Dimension-Minimality and Primality of Counter Nets

Shaull Almagor
Department of Computer Science, Technion, Israel

Guy Avni
Department of Computer Science, University of Haifa, Israel

Henry Sinclair-Banks
Centre for Discrete Mathematics and its Applications (DIMAP) & Department of Computer Science, University of Warwick, Coventry, UK

Asaf Yeshurun
Department of Computer Science, Technion, Israel

Abstract

A k-Counter Net (k-CN) is a finite-state automaton equipped with k integer counters that are not allowed to become negative, but do not have explicit zero tests. This language-recognition model can be thought of as labelled vector addition systems with states, some of which are accepting. Certain decision problems for k-CNs become easier, or indeed decidable, when the dimension k is small. Yet, little is known about the effect that the dimension k has on the class of languages recognised by k-CNs. Specifically, it would be useful if we could simplify algorithmic reasoning by reducing the dimension of a given CN.

To this end, we introduce the notion of dimension-primality for k-CN, whereby a k-CN is prime if it recognises a language that cannot be decomposed into a finite intersection of languages recognised by d-CNs, for some $d < k$. We show that primality is undecidable. We also study two related notions: dimension-minimality (where we seek a single language-equivalent d-CN of lower dimension) and language regularity. Additionally, we explore the trade-offs in expressiveness between dimension and non-determinism for CN.

2012 ACM Subject Classification Replace ccsdesc macro with valid one

Keywords and phrases Counter Net, Vector Addition System, VASS, Automata, Primality

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Funding *Shaull Almagor*: supported by the ISRAEL SCIENCE FOUNDATION (grant No. 989/22)
Guy Avni: supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1679/21)
Henry Sinclair-Banks: Supported by EPSRC Standard Research Studentship (DTP), grant number EP/T5179X/1.

1 Introduction

A *k-dimensional Counter Net* (k-CN) is a finite-state automaton equipped with k integer counters that are not allowed to become negative, but do not have explicit zero tests (see Figure 1a for an example). This language-recognition model can be thought of as an alphabet-labelled Vector Addition System with States (VASS), some of whose states are accepting [7]. A k-CN $\mathcal{A}$ over alphabet Σ *accepts* a word $w \in \Sigma^*$ if there is a run of $\mathcal{A}$ on w that ends in an accepting state in which the counters stay non-negative. The *language* of $\mathcal{A}$ is the set $\mathcal{L}(\mathcal{A})$ of words accepted by $\mathcal{A}$.

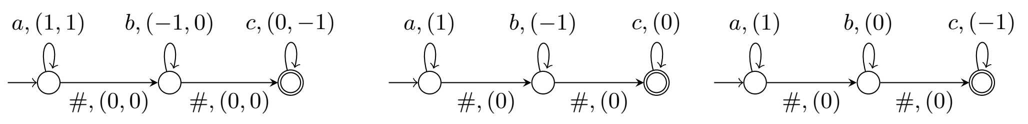
k-CNs are a natural model of concurrency and are closely related — and equivalent, in some senses — to labelled Petri Nets. These models have received significant attention over the years [6, 7, 13, 14, 16, 18, 27], with specific interest in the one-dimensional case, often referred to as One-Counter Nets [19, 20, 1, 2]. Unfortunately, most decision problems for k-CNs are notoriously difficult and are often undecidable [1, 2]. In particular, k-CNs subsume VASS and Petri nets, for which many problems are known to be Ackermann-complete, for example see the recent breakthrough in the complexity of reachability in VASS [11, 25].

In recent years it was noticed that for low dimensions, decision problems for VASS, sometimes with extensions, become more tractable [9, 8, 10, 15]. For example, reachability in dimensions one and two is NP-complete [17] and PSPACE-complete [4], respectively, when counter updates are encoded in binary.

A natural question, therefore, is whether we can *decrease* the dimension of a given a k-CN whilst maintaining its language, to facilitate reasoning about it. Since k-CNs are used for various purposes (usually in the form of Petri nets), a positive answer to this question could allow us to models systems of size previously untractable using richer parallelism. We tackle this question in this work by introducing two approaches. The first is straightforward *dimension-minimality*: given a k-CN, does there exist a d-CN $\mathcal{B}$ recognising the same language for some $d < k$?

The second approach is *primality*: given a k-CN, does there exist some $d < k$ and d-CNs $\mathcal{B}_1, \ldots, \mathcal{B}_n$ such that $L(\mathcal{A}) = \bigcap_{i=1}^{n} \mathcal{L}(\mathcal{B}_i)$? That is, we ask whether the language of $\mathcal{A}$ can be decomposed as an intersection of languages recognised by several lower-dimension CNs. We also consider *compositeness*, the dual of primality. Intuitively, in a composite k-CN the usage of the counters can be "split" across several lower-dimension CNs, allowing for properties (such as universality) to be checked on each conjunct separately.

▶ **Example 1.** We illustrate the model and the definition of compositionality. Consider the 2-CN $\mathcal{A}$ depicted in Figure 1a, and consider a word $w = a^m\#b^n\#c^k$. We have that $\mathcal{A}$ has an accepting run on w iff $m \geq n$ and $m \geq k$. Indeed, if $m < n$, the first counter drops below 0 while cycling in the second state and so the run is "stuck", and similarly if $m < k$. It is not hard to show that there is no 1-CN that recognizes the languages of $\mathcal{A}$. However, Figure 1b shows two 1-CNs $\mathcal{B}_1$ and $\mathcal{B}_2$ such that $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{B}_1) \cap \mathcal{L}(\mathcal{B}_2)$. Indeed, a word $w = a^m\#b^n\#c^k \in \mathcal{L}(\mathcal{B}_1)$ iff $m \geq n$, and $w \in \mathcal{L}(\mathcal{B}_2)$ iff $m \geq k$. Note that this decomposition is obtained by automata with the same structure as $\mathcal{A}$ and by "splitting" the counters between the two parts. However, as we show in Theorem 8, such a partition of counters does not in general yield a decomposition, but does work for deterministic 1-CNs, as we show in Proposition 4.



(a) A composite 2-CN. **(b)** Two 1-CNs showing compositeness of the 2-CN.

Figure 1 A composite 2-CN whose language is $\{a^m\#b^n\#c^k \mid m \geq n \wedge m \geq k\}$ and its decomposition into two 1-CNs recognising the languages $\{a^m\#b^n\#c^k \mid m \geq n\}$ and $\{a^m\#b^n\#c^k \mid m \geq k\}$.

The notion of primality has been studied for regular languages in [24, 23, 22], the exact complexity of deciding primality is still open. There, an automaton is composite if it can be written as an intersection of finite automata with fewer *states*. In this work we introduce primality for CNs. We focus on *dimension* as a measure of size, a notion which does not exist for regular languages. Thus, unlike regular languages, the differences between prime and composite CNs is not only in succinctness, but actually in expressiveness, as we later demonstrate.

We parameterise primality and compositeness by the dimension d and the number n of lower-dimension factors. Thus, a k-CN $\mathcal{A}$ is *(d, n)-composite* if it can be written as the intersection above. Then, $\mathcal{A}$ is *composite* if it is (d, n)-composite for some $d < k$ and $n \in \mathbb{N}$. Under this view, dimension-minimality is a special case of compositeness, namely $\mathcal{A}$ is dimension-minimal if it is not $(k-1, 1)$-composite. Another particular problem captured by compositeness is *regularity*. Indeed, $\mathcal{L}(\mathcal{A})$ is regular if and only if $\mathcal{A}$ is $(0, 1)$-composite, since 0-CNs are just NFAs. Since regularity is already undecidable for 1-CNs [2, 28], it follows that deciding whether a k-CN is (d, n)-composite is undecidable. Moreover, it follows that both primality and dimension-minimality are undecidable for 1-CNs.

The undecidability of the above problems is not surprising, as the huge difference in expressive power between 1-CNs and regular languages is well understood. In contract, even the expressive power difference between 1-CNs and 2-CNs is poorly understood, let alone what effect the dimension has on the expressive power beyond regular languages. Already, 1-VASS and 2-VASS are known to have *flat* equivalents with respect to reachability [26, 4], but the complexity differs greatly.

Our goal in this work is to shed light on these differences. In Section 4, we give a concrete example of a prime 2-CN, which turns out to be technically challenging. This example is the heart of our technical contribution, and we emphasize that we *do not* know of any examples of prime 3-CN, let alone for general k-CN. We consider this an interesting open problem. The technical intricacy in proving our example suggests that generalizing it is highly nontrivial. Using our example, we obtain in Section 5, the undecidability of primality and of dimension-minimality for 2-CNs. To complement this, we show in Section 6, that regularity of k-DCNs is decidable. In Section 7, we explore trade-offs in expressiveness of CNs with increasing dimension and with nondeterminism. We conclude with a discussion and open problems in Section 8. Due to space constraints, some proofs appear in the appendix.

2 Preliminaries

We denote the non-negative integers $\{0, 1, \ldots\}$ by $\mathbb{N}$. We write vectors in bold, e.g., $\boldsymbol{e} \in \mathbb{Z}^k$, and $\boldsymbol{e}[i]$ is the i-th coordinate. We use $[k] = \{1, \ldots, k\}$ for $k \geq 1$. We use Σ^* to denote the set of all words over an alphabet Σ, and $|w|$ is the length of $w \in \Sigma^*$.

A *k-dimensional Counter Net* (k-CN) $\mathcal{A}$ is a quintuple $\mathcal{A} = \langle \Sigma, Q, Q_0, \delta, F \rangle$ where Σ is a finite alphabet, Q is a finite set of states, $Q_0 \subseteq Q$ is the set of initial states, $\delta \subseteq Q \times \Sigma \times \mathbb{Z}^k \times Q$ is a set of transitions, and $F \subseteq Q$ are the accepting states. A k-CN is *deterministic*, denoted k-DCN, if $|Q_0| = 1$, and for every $p \in Q$ and $\sigma \in \Sigma$ there is at most one transition of the form $(p, \sigma, \boldsymbol{v}, q) \in \delta$. For a transition $(p, \sigma, \boldsymbol{v}, q) \in \delta$, we refer to $\boldsymbol{v} \in \mathbb{Z}^k$ as its *effect*.

An $\mathbb{N}$-configuration (resp. $\mathbb{Z}$-configuration) of a k-CN $\mathcal{A}$ is a pair $(q, \boldsymbol{v}) \in Q \times \mathbb{N}^k$ (resp. $(q, \boldsymbol{v}) \in Q \times \mathbb{Z}^k$) representing the current state and values of the counters. A transition $(p, \sigma, \boldsymbol{e}, q) \in \delta$ is *valid* from $\mathbb{N}$-configuration $(q, \boldsymbol{v})$ if $\boldsymbol{v} + \boldsymbol{e} \in \mathbb{N}^k$, i.e., if all k counters remain non-negative after the transition. A *$\mathbb{Z}$-run* ρ of $\mathcal{A}$ on w is a sequence of $\mathbb{Z}$-configurations $\rho = (q_0, \boldsymbol{v}_0), (q_1, \boldsymbol{v}_1), \ldots, (q_n, \boldsymbol{v}_n)$ such that $(q_i, \sigma_i, \boldsymbol{v}_{i+1} - \boldsymbol{v}_i, q_{i+1}) \in \delta$ for every $0 \leq i \leq n-1$, we may also say that ρ *reads* $w = \sigma_0\sigma_1 \cdots \sigma_n$. An *$\mathbb{N}$-run* is a $\mathbb{Z}$-run that visits only $\mathbb{N}$-configuration. Note that all the transitions in an $\mathbb{N}$-run are valid. We may omit $\mathbb{N}$ or $\mathbb{Z}$ from the run when it does not matter. For a run $\rho = (q_0, \boldsymbol{v}_0), (q_1, \boldsymbol{v}_1), \ldots, (q_n, \boldsymbol{v}_n)$ of $\mathcal{A}$, we denote $(q_0, \boldsymbol{v}_0) \xrightarrow{\rho} (q_n, \boldsymbol{v}_n)$. We define the *effect* of ρ to be $\mathsf{eff}(\rho) = \boldsymbol{v}_n - \boldsymbol{v}_0$.

