THE UNARY MULTIPLE-CONCATENATION BOUND IS NOT TIGHT, AND FAILS ONLY IN ITS CYCLE LENGTH

KOREA SUPERINTELLIGENCE LABS

Abstract. Jirásek and Jirásková (arXiv:2511.03814) close their study of the state complexity of the concatenation $L_1L_2\cdots L_k$ of k regular languages with the unary case, and give an upper bound for unary automata that may carry final states in their tails: if A_i has size (λ_i,μ_i) then $L(A_1)\cdots L(A_k)$ is recognised by a unary automaton of size (Λ, M) with $\Lambda = \mathrm{lcm}(\lambda_1,\dots,\lambda_k)$ and M a maximum, over subsets I of the indices, of $\sum_i \mu_i - k + 1 + d_I f(I)$, where d_I is the greatest common divisor of the λ_i with $i \in I$ and f is the modified Frobenius number. They exhibit one instance in which the bound is met — sizes $(12, 2)$, $(20, 2)$, $(30, 2)$, minimal size $(60, 124)$, 184 states — and write that "the tightness of this upper bound in a general case remains open". We settle it: the bound is *not* tight. At $k = 3$ with cycle lengths $(3, 2, 2)$ and tails $(2, 2, 2)$ the bound is $(6, 12)$, that is 18 states, and yet for *every* one of the 8192 choices of final-state sets the concatenation is recognised by a unary automaton with at most 15 states, because the cycle length $\mathrm{lcm}(3, 2, 2) = 6$ is never attained. The failure is confined to the cycle: at those same sizes the preperiod half $M = 12$ of the bound is exactly right, forced by a single witness. We also exhibit two size tuples at which the whole bound *is* met, both far smaller than the published example: $(2, 4, 4)$ with tails $(2, 2, 2)$ gives minimal size exactly $(4, 12)$, 16 states, and $(1, 4, 6)$ gives exactly $(12, 16)$, 28 states — the latter with globally coprime cycle lengths and with the maximum defining M attained on a proper pair of indices. An exhaustive census over all 216 cycle-length triples with $\lambda_i \le 6$ and $\mu_i = 2$ finds no violation of the bound, finds its preperiod half attained every time, and singles out a candidate closed form for the true cycle length, which we state as a conjecture. Finally we settle, in the unary case and for every k, a second remark the same paper leaves open: the concatenation of $k - 1$ unary languages of at most two states each with a one-state language is recognised by a k-state automaton, and k states are necessary. All theorems below are machine-checked in Lean 4.

1. Introduction

The quantity and the source. The *state complexity* of a regular language is the number of states of its minimal deterministic finite automaton. For the concatenation of two languages the answer is classical [6, 9]; for the concatenation $L_1L_2\cdots L_k$ of k languages the problem was raised by Caron, Luque and Patrou [1] and is the subject of the recent paper of Jirásek and Jirásková [3], which describes witness languages meeting the upper bound over alphabets of various sizes and proves that a ternary alphabet is optimal for $k = 3$.

Its last section treats the *unary* case, where the alphabet is a single letter a. Unary state complexity has a different flavour from the general case: the minimal automaton is a "lasso", concatenation becomes addition, and number theory — the Frobenius coin problem — enters the bounds [8]; the average behaviour of the operations on unary automata is different again [7]. For two unary languages the case is the subject of Pighizzini and Shallit [8], on which [3] leans throughout its unary section. For k languages, [3] proves tight bounds for cyclic unary languages (its Theorem 26) and for languages recognised by automata with no final state in the tail (its Corollary 27), and then gives an upper bound in the remaining, general case (Theorem 2.1 below). About that last bound it says, verbatim:

> We cannot obtain a tight upper bound here, nevertheless, we provide an example that meets our upper bound.

and, after the example,

> The tightness of this upper bound in a general case remains open.

The conclusions repeat the point, as "The tightness of this upper bound remains open." The paper's single example uses unary automata of sizes $(12, 2)$, $(20, 2)$, $(30, 2)$; the minimal automaton of the concatenation has size $(60, 124)$, that is, 184 states.

What is settled here. The bound is not tight. Section 4 exhibits a size tuple at which no choice of final states meets it, and the gap is not marginal: the bound asks for 18 states and the truth is at most 15, for all 8192 instances at once (Theorem 4.1). What fails is precisely the cycle length: the bound predicts a cycle of length $\mathrm{lcm}(3, 2, 2) = 6$, and 6 is never reached. The other half of the bound survives the counterexample. The preperiod M is built from the modified Frobenius number, and at those same sizes it is exactly right: a single explicit witness needs a tail of length 12 in every unary automaton that recognises it, and 12 is the value the bound predicts (Theorem 4.2).

Section 5 goes the other way. The published example is the only instance in [3] where the bound is met, and it gives no criterion for when that happens. We add two instances, both much smaller. At cycle lengths $(2, 4, 4)$ with tails $(2, 2, 2)$ the bound $(4, 12)$ is met exactly, on 16 states (Theorem 5.1). At cycle lengths $(1, 4, 6)$ the bound $(12, 16)$ is met exactly, on 28 states (Theorem 5.2); there the cycle lengths are globally coprime, $\gcd(1, 4, 6) = 1$, and the maximum defining M is attained not at the full index set — as in the published example, where $\gcd(12, 20, 30) = 2$ — but at the proper pair $\{2, 3\}$. Both are shapes the published example does not exhibit.

