A MONOTONE MEASURE FOR PATTERN-LANGUAGE INCLUSION: EVERY MORPHIC INCLUSION, EVERY PATTERN OF LENGTH AT MOST EIGHT, AND THE INCLUSION DEPTH IN THAT RANGE

KOREA SUPERINTELLIGENCE LABS

ABSTRACT. Luo's open-problem note from COLT 2005, reissued in 2026 as arXiv:2605.30389, asks for the *inclusion depth* $\mathrm{ID}_\Sigma(p)$ of a pattern p — the length of the longest strict chain of pattern-language inclusions joining $L_\Sigma(x_1)$ to $L_\Sigma(p)$ — and asks whether $\mathrm{ID}_\Sigma(p) = 2|p| - \#\mathrm{var}(p) - 1$. It reduces that formula to one monotonicity statement, its Conjecture 1: if $L(p) \subset L(q)$ then $2|p| - \#\mathrm{var}(p) > 2|q| - \#\mathrm{var}(q)$. The note reports that its author's program verified the inequality for patterns of length at most 7 and that longer patterns were "computationally impractical". Write $\mu(r) = 2|r| - \#\mathrm{var}(r)$. We prove that $\mu(\sigma(q)) \geq \mu(q)$ for every non-erasing substitution σ of patterns for variables and every pattern q, at every length, and that equality forces σ to act on the variables of q as an injective renaming, whose inverse substitution the proof produces; hence Conjecture 1 holds for every inclusion realised by a morphism, with no restriction on length. That is a genuine partial result: we exhibit a strict inclusion $L(001x_1x_2) \subsetneq L(x_1x_2x_3x_2)$ over $\{0,1\}$ realised by no morphism — a phenomenon credited to Angluin, and not new here. Independently, we verify Conjecture 1 for every pair of patterns of length at most 8 over a two-letter alphabet — one length beyond the note's own computation, so that any counterexample must involve a pattern of length at least 9. The verification is an exhaustive check over 115,974 canonical patterns, run behind a proof that the check quantifies over *all* patterns and not merely over the enumerated representatives. As a consequence the upper half of the note's formula becomes unconditional in that range: every strict chain from $L(x_1)$ to $L(p)$ through patterns of length at most 8 has at most $2|p| - \#\mathrm{var}(p) - 1$ steps, which together with the note's own displayed chain pins $\mathrm{ID}_\Sigma(0x_11) = 4$. Every theorem below is machine-checked in Lean 4.

1. INTRODUCTION

Patterns, their languages, and inclusion depth. A *pattern* is a finite non-empty word over the disjoint union of an alphabet Σ of terminals and a countable set $X = \{x_1, x_2, \dots\}$ of variables, and its language is the set of terminal words obtained by replacing every variable by a non-empty terminal word, the same word at each occurrence. Introduced by Angluin [1], pattern languages are one of the standard test objects of algorithmic learning theory, and their inclusion problem is undecidable [4, 5].

Luo's note [8] attaches to a pattern a quantity measuring how far its language sits below the top of the inclusion order. We quote its definition verbatim ([8], Definition 1):

> The *inclusion depth* of a pattern p with alphabet Σ, denoted by $\mathrm{ID}_\Sigma(p)$, is the length of the longest strict inclusion chain connecting the language of the universal pattern $L_\Sigma(x_1)$ and the language $L_\Sigma(p)$.

The motivation is learning-theoretic: inclusion depth is intended to capture the mind-change complexity of identifying a pattern from positive data, and a computable inclusion depth would yield a mind-change bound for that problem and a uniformly mind-change optimal learner ([8], §2; see [9]). The note poses two questions. Question 1 asks whether $\mathrm{ID}_\Sigma(p)$ is computable at all, and if so in polynomial time. Question 2 asks, verbatim, "Is it true that, for every alphabet Σ with at least two symbols, $\mathrm{ID}_\Sigma(p) = 2|p| - \#\mathrm{var}(p) - 1$, where $|p|$ is the length, namely the number of variables and constants in p, and $\#\mathrm{var}(p)$ is the number of distinct variables in p?" A positive answer to Question 2 answers Question 1 with a linear-time algorithm.

The conjecture, and what was known about it. The note reduces Question 2 to a single monotonicity statement, which we quote verbatim ([8], Conjecture 1):

> Let p and q be two patterns. If $L(p) \subset L(q)$, then $2|p| - \#\mathrm{var}(p) > 2|q| - \#\mathrm{var}(q)$.

Throughout we write

$$\mu(r) \;=\; 2|r| - \#\mathrm{var}(r),$$

so the conjecture reads: a proper inclusion of pattern languages strictly increases μ. The reference to Σ is dropped for the reason the note gives ([8], §3): if $L_\Sigma(p) \subseteq L_\Sigma(q)$ then $L_{\Sigma'}(p) \subseteq L_{\Sigma'}(q)$ for every $\Sigma' \subseteq \Sigma$, so only $|\Sigma| = 2$ need be considered. We work throughout with $\Sigma = \{0, 1\}$.

About the conjecture the note proves nothing. Its entire evidence is one sentence, which we quote verbatim because it is the boundary this paper moves ([8], §3):

> Exhaustive computation by the author's program showed that Equation (2) holds for patterns of length less than or equal to 7; computation for longer patterns was computationally impractical. Therefore, if two patterns p and q form a counterexample to Equation (2), one of them must be longer than 7.