An $\mathbb{N}$-run ρ is *accepting* if $q_0 \in Q_0$, $\boldsymbol{v}_0 = \boldsymbol{0}$, and $q_n \in F$. We say that $\mathcal{A}$ *accepts* w if there is an accepting $\mathbb{N}$-run of $\mathcal{A}$ on w. The *language* of $\mathcal{A}$ is $\mathcal{L}(\mathcal{A}) = \{w \in \Sigma^* \mid \mathcal{A} \text{ accepts } w\}$.

An infix $\pi = (q_k, \boldsymbol{v}_k), (q_{k+1}, \boldsymbol{v}_{k+1}), \ldots, (q_{k+n}, \boldsymbol{v}_{k+n})$ of a run ρ is a *cycle* if $q_k = q_{k+n}$ and is a *simple cycle* if it does not contain a cycle as a proper infix. When discussing an infix π of a 1-CN – we write that π is > 0, ≥ 0, or < 0 if $\mathsf{eff}(\pi) > 0$, $\mathsf{eff}(\pi) \geq 0$, or $\mathsf{eff}(\pi) < 0$, respectively.

3 Primality and Compositeness

We begin by presenting our main definitions, followed by some introductory properties.

▶ **Definition 2** (Compositeness, Primality, and Dimension-Minimality). *Consider a k-CN $\mathcal{A}$, and let $d, n \in \mathbb{N}$. We say that $\mathcal{A}$ is* (d, n)-composite *if there exist d-CNs $\mathcal{B}_1, \ldots, \mathcal{B}_n$ such that $\mathcal{L}(\mathcal{A}) = \bigcap_{i=1}^{n} \mathcal{L}(\mathcal{B}_i)$. If $\mathcal{A}$ is (d, n)-composite for some $d < k$ and $n \in \mathbb{N}$, we say $\mathcal{A}$ is* composite*. Otherwise, $\mathcal{A}$ is* prime*. If $\mathcal{A}$ is not $(k-1, 1)$-composite, we say that $\mathcal{A}$ is* dimension-minimal

▶ Remark 3. Note that the special case where $\mathcal{A}$ is $(0, n)$-composite coincides with the regularity of $\mathcal{L}(\mathcal{A})$, and hence also with being $(0, 1)$-composite.

Observe that in Figure 1 we in fact show a composite 2-DCN. We now show that every k-DCN is $(1, k)$-composite, by projecting to each of the counters separately. In particular, a k-DCN is prime only when $k = 1$ and it recognises a non-regular language, or when $k = 0$. Formally, consider a k-DCN $\mathcal{D} = \langle \Sigma, Q, Q_0, \delta, F \rangle$ and let $1 \leq i \leq k$. We define the *i-projection* to be the 1-DCN $\mathcal{D}|_i = \langle \Sigma, Q, Q_0, \delta|_i, F \rangle$ where $\delta|_i = \{(p, \sigma, \boldsymbol{v}[i], q) \mid (p, \sigma, \boldsymbol{v}, q) \in \delta\}$.

▶ **Proposition 4.** *Every k-DCN $\mathcal{D}$ is $(1, k)$-composite. Moreover, $\mathcal{L}(\mathcal{D}) = \bigcap_{i=1}^{k} \mathcal{L}(\mathcal{D}|_i)$.*

Proof. Let $w \in \mathcal{L}(\mathcal{D})$ and let ρ be the accepting run of $\mathcal{D}$ on w, then the projection of ρ on counter i induces an accepting run of $\mathcal{D}|_i$ on w, thus $w \in \bigcap_{i=1}^{k} \mathcal{L}(\mathcal{D}|_i)$. Note that this direction does not use the determinism of $\mathcal{D}$.

Conversely, let $w \in \bigcap_{i=1}^{k} \mathcal{L}(\mathcal{D}|_i)$, then each $\mathcal{D}|_i$ has an accepting run ρ_i on w. Since the structure of all the $\mathcal{D}|_i$ is identical to that of $\mathcal{D}$, all the runs ρ_i have identical state sequences, and therefore are also a $\mathbb{Z}$-run of $\mathcal{D}$ on w. Moreover, due to this being a single $\mathbb{N}$-run in each $\mathcal{D}|_i$, it follows that all counter values remain non-negative in the corresponding run of $\mathcal{D}$ on w. Hence, this is an accepting $\mathbb{N}$-run of $\mathcal{D}$ on w, so $w \in \mathcal{L}(\mathcal{D})$. ◀

▶ Remark 5 (Unambiguous Counter Nets are Composite). The proof of Proposition 4 applies also to *unambiguous* CNs. Thus, every unambiguous CN is $(1, k)$-composite.

Consider k-CNs $\mathcal{B}_1, \ldots, \mathcal{B}_n$. By taking their product, we can construct a kn-CN $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \bigcap_{i=1}^{n} \mathcal{L}(\mathcal{B}_i)$. In particular, if each $\mathcal{B}_i$ is a 1-DCN, then $\mathcal{A}$ is an n-DCN. Combining this with Proposition 4, we can deduce the following (proof in Appendix A.1).

▶ **Proposition 6.** *A k-DCN is dimension-minimal if and only if it is not $(1, k-1)$-composite.*

4 A Prime Two-Counter Net

In this section we present our main technical contribution, namely an example of a prime 2-CN. The technical difficulty arises from the need to prove that this example cannot be decomposed as an *intersection* of *nondeterministic* 1-CNs. Since intersection has a "universal flavour", and nondeterminism has an "existential flavour", we have a sort of "quantifier alternation" which is often a source of difficulty.

The importance of this example is threefold. First, it enables us to show in that primality is undecidable in Section 5. Second, it offers intuition on what makes a language prime. Third, we suspect that the techniques developed here will be useful in other settings when reasoning about nondeterministic automata, perhaps with counters.

We start by presenting the prime 2-CN, followed by an overview of the proof, before delving into the details.

▶ **Example 7.** Consider the 2-CN $\mathcal{P}$ over alphabet $\Sigma = \{a, b, c, \#\}$ depicted in Figure 2. Intuitively, $\mathcal{P}$ starts by reading segments of the form $a^m\#$, where in each segment it nondeterministically chooses whether to increase the first or second counter by m. Then, it reads $b^{m_b}c^{m_c}$ and accepts if the value of the first and second counter is at least m_b and m_c, respectively. Thus, $\mathcal{P}$ accepts a word if its $a^m\#$ segments can be partitioned into two sets I and $\overline{I}$ so that the combined lengths of the segments in I (resp. $\overline{I}$) is at least the length of the b segment (resp. c segment). For example, $a^{10}\#a^{20}\#a^{15}\#b^{15}c^{30} \in \mathcal{L}(\mathcal{P})$, since segments 1 and 2 have length 30, matching c^{30} and segment 3 matches b^{15}. However, $a^{10}\#a^{20}\#a^{15}\#b^{21}c^{21} \notin \mathcal{L}(\mathcal{P})$, since in any partition of $\{10, 20, 15\}$, one set will have sum lower than 21. More precisely, we have the following:

$$\mathcal{L}(\mathcal{P}) = \{a^{m_1}\#a^{m_2}\#\cdots\#a^{m_t}\#b^{m_b}c^{m_c} \mid \exists I \subseteq [t] \text{ s.t. } \sum_{i\in I} m_i \geq m_b \wedge \sum_{i\notin I} m_i \geq m_c\}$$

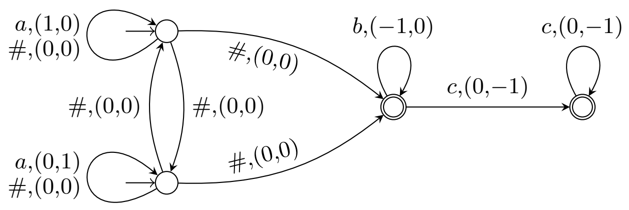


Figure 2 The prime 2-CN $\mathcal{P}$ for Example 7 and Theorem 8.

▶ **Theorem 8.** *$\mathcal{P}$ is prime.*

The high-level intuition behind Theorem 8 is that any 1-CN can either guess a subset of segments that covers m_b or m_c, but not both, and in order to make sure the choices between two 1-CNs form a partition, we need to fix the partition in advance. This is only possible if the number of segments is a priori fixed, which is not true (c.f., Remark 16). This intuition, however, is far from a proof.

4.1 Overview of the Proof of Theorem 8

Assume by way of contradiction that $\mathcal{P}$ is not a prime 2-CN. Thus, there exist 1-CNs $\mathcal{V}_1, \ldots \mathcal{V}_k$ such that $\mathcal{L}(\mathcal{P}) = \bigcap_{1\leq j\leq k} \mathcal{L}(\mathcal{V}_j)$.

Throughout the proof, we focus on words of the form $a^{m_1}\#a^{m_2}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$ for integers $\{m_i\}_{i=1}^{k+1}, m_b, m_c \in \mathbb{N}$. We index the a^{m_i} segments of these words, so a^{m_i} is the i-th segment. Note that we focus on words with $k+1$ many a segments, one more than the number of $\mathcal{V}_j$ factors in the intersection. It is useful to think about each segment as "paying" for either b or c. Then, a word is accepted if there is a way to choose for each segment whether it pays for b or c, such that there is sufficient budget for both.

Let $i \in [k+1]$ and $j \in [k]$. We say that the i-th segment is *bad* in $\mathcal{V}_j$ if, intuitively, we can pump the length m_i of segment i whilst pumping both m_b and m_c to unbounded lengths, such that the resulting words are accepted by $\mathcal{V}_j$ (see Definition 13 for the formal definition). For example, consider the word $a^{10}\#a^{10}\#a^{10}\#b^{20}c^{10} \in \mathcal{L}(\mathcal{P})$. If the second segment is bad for $\mathcal{V}_j$ then there exist $x, y, z > 0$ such that for every $t, t_b, t_c \in \mathbb{N}$ it holds that $a^{10}\#a^{10+tx}\#a^{10}\#b^{20+t_b y}c^{10+t_c z} \in \mathcal{L}(\mathcal{V}_j)$. Observe that such behaviour is undesirable, since for large enough t, t_b, t_c, the resulting word is not in $\mathcal{L}(\mathcal{P})$. Note, however, that the existence of such a bad segment is not a contradiction by itself, since the resulting pumped words might not be accepted by some other 1-CN $\mathcal{V}_{j'}$.

In order to reach a contradiction, we need to show the existence of a segment i that is bad for *every* $\mathcal{V}_j$. Moreover, we must also show that arbitrarily increasing m_i, m_b, m_c can be simultaneously achieved in all the $\mathcal{V}_j$ together (i.e., the above $x, y, z > 0$ are the same for all V_j). This would create a contradiction since all the $\mathcal{V}_j$ accept a word that is not in $\mathcal{L}(\mathcal{P})$. Our goal is therefore to establish a robust and precise definition of a "bad" segment, then find a word w comprising $k+1$ segments where one of the segments is bad for every $\mathcal{V}_j$, and pumping the words in each segment can be done synchronously.

4.2 Pumping Arguments in One-Counter Nets

In this section we establish some pumping results for 1-CN which will be used in the proof of Theorem 8. Throughout this section, we consider a 1-CN $\mathcal{V} = \langle \Sigma, Q, Q_0, \delta, F \rangle$.

Our first lemma states the intuitive fact that without > 0 cycles, the counter value of a run is bounded (proof in Appendix A.2).

▶ **Lemma 9.** *Let (q, n) be a configuration of $\mathcal{V}$, let W be the maximal positive update in $\mathcal{V}$, $\sigma \in \Sigma$, and $N \in \mathbb{N}$. If an $\mathbb{N}$-run ρ of $\mathcal{V}$ on σ^N from configuration (q, n) does not traverse any > 0 cycle, then the maximal possible counter value anywhere along ρ is $n + W|Q|$.*

The next lemma shows that long-enough runs must contain ≥ 0 cycles.

▶ **Lemma 10.** *Let $\sigma \in \Sigma$ and (q, n) be an $\mathbb{N}$-configuration of $\mathcal{V}$. Then there exists $N \in \mathbb{N}$ such that for all $N' \geq N$, every $\mathbb{N}$-run of $\mathcal{V}$ on $\sigma^{N'}$ from (q, n) traverses a ≥ 0 cycle.*

Proof. Let W be the maximal positive transition update in $\mathcal{V}$, we show that $N = |Q|(n + |Q| \cdot W)$ satisfies the requirements. Assume by way of contradiction that $\mathcal{V}$ can read σ^N via an $\mathbb{N}$-run $\rho = (q_0, n_0 = n) \xrightarrow{\rho} (q_N, n_N)$ that only traverses < 0 cycles.