Section 6 reports the exhaustive census that located these parameters, and states the conjecture it suggests: when every $\mu_i \geq 1$, the largest attainable cycle length is

$$\Lambda^* \;=\; \max\bigl\{\, \mathrm{lcm}_{i\in I}\, \lambda_i \;:\; \emptyset \neq I \subseteq \{1, \dots, k\},\; \gcd_{i\in I} \lambda_i \geq 2 \,\bigr\}$$

(and 1 when no such I exists). This matched in every case computed. Proving it would replace $\Lambda = \mathrm{lcm}(\lambda_1, \dots, \lambda_k)$ in Theorem 2.1 by the largest attainable cycle length; the preperiod half is attained throughout the window we searched. That would make each coordinate of the bound sharp, but not yet the pair: at some sizes the two are never attained together, and Section 9 exhibits one.

Section 7 settles, in the unary case and for every k, a second question left open in [3]: if $A_1, \dots, A_{k-1}$ have at most two states each and A_k has one, is $L(A_1) \cdots L(A_k)$ recognised by a k-state automaton? Over one letter the answer is yes, by an elementary argument, and k states are necessary. The remark in [3] is stated over a general alphabet, and that case is not proved here; Remark 7.2 records an exhaustive verification of it for $k \leq 6$ over every alphabet.

Section 8 records one more computation, on a different conjecture of [3]: that k symbols are necessary to describe witnesses for the concatenation of k languages, which [3] proves only for $k = 3$. At $k = 4$ with all component automata of two states, a ternary alphabet reaches 35 while the bound is 37, and the four-letter construction of [3] reaches 37; so four symbols are necessary in that case.

Section 10 describes the formal verification. Every theorem and proposition below is machine-checked in Lean 4; the statements labelled as remarks are the ones that are not, and each of them says so. One elementary step is carried out on paper rather than in Lean, and that section says which.

2. Preliminaries

Unary automata as eventually periodic sets. Fix the one-letter alphabet $\{a\}$ and identify the word a^n with the natural number n, so that a unary language is a subset of $\mathbb{N} = \{0, 1, 2, \dots\}$ and concatenation is the *sumset*

$$L_1 L_2 \;=\; \{\, m + n \;:\; m \in L_1,\; n \in L_2 \,\}.$$

Following [3], which in turn follows [8], a unary automaton of *size* (λ, μ) with $\lambda \geq 1$ has state set $q_0, \dots, q_{\mu-1}$ (the tail) and $p_0, \dots, p_{\lambda-1}$ (the cycle) with transitions

$$q_0 \to q_1 \to \cdots \to q_{\mu-1} \to p_0 \to p_1 \to \cdots \to p_{\lambda-1} \to p_0,$$

so $\lambda + \mu$ states in all. We number the states $0, 1, \ldots, \lambda + \mu - 1$ in that order and record the final states as a subset $F \subseteq \{0, \ldots, \lambda + \mu - 1\}$; this is the convention used for F_1, F_2, F_3 in the example of [3] reproduced below. The state reached by a^n is n if $n < \mu + \lambda$ and $\mu + ((n - \mu) \bmod \lambda)$ otherwise.

Say that $L \subseteq \mathbb{N}$ is *eventually periodic with preperiod m and period p* when

$$\forall n \geq m: \quad n \in L \iff n + p \in L.$$

The dictionary we use throughout is [8, Theorem 2], as quoted by [3]: L is recognised by a unary automaton of size (λ, μ) if and only if $\lambda \geq 1$ and L is eventually periodic with preperiod μ and period λ. Consequently no automaton interface is needed below: every statement is arithmetic on $\mathbb{N}$.

We say that a nonempty L has *minimal size exactly* (λ_0, μ_0) when L is recognised by an automaton of size (λ_0, μ_0) and every unary automaton of size (λ, μ) recognising L satisfies $\lambda_0 \mid \lambda$ and $\mu_0 \leq \mu$. This is the form in which all the exactness claims below are stated and verified; it is stronger than minimality of the state count $\lambda + \mu$, and it is what makes "the bound is met" a claim about both coordinates separately.

The Frobenius numbers and the source's bound. For positive integers $n_1, \ldots, n_\ell$ with $\gcd(n_1, \ldots, n_\ell) = 1$, let $g(n_1, \ldots, n_\ell)$ be the Frobenius number, the largest integer not representable as $x_1 n_1 + \cdots + x_\ell n_\ell$ with all $x_j \geq 0$, and let

$$f(n_1, \ldots, n_\ell) \;=\; g(n_1, \ldots, n_\ell) + n_1 + \cdots + n_\ell$$

be the *modified* Frobenius number, which [3] glosses as the largest integer not representable by *positive* integer linear combinations. Thus $f(1) = 0$ and $f(1, 1) = 1$, while $f(2, 3) = 1 + 5 = 6$ and $f(3, 5) = 7 + 8 = 15$.

Theorem 2.1 ([3, Theorem 29])**.** *For $i = 1, \ldots, k$ let A_i be a unary automaton of size (λ_i, μ_i). For each nonempty $I = \{i_1, \ldots, i_\ell\} \subseteq \{1, \ldots, k\}$ put $d_I = \gcd(\lambda_{i_1}, \ldots, \lambda_{i_\ell})$ and $f(I) = f(\lambda_{i_1}/d_I, \ldots, \lambda_{i_\ell}/d_I)$, and set $d_\emptyset = 1$, $f(\emptyset) = 0$. Then $L(A_1)L(A_2)\cdots L(A_k)$ is recognised by an automaton of size (Λ, M) where*

$$\Lambda = \operatorname{lcm}(\lambda_1, \ldots, \lambda_k), \qquad M = \max_{I \subseteq \{1, \ldots, k\}} \Big(\mu_1 + \cdots + \mu_k - k + 1 + d_I \cdot f(I)\Big).$$

We follow [3] in saying that this bound is *met* at a tuple of sizes $(\lambda_1, \mu_1), \ldots, (\lambda_k, \mu_k)$ when there are automata of exactly those sizes whose concatenation has minimal size exactly (Λ, M), and that the bound is *tight* when it is met at every tuple of sizes. This is the reading [3] itself fixes: its Theorem 26 and its Corollary 27 each end with the words "and this upper bound is tight", and each is proved tight by exhibiting a witness family for *every* tuple of sizes — languages $L_i = a^{n_i - 1}(a^{n_i})^*$ in Theorem 26 and $L_i = a^{\mu_i + \lambda_i - 1}(a^{\lambda_i})^*$ in Corollary 27.