"Equation (2)" there is the displayed inequality of Conjecture 1 above: in the source of v1 it is the second and last of the two numbered equations, the first being the formula of Question 2. The number is the note's own, and is unrelated to the equation numbers of this paper.

For ID itself the note offers one worked value. It displays the chain

$$L_\Sigma(x_1) \supset L_\Sigma(x_1x_2) \supset L_\Sigma(x_11) \supset L_\Sigma(x_1x_21) \supset L_\Sigma(0x_11), \tag{1}$$

remarks that "it is routine, though tedious, to verify that there exists no longer inclusion chain connecting the two", and concludes $\mathrm{ID}_\Sigma(0x_11) = 4$. The lower half of that sentence is the displayed chain; the upper half is asserted without argument.

Results.

- *A monotone measure for morphic inclusions, at every length* (Theorems 3.1, 3.3 and 3.4). Call an inclusion $L(p) \subseteq L(q)$ *morphic* when it is realised syntactically, that is when $p = \sigma(q)$ for a non-erasing substitution σ of patterns for the variables of q; such an inclusion always holds, and this is the standard sufficient condition for inclusion of pattern languages (Lemma 2.2). We prove that μ never decreases along it: $\mu(\sigma(q)) \geq \mu(q)$ for every non-erasing σ and every q. Equality forces σ to send every variable of q to a single variable, injectively, and the proof produces the inverse substitution ρ with $\rho(\sigma(q)) = q$, whence $L(\sigma(q)) = L(q)$. Consequently Conjecture 1 holds for every morphic inclusion, with no bound on the length of either pattern. The argument is elementary counting and uses no computation.
- *The morphic hypothesis is a real restriction* (Proposition 3.5). Over $\Sigma = \{0, 1\}$ the inclusion $L(001x_1x_2) \subsetneq L(x_1x_2x_3x_2)$ is strict, and no non-erasing substitution maps $x_1x_2x_3x_2$ to $001x_1x_2$. So Theorem 3.4 is a partial result and not the conjecture in disguise. That non-morphic non-erasing inclusions exist at all is credited to Angluin; the short witness above is a control, not a new phenomenon.
- *The conjecture at length at most* 8 (Theorem 4.4). For all patterns p, q over a two-letter alphabet with $|p| \leq 8$ and $|q| \leq 8$, a proper inclusion $L(p) \subset L(q)$ forces $\mu(q) < \mu(p)$. With the note's own alphabet reduction (quoted above; we do not re-prove it) this is Conjecture 1 itself in that range, so a counterexample must involve a pattern of length at least 9; the note's own sentence said "longer than 7". This claim rests on exhaustive computation, described precisely in §4.
- *The upper half of Question 2, unconditionally at length at most* 8 (Theorem 5.3). Every strict inclusion chain $L(x_1) \supset \cdots \supset L(p)$ whose members are patterns of length at most 8 has at most $2|p| - \#\mathrm{var}(p) - 1$ steps. The length hypothesis on the intermediate patterns is vacuous once $|p| \leq 8$ (Lemma 5.2). Combined with the chain (1) this turns the note's "routine, though tedious" into a proof for its own example: $\mathrm{ID}_\Sigma(0x_11) = 4$ (Corollary 5.4).

Supporting statements are of a different kind and are labelled as such: the standard sufficient condition for inclusion (Lemma 2.2, credited to Angluin and to Jiang, Kinber, Salomaa, Salomaa and Yu), the note's own membership example and chain (Propositions 2.1 and 5.1), negative controls bounding what the theorems say (Proposition 4.6), and the soundness layer under the exhaustive check (Lemmas 4.1, 4.2 and 4.3).

What is not claimed. We do not prove Conjecture 1. We do not answer Question 2: the matching lower bound $\mathrm{ID}_\Sigma(p) \geq 2|p| - \#\mathrm{var}(p) - 1$ — that a chain of that length always exists — is not proved here for general p, and the note asserts it only for its one example. We do not decide inclusion of pattern languages anywhere; that problem is undecidable in general [4, 5] and remains so over a two-letter alphabet [11], and every negative statement below is instead certified by an explicit witness word. We claim no novelty for the existence of non-morphic inclusions, nor for the value $\mathrm{ID}_\Sigma(0x_1 1) = 4$, which the note computes; what is new there is that the upper half now has a proof.

Provenance. The quotations above are from the arXiv source of [8], retrieved on 3 September 2026; that version is the author's own reissue of a two-page open-problem note published in the COLT 2005 proceedings, and its bibliographic data were confirmed against three independent records. The conjecture appears nowhere else that we could find. A bibliographic database reports exactly five works citing the 2005 note; all five were read or checked, and none attacks or resolves the conjecture, nor reports any verification beyond the note's own. The one published text that engages with inclusion depth at all is Freydenberger's thesis [3], which was read end to end: its Proposition 7.4 shows, in the *erasing* setting and for the reduced superpatterns of a language, that each such pattern has only finitely many strict superpatterns among them, the argument being the length bound $|\beta| \leq |\alpha|$ alone; the side note following that proposition defines the inclusion depth, observes that it is therefore finite, and cites [8] for the non-erasing case. No numeric bound is given there. The refinement $\mu(p) = 2|p| - \#\mathrm{var}(p)$ is exactly what turns that finiteness into a bound. A full-text search of a large snapshot of arXiv sources matched 1,902 distinct papers on pattern-language terms; sixteen are genuinely about pattern languages in Angluin's sense, the nine most relevant were read in full, and none contains a measure of a pattern that is monotone under non-erasing pattern substitution. "Inclusion depth" in this sense occurs in exactly one paper of that snapshot, namely [8] itself.