Since ρ visits $N+1$ states, then by the Pigeonhole Principle, there exists a state $p \in Q$ that is visited $m \geq (N+1)/|Q| > N/|Q|$ many times in ρ.

Consider all the indices $0 \leq i_1 < i_2 < \ldots < i_m \leq N$ such that $p = q_{i_1} = \ldots = q_{i_N}$. Each run segment $(q_{i_1}, n_{i_1}) \to (q_{i_2}, n_{i_2}), \ldots, (q_{i_{m-1}}, n_{i_{m-1}}) \to (q_{i_m}, n_{i_m})$ is a cycle in ρ, and therefore must have negative effect. Thus $n_{i_1} > n_{i_2} > \ldots > n_{i_m} \geq 0$, so in particular $n_{i_1} \geq n_{i_m} + m \geq 0$ (as each cycle has effect at most -1). Moreover, $n_{i_1} < n + |Q| \cdot W$ since the prefix $(q_0, n) \to (q_{i_1}, n_{i_1})$ cannot contain a non-negative cycle. However, since $m > N/|Q| = n + |Q| \cdot W$ and $n_{i_1} \geq n_{i_m} + m \geq m \geq n + |Q| \cdot W$, we get $n + |Q| \cdot W < n + |Q| \cdot W$ which is a contradiction. ◀

Next, we show that runs with ≥ 0 and > 0 cycles have "pumpable" infixes.

▶ **Lemma 11.** *Let $\sigma \in \Sigma$ and consider a > 0 (resp. ≥ 0) cycle $\pi = (q_0, c_0) \xrightarrow{\sigma} (q_1, c_1) \xrightarrow{\sigma} \ldots (q_n = q_0, c_n)$ on σ^n that induces an $\mathbb{N}$-run. Then, there is a sequence of (not necessarily contiguous) indices $0 \leq i_1 \leq \ldots \leq i_k \leq n$ such that $q_{i_1} \xrightarrow{\sigma} q_{i_2} \xrightarrow{\sigma} \cdots q_{i_k}$ is a simple > 0 (resp. ≥ 0) cycle with some effect $e > 0$ (resp. $e \geq 0$). In addition, this simple cycle is "pumpable"* *from the first occurrence of q_{i_1} in π; namely, for all $m \in \mathbb{N}$ there is a run π_m obtained from π by traversing the cycle m times so that $\mathsf{eff}(\pi_m) = \mathsf{eff}(\pi) + em$.*

Proof. We prove the ≥ 0 case, the > 0 case can be proved mutatis mutandis.

We define $\pi_m = (q_0, c_0) \xrightarrow{\sigma} \dots (q_{i_1}, c_{i_1}) \xrightarrow{\sigma} \dots (q_{i_1}, c_{i_1} + km) \xrightarrow{\sigma} \dots (q_n, c_n + km)$. The proof is now by induction on the length of π.

The base of the induction is a cyclic $\mathbb{N}$-run of length 2. In this case $\pi = (q_0, c_0) \xrightarrow{\sigma} (q_1 = q_0, c_1)$ is itself a ≥ 0 simple cycle that is infinitely pumpable from (q_0, c_0).

We now assume correctness for length n, and discuss $\pi = (q_0, c_0) \xrightarrow{\sigma} (q_1, c_1) \xrightarrow{\sigma} \dots (q_n = q_0, c_n)$ of length $n + 1$. Let $0 \leq j_1 < j_2 \leq n$ be indices such that $q_{j_1} = q_{j_2}$, for a maximal j_1. Note that the cycle $\tau = (q_{j_1}, c_{j_1}) \xrightarrow{\sigma} \dots (q_{j_2}, c_{j_2})$ must be simple. If $j_1 = 0$ and $j_2 = n$, then π itself is a simple ≥ 0 cycle, and the pumping argument is straightforward. Otherwise τ is nested. We now split into two cases, based on whether $\mathsf{eff}(\tau) \geq 0$.

1. τ is ≥ 0: then the induction hypothesis applies on τ. We take the guaranteed constants $j_1 \leq i_1 \leq \ldots \leq i_k \leq j_2$, which apply to π as well.
2. τ is < 0: then we remove τ from π to obtain $\pi' = (q_0, c_0) \xrightarrow{\sigma} \dots (q_{j_1}, c_{j_1}) \xrightarrow{\sigma} (q_{j_2+1}, c'_{j_2+1}) \xrightarrow{\sigma} \dots (q_n, c'_n)$, such that $c'_i \geq c_i$ for all $j_2 + 1 \leq i \leq n$. The induction hypothesis applies on π', so let $i_1, \ldots, i_k$ be the guaranteed constants. Note that $i_1 \leq j_1$, since the cycle removed when obtaining π' from π is the last occurrence of a repetition of states in π. We therefore know that $q_{i_1} \xrightarrow{\sigma} q_{i_2} \xrightarrow{\sigma} \cdots q_{i_k}$ is a simple ≥ 0 cycle in π' – which applies to π as well. In addition, it is infinitely pumpable from $\mathbb{N}$-configuration (q_{i_1}, c_{i_1}) in π' for $i_1 \leq j_1$. Indeed, since π and π' coincide up to and including (q_{j_1}, c_{j_1}) between π and π' - this cycle is infinitely pumpable in π as well. ◀

The simple cycle in Lemma 11 has length $k < |Q|$. By pumping it $\frac{|Q|!}{k}$ times we obtain a pumpable cycle of length $|Q|!$, allowing us to conclude with the following.

▶ **Corollary 12.** *Let ρ be an $\mathbb{N}$-run of $\mathcal{V}$ on σ^n that traverses a ≥ 0 cycle. For every $m \in \mathbb{N}$, we can construct an $\mathbb{N}$-run ρ' of $\mathcal{V}$ on $\sigma^{n+m|Q|!}$ such that $\mathsf{eff}(\rho') \geq \mathsf{eff}(\rho)$ by pumping a ≥ 0 simple cycle in ρ.*

4.3 Good and Bad Segments

We lift the colour scheme[1] of > 0 and ≥ 0 to words and runs as follows. For a word $w = uv$ and a run ρ, we write e.g., uv to denote that ρ traverses a > 0 cycle when reading u, then a ≥ 0 cycle when reading v. Note that this does not preclude other cycles, e.g., there could also be negative cycles in the u part, etc. That is, the colouring is not unique, but represents elements of the run.

Recall our assumption that $\mathcal{L}(\mathcal{P}) = \bigcap_{1 \leq j \leq k} \mathcal{L}(\mathcal{V}_i)$, and denote $\mathcal{V}_j = \langle \Sigma, Q_j, I_j, \delta_j, F_j \rangle$ for all $j \in [k]$. Let $Q_{\max} = \max\{|Q_j|\}_{j=1}^{k}$ and denote $\alpha = Q_{\max}!$. Further recall that we focus on words of the form $a^{m_1}\#a^{m_2}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$ for integers $\{m_i\}_{i=1}^{k+1}, m_b, m_c \in \mathbb{N}$, and that we refer to the infix a^{m_i} as the i-th segment, for $1 \leq i \leq k$. We proceed to formally define good and bad segments.

▶ **Definition 13** (Good and Bad Segments)**.** *The i-th segment is* bad *in $\mathcal{V}_j$ if there exist constants $\{m_i\}_{i=1}^{k+1}, m_b, m_c \in \mathbb{N}$ such that the following hold.*

(a) *$\{m_i\}_{i=1}^{k+1}, m_b, m_c$ are multiples of α, and*

[1] The colours were chosen as accessible for the colourblind. For a greyscale-friendly version, see Appendix B.

(b) *there is an accepting $\mathbb{N}$-run ρ of $\mathcal{V}_j$ on $w = a^{m_1}\#a^{m_2}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$ that adheres to one of the three forms:*

(i) $a^{m_1}\#a^{m_2}\#\cdots a^{m_{i-1}}\#a^{m_i}\#a^{m_{i+1}}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$,

(ii) $a^{m_1}\#a^{m_2}\#\cdots a^{m_{i-1}}\#a^{m_i}\#a^{m_{i+1}}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$, *or*

(iii) $a^{m_1}\#a^{m_2}\#\cdots a^{m_{i-1}}\#a^{m_i}\#a^{m_{i+1}}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$.

The i-th segment is good *in $\mathcal{V}_j$ if it is not bad in $\mathcal{V}_j$.*

Lemma 14 formalises the intuition that a bad segment can be pumped simultaneously with both the b and c segments, giving rise to a word accepted by $\mathcal{V}_j$ but rejected by $\mathcal{P}$.

Intuitively, Forms (ii) and (iii) indicate that all segments are bad. Indeed, the i-th segment has a ≥ 0 cycle, so it can be pumped safely, and in Form (ii) both b and c can be pumped using ≥ 0 cycles. Whereas in Form (iii) we can pump b using a > 0 cycle, and can use it to compensate for pumping c, even if the latter requires iterating a negative cycle.

Form (i) is the interesting case, where we use a > 0 cycle in the i-th segment to compensate for pumping both b and c. The requirement that all segments up to the i-th are ≥ 0 is at the core of our proof and is explained in Section 4.4.

▶ **Lemma 14.** *Suppose the l-th segment is bad in $\mathcal{V}_j$, then there exist $x, y, z \in \mathbb{N}$, that are multiples of α, such that for every $n \in \mathbb{N}$ the following word w is accepted by $\mathcal{V}_j$.*

$$w_n = a^{m_1}\#a^{m_2}\#\cdots\#a^{m_{l-1}}\#a^{m_l+xn}\#a^{m_{l+1}}\#\cdots\#a^{m_{k+1}}\#b^{m_b+yn}c^{m_c+zn}$$

Proof. We can choose $z = \alpha$, then take y to be large enough so that Form (iii) runs can compensate for negative cycles in c^z using > 0 cycles in b^y, whilst not decreasing the counters in Form (ii) runs. We can indeed find such a $y \in \mathbb{N}$ that is a multiple of α, since α is divisible by all lengths of simple cycles. Finally, we choose x so that Form (i) runs can compensate for c^z and b^y using > 0 cycles on a^x in the l-th segment, again whilst not decreasing the counters in Forms (ii) and (iii). ◀

Recall that our goal is to show that there is a segment $l \in [k+1]$ that is bad in *every* V_j, for $j \in [k]$. In Lemma 15, We show that each V_j has at most one good segment. Therefore, there are at most k good segments in total, leaving at least one segment that is bad in every $\mathcal{V}_j$, as desired.

▶ **Lemma 15.** *Let $j \in [k]$ and $0 \leq r < s \leq k+1$. Then the r-th or s-th segment is bad in $\mathcal{V}_j$.*

Proof. Since j is fixed, denote $\mathcal{V}_j = \langle \Sigma, Q, Q_0, \delta, F \rangle$. We inductively define constants $\{n_i\}_{i=1}^{k+1}, n_b, n_c \in \mathbb{N}$ as follows. Suppose that n_1 is a large-enough multiple of α so that Lemma 10 guarantees a ≥ 0 cycle in any accepting run of $\mathcal{V}_j$ on a^{n_1} from some $(q_0, 0)$ with $q_0 \in Q_0$. Now, assume that we have defined $n_1, \ldots n_{l-1}$, and consider the word $u = a^{n_1}\#a^{n_2}\#\cdots\#a^{n_{l-1}}\#$. Define $n = |u| \cdot W$ where W is the maximal update of any transition of $\mathcal{V}_j$. Since u consists of $\frac{n}{W}$ letters, $n+1$ is greater than any counter value that can be observed in any run of $\mathcal{V}_j$ on u. We define n_l to be a multiple of α large enough so that Lemma 10 guarantees a ≥ 0 cycle when reading a^{n_l} from any configuration of the form $\{(q, n') \mid q \in Q,\ n' \leq n+1\}$. We set $n_b = n_c = \alpha$, the choice of n_b, n_c is somewhat arbitrary. Finally, we set $w = a^{n_1}\#\cdots\#a^{n_{k+1}}\#b^{n_b}c^{n_c}$.

