $0 \leq j \leq \beta$. Thus Lemma 2.2 is exhaustive over all label words; it does not presume that a prefix follows the intended tag simulation.

2.2 Queue histories

Appending one final 1 completes the dangling marker: $M1 = H(b)$. Direct expansion of a block history gives

$$U(B(S)) = H(C(S)), \quad (1)$$

$$V(B(s_0)B(S'))1 = H(cQ(\text{head } s_0) b P(S')), \quad (2)$$

where $Q(c) = q$ and $Q(b) = \varepsilon$. Since H is injective, the terminal equation and Equations (1)–(2) first force $\text{head } s_0 = c$. If $s_0 = cz$ with $|z| = \beta - 1$, cancellation then yields

$$z C(S')b = q b P(S'). \quad (3)$$

Both z and q are prefixes of the same word, and $|z| = \beta - 1 \leq |q|$. Hence $z = \text{take}_{\beta-1}(q)$. Cancelling it from (3) gives

$$C(S')b = I P(S'). \quad (4)$$

Lemma 2.3 (History soundness). *Let a tag system have deletion width β . If a stroke history S , initial queue I , and word F with $|F| < \beta$ satisfy*

$$C(S)F = I P(S),$$

then the tag system halts from I .

Proof. Induct on S . For the empty history, $I = F$ is already short. Otherwise, if I is short, stop. If not, the first stroke and I are both prefixes of the common word in the displayed equality, while the stroke has length $\beta \leq |I|$. It is therefore the first β symbols of I . Delete it, append the rule word selected by its head, cancel the common prefix in the global equation, and invoke the induction hypothesis. Notice that the proof deliberately stops if a short queue occurs before the prescribed end of the history. $\square$

Taking $F = b$ in Lemma 2.3 and using (4) proves the soundness direction of Theorem 2.1.

For the converse, record the β deleted symbols of each lawful tag step as a stroke. Every resulting queue ends in b . Moreover,

$$|I| \equiv 1 \pmod{\beta - 1}, \quad |b| \equiv |qb| \equiv 1 \pmod{\beta - 1},$$

so every reachable queue has length congruent to 1 modulo $\beta - 1$. A reachable queue of length less than β must consequently be the one-letter queue b . The execution therefore supplies a history satisfying (4). Prepending the initial stroke $c \text{take}_{\beta-1}(q)$ and reversing the calculation above constructs a terminal match. This completes the proof of Theorem 2.1.

2.3 The undecidable source family

Neary's Lemma 9 [6] proves halting undecidable for exactly these systems. In his notation $\beta = 10p$ and the whole c -appendant qb has length βs . The padding parameter may be chosen as $s = x(\beta - 1) + 1$. Consequently

$$|q| = \beta[x(\beta - 1) + 1] - 1 = (x\beta + 1)(\beta - 1),$$

so the hypotheses of Theorem 2.1 hold, and his input $q_{\beta-1}q_{\beta}\cdots b$ is precisely I . We use Neary's tag-universality lemma here, but no soundness claim from his subsequent five-pair PCP proof.

Proof of Theorem 1.1. Use the four labels as the GPCP source alphabet; use U, V as the two morphisms; take both left boundaries empty, and take M, ε as the right boundaries. The GPCP equation is exactly $U(w)M = V(w)$. The construction is computable and all eight ordinary tile words are nonempty. The empty source word is not a solution because $M \neq \varepsilon$. Theorem 2.1 and Neary's Lemma 9 now give undecidability. $\square$

3 A corrected five-pair PCP source

Although GPCP and mortality need only Theorem 2.1, the same argument repairs the five-pair PCP theorem cleanly. Introduce a fresh symbol $\#$, lift every ordinary tile word from $\{0, 1\}^*$ into $\{0, 1, \#\}^*$, and add

$$(M\#, \#).$$

Finally return to a binary target alphabet with the fixed-length injective code

$$\eta(0) = 00, \quad \eta(1) = 01, \quad \eta(\#) = 11.$$

Theorem 3.1. *The resulting binary five-pair PCP instance is solvable exactly when $T_{\beta,q}$ halts. Every primitive solution contains the fifth pair exactly once, in its final position.*