Two limits on this provenance should be stated plainly. First, [1] and [4] could not be obtained: both publishers refused the full text to us. Every statement credited to them below is credited as [3] credits it — that thesis is the version deposited on its university's dissertation server, and its numbering was read off that deposit — and no numbered result of [1] or [4] is quoted here. Second, the printed 2005 note could not be inspected either, the publisher's chapter page being behind a bot challenge. Its title, sole author, venue, pages and DOI agree across Crossref, DBLP and the bibliography of [3], and [8] itself opens with a publication note declaring it the author's version of that note; but we have not compared the printed text word for word, so every quotation in this paper is attributed to the arXiv version v1 and to no other.

2. Preliminaries

Patterns and their languages. We reproduce the objects of [8] verbatim ([8], §1):

> Let X be a set of variables, for example $x_1, x_2, \ldots$, and let Σ be a finite alphabet containing at least two symbols, for example $\{0, 1\}$. A *pattern*, denoted by p, q, and so on, is a finite non-empty sequence over $X \cup \Sigma$. The language of a pattern p with alphabet Σ, denoted by $L_\Sigma(p)$, is the set of ground strings that are consequences of p by substituting each variable in p with a string in Σ^+. For example, if $\Sigma = \{0, 1\}$, then the strings 010 and 10110 are in $L_\Sigma(x_1 x_2 x_1)$, but $1010 \notin L_\Sigma(x_1 x_2 x_1)$.

Two conventions are fixed by that paragraph and both are load-bearing. The substitution is *non-erasing*: images lie in Σ^+, never empty. And it is a *morphism*: every occurrence of a variable receives the same word. The non-example 1010 is exactly what pins the second convention, since $1010 = 10 \cdot 1 \cdot 0$ would be a member if the two occurrences of x_1 could be filled differently.

Proposition 2.1. *With $\Sigma = \{0, 1\}$: $010 \in L(x_1x_2x_1)$, $10110 \in L(x_1x_2x_1)$, and $1010 \notin L(x_1x_2x_1)$.*

Proof. Take $x_1 \mapsto 0,\ x_2 \mapsto 1$ and $x_1 \mapsto 10,\ x_2 \mapsto 1$ for the first two. For the third, a substitution τ with $\tau(x_1)\tau(x_2)\tau(x_1) = 1010$ has $2|\tau(x_1)| + |\tau(x_2)| = 4$ with both images non-empty, so $|\tau(x_1)| = 1$; then $\tau(x_1)$ is both the first and the last letter of 1010, which differ. □

Fix $\Sigma = \{0, 1\}$ from here on, as [8] permits. For a pattern r write $|r|$ for its length and $\#\mathrm{var}(r)$ for the number of distinct variables occurring in it, and set $\mu(r) = 2|r| - \#\mathrm{var}(r)$. Since $\#\mathrm{var}(r) \leq |r|$ we have $\mu(r) \geq |r| \geq 1$, so all the inequalities below may equivalently be read in the additive form that avoids subtraction, and that is the form in which they are proved: for instance $\mu(p) > \mu(q)$ is $2|p| + \#\mathrm{var}(q) > 2|q| + \#\mathrm{var}(p)$.

Substitutions of patterns, and morphic inclusions. A *pattern substitution* is a map σ from variables to patterns, extended to a morphism on patterns by fixing the terminals; it is *non-erasing* when every $\sigma(x)$ is non-empty. Composing substitutions is again a substitution, which gives at once the standard sufficient condition for inclusion.

Lemma 2.2. *For every non-erasing pattern substitution σ and every pattern q, $L(\sigma(q)) \subseteq L(q)$.*

Proof. If $w = \tau(\sigma(q))$ for a non-erasing ground substitution τ, then $w = (\tau \circ \sigma)(q)$ and $x \mapsto \tau(\sigma(x))$ is again non-erasing, because a non-erasing ground substitution maps a non-empty pattern to a non-empty word. □

Lemma 2.2 is classical — it is the direction of the inclusion criterion for pattern languages that holds without restriction. We take it in the form of Theorem 3.5 of [3], which reads: *Let Σ be an alphabet and $\alpha, \beta \in \mathrm{Pat}_\Sigma$. If there is a terminal-preserving morphism $\varphi : (\Sigma \cup X)^* \to (\Sigma \cup X)^*$ with $\varphi(\beta) = \alpha$, then $L_{E,\Sigma}(\alpha) \subseteq L_{E,\Sigma}(\beta)$. If φ is also nonerasing, then $L_{NE,\Sigma}(\alpha) \subseteq L_{NE,\Sigma}(\beta)$.* It is the second sentence that is Lemma 2.2. [3] credits that theorem to [6] and to [1], in that order; note that [6] is the five-author paper, not the four-author Jiang–Salomaa–Salomaa–Yu papers [4, 5] cited elsewhere here. Call an inclusion $L(p) \subseteq L(q)$ *morphic* when $p = \sigma(q)$ for some non-erasing pattern substitution σ. The converse of Lemma 2.2 fails: non-erasing inclusions realised by no such σ exist, a fact [3] credits to [1], and Proposition 3.5 below exhibits a short one.

3. The measure under substitution

This section proves Conjecture 1 for every morphic inclusion, with no restriction on length, and determines exactly when it is tight.