Two special values are used repeatedly. At $k = 3$ with all $\mu_i = 2$ the additive constant is $\mu_1 + \mu_2 + \mu_3 - k + 1 = 4$, so $M = 4 + \max_I d_I f(I)$. And for a singleton $I = \{i\}$ one has $d_I = \lambda_i$ and $f(I) = f(1) = 0$, so singletons never contribute.

Three lemmas. Everything below rests on three elementary facts, all proved in full generality (no computation) in the formal development.

Lemma 2.2. *If A is eventually periodic with preperiod α and period p, and B with preperiod β and period q, then the sumset $A + B$ is eventually periodic with preperiod $\alpha + \beta + \operatorname{lcm}(p, q)$ and period $\operatorname{lcm}(p, q)$.*

This preperiod is much larger than the one Theorem 2.1 provides; it is used only as a *coarse* starting point, from which a finite window check upgrades to the sharp statement:

Lemma 2.3. *Let L be eventually periodic with preperiod Γ and period $\Lambda \geq 1$, let $m \leq \Gamma$, and let $p \geq 0$. If $n \in L \iff n + p \in L$ for every n in the finite window $[\,m, \Gamma + \Lambda)$, then L is eventually periodic with preperiod m and period p.*

Lemma 2.4. *Let L be eventually periodic with preperiod Γ and period $\Lambda \geq 1$. If L is also eventually periodic with period p and some preperiod, then L is eventually periodic with preperiod Γ and period $\gcd(p, \Lambda)$.*

Lemma 2.4 is what turns a handful of refutations into a statement about *all* automata: to show that every eventual period of L is a multiple of Λ, it suffices to refute the maximal proper divisors of Λ, since any period p with $\gcd(p, \Lambda) \neq \Lambda$ has $\gcd(p, \Lambda)$ dividing one of them. Likewise, if L is Λ-periodic from Γ on and is recognised by an automaton of size (λ, μ), then L is Λ-periodic from μ on; so a single failure of Λ-periodicity at some $n < \mu_0$ shows that every recognising automaton has tail at least μ_0.

3. The published example, re-derived

Before the new parameters, the one instance printed in [3] was recomputed from the definitions, as a check on the model and on the reading of the size and final-state conventions.

Proposition 3.1. *Let A_1, A_2, A_3 be the unary automata of sizes $(12, 2)$, $(20, 2)$, $(30, 2)$ with $F_1 = \{0, 13\}$, $F_2 = \{0, 21\}$, $F_3 = \{0, 31\}$. Then $L(A_1)L(A_2)L(A_3)$ has minimal size exactly $(60, 124)$: it is recognised by an automaton of size $(60, 124)$, every eventual period of it is a multiple of 60, and every unary automaton recognising it has tail at least 124.*

Proof sketch. Each $L(A_i)$ is eventually periodic with preperiod 2 and period λ_i; two applications of Lemma 2.2 make the sumset 60-periodic from 126 on, with no computation. Lemma 2.3 with $\Gamma = 126$, $\Lambda = 60$, $m = 124$ then reduces recognisability by $(60, 124)$ to checking $n \in L \iff n + 60 \in L$ for the 62 values $n \in [124, 186)$, which is an exhaustive finite check. For the tail, one evaluation suffices: $123 \notin L$ and $183 \in L$ differ, so L is not 60-periodic from 123 on, and by the transport described after Lemma 2.4 no recognising automaton has tail below 124. For the cycle, the maximal proper divisors of 60 are 30, 20 and 12; each is refuted by a single evaluation ($n = 126$, 128, 132 respectively), and Lemma 2.4 upgrades the three refutations to: every eventual period is a multiple of 60. The arithmetic $\Lambda = \operatorname{lcm}(12, 20, 30) = 60$ and $M = 2+2+2-3+1+\max\{60, 120\} = 124$ printed in [3], and the intermediate values $4f(3, 5) = 6f(2, 5) = 10f(2, 3) = 60$ and $2f(6, 10, 15) = 120$, were recomputed independently and agree. $\square$

4. The bound is not tight

Take $k = 3$, cycle lengths $(\lambda_1, \lambda_2, \lambda_3) = (3, 2, 2)$ and tails $\mu_1 = \mu_2 = \mu_3 = 2$. Theorem 2.1 gives $\Lambda = \operatorname{lcm}(3, 2, 2) = 6$ and, since

$$d_{\{1,2\}} f(\{1, 2\}) = f(3, 2) = 6, \quad d_{\{2,3\}} f(\{2, 3\}) = 2f(1, 1) = 2, \quad d_{\{1,2,3\}} f(\{1, 2, 3\}) = f(3, 2, 2) = 8,$$

with the singletons contributing 0, the maximum is 8 and $M = 4 + 8 = 12$. So the bound predicts a minimal size of $(6, 12)$, that is 18 states. The three automata have $3+2 = 5$, $2+2 = 4$ and $2+2 = 4$ states, so there are $2^5 \cdot 2^4 \cdot 2^4 = 8192$ choices of final-state sets in all.