Now, for every $x \in \mathbb{N}$, we obtain from w a word w_x by pumping $x\alpha$ many a's in the r-th and s-th segments and pumping $x\alpha$ many b's and c's in their segments. That is, let $n'_i = n_i + x\alpha$ for $i \in \{r, s\}$ and $n'_i = n_i$ for $i \notin \{r, s\}$, and let $n'_b = n_b + x\alpha$ and $n'_c = n_c + x\alpha$, then $w_x = a^{n'_1}\#\cdots\#a^{n'_{k+1}}\#b^{n'_b}c^{n'_c}$. Observe that $w_x \in \mathcal{L}(\mathcal{P})$. Indeed, since $n_r \geq n_b = \alpha$ and $n_s \geq n_c = \alpha$ we have that $n_r + x\alpha \geq n_b + x\alpha$ and $n_s + x\alpha \geq n_c + x\alpha$, so the r-th and

s-th segments can already pay for the b's and c's, respectively. In particular, $w_x \in \mathcal{L}(\mathcal{V}_j)$ via some accepting $\mathbb{N}$-run ρ_x.

We choose a particular value of x, as follows. Consider x and suppose some accepting $\mathbb{N}$-run ρ_x as above does not traverse a > 0 cycle neither in r-th nor s-th segment. By Lemma 9, the maximal possible counter value of ρ_x after reading

$$a^{n_1}\#\cdots\#a^{n_r+x\alpha}\#\cdots\#a^{n_s+x\alpha}\#\cdots\#a^{n_{k+1}}\#$$

is $M_b = (k+1+\sum_{z\in[k+1]\setminus\{r,s\}} n_z)\cdot W + 2|Q|\cdot W$. Crucially, this value does not depend on x. Further, if there is no > 0 cycle in the segment of b's as well, again the maximal counter value of ρ up to the c segment is bounded by $M_c = (k+2+\sum_{z\in[k+1]\setminus\{r,s\}} n_z)\cdot W + 3|Q|\cdot W$, that is independent of x and M_b. By Lemma 10, we can now choose x large enough to satisfy that for every accepting $\mathbb{N}$-run ρ_x on w_x:

1. If ρ_x does not traverse any > 0 cycle in the r-th or s-th segments, then ρ_x has a ≥ 0 cycle reading $b^{(n_b+x\alpha)}$ from any configuration in $\{(q, M') \mid q \in Q,\ M' \leq M_b\}$.
2. If ρ_x does not traverse any > 0 cycle in the r-th or s-th segment, nor in the b segment, then ρ_x has a ≥ 0 cycle reading $c^{(n_c+x\alpha)}$ from any configuration in $\{(q, M')|q \in Q,\ M' \leq M_c\}$.

Having fixed x, we claim that for the constants of w_x, one of the r-th or s-th segment is bad in $\mathcal{V}_j$. By construction, Lemma 10 guarantees that ρ_x has ≥ 0 cycles in segments $1, \ldots r-1$. If ρ_x has a > 0 cycle in segment r, then ρ_x is of Form (i):

$$a^{n_1}\#a^{n_2}\#\cdots\#a^{n_{r-1}}\#a^{n_r+x\alpha}\#\cdots\#a^{n_s+x\alpha}\#\cdots\#a^{n_{k+1}}\#b^{n_b+x\alpha}c^{n_c+x\alpha}$$

and so the r-th segment must be bad in $\mathcal{V}_j$.

Otherwise, if ρ_x does not have a > 0 cycle in the r-th segment, then the construction in Lemma 10 guarantees ≥ 0 cycles in segments indexed $r, r+1, \ldots, s-1$. Indeed, for the r-th segment, we are guaranteed a ≥ 0 cycle reading a^{n_r}, all the more for $a^{n_r+x\alpha}$. As for segments indexed $r+1, \ldots s-1$, if ρ_x does not have a > 0 cycle in the r-th segment, then the maximal effect of segment r is $|Q|\cdot W$. However, n_{r+1} was constructed to guarantee a ≥ 0 cycle even in case the effect of segment r is $Wn_r \geq W\alpha \geq W|Q|$.

If there is a > 0 cycle in segment s, then ρ_x is again of Form (i):

$$a^{n_1}\#a^{n_2}\#\cdots\#a^{n_{s-1}}\#a^{n_s+x\alpha}\#a^{n_{s+1}}\#\cdots\#a^{n_{k+1}}\#b^{n_b+x\alpha}c^{m_c+x\alpha}$$

and so the s-th segment must be bad in $\mathcal{V}_j$.

Otherwise, using the same arguments as for the r-th segment, we have that segments indexed $s+1, \ldots, i_{k+1}$ each contain a ≥ 0 cycle. In this case we are left with the b and c segments. The choice of x guarantees a ≥ 0 cycle in the b segment. If ρ_x traverses a > 0 cycle in the b segment, then w_x is of Form (iii).

$$a^{n_1}\#a^{n_2}\#\cdots\#a^{n_{k+1}}\#b^{n_b+x\alpha}c^{n_c+x\alpha}$$

Finally, if there are no > 0 cycles in the b segment, then the choice of x again guarantees a ≥ 0 cycle in the c segment, so w_x is of Form (ii).

$$a^{n_1}\#a^{n_2}\#\cdots\#a^{n_{k+1}}\#b^{n_b+x\alpha}c^{n_c+x\alpha}$$

In the two latter cases, both the r-th and the s-th segments are bad in $\mathcal{V}_j$. ◀

4.4 Proof of Theorem 8

Given Lemma 15, we now know that each $\mathcal{V}_j$ has at most one good segment. Therefore, all 1-CNs $\mathcal{V}_1, \ldots, \mathcal{V}_k$ together have at most k good segments. Recall that the words we focus on have $k+1$ segments, and therefore there is at least one segment, say the l-th segment, that is bad in every $\mathcal{V}_j$. Note, however, that this segment may correspond to different constants in each $\mathcal{V}_j$. That is, there exists constants $\{m_i^j, m_b^j, m_c^j \mid i \in [k+1], j \in [k]\}$ witnessing that the l-th segment is bad for each $\mathcal{V}_j$. We group the $\mathcal{V}_j$ according to the form of their accepting runs ρ_j (see Definition 13):

(i) $a^{m_1^j} \# a^{m_2^j} \# \cdots \# a^{m_l^j} \# a^{m_{l+1}^j} \# \cdots \# a^{m_{k+1}^j} \# b^{m_b^j} c^{m_c^j}$,

(ii) $a^{m_1^j} \# a^{m_2^j} \# \cdots \# a^{m_l^j} \# a^{m_{l+1}^j} \# \cdots \# a^{m_{k+1}^j} \# b^{m_b^j} c^{m_c^j}$, or

(iii) $a^{m_1^j} \# a^{m_2^j} \# \cdots \# a^{m_l^j} \# a^{m_{l+1}^j} \# \cdots \# a^{m_{k+1}^j} \# b^{m_b^j} c^{m_c^j}$.

We now find constants resulting in a single word for which the l-th segment is bad in every $\mathcal{V}_j$. First, for $i \in [k+1] \setminus \{l\}$, we define $M_i = \max\{m_i^j \mid j \in [k]\}$, note that these values are still multiples of α. Similarly, we define $M_c = \max\{m_c^j \mid j \in [k]\}$. It remains to fix new constants L and B, which we do in phases in the following. The resulting word is then
$w = a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^L \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^B c^{M_c}$.

Most steps in the analysis below are based on Lemma 11 and Corollary 12. We first, partially, handle Form (iii) runs. For such $\mathcal{V}_j$, there is an accepting $\mathbb{N}$-run ρ_j on
$a^{m_1^j} \# \cdots \# a^{m_{l-1}^j} \# a^{m_l^j} \# a^{m_{l+1}^j} \# \cdots \# a^{m_{k+1}^j} \# b^{m_b^j} c^{m_c^j}$
By pumping ≥ 0 cycles as per Corollary 12 in all segments except l we obtain an accepting $\mathbb{N}$-run ρ'_j on
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^{m_l^j} \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^{m_b^j} c^{m_c^j}$.
We now pump arbitrary cycles in the c segment to construct a $\mathbb{Z}$-run ρ''_j on
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^{m_l^j} \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^{m_b^j} c^{M_c}$.
Next, we compensate for possible negative cycles in the c segment by pumping a > 0 cycle in the b segment. Thus, we construct an $\mathbb{N}$-run ρ'''_j on
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^{m_l^j} \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^B c^{M_c}$,
where B is chosen to be large enough such that ρ'''_j is an $\mathbb{N}$-run for all $\mathcal{V}_j$, $j \in [k]$. Note that it remains to fix L.

We now turn to Form (i) with a similar process We start with an accepting $\mathbb{N}$-run ρ_j on
$a^{m_1^j} \# \cdots \# a^{m_{l-1}^j} \# a^{m_l^j} \# a^{m_{l+1}^j} \# \cdots \# a^{m_{k+1}^j} \# b^{m_b^j} c^{m_c^j}$.
Pump ≥ 0 cycles in segments indexed $1, \ldots, l-1$ to obtain an accepting $\mathbb{N}$-run ρ'_j on
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^{m_l^j} \# a^{m_{l+1}^j} \# \cdots \# a^{m_{k+1}^j} \# b^{m_b^j} c^{m_c^j}$.
Now, obtain a $\mathbb{Z}$-run ρ''_j by pumping arbitrary cycles in the remaining segments, including the b segment.
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^{m_l^j} \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^B c^{M_c}$
Again, compensate for negative cycles by taking L large enough so that pumping > 0 cycles in the l-th segment yields an accepting $\mathbb{N}$-run ρ'''_j on
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^L \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^B c^{M_c}$.

We now return to Form (iii) and fix the l-th segment by pumping ≥ 0 cycles to construct an accepting $\mathbb{N}$-run on
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^L \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^B c^{M_c}$.

We are left with Form (ii), which are the most straightforward to handle. We simply pump ≥ 0 cycles in all segments to construct an accepting $\mathbb{N}$-run ρ'_j on
$a^{M_1} \# \cdots \# a^{M_{l-1}} \# a^L \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^B c^{M_c}$.

Note that the requirement for all segments before the l-th to be ≥ 0 is crucial, otherwise we won't be able to pump all the cycles in all forms simultaneously.

We now have that w is accepted by every $\mathcal{V}_j$, and the l-th segment is bad for all $\mathcal{V}_j$. By applying Lemma 14 for each of the $\mathcal{V}_j$ and taking global constants to be the products of the respective constants $x, y, z > 0$ for each $\mathcal{V}_j$, we now obtain $X, Y, Z \in \mathbb{N}$, multiples of α, such that for every $n \in \mathbb{N}$ the word

$$w_n = a^{M_1}\# \cdots \# a^{M_{l-1}} \# a^{L+Xn} \# a^{M_{l+1}} \# \cdots \# a^{M_{k+1}} \# b^{B+Yn} c^{M_c+Zn} \in \mathcal{L}(\mathcal{V}_j) \ \ \forall j \in [k].$$

is accepted by every $\mathcal{V}_j$.

Finally, we choose n large enough to satisfy $\sum_{i \in [k+1] \setminus \{l\}} M_i < \min\{B + Yn, M_c + Zn\}$, so that $w_n \notin \mathcal{L}(\mathcal{P})$. This is possible because, w.l.o.g, the l-th segment can only pay for b, and the remaining segments $[k+1] \setminus \{l\}$ cannot pay for c. This contradicts the assumption that $\mathcal{L}(\mathcal{P}) = \bigcap_{j \in [k]} \mathcal{L}(\mathcal{V}_j)$, concluding the proof of Theorem 8. ◀

▶ Remark 16 (Unbounded Compositeness). The proof of Theorem 8 shows that if words with $k+1$ segments are allowed, then the language is not $(1, k)$-composite, we use this to establish primality. By intersecting $\mathcal{L}(\mathcal{P})$ with words that allow at most $k+1$ segments, we obtain a language that is not $(1, k)$-composite, but it is not hard to show that it is $(1, 2^{k+1})$-composite. This demonstrates that a 2-CN can be composite, but may require unboundedly many factors.