Proof. The fixed-length code preserves equality and prefix comparability at tile boundaries. Every ordinary upper word ends in 1, whereas every ordinary lower word ends in 0; hence an ordinary-only label word cannot be a PCP solution. Every solution therefore uses the fifth pair. At its first occurrence, prefix comparability requires

$$U(w)M\# \quad \text{and} \quad V(w)\#$$

to be prefix-comparable. Since $\#$ is fresh, words $x\#$ and $y\#$ are prefix-comparable only when $x = y$. Thus $U(w)M = V(w)$, and the prefix through this first terminal tile is already a PCP solution. If a nonempty suffix remained, cancellation would make that suffix another PCP solution, contradicting primitivity. The converse follows by appending the terminal pair to any terminal match and applying η . $\square$

This synchronization replaces the quantitative unary guard proposed by Rote. In particular, Theorem 3.1 remains valid even if malformed ordinary prefixes wander through prefix-comparable residuals after a simulated halt.

4 The matrix compiler

Recode binary letters as nonzero ternary digits, $0 \mapsto 1$ and $1 \mapsto 2$. For a digit word x , let $\sigma(x)$ be its base-3 value, with $\sigma(\varepsilon) = 0$, and define

$$\Psi(x, y) = \begin{pmatrix} 1 & \sigma(y) & \sigma(x) - \sigma(y) \\ 0 & 3^{|y|} & 3^{|x|} - 3^{|y|} \\ 0 & 0 & 3^{|x|} \end{pmatrix}.$$

Lemma 4.1. *For binary-coded words,*

$$\begin{aligned} \Psi(xx', yy') &= \Psi(x, y)\Psi(x', y'), \\ e_1^\top \Psi(x, y) e_3 &= 0 \iff x = y, \\ \det \Psi(x, y) &= 3^{|x|+|y|}. \end{aligned}$$

Proof. The first identity follows from $\sigma(xx') = 3^{|x'|}\sigma(x) + \sigma(x')$ by multiplication. The selected entry is $\sigma(x) - \sigma(y)$. Positional notation over the nonzero digits 1, 2 is injective even between unequal lengths. The determinant is the product of the diagonal entries. $\square$

For the four ordinary labels put $X_i = \Psi(U_i, V_i)$ after ternary recoding, and put

$$X_\star = \Psi(M, \varepsilon), \quad c = X_\star e_3,$$

where words in the latter display are likewise recoded. Define five integer matrices by

$$A_i = X_i \quad (1 \leq i \leq 4), \quad A_\star = X_\star e_3 e_1^\top = c e_1^\top.$$

Each X_i is upper triangular and nonsingular over $\mathbb{Q}$. The third coordinate of c is $3^{|M|}$, so $A_\star$ is nonzero and has rank one.

Lemma 4.2 (Rank-one boundary compiler). *The five emitted matrices are mortal if and only if*

$$\exists w \in \{1, 2, 3, 4\}^* : U(w)M = V(w).$$

Proof. A terminal match gives

$$A_\star X_w A_\star = c(e_1^\top X_w X_\star e_3)e_1^\top = 0$$

by Lemma 4.1.

Conversely, a product without $A_\star$ is nonsingular. Every product containing $t \geq 1$ copies has a unique decomposition

$$P_0 A_\star P_1 A_\star \cdots A_\star P_t,$$

where each P_j is a possibly empty product of ordinary matrices. Expanding the outer products gives

$$(P_0 c) \left(\prod_{j=1}^{t-1} e_1^\top P_j c \right) (e_1^\top P_t).$$

The exterior column and row are nonzero because P_0, P_t are nonsingular. For $t = 1$ the outer product is nonzero. For $t \geq 2$ the product vanishes exactly when some internal scalar vanishes. Writing the corresponding block as $P_j = X_w$, Lemma 4.1 gives

$$e_1^\top P_j c = 0 \iff e_1^\top X_w X_\star e_3 = 0 \iff U(w)M = V(w).$$

This includes empty exterior or interior blocks and adjacent separators. $\square$

The construction is computable. Theorem 2.1 and Neary's undecidable source family therefore prove Theorem 1.2.