Theorem 4.1. *Let A_1 have size $(3, 2)$ and A_2, A_3 size $(2, 2)$, with arbitrary final-state sets $F_1 \subseteq \{0, \ldots, 4\}$ and $F_2, F_3 \subseteq \{0, \ldots, 3\}$. Then $L(A_1)L(A_2)L(A_3)$ is eventually periodic with preperiod 12 and period 2, or with preperiod 12 and period 3. Consequently it is recognised by a unary automaton with at most 15 states, for every one of the 8192 choices of final-state sets, whereas the bound of Theorem 2.1 at these sizes is $(6, 12)$, that is 18 states. In particular the cycle length $\operatorname{lcm}(3, 2, 2) = 6$ is never attained, and the upper bound of Theorem 2.1 is not tight.*

Proof sketch. For every choice of final-state sets, two applications of Lemma 2.2 make the concatenation 6-periodic from 18 on; this half is general and involves no computation. Lemma 2.3 with $\Gamma = 18$, $\Lambda = 6$, $m = 12$ then reduces "2-periodic from 12 on" and "3-periodic from 12 on" to finite checks on the window $[12, 24)$, twelve values each. The claim is that at least one of the two checks succeeds, and this is verified by exhaustive computation over all 8192 triples of final-state sets: for each triple, the membership test $n \in L(A_1)L(A_2)L(A_3)$ is recomputed from the sumset definition for

$n \le 26$ and the two window checks are run. This is the only exhaustive search the theorem depends on; the rest of the argument is the two general lemmas. Given the conclusion, recognisability by a unary automaton of size $(2, 12)$ or $(3, 12)$ — 14 or 15 states — is immediate from the dictionary of Section 2. □

Theorem 4.1 refutes tightness by a gap of at least three states, uniformly over all instances at those sizes. It also isolates *where* the bound goes wrong: the predicted cycle length is 6 and the true one is at most 3. The next theorem shows that the other coordinate is not merely close but exactly right at the same sizes.

The choice $k = 3$ is not incidental. At $k = 2$ the maximum in Theorem 2.1 is always attained at the full index set: writing $d = \gcd(\lambda_1, \lambda_2)$, Sylvester's value $f(a, b) = ab$ for coprime a, b gives $d \cdot f(\lambda_1/d, \lambda_2/d) = \lambda_1\lambda_2/d = \mathrm{lcm}(\lambda_1, \lambda_2)$, so the bound is always $(\mathrm{lcm}(\lambda_1, \lambda_2),\ \mu_1+\mu_2-1+\mathrm{lcm}(\lambda_1, \lambda_2))$, a total of $2\,\mathrm{lcm}(\lambda_1, \lambda_2) + \mu_1 + \mu_2 - 1$ states. Over the whole window $\lambda_i \le 8$, $\mu_i \le 4$ the census of Section 6 measured the largest total state count in the cases with $\gcd(\lambda_1, \lambda_2) = 1$ as $\lambda_1\lambda_2 + \mu_1 + \mu_2$, smaller by $\lambda_1\lambda_2 - 1$: at $k = 2$ the bound is already not tight. That is no surprise, and we claim no priority for it — the two-language unary case, automata with final states in their tails included, is the subject of the results of [8] that [3] cites for exactly this point, namely that the cycle of a two-language unary concatenation can be the least common multiple of the two given cycles when the automata have final states in their tails. (The value $\lambda_1\lambda_2 + \mu_1 + \mu_2$ quoted above is our own measurement over that window, not a quotation from [8].) What Theorem 2.1 introduces, and what [3] records as open, is the case $k \ge 3$ with final states in the tails; that is where Theorem 4.1 sits.

Theorem 4.2. *At the sizes of Theorem 4.1, take $F_1 = \{4\}$, $F_2 = \{3\}$, $F_3 = \{3\}$. Then every unary automaton recognising $L(A_1)L(A_2)L(A_3)$ has tail length at least 12. Since Theorem 4.1 recognises this language with tail exactly 12, its minimal preperiod is exactly $12 = M$: the preperiod half of the bound of Theorem 2.1 is attained at these sizes, and the failure of the bound is confined to its cycle length.*

Proof sketch. Write L for the concatenation at this choice of final states. A single evaluation shows $11 \notin L$ and $17 \in L$ differ, so L is not 6-periodic from 11 on. Suppose L were recognised by an automaton of size (λ, μ) with $\mu \le 11$. Then L is λ-periodic from μ on and, by Lemma 2.2, 6-periodic from 18 on; the transport described after Lemma 2.4 yields 6-periodicity from $\mu \le 11$ on, a contradiction. The finite input to the argument is the single membership pair $(11, 17)$, evaluated from the sumset definition. □

5. Two size tuples at which the bound is met

The example of [3] reproduced in Proposition 3.1 is the only instance in that paper where the bound is met, and no criterion is offered for when this happens. Two more instances follow, both an order of magnitude smaller in state count. They were located by the census of Section 6 and then verified exactly.

Theorem 5.1. *Let A_1 have size $(2, 2)$ with $F_1 = \{0, 3\}$, and let A_2, A_3 have size $(4, 2)$ with $F_2 = F_3 = \{5\}$. Then $L(A_1)L(A_2)L(A_3)$ has minimal size exactly $(4, 12)$, which is the bound of Theorem 2.1 at these sizes: 16 states, against the 184 states of the published example.*

Proof sketch. At cycle lengths $(2, 4, 4)$ and tails $(2, 2, 2)$ the bound reads $\Lambda = \mathrm{lcm}(2, 4, 4) = 4$ and $M = 4 + \max\{\,2f(1, 2),\ 4f(1, 1),\ 2f(1, 2, 2)\,\} = 4 + \max\{4, 4, 8\} = 12$. Lemma 2.2 makes the concatenation 4-periodic from 14 on, without computation. Lemma 2.3 with $\Gamma = 14$, $\Lambda = 4$, $m = 12$ reduces recognisability by $(4, 12)$ to a check on the six values $n \in [12, 18)$. For minimality of the tail, one evaluation: $11 \notin L$ and $15 \in L$ differ, so L is not 4-periodic from 11 on, and as in Theorem 4.2 no recognising automaton has tail below 12. For minimality of the cycle, the only maximal proper divisor of 4 is 2, refuted by the single evaluation at $n = 14$; Lemma 2.4 then gives that every eventual period is a multiple of 4. □