5 Primality of Counter Nets is Undecidable

In this section we consider the *primality* and *dimension-minimality* decision problems: given a k-CN $\mathcal{A}$, decide whether $\mathcal{A}$ is prime and whether $\mathcal{A}$ is dimension-minimal, respectively.

We use our prime 2-CN from Example 7 and the results of Section 4 to show that both problems are undecidable. Our proof is by reduction from the containment problem[2] for 1-CN: given two 1-CN $\mathcal{A}, \mathcal{B}$ over alphabet Σ, decide whether $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$. This problem was shown to be undecidable in [19].

We begin by describing the reduction that applies to both problems. Consider an instance of 1-CN containment with two 1-CNs $\mathcal{A}$ and $\mathcal{B}$ over the alphabet Σ. We construct a 2-CN $\mathcal{C}$ as follows. Let Λ be the alphabet of the 2-CN from Example 7 and Theorem 8, and let $\$ \notin \Sigma \cup \Lambda$ be a fresh symbol. Intuitively, $\mathcal{C}$ accepts words of the form $u\$v$ when either $u \in \mathcal{L}(\mathcal{A})$ and v is accepted by $\mathcal{P}$ starting from the maximal counter $\mathcal{A}$ ended with on u, or when $u \in \mathcal{L}(\mathcal{B})$ and $v \in \Lambda^*$.

Formally, we convert $\mathcal{A}$ and $\mathcal{B}$ to 2-CNs $\mathcal{A}'$ and $\mathcal{B}'$ by adding a counter and never modifying its value, so a transition (p, σ, v, q) in $\mathcal{A}$ becomes $(p, \sigma, (v, 0), q))$ in $\mathcal{A}'$, for example. We construct a 2-CN $\mathcal{C}$ as follows (see Figure 3). We take $\mathcal{A}'$, $\mathcal{B}'$, and $\mathcal{P}$, and for every accepting state q of $\mathcal{A}'$ we introduce a transition $(q, \$, \mathbf{0}, p_0)$ where p_0 is an initial state of $\mathcal{P}$. We then add a new accepting state $q_\top$ and add the transitions $(q_\top, \lambda, \mathbf{0}, q_\top)$ for every letter $\lambda \in \Lambda$, in other words $q_\top$ is an accepting sink for Λ. We also add transitions $(s, \$, \mathbf{0}, q_\top)$ from every accepting state s of $\mathcal{B}'$. The initial states are those of $\mathcal{A}'$ and $\mathcal{B}'$, and the accepting states are those of $\mathcal{P}$ and $q_\top$.

▶ **Theorem 17.** *Primality and dimension-minimality are undecidable, already for* 2*-CN.*

Proof. We prove the theorem by establishing that $\mathcal{C}$ is not prime if and only if $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$, and $\mathcal{C}$ is not dimension-minimal if and only if $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$.

[2] Actually, the complement thereof.

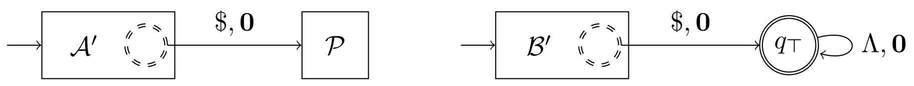


Figure 3 The reduction from 1-CN non-containment to 2-CN primality and dimension-minimality. The dashed accepting states are those of $\mathcal{A}'$ and $\mathcal{B}'$, and are not accepting in the resulting construction.

Assume that $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$, then the component of $\mathcal{C}$ containing $\mathcal{A}'$ and $\mathcal{P}$ (left-hand side of Figure 3) becomes redundant. Since the component containing $\mathcal{B}'$ and $q_\top$ only makes use of one counter, $\mathcal{C}$ is composite. Formally, we claim that $\mathcal{L}(\mathcal{C}) = \{u\$v \mid u \in \mathcal{L}(\mathcal{B}) \wedge v \in \Lambda^*\}$. Indeed, if $w \in \mathcal{L}(\mathcal{C})$ then $w = u\$v$ so either $u \in \mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$ or $u \in \mathcal{L}(\mathcal{B})$, but since $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$, this is equivalent to $u \in \mathcal{L}(\mathcal{B})$, and in this case there is simply no condition on $v \in \Lambda^*$. Since the second counter is not used in component containing $\mathcal{B}'$ and $q_\top$ (right-hand side of Figure 3), we can construct a 1-CN equivalent to $\mathcal{C}$ by projecting on the first counter and just deleting the component containing $\mathcal{A}'$ and $\mathcal{P}$ completely. It follows that in this case $\mathcal{C}$ is not dimension-minimal, and therefore is not prime either.

For the converse, assume that $\mathcal{L}(\mathcal{A}) \not\subseteq \mathcal{L}(\mathcal{B})$, and let $u \in \mathcal{L}(\mathcal{A}) \setminus \mathcal{L}(\mathcal{B})$. Denote $m = \max\{\mathsf{eff}(\rho) \mid \rho \text{ is an accepting run of } \mathcal{A} \text{ on } u\}$. Thus, for a word $v \in \Lambda^*$ we have that $u\$v \in \mathcal{L}(\mathcal{C})$ if and only if v is accepted in $\mathcal{P}$ with initial counter m. Assume by way of contradiction that $\mathcal{C}$ is not prime, then we can write $\mathcal{L}(\mathcal{C})$ as an intersection of languages of 1-CNs. Loosely speaking, this will create a contradiction as we will be able to argue that $\mathcal{P}$ is not prime. More precisely, take $v = a^{m_1}\#a^{m_2}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$ for integers $\{m_i\}_{i=1}^{k+1}, m_b, m_c \in \mathbb{N}$ and consider words of the form $u\$v$. Our analysis from Section 4—specifically the arguments used in the proof Lemma 15—on $u\$v$ can show, mutatis mutandis, that the language of $\mathcal{P}$ is not composite regardless of any fixed initial counter value (an analogue of Theorem 8).

We thus have that $\mathcal{C}$ is prime, and in particular $\mathcal{C}$ is dimension-minimal, concluding the correctness of the reduction. ◀

6 Regularity of Deterministic Counter Nets is Decidable

We now turn our attention to decidable fragments of primality. A natural candidate for decidability is the deterministic fragment. Recall that by Proposition 4, a k-DCN is dimension-minimal if and only if it is not $(1, k-1)$-composite. Thus, dimension-minimality "captures" primality, and is our focus. Following, we show that regularity, equivalent to being $(0, 1)$-composite, is decidable for k-DCN.

Our proof approach is to reduce the problem to regularity of Vector Addition Systems (VAS). In our context, a VAS is a single-state k-DCN. Given the single state and determinism, we can then associate the letter corresponding to each transition with the transition itself. Regularity of VAS was shown to be decidable, and in fact in EXPSPACE, in [3, Theorem 4.5].

▶ Remark 18. Regularity of VAS with states (VASS) was shown to be decidable in [12]. However, the model there does not have accepting states. It appears to be possible to extend the techniques used there to our setting, but would this require re-proving several results. Therefore, we take the following approach through VAS. Additionally, we note that a stronger result appears in an unpublished manuscript (at the time of writing of this paper) [5].

Throughout this section, we consider a k-DCN $\mathcal{D} = \langle \Sigma, Q, Q_0, \delta, F \rangle$. Our construction proceeds as follows. First, we obtain from $\mathcal{D}$ a k-DCN $\mathcal{C}$ with the same structure, but such that its transitions are distinctly labelled. Next, we invoke a technique of Hopcroft and Pansiot [21]; we will construct a single-state $(k+3)$-DCN (namely a VAS) $\mathcal{U}$ whose language "approximates" that of $\mathcal{C}$, and in particular preserves regularity. Finally, we invoke the deciability of regularity for deterministic VAS [3]. The crux of the argument is to maintain regularity throughout, even though the language changes in each step. In the following, a *DFA* (resp. NFA) is a 0-DCN (resp. 0-CN).

▶ **Lemma 19.** *Let $\mathcal{D}$ be as above, and define $\mathcal{C} = \langle \Gamma, Q, Q_0, \delta', F \rangle$ where $\Gamma = \{\gamma_t \mid t \in \delta\}$ with $(p, \gamma_t, \boldsymbol{v}, q) \in \delta'$ iff $t = (p, \sigma, \boldsymbol{v}, q) \in \delta$. Then $\mathcal{L}(\mathcal{D})$ is regular if and only if $\mathcal{L}(\mathcal{C})$ is regular.*

Proof. For the first direction, assume $\mathcal{L}(\mathcal{C})$ is regular, we show that $\mathcal{L}(\mathcal{D})$ is regular. Let $\mathcal{A}$ be a DFA such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{C})$. Now consider the process of reverting the labelling, to construct an NFA $\mathcal{A}'$ with the same state space and same transition structure, but different labels. For a transition (s, γ_t, t) in $\mathcal{A}$, let $t = (p, \sigma, \boldsymbol{v}, q)$ be the transition in $\mathcal{D}$ corresponding to γ_t, then we introduce in $\mathcal{A}'$ the transition (s, σ, t). Note that $\mathcal{A}'$ may be nondeterministic.

Now, since $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{C})$, we claim that $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{D})$. Consider $w \in \mathcal{L}(\mathcal{A}')$ with $w = \sigma_1 \cdots \sigma_n$, then there exist transitions $t_1, \ldots, t_n$ where each t_i is labelled σ_i such that $\gamma_{t_1} \cdots \gamma_{t_n} \in \mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{C})$. The sequence $t_1, \ldots, t_n$ represents an accepting run of $\mathcal{D}$, which is labelled by $w = \sigma_1 \cdots \sigma_n$, so $w \in \mathcal{L}(\mathcal{D})$. Conversely, consider $w \in \mathcal{L}(\mathcal{D})$ with $w = \sigma_1 \cdots \sigma_n$, then the sequence of transitions in the accepting run of $\mathcal{D}$ on w is $t_1, \ldots, t_n$ such that $\gamma_{t_1} \cdots \gamma_{t_n} \in \mathcal{L}(\mathcal{C}) = \mathcal{L}(\mathcal{A})$, so by the construction of $\mathcal{A}'$ we have that $w = \sigma_1 \cdots \sigma_n \in \mathcal{L}(\mathcal{A}')$. We conclude that if $\mathcal{L}(\mathcal{C})$ is regular, then so is $\mathcal{L}(\mathcal{D})$.

For the second direction, assume $\mathcal{L}(\mathcal{D})$ is regular, we show that $\mathcal{L}(\mathcal{C})$ is regular. Let $\mathcal{A}$ be a DFA such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{D})$. We consider a product DFA $\mathcal{A} \times \mathcal{D}$ over the alphabet Γ as follows. When reading letter γ_t, for $t = (p, \sigma, \boldsymbol{v}, q)$, it transitions from state (r, p) to (r', q), where r' is reached when reading σ from r in $\mathcal{A}$ and where $q \in Q$ is reached when taking transition t from $p \in Q$. Acceptance is determined by $\mathcal{A}$. Intuitively, $\mathcal{A} \times \mathcal{D}$ simulates $\mathcal{A}$ while ensuring that the transitions being read actually form a $\mathbb{Z}$-run of $\mathcal{D}$.

We claim that $\mathcal{L}(\mathcal{A} \times \mathcal{D}) = \mathcal{L}(\mathcal{C})$. Indeed, $\mathcal{C}$ accepts a sequence $\gamma_{t_1} \cdots \gamma_{t_n}$ if and only if $t_1 \cdots t_n$ forms an accepting run in $\mathcal{D}$, if and only if $t_1 \cdots t_n$ forms a $\mathbb{Z}$-run of $\mathcal{D}$ and the letters on the transitions are $\sigma_1 \cdots \sigma_n \in \mathcal{L}(\mathcal{D}) = \mathcal{L}(\mathcal{A})$. Note that this last double implication relies on the fact that $\mathcal{D}$ is deterministic, hence there is a single run of $\mathcal{D}$ on any given word. ◀

We now employ a well-known technique of Hopcroft and Pansiot [21, Lemma 2.1] that converts a k-VASS to a $(k+3)$-VAS by simulating the finite control states using three extra dimensions. The first k dimensions just mirror the k dimensions in the original VASS and the last three dimensions operate the state simulation; one of the three extra dimensions explicitly maintains a value corresponding to the current state. In the unlabelled (VASS) setting, the construction takes a transition $(p, \boldsymbol{x}, q)$ and spawns three transitions $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$ in the corresponding VAS. Crucially, these vectors are chosen in such a way that if $(p, \boldsymbol{x}, q)$ is taken in the original VASS, then $\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3$ can be taken in sequence. Moreover, if $\boldsymbol{v}_1$ is taken, then the only transition that could follow is $\boldsymbol{v}_2$, and then the only transition that could follow is $\boldsymbol{v}_3$; ending with the last three dimensions taking values corresponding to q.