Theorem 3.1. *Let σ be a non-erasing pattern substitution and q a pattern. Then*

$$2|q| + \#\mathrm{var}(\sigma(q)) \ \leq\ 2|\sigma(q)| + \#\mathrm{var}(q), \qquad \textit{that is,} \qquad \mu(\sigma(q)) \ \geq\ \mu(q).$$

Proof. For a variable x occurring in q write $k_x \geq 1$ for the number of its occurrences and $\ell_x = |\sigma(x)| \geq 1$. Splitting the positions of q into variable and terminal positions, and writing c for the number of terminal positions,

$$|q| = c + \sum_x k_x, \qquad |\sigma(q)| = c + \sum_x k_x \ell_x,$$

both sums being over the *distinct* variables of q; terminals contribute 1 to each and cancel. The variables occurring in $\sigma(q)$ are those occurring in the images $\sigma(x)$, so

$$\#\mathrm{var}(\sigma(q)) \ \leq\ \sum_x \#\mathrm{var}(\sigma(x)) \ \leq\ \sum_x \ell_x, \qquad \#\mathrm{var}(q) = \sum_x 1.$$

Substituting these four identities, the asserted inequality follows by summing over the distinct variables of q the pointwise inequality

$$2k+\ell \;\le\; 2k\ell+1 \qquad (k,\ell \ge 1), \tag{2}$$

which is $(2k-1)(\ell-1) \ge 0$. $\square$

Remark 3.2. The sums must be over distinct variables, not over positions. Bounding $\#\mathrm{var}(\sigma(q)) \le \sum_{s\in q}\#\mathrm{var}(\sigma(s))$ position-wise is true but too weak: for $q = x_1x_1$ and $\sigma(x_1) = x_2$ it gives 2 where the correct value is 1, and the resulting inequality is false.

Theorem 3.3. *Let σ be a non-erasing pattern substitution and q a pattern, and suppose that equality holds in Theorem 3.1, that is $2|\sigma(q)| + \#\mathrm{var}(q) \le 2|q| + \#\mathrm{var}(\sigma(q))$. Then there is a non-erasing pattern substitution ρ with $\rho(\sigma(q)) = q$. Consequently $L(\sigma(q)) = L(q)$.*

Proof. Equality in the proof of Theorem 3.1 forces equality in (2) at every distinct variable x of q, and since $k_x \ge 1$ this gives $\ell_x = 1$: each $\sigma(x)$ is a single symbol. Hence σ acts on q as a symbol map T, with $\sigma(q) = T(q)$ letterwise and T the identity on terminals. Equality also forces $\#\mathrm{var}(\sigma(q)) = \#\mathrm{var}(q)$. Write V for the set of distinct variables of q; the variables of $\sigma(q)$ are precisely those elements of $T(V)$ that are variables, so

$$\#\mathrm{var}(q) \;=\; \#\mathrm{var}(\sigma(q)) \;\le\; |T(V)| \;\le\; |V| \;=\; \#\mathrm{var}(q),$$

and both inequalities are equalities. The second says T is injective on V; the first says every element of $T(V)$ is a variable. Define ρ on a variable y by $\rho(y) = x$ if $y = T(x)$ for a (then unique) variable x of q, and $\rho(y) = y$ otherwise; ρ is non-erasing and $\rho(\sigma(q)) = q$ symbol by symbol. The last sentence is Lemma 2.2 applied to σ and to ρ in turn. $\square$

Theorem 3.4. *Conjecture 1 of* [8] *holds for every morphic inclusion, at every length: if σ is a non-erasing pattern substitution, q a pattern, and the inclusion $L(\sigma(q)) \subseteq L(q)$ of Lemma 2.2 is* strict*, then*

$$\mu(q) \;<\; \mu(\sigma(q)).$$

Equivalently: if $L(p) \subsetneq L(q)$ and $p = \sigma(q)$ for some non-erasing σ, then $2|p| - \#\mathrm{var}(p) > 2|q| - \#\mathrm{var}(q)$.

Proof. By Theorem 3.1, $\mu(\sigma(q)) \ge \mu(q)$. If equality held then Theorem 3.3 would give $L(\sigma(q)) = L(q)$, contradicting strictness. $\square$

Neither theorem uses any computation, and neither restricts the alphabet, the lengths, or the number of variables.

The morphic hypothesis is a restriction.

Proposition 3.5. *Over $\Sigma = \{0,1\}$, put $p = 001x_1x_2$ and $q = x_1x_2x_3x_2$. Then*

$$L(p) \subsetneq L(q), \qquad \textit{no non-erasing } \sigma \textit{ satisfies } \sigma(q) = p, \qquad \mu(p) = 8 > 5 = \mu(q).$$

Proof. Inclusion. A word of $L(p)$ is $001u$ with $|u| \ge 2$; write $u = u'c$ with c its last letter. If $c = 0$, then $001u'0 = (0)(0)(1u')(0)$ is of the form $u_1u_2u_3u_2$ with all four blocks non-empty. If $c = 1$, then $001u'1 = (00)(1)(u')(1)$ is again of that form, u' being non-empty. *Strictness.* $1010 = (1)(0)(1)(0) \in L(q)$, while every word of $L(p)$ begins with 001. *No morphism.* Suppose $\sigma(q) = p$ with σ non-erasing. Then $|\sigma(x_1)| + 2|\sigma(x_2)| + |\sigma(x_3)| = 5$ with all three images non-empty, so $|\sigma(x_2)| = 1$ and $|\sigma(x_1)| \in \{1,2\}$. But $\sigma(x_2)$ is both the last symbol of p, namely x_2, and the symbol of p at position $|\sigma(x_1)|+1$, namely 0 or 1 according to the two cases — a terminal either way, and never the variable x_2. *Measures.* $\mu(p) = 2\cdot 5 - 2$ and $\mu(q) = 2\cdot 4 - 3$. $\square$