Theorem 5.2. *Let A_1 have size $(1,2)$ with $F_1 = \{1\}$, let A_2 have size $(4,2)$ with $F_2 = \{0,5\}$, and let A_3 have size $(6,2)$ with $F_3 = \{0,7\}$. Then $L(A_1)L(A_2)L(A_3)$ has minimal size exactly $(12,16)$, which is the bound of Theorem 2.1 at these sizes: 28 states.*

Proof sketch. At cycle lengths $(1,4,6)$ and tails $(2,2,2)$ the bound reads $\Lambda = \mathrm{lcm}(1,4,6) = 12$ and, from $f(1,4) = 4$, $f(1,6) = 6$, $2f(2,3) = 12$ and $f(1,4,6) = 10$, the maximum is 12 and $M = 4 + 12 = 16$. Lemma 2.2 makes the concatenation 12-periodic from 22 on; Lemma 2.3 with $\Gamma = 22$, $m = 16$ reduces recognisability by $(12,16)$ to a check on the eighteen values $n \in [16,34)$. The tail is minimal because $15 \notin L$ and $27 \in L$ differ, so L is not 12-periodic from 15 on. The cycle is minimal because the maximal proper divisors of 12 are 6 and 4, refuted by single evaluations at $n = 22$ and $n = 28$; Lemma 2.4 upgrades these to: every eventual period is a multiple of 12. □

Theorem 5.2 is worth separating from Theorem 5.1 for two reasons. First, $\gcd(1,4,6) = 1$: the bound can be met even when the cycle lengths are globally coprime, whereas the example of [3] has $\gcd(12,20,30) = 2$. Second, the maximum defining M is attained at the proper pair $I = \{2,3\}$, where $d_I = \gcd(4,6) = 2$ and $f(2,3) = 6$ give $d_I f(I) = 12$, and *not* at the full index set, which contributes only 10; in the example of [3] the maximum sits at the full index set. So the subset structure of the maximum in Theorem 2.1 is genuinely used.

6. The census, and a conjecture for the true cycle length

The parameters of Sections 4 and 5 were not guessed. The following is a computation carried out outside the formal development, by an independent program in C that recomputes the sumset from the definitions and never uses the formula of Theorem 2.1 except to compare against it. It is reported here as measurement, not as a theorem.

Remark 6.1 (an exhaustive census at $k = 3$; computation outside the formal development). For every one of the 216 cycle-length triples with $\lambda_i \le 6$, with all tails $\mu_i = 2$, and for every one of the $2^{\lambda_1+2} \cdot 2^{\lambda_2+2} \cdot 2^{\lambda_3+2}$ choices of final-state sets — 128 024 064 instances in total, 6.9 seconds of processor time — the exact minimal size of the concatenation was computed. The findings were: (a) the bound of Theorem 2.1 is never violated, an independent check of that theorem over the window; (b) the preperiod half M is attained at every one of the 216 triples, so Theorem 4.2 is not an accident of its parameters; (c) the full pair (Λ, M) is attained at exactly 12 cycle-length multisets, namely the six constant ones $\{1,1,1\}, \ldots, \{6,6,6\}$ — where all three cycle lengths agree, so that the bound of Theorem 2.1 coincides with the one [3, Corollary 27] already states to be tight — and the six further multisets $\{2,4,4\}$, $\{2,4,6\}$, $\{1,4,6\}$, $\{2,6,6\}$, $\{3,3,6\}$, $\{3,6,6\}$; Theorems 5.1 and 5.2 are the two smallest of the six. A companion run at $k = 2$ over all 1600 size pairs with $\lambda_i \le 8$ and $\mu_i \le 4$ produced the two-language measurements quoted in Section 4.

Conjecture 6.2. Let A_i be unary automata of sizes (λ_i, μ_i) with $\mu_i \ge 1$ for every i. Then the largest cycle length attained by $L(A_1) \cdots L(A_k)$ over all choices of final-state sets is

$$\Lambda^* \;=\; \max\bigl\{\, \mathrm{lcm}_{i\in I}\, \lambda_i \;:\; \emptyset \ne I \subseteq \{1,\ldots,k\},\ \gcd_{i\in I} \lambda_i \ge 2 \,\bigr\},$$

and $\Lambda^* = 1$ when no such I exists.

The evidence is the census: Λ^* matched the measured largest attainable cycle length in all 216 cases at $k = 3$ in the window above, and in all 1024 size pairs at $k = 2$ with $\lambda_i \le 8$ and $1 \le \mu_i \le 4$. The hypothesis $\mu_i \ge 1$ cannot be dropped, and what goes wrong is already visible in the source: for cyclic languages, that is $\mu_i = 0$ throughout, [3, Theorem 26] makes the cycle of the concatenation exactly $\gcd(\lambda_1, \ldots, \lambda_k)$, so at $\lambda = (2,3)$ with $\mu = (0,0)$ the true cycle is 1 while $\Lambda^* = 3$. Of the 576 two-language size pairs in that window with some $\mu_i = 0$, the formula fails in 296.

Conjecture 6.2 explains both halves of the picture above. It gives $\Lambda^* = 3$ at $(3,2,2)$, matching Theorem 4.1; and it gives $\Lambda^* = \mathrm{lcm}(\lambda_1, \ldots, \lambda_k)$ exactly when some subset with gcd ≥ 2 already has the full least common multiple, which holds at $(2,4,4)$ and at $(1,4,6)$ — Theorems 5.1 and 5.2.