Following, in Lemma 20 (proof in Appendix A.3), we show that one can replicate this construction starting with the distinctly-labelled k-DCN $\mathcal{C}$ and obtaining a single-state $(k+3)$-DCN $\mathcal{U}$. In particular, if a σ labelled transition, $(p, \sigma, \boldsymbol{v}, q)$, is present in the k-DCN $\mathcal{C}$, then labelled transitions $(\sigma_1, \boldsymbol{v}_1)$, $(\sigma_2, \boldsymbol{v}_2)$, and $(\sigma_3, \boldsymbol{v}_3)$ are present in the constructed $(k+3)$-DCN $\mathcal{U}$.

▶ **Lemma 20** (Corollary of [21, Lemma 2.1])**.** *Let $\mathcal{C}$ be a distinctly-labelled k-DCN over alphabet Γ (obtained as per Lemma 19), then there exists a single-state $(k+3)$-DCN $\mathcal{U}$ over alphabet $\Upsilon = \{\gamma_1, \gamma_2, \gamma_3 \mid \gamma \in \Gamma\}$ such that $\mathcal{L}(\mathcal{U}) = \{a_1a_2a_3\, b_1b_2b_3 \cdots c_1,\ a_1a_2a_3\, b_1b_2b_3 \cdots c_1c_2,\ a_1a_2a_3\, b_1b_2b_3 \cdots c_1c_2c_3 \mid a\, b \cdots c \in \mathcal{L}(\mathcal{C}) \wedge a, b, \ldots, c \in \Gamma\}$.*

▶ **Proposition 21.** *Let $L \subseteq \{a, b, \dots, c\}^*$ and $L' \subseteq \{a_1, a_2, a_3, b_1, b_2, b_3, \dots, c_1, c_2, c_3\}^*$ such that $L' = \{a_1a_2a_3\, b_1b_2b_3 \cdots c_1,\ a_1a_2a_3\, b_1b_2b_3 \cdots c_1c_2,\ a_1a_2a_3\, b_1b_2b_3 \cdots c_1c_2c_3 \mid a\, b \cdots c \in L\}$, then L is regular if and only if L' is regular.*

Combining Lemma 19, Lemma 20, Proposition 21, and [3, Theorem 4.5], we can conclude that regularity is decidable for k-DCNs.

▶ **Theorem 22.** *Regularity of k-DCN is decidable and is in* EXPSPACE*.*

7 Expressiveness Trade-Offs between Dimensions and Nondeterminism

Theorem 8 shows that, unsurprisingly, increasing the dimension increases expressiveness, and that nondeterministic models are more expressive than deterministic ones. In particular, a k-DCN can be decomposed by projection (Proposition 4), and have decidable regularity (Theorem 22). It is therefore interesting to study the interplay between increasing the dimension and introducing nondeterminism. In this section we show that the two notions are incomparable, in a sense.

We start by showing that nondeterminism can sometimes compensate for dimension. Let $k \in \mathbb{N}$ and $\Sigma = \{a_1, \dots, a_k, b_1, \dots, b_k, c\}$; consider the language $L_k = \{a_1^{n_1} a_2^{n_2} \cdots a_k^{n_k} b_i c^m \mid i \in [k] \wedge n_i \geq m\}$. It is easy to construct a k-DCN as well as a 1-CN for L_k, as depicted by Figures 4 and 5 for $k = 3$. To construct a 1-CN we guess which b_i will be later read, and verify the guess using the single counter in the $a_i^{n_i}$ part.

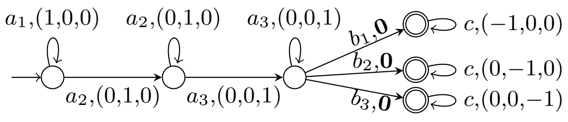


Figure 4 A 3-DCN for $L_3 = \{a_1^{n_1} a_2^{n_2} a_3^{n_3} b_i c^m \mid i \in [3] \wedge n_i \geq m\}$. Intuitively, the 3-DCN counts the number of occurrences of each letter, and decreases the appropriate counter once the letter b_i selects it.

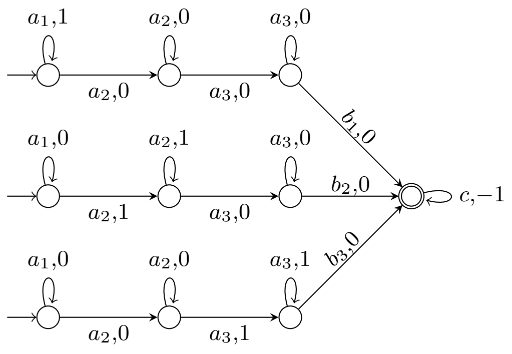


Figure 5 A 1-CN for $L_3 = \{a_1^{n_1} a_2^{n_2} a_3^{n_3} b_i c^m \mid i \in [3] \wedge n_i \geq m\}$. Intuitively, the 1-CN guesses which b_i will be seen, and counts the respective occurrences of the letter a_i. Then, once b_i is seen, the counter is decreased on c.

However, we show that the dimension cannot be minimised whilst maintaining determinism.

▶ **Proposition 23.** *L_k is not recognisable by a $(k-1)$-DCN.*

Proof. Assume by way of contradiction that there exists a $(k-1)$-DCN $\mathcal{D} = \langle \Sigma, Q, Q_0, \delta, F \rangle$ such that $\mathcal{L}(\mathcal{D}) = L_k$. Let $n > |Q|$ and for every $i \in [k]$ consider the word $w_i = a_1^n a_2^n \cdots a_k^n b_i c^n \in L_k$. Since $\mathcal{D}$ is deterministic and $n > |Q|$, all of the accepting runs on the w_i coincide up to the b_i part and have cycles in each a_i^n segment as well as in the c^n segment (the latter may differ according to i). Let M be the product of the lengths of all these cycles.

First, observe that the cycles in all of the a_i^n segments cannot decrease any counter. Indeed, otherwise by pumping such a cycle for large enough $t > 0$ times, there would not exist an $\mathbb{N}$-run on words with the prefix $a_1^n \cdots a_{i-1}^n a_i^{n+tM}$. This creates a contradiction since, with an appropriate suffix, such words can be accepted.

Thus, all a_i cycles have non-negative effects for all counters. Indeed, for each counter i – associate with i the minimal segment index whose cycle strictly increases i. Since there are $k-1$ counters and k segments this map is not surjective, in other words, there is a segment (without loss of generality, the a_k segment) such that every counter that is increased in the a_k cycle is also increased in a previous segment. Therefore, there exist $s, t > 0$ such that the word

$$a_1^{n+sM} a_2^{n+sM} \cdots a_{k-1}^{s+sM} a_k^n b_k c^{n+tM} \notin L_k$$

is accepted by $\mathcal{D}$, which is a contradiction. ◀

We now turn to show that conversely, dimension can sometimes compensate for non-determinism. Consider the language $K_2 = \{a^k b^\ell c^m d^n \mid k \geq m \wedge \ell \geq n\}$. Standard cycle analysis enables us to argue the following (proof in Appendix A.6).

▶ **Proposition 24.** *K_2 can be recognised by a 2-DCN, but not by a 1-CN.*

8 Discussion

Broadly, this work explores the interplay between a CN's dimension and its expressive power. This is done by studying the *dimension-minimality* problem, where we ask whether a CN's dimension can be decreased while preserving its language, and by the more involved *primality* problem, which allows a decomposition to multiple CNs of lower dimension. We show that both primality and dimension-minimality are undecidable. Moreover, they remain undecidable even when we discard the degenerate dimension 0 case, which corresponds to finite memory, i.e., regular languages. On the other hand, this degenerate case is one where we can show decidability for DCNs.

We conclude with two open problems for future research.

Example of a prime k-CN: We demonstrate a prime 2-CN. A natural question is whether we can find, and indeed prove correctness of, a prime k-CN for every $k \in \mathbb{N}$? And more generally, for every $d < k \in \mathbb{N}$ a k-CN that is (d, n)-composite but not $(d-1, m)$-composite, for some $n, m \in \mathbb{N}$? The difficulty involved in proving Theorem 8 seems to indicate that this might be highly nontrivial.

Decidability of dimension-minimality for k-DCN: In Section 6 we show that regularity is decidable for DCN, but this relies heavily on the finite-memory property of regular languages. Extending this to deciding whether a k-DCN has an equivalent d-DCN for $d < k$ seems to be technically challenging, even for $k = 2$ and $d = 1$.

References

1 Shaull Almagor, Udi Boker, Piotr Hofman, and Patrick Totzke. Parametrized universality problems for one-counter nets. In *31st International Conference on Concurrency Theory (CONCUR 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

2 Shaull Almagor and Asaf Yeshurun. Determinization of one-counter nets. In *33rd International Conference on Concurrency Theory (CONCUR 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

3 Michel Blockelet and Sylvain Schmitz. Model checking coverability graphs of vector addition systems. In Filip Murlak and Piotr Sankowski, editors, *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, volume 6907 of *Lecture Notes in Computer Science*, pages 108–119. Springer, 2011. doi:10.1007/978-3-642-22993-0_13.

4 Michael Blondin, Matthias Englert, Alain Finkel, Stefan Göller, Christoph Haase, Ranko Lazić, Pierre McKenzie, and Patrick Totzke. The reachability problem for two-dimensional vector addition systems with states. *Journal of the ACM (JACM)*, 68(5):1–43, 2021.

5 Sougata Bose, David Purser, and Patrick Totzke. History-deterministic vector addition systems. *arXiv preprint arXiv:2305.01981*, 2023.

6 Maria Paola Cabasino, Alessandro Giua, and Carla Seatzu. Diagnosability of discrete-event systems using labeled petri nets. *IEEE Transactions on Automation Science and Engineering*, 11(1):144–153, 2013.

7 Wojciech Czerwiński, Diego Figueira, and Piotr Hofman. Universality problem for unambiguous vass. In *31st International Conference on Concurrency Theory (CONCUR 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

8 Wojciech Czerwiński, Sławomir Lasota, Ranko Lazić, Jérôme Leroux, and Filip Mazowiecki. Reachability in fixed dimension vector addition systems with states. In *31st International Conference on Concurrency Theory (CONCUR 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

9 Wojciech Czerwiński, Slawomir Lasota, Christof Löding, and Radoslaw Piórkowski. New pumping technique for 2-dimensional vass. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

10 Wojciech Czerwiński, Sławomir Lasota, and Lukasz Orlikowski. Improved lower bounds for reachability in vector addition systems. In *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

11 Wojciech Czerwiński and Lukasz Orlikowski. Reachability in vector addition systems is ackermann-complete. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1229–1240. IEEE, 2022.

12 Stéphane Demri. On selective unboundedness of VASS. *J. Comput. Syst. Sci.*, 79(5):689–713, 2013. doi:10.1016/j.jcss.2013.01.014.

13 Javier Esparza. Decidability and complexity of petri net problems—an introduction. *Lectures on Petri Nets I: Basic Models: Advances in Petri Nets*, pages 374–428, 2005.

14 Diego Figueira. *Co-finiteness of VASS coverability languages.* PhD thesis, LaBRI, CNRS UMR 5800, 2019.

15 Alain Finkel, Jérôme Leroux, and Grégoire Sutre. Reachability for two-counter machines with one test and one reset. In *FSTTCS 2018-38th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 122, pages 31–1. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

16 Sheila A. Greibach. Remarks on blind and partially blind one-way multicounter machines. *Theoretical Computer Science*, 7(3):311–324, 1978.