Proposition 3.5 is the non-vacuity control for Theorem 3.4: the morphic hypothesis excludes something, so Theorem 3.4 is not a restatement of Conjecture 1. The *existence* of non-morphic non-erasing inclusions is not new: it is Proposition 3.7 of [3], credited there to [1], and asserts that for every finite alphabet there are patterns α with one variable and β with two such that $L_{NE}(\alpha) \subseteq L_{NE}(\beta)$ while no non-erasing terminal-preserving morphism maps β to α. The proof given there builds α from the alphabet, so for $|\Sigma| = 2$ its witness is the pair $\alpha = a_1 x a_2 a_1 x x a_2$, $\beta = xxy$, of lengths 7 and 3. The pair above is a shorter instance found by exhaustive search, and is recorded here only for the role just described.

Remark 3.6 (a census, computed outside the formal development). The following counts were produced by a stand-alone program and are *not* part of the machine-checked development; nothing below depends on them. For ordered pairs (p, q) of canonical patterns over $\Sigma = \{0, 1\}$ with $|q| \le |p| \le L$ and $p \ne q$, classify each pair as *witnessed* (a word of $L(p) \setminus L(q)$ was found, so there is no inclusion), *morphic* ($\sigma(q) = p$ for some non-erasing σ), or neither:

L	pairs	witnessed	morphic	neither
4	32,139	29,442	2,697	0
5	621,215	596,076	25,003	136
6	14,120,246	13,866,352	248,298	5,596

No pair is both witnessed and morphic in any row, which is the census's internal consistency check. Two readings. Every inclusion with $|p| \le 4$ over this alphabet is morphic, and the shortest non-morphic ones have $|p| = 5$ — the pair of Proposition 3.5 is one of 136. And at $|p| \le 6$ the morphic criterion covers 248,298 of the 253,894 candidate inclusions, about 97.8 %; the remaining 2.2 % measures the distance between Theorem 3.4 and Conjecture 1.

4. The conjecture for patterns of length at most eight

Why the finite check is certificate-shaped. Inclusion of pattern languages is undecidable in general [4, 5], and — what matters here, because we work over a fixed two-letter alphabet — it is undecidable already for terminal-free non-erasing patterns over an alphabet of two terminals ([11], Corollary 12). So no program can decide the hypothesis $L(p) \subset L(q)$ of Conjecture 1, and a finite verification cannot proceed by deciding inclusion pair by pair. It does not have to. A counterexample is a pair with $\mu(p) \le \mu(q)$ *and* $L(p) \subsetneq L(q)$; to eliminate a pair it therefore suffices either to observe $\mu(q) < \mu(p)$, or to observe $L(p) = L(q)$, or to exhibit a single word

$$w \in L(p) \setminus L(q),$$

which refutes inclusion outright. Both halves of such a witness are finite and checkable: $w \in L(p)$ is witnessed by the substitution that produced it, and $w \notin L(q)$ by the absence of any way of cutting w into $|q|$ non-empty blocks compatible with q — terminals carrying their own letter and equal variables carrying equal blocks. The check below never decides inclusion; it only ever refutes it, and every refutation carries its certificate.

The enumeration. Renaming variables does not change a pattern's language or its measure, so it is enough to sweep patterns in *canonical form*, with variables numbered $x_1, x_2, \ldots$ in order of first occurrence. The number of canonical patterns of length L over a two-letter alphabet is $\sum_k \binom{L}{k} 2^{L-k} B_k$ with B_k the Bell numbers:

length L	1	2	3	4	5	6	7	8
canonical patterns	3	10	37	151	674	3,263	17,007	94,828

so 115,974 canonical patterns of length at most 8, counting the empty one. The enumerator used in the verification was checked against this closed form.

Three soundness statements carry the check. They are supporting lemmas, not new mathematics, but the theorem below means what it says only because of them.

Lemma 4.1. *The non-membership certificate is sound: if the enumeration of decompositions of w into $|q|$ non-empty blocks contains no block sequence compatible with q, then $w \notin L(q)$. Likewise the tabulated form of that certificate over all words of a fixed length is sound, and every word listed as a one-letter substitution instance of p does lie in $L(p)$.*

Proof sketch. The content is completeness of the block enumerator: if $w = \tau(q)$ for a non-erasing τ, then the block sequence obtained by cutting w at the images of the symbols of q is a decomposition into $|q|$ non-empty blocks that is compatible with q, so it is enumerated and the certificate fails. The enumerator bounds the first block by $|w| - (|q| - 1)$, which is exactly the room the remaining non-empty blocks need, and completeness survives that bound. □

Lemma 4.2. *Canonicalisation — relabelling the variables of p in order of first occurrence — changes neither the length, nor the language, nor the measure of p; and every pattern of length at most n has its canonical form in the enumeration of canonical patterns of length at most n.*

Proof sketch. Canonicalisation is a renaming, exhibited together with its inverse, so Lemma 2.2 applied in both directions gives equality of languages; equality of measures then follows from Theorem 3.1 applied in both directions together with the equality of lengths. Completeness of the enumeration is an induction on p carrying the list of variables seen so far. □