A mechanism is visible in the proof of Theorem 2.1 in [3], which decomposes the concatenation as a union over subsets I of $\left(\prod_{j\notin I} X_j\right)\left(\prod_{i\in I} a^{\mu_i} Y_i\right)$: the piece indexed by I carries cycle d_I, and a piece with $d_I = 1$ is eventually all of $\mathbb{N}$, collapsing everything after it to period 1. Only subsets with $d_I \geq 2$ can carry a long cycle. Proving Conjecture 6.2 would replace Λ in Theorem 2.1 by the largest attainable cycle length, which together with the preperiod half — attained throughout the window — makes each coordinate of the bound sharp. Whether both can be met by one choice of final states is a further question; Section 9 shows that it is not automatic.

Widening the census is a matter of cost. At $k = 3$ the same search with $\mu_i \leq 3$ is about 6.8×10^9 instances, some fifty times the run above, and is the obvious next step; at $k = 4$ over the same cycle-length window $\lambda_i \leq 6$ with $\mu_i = 2$ it is about 6.5×10^{10} instances, and was not run.

7. The k-state question in the unary case

Separately from the unary section, [3] records a second open remark, in the middle of its treatment of automata with one state, in the paragraph that follows its Proposition 10 and the example illustrating it. We quote it verbatim:

> However, it looks like these upper bounds can further be decreased if a two-state automaton precedes the one-state automaton. If all automata except for A_k have two states, and A_k has one, it looks like their concatenation is recognized by a k-state DFA. We leave the problem of finding a tight upper bound for such cases open.

The remark is stated over a general alphabet Σ, where $L(A_k) \in \{\emptyset, \Sigma^*\}$. Over one letter it becomes elementary, and holds for every k.

Proposition 7.1. *Let $L_1, \dots, L_{k-1} \subseteq \mathbb{N}$ each be recognised by a unary automaton with at most two states, and let $L_k = \mathbb{N}$. Then $L_1 L_2 \cdots L_k$ is recognised by a unary automaton of size $(1, k-1)$, that is, by a k-state automaton. The bound is attained: taking each of $L_1, \dots, L_{k-1}$ to be the set of odd numbers gives $L_1 \cdots L_k = \{n : n \geq k-1\}$, which is not eventually periodic with preperiod $k-2$ and period 1.*

Proof sketch. No computation is involved. A unary language with at most two states is either empty or contains an element at most 1: its automaton has size $(1, 0)$, $(1, 1)$ or $(2, 0)$, and in each case the states reachable by a^0 and a^1 exhaust the state set. If some L_i is empty the concatenation is empty and the claim is trivial. Otherwise pick $e_i \in L_i$ with $e_i \leq 1$ for $i \leq k-1$; then $e_1 + \cdots + e_{k-1} \leq k-1$ lies in $L_1 \cdots L_{k-1}$, and adding $L_k = \mathbb{N}$ makes the concatenation the upward closure of a set meeting $\{0, \dots, k-1\}$, hence of the form $\{n : n \geq m\}$ with $m \leq k-1$. Such a set is eventually periodic with preperiod $m \leq k-1$ and period 1, so it is recognised by an automaton of size $(1, k-1)$. For attainment, the sumset of r copies of the odd numbers contains no element below r and contains r, by induction on r; with $L_k = \mathbb{N}$ the concatenation is $\{n : n \geq k-1\}$, which is not 1-periodic from $k-2$ on, so its preperiod is exactly $k-1$. □

That last clause gives the lower bound in the form the formal development checks. It extends to every automaton size in one line, on paper: an automaton of size (λ, μ) with $\lambda + \mu < k$ has $\lambda \geq 1$ and $\mu \leq k-2$, while $k-2 \notin \{n : n \geq k-1\}$ and $k-2+\lambda \in \{n : n \geq k-1\}$, so it fails to be λ-periodic from μ on. Hence k states really are necessary.

Remark 7.2 (the general alphabet; computation outside the formal development). Proposition 7.1 does not settle the remark of [3], which is over an arbitrary alphabet. That case was verified exhaustively for $k \leq 6$, over *every* alphabet, by the following reduction, again outside the formal development. State complexity does not decrease when the alphabet is enlarged, and every r-tuple of two-state automata over any alphabet factors through the universal action alphabet $\Sigma_r = \{\mathrm{id}, \mathrm{swap}, \mathrm{const}_0, \mathrm{const}_1\}^r$ of size 4^r, by sending each letter to the tuple of the actions it induces. So enumerating Σ_r with all 4^r choices of final-state sets dominates every alphabet and every choice of automata. For $r = k-1 = 1, \dots, 5$ the maximum state complexity of $L_1 \cdots L_r \Sigma^*$ was measured as

$2, 3, 4, 5, 6$ respectively — that is, exactly k — and in each case the maximum is attained by exactly one choice of final-state sets, the one in which every A_i has its initial state non-final. A proof for general k along these lines, taking the maximal non-empty level of the underlying nondeterministic automaton as a bisimulation invariant, is plausible but is not carried out here.

8. A cell of the k-letter conjecture

The paper [3] also conjectures, about a different question, that "k symbols are necessary for describing witnesses for concatenation of k languages", and proves this only at $k = 3$ (its Theorem 19, about the witnesses of its Theorem 17), adding in its conclusions that "we cannot see any generalization of the proof". The relevant upper bound is the number $\#\tau_k$ of *valid states* of the subset automaton defined in [3] and computed there by the recurrence of [3, Theorem 6]; with all component automata on two states that recurrence gives $\#\tau_3 = 15$, $\#\tau_4 = 37$, $\#\tau_5 = 90$ and $\#\tau_6 = 218$.