17 Christoph Haase, Stephan Kreutzer, Joël Ouaknine, and James Worrell. Reachability in succinct and parametric one-counter automata. In *CONCUR 2009-Concurrency Theory: 20th*

International Conference, CONCUR 2009, Bologna, Italy, September 1-4, 2009. Proceedings 20, pages 369–383. Springer, 2009.

18 Michel Henri Theódore Hack. Petri net language. *Computation Structures Group Memo 124*, 1976. URL: `http://publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TR-159.pdf`.

19 Piotr Hofman, Richard Mayr, and Patrick Totzke. Decidability of weak simulation on one-counter nets. In *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 203–212. IEEE, 2013.

20 Piotr Hofman and Patrick Totzke. Trace inclusion for one-counter nets revisited. In *Reachability Problems: 8th International Workshop, RP 2014, Oxford, UK, September 22-24, 2014. Proceedings 8*, pages 151–162. Springer, 2014.

21 John E. Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theor. Comput. Sci.*, 8:135–159, 1979. `doi:10.1016/0304-3975(79)90041-0`.

22 I. Jecker, N. Mazzocchi, and P. Wolf. Decomposing permutation automata. In *Proc. 32nd CONCUR*, volume 203 of *LIPIcs*, pages 18:1–18:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

23 Ismael R Jecker, Orna Kupferman, and Nicolas Mazzocchi. Unary prime languages. In *45th International Symposium on Mathematical Foundations of Computer Science*, volume 170, 2020.

24 Orna Kupferman and Jonathan Mosheiff. Prime languages. *Information and Computation*, 240:90–107, 2015.

25 Jérôme Leroux. The reachability problem for petri nets is not primitive recursive. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1241–1252. IEEE, 2022.

26 Jérôme Leroux and Grégoire Sutre. On flatness for 2-dimensional vector addition systems with states. In *CONCUR*, volume 4, pages 402–416. Springer, 2004.

27 Elaine Render and Mark Kambites. Rational subsets of polycyclic monoids and valence automata. *Information and Computation*, 207(11):1329–1339, 2009.

28 Rüdiger Valk and Guy Vidal-Naquet. Petri nets and regular languages. *Journal of Computer and system Sciences*, 23(3):299–325, 1981.

A Missing Proofs

A.1 Proof of Proposition 6

If $\mathcal{D}$ is $(1, k-1)$-composite, then there exist 1-DCNs $\mathcal{B}_1, \ldots, \mathcal{B}_{k-1}$ such that $\mathcal{L}(\mathcal{D}) = \bigcap_{i=1}^{k-1} \mathcal{L}(\mathcal{B}_i)$. Define $\mathcal{B}$ to be the product $(k-1)$-CN of $\mathcal{B}_1, \ldots, \mathcal{B}_{k-1}$, then $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{D})$, so $\mathcal{D}$ is not dimension-minimal.

Conversely, if $\mathcal{D}$ is not dimension-minimal, there exists w.l.o.g., a $k-1$-CN $\mathcal{B}$ such that $\mathcal{L}(\mathcal{B}) = \mathcal{L}(\mathcal{D})$. Then, we have $\mathcal{L}(\mathcal{B}) = \bigcap_{i=1}^{k-1} \mathcal{L}(\mathcal{B}|_i)$, so $\mathcal{D}$ is $(1, k-1)$-composite. ◀

A.2 Proof of Lemma 9

We prove the contra-positive: assume ρ has counter value $n + W|Q|$, w.l.o.g. we can assume this occurs at the end. Since the maximal counter increase along a transition is W, it follows that there are at least $|Q|$ indices where the counter increases beyond any previous level. Thus, we can find indices $i_0, i_1, \ldots, i_{|Q|}$ such that the counter increases in the infix of ρ between i_j and i_{j+1}. Since these are $|Q| + 1$ states, then there exists a state that is visited twice, which forms a > 0 cycle in ρ. ◀

A.3 Proof of Lemma 20

We give the construction of the VAS, this proof only differs from the proof given by Hopcroft and Pansoit [21] by the inclusion of transition labels. Assume that $\mathcal{C}$ has n states $Q = \{q_1, \dots, q_n\}$. For each $i \in \{1, \dots, n\}$, let $a_i = i$ and $b_i = (n+1)(n+1-i)$. For each transition $t = (q_i, \sigma, \boldsymbol{x}, q_j)$ in $\mathcal{C}$, there are three transitions $t_1 = \sigma_1, (\mathbf{0}, -a_i, a_{n+1-i} - b_i, b_{n+1-i})$, $t_2 = \sigma_2, (\mathbf{0}, b_i, -a_{n+1-i})$, and $t_3 = \sigma_3, (\boldsymbol{x}, a_j - b_i, b_j, -a_i)$ in $\mathcal{U}$ (we omit the states from the transitions, since there is only one state).

Observe that since each transition in $\mathcal{C}$ has a distinct label, then each of the transitions in $\mathcal{U}$ also has a distinct label, making $\mathcal{U}$ a DCN. It remains to show that $\mathcal{U}$ recognises the language given in the statement of this lemma. Suppose the word $\sigma_1 \sigma_2 \cdots \sigma_k \in \mathcal{L}(\mathcal{C})$, then there is a path $\pi = (t_i)_{i=1}^{k}$ in $\mathcal{C}$, so $t_i = (q_i, \sigma_i, \boldsymbol{x}_i, q_{i+1})$. Since $\mathcal{U}$ faithfully simulates $\mathcal{C}$, the path $\pi' = (t_{i,1}\, t_{i,2}\, t_{i,3})_{i=1}^{k}$ in $\mathcal{U}$ witnesses the word $\sigma_{1,1}\, \sigma_{1,2}\, \sigma_{1,3}\, \sigma_{2,1}\, \sigma_{2,2}\, \sigma_{2,3} \cdots \sigma_{k,1}\, \sigma_{k,2}\, \sigma_{k,3}$. In fact, the last three transitions need not all be executed; indeed $\sigma_{1,1}\, \sigma_{1,2}\, \sigma_{1,3} \cdots \sigma_{k,1} \in \mathcal{L}(\mathcal{U})$, $\sigma_{1,1}\sigma_{1,2}\sigma_{1,3} \cdots \sigma_{k,1}\, \sigma_{k,2} \in \mathcal{L}(\mathcal{U})$, and $\sigma_{1,1}\, \sigma_{1,2}\, \sigma_{1,3} \cdots \sigma_{k,1}\, \sigma_{k,2}\, \sigma_{k,3} \in \mathcal{L}(\mathcal{U})$. ◀

A.4 Proof of Proposition 21

This is a basic exercise in automata. The direction L is regular $\implies$ L' is regular is proved by constructing a DFA for L' that verifies letters are read in triplets (i.e., $a_1a_2a_3$, possibly stopping in the middle), and after every triplet simulates the corresponding transition in a DFA for L.

The direction L' is regular $\implies$ L is regular is proved by constructing an NFA for L that given letter a simulates the transition of a DFA for L' on $a_1a_2a_3$. ◀

A.5 Proof of Theorem 22

Given a k-CN $\mathcal{D}$, construct a distinctly-labelled k-DCN $\mathcal{C}$ as per Lemma 19. Next, apply Lemma 20 to obtain a single-state, distinctly-labelled $k + 3$-DVAS $\mathcal{U}$. Finally, treat $\mathcal{U}$ as a VAS. Indeed, regularity of VAS is with respect to the transition "names", which are unique in $\mathcal{U}$ due to the distinct labels. We can then decide the regularity of $\mathcal{U}$ using [3, Theorem 4.5].

We have that $\mathcal{U}$ is regular iff $\mathcal{L}(\mathcal{D})$ is regular by Lemmas 19 and 20 and Proposition 21. Moreover, our construction can clearly be implemented in polynomial time, thus giving the same complexity bound. ◀

A.6 Proof of Proposition 24

Constructing a 2-DCN for K_2 is easy – the first counter is increased on a and decreased on c, and the second is increased on b and decreased on d.

We now argue that there does not exist a 1-CN recognising K_2. Assume by way of contradiction that there exists a 1-CN $\mathcal{A}$ recognising K_2. Suppose that $\mathcal{A}$ comprises n states and that M is the absolute value of the largest transition effect. We analyse the behaviour of $\mathcal{A}$ when accepting the word $w = a^m b^m c^m d^m$ for $m > n \cdot M$. Since $m > n$, any accepting run on w must take some cycle reading 'a's, some cycle reading 'b's, some cycle reading 'c's, and some cycle reading 'd's.

First, it is never the case that a cycle reading 'c's or 'd's has non-negative effect, otherwise $\mathcal{A}$ accepts a word $a^m b^m c^{m+x} d^m \notin K_2$ or $a^m b^m c^m d^{m+x} \notin K_2$, for some $x > 0$. Now, among all accepting runs on w, it must be the case that either a positive cycle reading 'a's or a positive cycle reading 'b's was taken, otherwise we can remove the cycle and accept a word with too few 'a's or 'b's.

Assume that there exists a positive cycle π reading 'a's on some accepting run, and recall that there also exists a negative cycle ν reading 'd's. We can arbitrarily increase the counter value using π to then take an additional iteration of ν later in the run. Hence, $a^{m+x} \# b^m \# c^m \# d^{m+y}$ is accepted by $\mathcal{A}$ for some $y > 0$ and some x large enough, however $a^{m+x} \# b^m \# c^m \# d^{m+y} \notin K_2$. Symmetrically, the same argument holds if instead there does not exist a positive cycle reading 'a's but there does exist a positive cycle reading 'b's. Therefore, we are able to conclude that no such 1-CN $\mathcal{A}$ recognising K_2 exists. ◀

B Greyscale-Friendly Version of Section 4

Here, we repeat Sections 4.3 and 4.4 with augmented symbols for positive and nonnegative cycles.

B.1 Good and Bad Segments

We lift the colour scheme of > 0 and ≥ 0 to words and runs as follows. For a word $w = uv$ and a run ρ, we write e.g., $\widehat{u}\widetilde{v}$ to denote that ρ traverses a > 0 cycle when reading u, then a ≥ 0 cycle when reading v. Note that this does not preclude other cycles, e.g., there could also be negative cycles in the u part, etc. That is, the colouring is not unique, but represents elements of the run.

Recall our assumption that $\mathcal{L}(\mathcal{P}) = \bigcap_{1\leq j\leq k} \mathcal{L}(\mathcal{V}_i)$, and denote $\mathcal{V}_j = \langle \Sigma, Q^j, I^j, \delta^j, F^j \rangle$ for all $j \in [k]$. Let $Q_{\max} = \max\{|Q_j|\}_{j=1}^k$ and denote $\alpha = Q_{\max}!$. Further recall that we focus on words of the form $a^{m_1}\#a^{m_2}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$ for integers $\{m_i\}_{i=1}^{k+1}, m_b, m_c \in \mathbb{N}$, and that we refer to the infix a^{m_i} as Segment i, for $1 \leq i \leq k$. We proceed to formally define good and bad segments.

▶ **Definition 25** (Good and Bad Segments). *Segment i is* bad *in $\mathcal{V}_j$ if there exist constants $\{m_i\}_{i=1}^{k+1}, m_b, m_c \in \mathbb{N}$ such that all the following hold.*

- *$\{m_i\}_{i=1}^{k+1}, m_b, m_c$ are multiples of α.*
- *There is an accepting $\mathbb{N}$-run ρ of $\mathcal{V}_j$ on $w = a^{m_1}\#a^{m_2}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$ that adheres to one of the following three* forms*:*
  1. $\widetilde{a^{m_1}}\#\widetilde{a^{m_2}}\#\cdots\widetilde{a^{m_{i-1}}}\#\widehat{a^{m_i}}\#a^{m_{i+1}}\#\cdots\#a^{m_{k+1}}\#b^{m_b}c^{m_c}$
  2. $\widetilde{a^{m_1}}\#\widetilde{a^{m_2}}\#\cdots\widetilde{a^{m_{i-1}}}\#\widetilde{a^{m_i}}\#\widetilde{a^{m_{i+1}}}\#\cdots\#\widetilde{a^{m_{k+1}}}\#\widetilde{b^{m_b}}\widetilde{c^{m_c}}$
  3. $\widetilde{a^{m_1}}\#\widetilde{a^{m_2}}\#\cdots\widetilde{a^{m_{i-1}}}\#\widetilde{a^{m_i}}\#\widetilde{a^{m_{i+1}}}\#\cdots\#\widetilde{a^{m_{k+1}}}\#\widehat{b^{m_b}}c^{m_c}$.