Lemma 4.3. *If the check described below returns "true" for the bound N, then Conjecture 1 holds for every pair of patterns of length at most N — arbitrary patterns, not merely the canonical representatives enumerated.*

Proof sketch. Given arbitrary p, q with $L(p) \subset L(q)$, replace them by their canonical forms, which by Lemma 4.2 lie in the enumeration and have the same languages and measures. The check tests, for that pair, a disjunction of three conditions. The first is the conclusion $\mu(q) < \mu(p)$. The second says the two canonical forms coincide, which contradicts strictness. The third produces a word of $L(p)$ certified outside $L(q)$ by Lemma 4.1, contradicting $L(p) \subseteq L(q)$. □

Theorem 4.4. *Let p and q be patterns over a two-letter alphabet with $|p| \leq 8$ and $|q| \leq 8$. If $L(p) \subset L(q)$ then $2|p| - \#\mathrm{var}(p) > 2|q| - \#\mathrm{var}(q)$. Equivalently: any counterexample to Conjecture 1 involves a pattern of length at least 9.*

Proof sketch. By Lemma 4.3 it suffices to run the check at $N = 8$, and this is the one claim of this paper that rests on exhaustive computation. The search is the following, and nothing else. For each word length $n \leq 8$ and each of the 115,974 canonical patterns q of length at most 8, tabulate over all 2^n words of length n whether the non-membership certificate of Lemma 4.1 succeeds. For each canonical pattern p, list the words obtained from p by substituting a single letter for each variable — there are $2^{\#\mathrm{var}(p)} \leq 256$ of them, all of length $|p|$, and these are exactly the shortest words of $L(p)$. Then test every ordered pair of canonical patterns of length at most 8, that is $115{,}974^2 \approx 1.34 \times 10^{10}$ pairs, for the disjunction of Lemma 4.3. The check returns "true".

Two features of the run are worth stating because they bound what it shows. First, the only refutation device the check owns is a one-letter substitution instance of p: it never searches longer witnesses. That the run succeeds therefore also says that in this range every pair needing a witness admits one of that shape. Second, no pruning by length or by measure is applied: all ordered pairs are tested, including those where the conclusion is immediate. □

Remark 4.5 (an independent reproduction, outside the formal development). Before the verified run, the same claim was checked by a separate program written directly from the definitions of §2, using the complementary pruning — only pairs with $|q| \leq |p|$, $\mu(p) \leq \mu(q)$ and $p \neq q$, of which there are 204,391,773 at length at most 7 and 6,262,426,110 at length at most 8. All were refuted, in 13.8 and 441.9 seconds of processor time respectively; the first of those two figures reproduces the note's own reported computation independently. Every one of the 6.26×10^9 pairs was refuted by a one-letter substitution witness, 6,224,301,344 of them by one of the two constant substitutions $x \mapsto 0$ and

$x \mapsto 1$; the program's escalation to two- and three-letter images was never reached. These figures are measurements, not theorems, and Theorem 4.4 does not rest on them.

Length 9 was priced and not attempted. The candidate-pair count grows by a factor of about 30 per length in this range, putting length 9 near 2.2×10^{11} pairs, about 4.3 processor-hours for the stand-alone program and about 20 for the verified run.

Controls.

Proposition 4.6. (1) (The inequality cannot be strengthened.) *$L(x_1x_2) \subsetneq L(x_1)$ and $\mu(x_1x_2) = \mu(x_1) + 1$, so Conjecture 1 cannot be sharpened to $\mu(p) \geq \mu(q) + 2$.*
(2) (The converse fails.) *$\mu(00) = 4 > 2 = \mu(1)$, yet $L(00) \not\subseteq L(1)$.*

Proof. (1) The inclusion is the first step of the chain (1), proved in Proposition 5.1; the measures are 2 and 1. (2) $L(1) = \{1\}$ and $00 \in L(00)$. □

Item (1) says the gap in Conjecture 1 is attained, so the conjecture is sharp; item (2) says the implication is strictly one-way, so μ orders the languages only in the direction claimed. Both are recorded because a monotonicity statement with no such controls can be satisfied by a misplaced quantifier.

5. Inclusion depth in the verified range

Proposition 5.1. *All four inclusions of the chain* (1) *displayed in* [8] *are strict:*

$$L(x_1) \supsetneq L(x_1x_2) \supsetneq L(x_11) \supsetneq L(x_1x_21) \supsetneq L(0x_11),$$

and the measures along it are $1, 2, 3, 4, 5$. *Hence* $\mathrm{ID}_\Sigma(0x_11) \geq 4$, *and* $2|0x_11| - \#\mathrm{var}(0x_11) - 1 = 4$.

Proof. Each inclusion is morphic, realised respectively by the substitutions $x_1 \mapsto x_1x_2$ (applied to x_1); $x_1 \mapsto x_1$, $x_2 \mapsto 1$ (applied to x_1x_2); $x_1 \mapsto x_1x_2$ (applied to x_11); and $x_1 \mapsto 0$, $x_2 \mapsto x_1$ (applied to x_1x_21), so Lemma 2.2 gives the containments. Strictness in each case is a one-word separation: $0 \in L(x_1) \setminus L(x_1x_2)$ on length grounds; $00 \in L(x_1x_2) \setminus L(x_11)$, whose words end in 1; $01 \in L(x_11) \setminus L(x_1x_21)$ on length grounds; and $101 \in L(x_1x_21) \setminus L(0x_11)$, whose words begin with 0. The measures are $2-1$, $4-2$, $4-1$, $6-2$, $6-1$. □

The chain and the value $\mathrm{ID}_\Sigma(0x_11) = 4$ are the note's own ([8], §1); only their verification is ours. Note that each of its four steps raises μ by exactly 1, which is Proposition 4.6(1).