Remark 8.1 (computation outside the formal development). All tuples of two-state automata over an alphabet of m letters were enumerated, the subset automaton constructed and minimised, for $m \leq 3$. With all $n_i = 2$: at $k = 3$ the largest state complexity of $L_1L_2L_3$ over a binary alphabet is 13, strictly below $\#\tau_3 = 15$, and over a ternary alphabet it is exactly 15 — which reproduces, at this instance, the ternary-optimality theorem [3, Theorem 19]. At $k = 4$ the largest is 18 over a binary alphabet and 35 over a ternary one, strictly below $\#\tau_4 = 37$, from 2.7×10^8 instances and about 15 minutes of processor time; and the four-letter all-two-state construction of [3] preceding its Proposition 21 reaches exactly 37. Hence four symbols are necessary for witnesses at $k = 4$ with all $n_i = 2$, a case the conjecture of [3] leaves open. Following the remark of [3] before its Proposition 5 — that to reach as many valid states as possible each A_i with $i \leq k - 1$ should have a single final state, different from its initial state — the search fixes $F_i \in \{\{s_i\}, \{f_i\}\}$; the two excluded choices are harmless, since $F_i = \emptyset$ makes the concatenation empty and $F_i = \{s_i, f_i\}$ makes $L_i = \Sigma^*$, a case in which [3, Propositions 8 and 10] bound the complexity far below 35 — at most 15, and less when more than one L_i equals Σ^*. The next case, $k = 5$ over a four-letter alphabet, is about 1.1×10^{12} tuples — some four thousand times the $k = 4$ run, a thousand hours of processor time at the rate just quoted — and was not run.

Remark 8.2 (a discrepancy in the source text). The all-two-state construction of [3] is introduced over the alphabet $\{b, c, a_2, a_3, \ldots, a_{k-1}\}$, which has k letters, while its Proposition 21, immediately following, lists the alphabet as $\{b, c, a_2, a_3, \ldots, a_k\}$, which has $k + 1$. The k-letter list is the one the construction is written for: the figure that accompanies it, drawn for $k = 6$, carries exactly the letters b, c, a_2, a_3, a_4, a_5, with $a_5 = a_{k-1}$ acting on both A_5 and A_6 as the construction prescribes, and the whole point of the section is to produce witnesses over k letters. Read that way the construction meets $\#\tau_k$ at $k = 4, 5, 6$ (the values 37, 90, 218 were recomputed), but degenerates at $k = 3$, where the index range $i = 2, \ldots, k - 2$ of its first clause is empty and the letters a_2 and a_{k-1} coincide; the resulting complexity is 7, not $\#\tau_3 = 15$. Taking the alphabet as Proposition 21 prints it does not repair this: none of the four clauses assigns an action to a_k, so the extra letter acts as the identity in every component and the complexity is again 7. None of this is a counterexample to a claim of [3]: exhaustive search finds 390 ternary all-two-state witnesses meeting $\#\tau_3 = 15$, so three letters do suffice at $k = 3$; only the construction as printed does not produce one.

9. Open problems

The tight cycle length. Conjecture 6.2 is the main question left. With the preperiod half of Theorem 2.1 attained throughout the census window (Remark 6.1(b)), a proof of Conjecture 6.2 would make each coordinate of Theorem 2.1 — with Λ replaced by Λ^* — attained by some choice of final states, at every tuple of sizes with all $\mu_i \geq 1$ in the window. It would not by itself make the *pair* attained, as the next paragraph shows.

A criterion for the pair. Which size tuples meet the full bound? At $k = 2$ the census gives a clean answer over its window — the bound is met exactly when $\operatorname{lcm}(\lambda_1, \lambda_2) = 1$, or $\gcd(\lambda_1, \lambda_2) \geq 2$ and

$\mu_i \geq 2$ for every i with $\lambda_i \neq \mathrm{lcm}(\lambda_1, \lambda_2)$ — but that criterion already fails at $k = 3$, for instance at cycle lengths $(2, 2, 4)$. The sharpest open cell in the window is $(3, 4, 6)$ with tails $(2, 2, 2)$: there the bound is $(12, 22)$, and each coordinate is attained by some choice of final states, but the measured maximum state count is 28 rather than 34, so they are never attained together.

The general-alphabet k-state remark. Remark 7.2 verifies it to $k = 6$; a proof for every k would settle the second open question of [3] in full.

10. Formal verification

Every theorem and proposition stated above — Propositions 3.1 and 7.1 and Theorems 4.1, 4.2, 5.1 and 5.2 — is formalised and machine-checked in Lean 4 [4] on top of Mathlib [5]. The material stated as remarks is exactly the material that is not: Remarks 6.1, 7.2, 8.1 and 8.2, together with Conjecture 6.2, are computations and observations outside the formal development. One further step is outside it: the paragraph after Proposition 7.1, which carries that proposition's lower bound from automata with a one-state cycle to automata of every size; it is a one-line argument, not a computation. The development is four files: the dictionary of Section 2 — eventual periodicity, the sumset, the language of a unary automaton of size (λ, μ) with a given final-state set, and Lemmas 2.2, 2.3 and 2.4 — then one file each for Section 3, for Sections 4–5, and for Section 7. There is no incomplete proof and no hand-declared axiom. The axiom set of every statement above was printed. The whole of the first file and the whole of Section 7 are free of compiled evaluation, using no axiom beyond the three of Lean's own logic; the first file uses only two of those, propositional extensionality and quotient soundness. That is the formal counterpart of their being proofs for all parameters rather than checks over a range. The finite-range statements depend additionally on compiled evaluation, through one named axiom for each named computation: five for Proposition 3.1 (the window check and the four single-point refutations), one for Theorem 4.1 (the 8192-instance check), one for Theorem 4.2, three for Theorem 5.1 and four for Theorem 5.2. Each of those runs a check and then uses its output only through Lemma 2.3 or through a single membership disagreement, so no search procedure is trusted. Negative controls are included: that a tail one shorter fails in each exactness claim, that the quantifier ranges of Theorem 4.1 are inhabited, that the witness languages are not eventually constant, and that $k - 1$ states do not suffice in Proposition 7.1. The whole development checks in a few minutes, and the computations reported in Remarks 6.1–8.1 took well under one hour of processor time in total, on the independent programs described there. The repository is private; repository reference to be added at submission.