5 Frontier consequences

Padding each generator by a zero direct summand preserves mortality. Cassaigne et al.’s trade

$$M_d(hk + 1) \leq_m M_{kd}(h + 1)$$

applied to $M_3(5)$ with $(h, k) = (2, 2)$ and $(1, 4)$ gives

$$M_3(5) \leq_m M_6(3), \quad M_3(5) \leq_m M_{12}(2).$$

Together with their independent $M_5(4)$ result, the minimal known undecidable antichain becomes

$$\boxed{M_3(5), \quad M_5(4), \quad M_6(3), \quad M_{12}(2)}.$$

There are parallel consequences in the notation of [2]. Their $Z_d(k)$ is scalar zero reachability: an instance supplies a row L , a column C , and at most k matrices, and asks whether $LYC = 0$ for some nonempty product Y . Their $R_d(k)$ asks for a zero in the upper-right corner. Theorem 1.1 and their reductions give the antichains

$$Z_3(4), \quad Z_5(3), \quad Z_7(2), \quad R_3(5), \quad R_4(4), \quad R_6(3), \quad R_8(2).$$

6 Formal verification and trust boundary

The accompanying Lean 4 development is pinned to Lean and mathlib version 4.12.0. It formalizes:

1. the exact fixed-width queue semantics and the history-soundness lemma;
2. the pulse automaton and its exhaustive block-forcing theorem;
3. both directions of Theorem 2.1 for every admissible β, q ;
4. the fresh-marker construction, fixed-length binary recoding, solvability, and primitive terminality of Theorem 3.1;
5. the ternary morphism, exact integer matrices, arbitrary-product converse, upper-triangularity, nonsingularity, nonzeroness, and rational rank one;
6. the composed equivalences between tag halting, four-generator GPCP, corrected five-pair PCP, and five-matrix mortality.

The matrix label type is `Option(Fin 4)`, which has exactly five elements. Integer products are cast entrywise to rational matrices; the development proves that this cast preserves products and reflects zero.

The source contains no `sorry`, `admit`, or custom axiom. Lean’s `#print axioms` command reports only `propext`, `Classical.choice`, and `Quot.sound` for the principal theorems. Mathlib is not an additional oracle: its theorem constants carry proof terms rechecked by Lean’s kernel. The trusted base is therefore the Lean kernel and toolchain implementation, together with Lean’s standard classical quotient type theory and the hardware/runtime executing the checker.

The formal artifact does not presently prove Neary’s Lemma 9 from a machine model already known undecidable in mathlib, nor does it formalize the bibliographic frontier reductions. Thus the novel source and matrix equivalences are machine-checked, while the final computability-theoretic wrapper imports Neary’s peer-reviewed tag-universality theorem. Formalizing his Table 2 compiler would remove that remaining external mathematical dependency; adding it as an axiom would merely conceal, rather than reduce, the trust base.

References

- [1] P. C. Bell, I. Potapov, and P. Semukhin. On the mortality problem: From multiplicative matrix equations to linear recurrence sequences and beyond. *Information and Computation*, 281:104736, 2021. <https://doi.org/10.1016/j.ic.2021.104736>.
- [2] J. Cassaigne, V. Halava, T. Harju, and F. Nicolas. Tighter undecidability bounds for matrix mortality, zero-in-the-corner problems, and more. arXiv:1404.0644v3, 2014. <https://arxiv.org/abs/1404.0644>.
- [3] R. Dong. Recent advances in algorithmic problems for semigroups. *ACM SIGLOG News*, 10(4):3–23, 2023. <https://doi.org/10.1145/3636362.3636365>.
- [4] V. Halava, T. Harju, and M. Hirvensalo. Undecidability bounds for integer matrices using Claus instances. *International Journal of Foundations of Computer Science*, 18(5):931–948, 2007. <https://doi.org/10.1142/S0129054107005066>.
- [5] T. Neary. Undecidability in binary tag systems and the Post correspondence problem for four pairs of words. arXiv:1312.6700v1, 2013. Superseded claim. <https://arxiv.org/abs/1312.6700>.
- [6] T. Neary. Undecidability in binary tag systems and the Post correspondence problem for five pairs of words. In *STACS 2015*, LIPIcs 30, pages 649–661, 2015. <https://doi.org/10.4230/LIPIcs.STACS.2015.649>.
- [7] F. Nicolas. (Generalized) Post correspondence problem and semi-Thue systems. arXiv:0802.0726v5, 2008. <https://arxiv.org/abs/0802.0726>.
- [8] G. Rote. Probabilistic finite automaton emptiness is undecidable for a fixed automaton. In *MFCS 2025*, LIPIcs 345, article 86, 2025. <https://doi.org/10.4230/LIPIcs.MFCS.2025.86>.