Lemma 5.2. *If $L(p) \subseteq L(q)$ then $|q| \leq |p|$.*

Proof. Substituting a single letter for every variable of p produces a word of $L(p)$ of length exactly $|p|$; every word of $L(q)$ has length at least $|q|$, since a non-erasing substitution never shortens. □

Theorem 5.3. *Let p be a pattern and let $L(x_1) \supsetneq L(r_1) \supsetneq \cdots \supsetneq L(r_m) = L(p)$ be a strict inclusion chain, all of whose members $x_1, r_1, \ldots, r_m$ are patterns of length at most* 8. *Then*

$$m \;\leq\; 2|p| - \#\mathrm{var}(p) - 1.$$

Proof. By Theorem 4.4, each strict step of the chain raises μ by at least 1, so $\mu(x_1) + m \leq \mu(p)$ by induction along the chain. Since $\mu(x_1) = 1$ this is $m \leq \mu(p) - 1 = 2|p| - \#\mathrm{var}(p) - 1$. □

Corollary 5.4. *For every pattern p with $|p| \leq 8$ over a two-letter alphabet, every strict inclusion chain from $L(x_1)$ to $L(p)$ has at most $2|p| - \#\mathrm{var}(p) - 1$ steps, so $\mathrm{ID}_\Sigma(p) \leq 2|p| - \#\mathrm{var}(p) - 1$. In particular*

$$\mathrm{ID}_\Sigma(0x_11) \;=\; 4.$$

Proof. Every member r of a chain descending to $L(p)$ has $L(p) \subseteq L(r)$, so $|r| \leq |p| \leq 8$ by Lemma 5.2, and the hypothesis of Theorem 5.3 is automatic. For $p = 0x_11$ the upper bound is 4 and the lower bound is the chain of Proposition 5.1. □

This is the upper half of Question 2 of [8], unconditional for $|p| \leq 8$. The note poses the equality as a question and proves no bound in either direction; the only published engagement with inclusion depth we could find [3] proves finiteness only, in the erasing setting and from the length alone (its Proposition 7.4), and gives no numeric bound. For its own example the note calls the upper half "routine, though tedious"; the value $\mathrm{ID}_\Sigma(0x_1 1) = 4$ is the note's, and what is added here is a proof of the half it asserted.

6. What remains open

Conjecture 1 itself. Theorem 3.4 settles it for the morphic inclusions at every length, and by Remark 3.6 those are about 98 % of the candidate inclusions at $|p| \leq 6$ but not all of them. What is missing is a structural description of the non-morphic non-erasing inclusions over a binary alphabet. The examples credited to Angluin show they exist; we know of no invariant that handles them.

Length 9 *and beyond.* Theorem 4.4 moves the exhaustive frontier by one length, and the cost of the next step is given in Remark 4.5. The obvious lever is the block of pairs with $|p| = |q|$. We expect an inclusion between patterns of equal length to be forced to be morphic, since a substitution that does not lengthen must send every variable to a single symbol; if so, that whole block is already covered by Theorem 3.4 and could be dropped from the search, removing its dominant term. We have neither proved that nor carried the reduction out.

The lower half of Question 2. Corollary 5.4 is one inequality; $\mathrm{ID}_\Sigma(p) \geq 2|p| - \#\mathrm{var}(p) - 1$, that a chain of that length always exists, is not proved here for general p. Building such a chain uniformly — deleting a terminal, splitting a variable — looks routine and would, with Corollary 5.4, settle Question 2 for $|p| \leq 8$.

Question 1. Computability of ID_Σ is untouched. No route through "decide inclusion" is available here: inclusion of pattern languages is undecidable in general [4, 5]; it remains undecidable for a fixed terminal alphabet of at least two letters in the erasing setting and at least four in the non-erasing one ([3], Theorem 3.3; see also [2]); and over a two-letter alphabet it is undecidable even for terminal-free non-erasing patterns ([11], Corollary 12), which is the case that applies to us. Every negative statement in this paper is therefore certificate-shaped by necessity rather than by preference.

7. Formal verification

Every lemma, proposition, theorem and corollary stated above is formalised and machine-checked in Lean 4 [7] against Mathlib [10]; the exceptions, labelled as such in the text, are Remarks 3.2, 3.6 and 4.5, the counts of canonical patterns in §4 (the enumerator is verified complete, but its agreement with the closed form is a computation), and the bibliographic and search statements of §1. A pattern is encoded as a list of symbols, each symbol being either a variable index or one of the two terminals, and a word as a list of bits; the language, the measure and Conjecture 1 are transcribed from the quotations in §2; the membership example of Proposition 2.1 is checked first, as a guard against transcribing the erasing convention or a non-morphic substitution by mistake. There is no incomplete proof. Propositions 2.1, 3.5, 4.6 and 5.1, Lemmas 2.2, 4.1, 4.2, 4.3 and 5.2, and Theorems 3.1, 3.3 and 3.4 depend only on the standard axioms of the system, with no compiled evaluation — which is the formal counterpart of their holding at every length. Theorem 4.4, and through it Theorem 5.3 and Corollary 5.4, additionally depend on compiled evaluation, through thirteen such axioms, one per chunk of the search of §4; the search was split into chunks only so that no single evaluation ran too long, and Lemma 4.3, which is what makes the search mean the theorem, uses none of them. The whole search costs about 43 minutes of processor time — the figure timed on the three modules after the final run, against about 35 predicted from the components; the excess is the cost of compiling and linking each goal separately — and under a gigabyte of memory per chunk. One step of Corollary 5.4 — the observation that Lemma 5.2 makes the length hypothesis of Theorem 5.3 automatic — is a one-line argument combining two machine-checked statements and is not itself machine-checked. The repository is private; repository reference to be added at submission.