References

[1] P. Caron, J.-G. Luque and B. Patrou, *State Complexity of Multiple Catenations*, Fund. Inform. **160** (2018), no. 3, 255–279; preprint arXiv:1607.04031; DOI `10.3233/FI-2018-1683`. The paper that poses the multiple-concatenation problem addressed by [3]. Authors, title, volume, issue, pages and year verified against Crossref on 3 September 2026. Its e-print was read on the same date and checked for a unary section; it has none, the word "unary" occurring there only in the sense of a one-argument operation.

[2] J. Jirásek and G. Jirásková, *Multiple Concatenation and State Complexity (Extended Abstract)*, in: Descriptional Complexity of Formal Systems (DCFS 2020), Lecture Notes in Computer Science **12442**, Springer, 2020, pp. 78–90; DOI `10.1007/978-3-030-62536-8_7`. The preliminary version of [3], as the title footnote of [3] records. Authors, title, series, volume, pages and year verified against dblp and Crossref on 3 September 2026; its text was not consulted, and nothing in this paper depends on its contents.

[3] J. Jirásek and G. Jirásková, *State Complexity of Multiple Concatenation*, arXiv:2511.03814v2 (26 November 2025), cs.FL; 28 pages, 17 figures, MSC 68QXX. All quotations and all references to numbered results are to the LaTeX source of v2, retrieved on 3 September 2026. On that date the arXiv abstract page listed v1 (5 November 2025) and v2 (26 November 2025, the current version) and carried no journal reference. Numbered results of [3] are cited by the numbers they receive in a compilation of that source, which was carried out here; the ones used are Proposition 5 and Theorem 6 (the count of valid states), Proposition 8 (one-state components), Proposition 10 and the remark following its Example 11 (the k-state question), Theorem 17 and Theorem 19 (k-letter witnesses and ternary optimality), Proposition 21 with the construction and Figure 13 preceding it (all-two-state witnesses), Theorem 26 and Corollary 27 (the two tight unary bounds) and Theorem 29 (the unary bound studied here). Since these numbers are not printed in the LaTeX source, a reader of a later version should re-derive them from the labels

`prop:ineq`, `thm:upper`, `prop:jeden1-state`, `prop:more-one-state`, `thm:k`, `fig:2stateDFAs`, `a:thm:cyclic_unary`, `cor_unary` and `thm:unary_final_tails`.

[4] L. de Moura and S. Ullrich, *The Lean 4 theorem prover and programming language*, in: Automated Deduction — CADE 28, Lecture Notes in Computer Science, Springer, 2021, pp. 625–635; DOI `10.1007/978-3-030-79876-5_37`.

[5] The mathlib Community, *The Lean mathematical library*, in: Certified Programs and Proofs (CPP 2020), ACM, 2020, pp. 367–381; DOI `10.1145/3372885.3373824`.

[6] A. N. Maslov, *Estimates of the number of states of finite automata*, Soviet Math. Doklady **11** (1970), 1373–1375; translated from Dokl. Akad. Nauk SSSR **194** (1970), 1266–1268. Cited for the classical concatenation bound; bibliographic data verified against zbMATH (Zbl 0222.94064) on 3 September 2026. The issue number recorded in the bibliography file distributed with the source of [3] is omitted here, having no independent confirmation.

[7] C. Nicaud, *Average State Complexity of Operations on Unary Automata*, in: Mathematical Foundations of Computer Science 1999 (MFCS 1999), Lecture Notes in Computer Science **1672**, Springer, 1999, pp. 231–240; DOI `10.1007/3-540-48340-3_21`. Cited only for the general point that the unary case has its own average-case theory; no result of it is used. Authors, title, series, volume, pages, year and DOI verified against dblp and Crossref on 3 September 2026.

[8] G. Pighizzini and J. Shallit, *Unary Language Operations, State Complexity and Jacobsthal's Function*, Internat. J. Found. Comput. Sci. **13** (2002), no. 1, 145–159; DOI `10.1142/S012905410200100X`. Source of the dictionary between unary automata of size (λ, μ) and eventually periodic subsets of $\mathbb{N}$ used throughout Section 2, and of the two-language unary concatenation results [3] invokes for the case with final states in the tails. Authors, title, journal, volume, issue, pages and year verified against Crossref on 3 September 2026. The article is not openly available and its text was not consulted: the result numbers attached to it here (Theorem 2 for the dictionary, Theorems 10 and 11 for two-language unary concatenation) are the numbers under which [3] cites it, and nothing in this paper depends on them — the dictionary is re-proved from the definitions in the formal development, and every $k = 2$ figure quoted here is our own measurement.

[9] S. Yu, Q. Zhuang and K. Salomaa, *The State Complexities of Some Basic Operations on Regular Languages*, Theoret. Comput. Sci. **125** (1994), no. 2, 315–328; DOI `10.1016/0304-3975(92)00011-F`. Cited for the classical two-language state complexity of concatenation; it is the reference [3] gives, in its conclusions, for the value $n_1 n_2$ attained by cyclic unary languages when $k = 2$ and $\gcd(n_1, n_2) = 1$. Authors, title, journal, volume, issue, pages and year verified against Crossref on 3 September 2026; its text was not consulted.