References

[1] D. Angluin, *Finding patterns common to a set of strings*, Journal of Computer and System Sciences **21** (1980), no. 1, 46–62; DOI 10.1016/0022-0000(80)90041-0 (title, authorship, volume, issue, pages and year confirmed against Crossref on 3 September 2026). Not read directly; see the caveat in §1.

[2] D. D. Freydenberger and D. Reidenbach, *Bad news on decision problems for patterns*, Information and Computation **208** (2010), no. 1, 83–96; DOI 10.1016/j.ic.2009.04.002 (volume, issue, pages and year confirmed against Crossref on 3 September 2026, and agreeing with the reference lists of [3] and [11]). A conference version appeared in Developments in Language Theory (DLT 2008), Lecture Notes in Computer Science **5257**, Springer, 2008, pp. 327–338; DOI 10.1007/978-3-540-85780-8_26. Not read directly.

[3] D. D. Freydenberger, *Inclusion of Pattern Languages and Related Problems*, dissertation, Fachbereich Informatik und Mathematik, Johann Wolfgang Goethe-Universität, Frankfurt am Main, 2011, 167 pp. Read in full on 3 September 2026 in the copy carried by the university's dissertation server; all numbering cited here (Theorems 2.3, 3.3, 3.4 and 3.5, Proposition 3.7, Proposition 7.4) was read off that copy.

[4] T. Jiang, A. Salomaa, K. Salomaa and S. Yu, *Inclusion is undecidable for pattern languages*, in: Automata, Languages and Programming (ICALP 1993), Lecture Notes in Computer Science **700**, Springer, 1993, pp. 301–312; DOI 10.1007/3-540-56939-1_81 (authorship, title, series volume, pages, year and DOI confirmed against Crossref and DBLP on 3 September 2026). Not read directly.

[5] T. Jiang, A. Salomaa, K. Salomaa and S. Yu, *Decision problems for patterns*, Journal of Computer and System Sciences **50** (1995), no. 1, 53–63; DOI 10.1006/jcss.1995.1006 (confirmed against Crossref on 3 September 2026). Not read directly.

[6] T. Jiang, E. Kinber, A. Salomaa, K. Salomaa and S. Yu, *Pattern languages with and without erasing*, International Journal of Computer Mathematics **50** (1994), no. 3–4, 147–163; DOI 10.1080/00207169408804252 (confirmed against Crossref on 3 September 2026). Not read directly; this is the paper that [3] cites, alongside [1], for its Theorem 3.5.

[7] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction — CADE 28, Lecture Notes in Computer Science **12699**, Springer, 2021, pp. 625–635; DOI 10.1007/978-3-030-79876-5_37 (confirmed against Crossref and DBLP on 3 September 2026).

[8] W. Luo, *The Inclusion Depth of Pattern Languages: An Open Problem in Algorithmic Learning Theory*, arXiv: 2605.30389v1 (28 May 2026), cs.FL, cross-listed cs.LG. Author-prepared version of *Compute Inclusion Depth of a Pattern*, in: Learning Theory: 18th Annual Conference on Learning Theory (COLT 2005), Lecture Notes in Computer Science **3559**, Springer, 2005, pp. 689–690; DOI 10.1007/11503415_48. All quotations in this paper are from the arXiv source of v1, retrieved on 3 September 2026, which is the only version; the title, authorship, series volume, pages and DOI of the 2005 note were confirmed on the same day against Crossref, against DBLP, and against the reference list of [3], which cites it as entry [66]. The arXiv metadata title is the longer one above, while the title inside the source is the original one. The printed note itself was not read; see §1.

[9] W. Luo and O. Schulte, *Mind change efficient learning*, Information and Computation **204** (2006), 989–1011; DOI 10.1016/j.ic.2006.02.004 (confirmed against Crossref on 3 September 2026). A conference version appeared in COLT 2005, Lecture Notes in Computer Science **3559**, Springer, 2005, pp. 398–412; DOI 10.1007/11503415_27.

[10] The mathlib Community, *The Lean mathematical library*, in: Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs (CPP 2020), ACM, 2020, pp. 367–381; DOI 10.1145/3372885.3373824 (title, corporate authorship, venue, pages and year confirmed against Crossref on 3 September 2026).

[11] A. Saarela, *Hardness results for constant-free pattern languages and word equations*, in: 47th International Colloquium on Automata, Languages, and Programming (ICALP 2020), Leibniz International Proceedings in Informatics **168**, Schloss Dagstuhl, 2020, article 140, 15 pp.; DOI 10.4230/LIPIcs.ICALP.2020.140. Read in full on 3 September 2026 in the open-access version. Its Corollary 12 is the statement quoted in §4 and §6: the inclusion problem for constant-free (that is, terminal-free) non-erasing pattern languages is undecidable over an alphabet of two terminals.