Segment i is good *in $\mathcal{V}_j$ if it is not bad.*

Lemma 26 below formalises the intuition that a bad segment can be pumped simultaneously with both the b and c segments, eventually generating a word that is not in $\mathcal{L}(\mathcal{P})$, but is accepted by $\mathcal{V}_j$.

Intuitively, Forms 2 and 3 mean that all segments are bad. Indeed, Segment i has a ≥ 0 cycle, so it can be pumped safely, and in Form 2 both b and c can be pumped using ≥ 0 cycles, whereas in Form 3 we can pump b using a > 0 cycle, and use it to compensate for pumping c, even if the latter requires pumping a negative cycle.

Form 1 is the interesting case, where we use a > 0 cycle in Segment i to compensate for pumping both b and c. The requirement that all segments up to i are ≥ 0 is at the heart of our proof, and is explained in Appendix B.2. Formally, we have the following.

▶ **Lemma 26.** *Let l be a bad segment in $\mathcal{V}_j$, then there exist $x, y, z \in \mathbb{N}$ multiples of α such that for every $n \in \mathbb{N}$ the following word w is accepted by $\mathcal{V}_j$.*

$$w_n = a^{m_1}\#a^{m_2}\#\cdots\#a^{m_l+xn}\#a^{m_{l+1}}\#\cdots\#a^{m_{k+1}}\#b^{m_b+yn}c^{m_c+zn}$$

Proof. We can choose $z = \alpha$, then take y to be large enough so that Form 3 runs can compensate for negative cycles in c^z using > 0 cycles in b^y, while not decreasing the counters in Form 2 runs (we can find such y that is a multiple of α, since α is divisible by all lengths of simple cycles). Finally, we choose x so that Form 1 runs can compensate for c^z and b^y using > 0 cycles in a^x in Segment l, again while not decreasing the counters in Forms 2 and 3. ◀

Recall that our goal is to show that there is a segment $l \in [k+1]$ that is bad in *every* V_j, for $j \in [k]$. Lemma 27 below shows that each V_j has at most one good segment. Therefore, there are at most k good segments in total, leaving at least one bad segment as desired.

▶ **Lemma 27.** *Let $j \in [k]$ and $r < s \in [k+1]$. Then either Segment r or s is bad in $\mathcal{V}_j$.*

Proof. Since j is fixed, denote $\mathcal{V}_j = \langle \Sigma, Q, Q_0, \delta, F \rangle$. We inductively define constants $\{n_i\}_{i=1}^{k+1}, n_b, n_c \in \mathbb{N}$ as follows. n_1 is a large-enough multiple of α so that Lemma 10 guarantees a ≥ 0 cycle in any accepting run of $\mathcal{V}_j$ on a^{n_1} from some $(q_0, 0)$ with $q_0 \in Q_0$. Assume we have defined $n_1, \ldots n_{l-1}$, and consider the word $u = a^{n_1}\#a^{n_2}\#\cdots\#a^{n_{l-1}}\#$. Define $n = |u|W$ where W is the maximal value in any transition of $\mathcal{V}_j$. Since u consists of $\frac{n}{W}$ letters, $n+1$ is greater than any counter value that can be reached by $\mathcal{V}_j$ by reading u. We define n_l to be a multiple of α large enough so that Lemma 10 guarantees a ≥ 0 cycle when reading a^{n_l} from any configuration of the form $\{(q, n') \mid q \in Q,\ n \leq n+1\}$. We set $n_b = n_c = \alpha$ (the choice of n_b, n_c is slightly arbitrary). Finally, we set $w = a^{n_1}\#\cdots\#a^{n_{k+1}}\#b^{n_b}c^{n_c}$.

Now, for every $x \in \mathbb{N}$, we obtain from w a word w_x by pumping $x\alpha$ a's to segments r, s and to the b and c segments. That is, let $n'_i = n_i + x\alpha$ for $i \in \{r, s\}$ and $n'_i = n_i$ for $i \notin \{r, s\}$, and let $n'_b = n_b + x\alpha$ and $n'_c = n_c + x\alpha$, then $w_x = a^{n'_1}\#\cdots\#a^{n'_{k+1}}\#b^{n'_b}c^{n'_c}$. Observe that $w_x \in \mathcal{L}(\mathcal{P})$. Indeed, since $n_r \geq n_b = \alpha$ and $n_s \geq n_c = \alpha$ we have that $n_r + x\alpha \geq n_b + x\alpha$ and $n_s + x\alpha \geq n_c + x\alpha$, so segments r and s can already pay for the b's and c's, respectively. In particular, $w_x \in \mathcal{L}(\mathcal{V}_j)$ with some accepting $\mathbb{N}$-run ρ_x.

We choose a particular value of x, as follows. Consider x and suppose some accepting $\mathbb{N}$-run ρ_x as above does not traverse a > 0 cycle neither in segment r nor in s. By Lemma 9, the maximal possible counter value of ρ_x after reading

$$a^{n_1}\#\cdots\#a^{n_r+x\alpha}\#\cdots\#a^{n_s+x\alpha}\#\cdots\#a^{n_{k+1}}\#$$

is $M_b = (k+1+\sum_{z\in[k+1]\setminus\{r,s\}} n_z)W + 2|Q|W$. Crucially, this value does not depend on x. Further, if there is no > 0 cycle in the segment of b's as well, again the maximal counter value of ρ up to the c segment is bounded by $M_c = (k+2+\sum_{z\in[k+1]\setminus\{r,s\}} n_z)W + 3|Q|W$, independent of x and M_b.

By Lemma 10, we can now choose x large enough such that for every accepting $\mathbb{N}$-run ρ_x on w_x:

1. If ρ_x does not traverse any > 0 cycle in segments r, s, then ρ_x has a ≥ 0 cycle reading $b^{(n_b+x\alpha)}$ from any configuration of the form $\{(q, M') \mid q \in Q,\ M' \leq M_b\}$.
2. If ρ_x does not traverse any > 0 cycle in segment r nor s, nor in the b segment, ρ_x has a ≥ 0 cycle reading $c^{(n_c+x\alpha)}$ from any configuration of the form $\{(q, M') | q \in Q,\ M' \leq M_c\}$.

Having fixed x, we claim that one of r, s is bad for the constants in w_x.

By construction, Lemma 10 guarantees that ρ_x has a ≥ 0 cycles in segments $1, \dots r-1$. If ρ_x has a > 0 cycle in segment r, then ρ_x is of Form 1:

$$\widetilde{a^{n_1}}\#\widetilde{a^{n_2}}\#\cdots\widetilde{a^{n_{i-1}}}\#\widehat{a^{n_r+x\alpha}}\#\cdots\#a^{n_s+x\alpha}\#\cdots\#a^{n_{k+1}}\#b^{n_b+x\alpha}c^{n_c+x\alpha}$$

so r is bad in $\mathcal{V}_j$.

If ρ_x does not have a > 0 cycles in segment r, then again by construction Lemma 10 guarantees ≥ 0 cycles in segments $r, r+1, \dots, s-1$. Indeed, for r – we are guaranteed a ≥ 0 cycle reading a^{n_r}, all the more so for $a^{n_r+x\alpha}$. As for $r+1, \dots s-1$ – if ρ_x does not have a > 0 cycle in segment r, then the maximal effect of segment r is $W|Q|$. However, n_{r+1} was constructed to guarantee a ≥ 0 cycle even in case the effect of segment r is $Wn_r \geq W\alpha \geq W|Q|$.

If there is a > 0 cycle in segment s - then ρ_x adheres to Form 1:

$$\widetilde{a^{n_1}}\#\widetilde{a^{n_2}}\#\cdots\widetilde{a^{n_{s-1}}}\#\widehat{a^{n_s+x\alpha}}\#a^{n_{s+1}}\#\cdots\#a^{n_{k+1}}\#b^{n_b+x\alpha}c^{m_c+x\alpha}$$

and s is bad in $\mathcal{V}_j$.

Otherwise, using the same arguments as for segment r, we have that segments $s+1, \dots, i_{k+1}$ contain ≥ 0 cycles. In this case we are left with the b and c segments. The choice of x guarantees a ≥ 0 cycle in the segment of b's. If ρ_x traverses a > 0 cycle in the b segment, then w_x is of Form 3:

$$\widetilde{a^{n_1}}\#\widetilde{a^{n_2}}\#\cdots\widetilde{a^{n_{k+1}}}\#\widehat{b^{n_b+x\alpha}}c^{n_c+x\alpha}$$

Finally, if there are no > 0 cycles in the b segment, then the choice of x again guarantees a ≥ 0 cycle in the c segment, so w_x is of Form 2:

$$\widetilde{a^{n_1}}\#\widetilde{a^{n_2}}\#\cdots\widetilde{a^{n_{k+1}}}\#\widetilde{b^{n_b+x\alpha}}\widetilde{c^{n_c+x\alpha}}$$

In the two latter cases both r and s are bad in $\mathcal{V}_j$. ◀

B.2 Proof of Theorem 8

We now have that each $\mathcal{V}_j$ has at most one good segment (otherwise we would have two good segments r and s, contradicting Lemma 27). Therefore, all 1-CNs $\mathcal{V}_1, \dots, \mathcal{V}_k$ have at most k good segments combined. Recall that our words have $k+1$ segments, and therefore there is at least one segment l that is bad in every $\mathcal{V}_j$. Note, however, that this segment may correspond to different constants in each $\mathcal{V}_j$. That is, there exists constants $\{m_i^j, m_b^j, m_c^j \mid i \in [k+1], j \in [k]\}$ witnessing that segment l is bad for each $\mathcal{V}_j$. We group the $\mathcal{V}_j$ according to the Form of their accepting runs ρ_j, as follows.

- Form 1 are $\widetilde{a^{m_1^j}}\#\widetilde{a^{m_2^j}}\#\cdots\#\widehat{a^{m_l^j}}\#a^{m_{l+1}^j}\#\cdots\#a^{m_{k+1}^j}\#b^{m_b^j}c^{m_c^j}$.
- Form 2 are $\widetilde{a^{m_1^j}}\#\widetilde{a^{m_2^j}}\#\cdots\#\widetilde{a^{m_l^j}}\#\widetilde{a^{m_{l+1}^j}}\#\cdots\#\widetilde{a^{m_{k+1}^j}}\#\widetilde{b^{m_b^j}}\widetilde{c^{m_c^j}}$.
- Form 3 are $\widetilde{a^{m_1^j}}\#\widetilde{a^{m_2^j}}\#\cdots\#\widetilde{a^{m_l^j}}\#\widetilde{a^{m_{l+1}^j}}\#\cdots\#\widetilde{a^{m_{k+1}^j}}\#\widehat{b^{m_b^j}}c^{m_c^j}$.

We now find constants such that the resulting (single) word has l as a bad segment in all $\mathcal{V}_j$. First, for $i \in [k+1] \setminus \{l\}$, define $M_i = \max\{m_i^j\}_{1 \leq j \leq k}$ (note that these are still multiples of α). Similarly, define $M_c = \max\{m_c^j\}_{1 \leq j \leq k}$. It remains to fix constants L and B, which we do in phases in the following. The resulting word is then

$$w = a^{M_1}\#\cdots\#a^L\#a^{M_{l+1}}\#\cdots\#a^{M_{k+1}}\#b^B c^{M_c}$$

Most steps in the flow of the analysis below are based on Lemma 11 and Corollary 12. We first (partially) handle Form 3. For such $\mathcal{V}_j$, there is an accepting $\mathbb{N}$-run ρ_j on

$$\widetilde{a^{m^j_1}}\#\cdots\#\widetilde{a^{m^j_l}}\#\widetilde{a^{m^j_{l+1}}}\#\cdots\#\widetilde{a^{m^j_{k+1}}}\#\widehat{b^{m^j_b}}c^{m^j_c}$$

By pumping ≥ 0 cycles as per Corollary 12 in all segments by l we obtain an accepting $\mathbb{N}$-run ρ'_j on

$$\widetilde{a^{M_1}}\#\cdots\#\widetilde{a^{m^j_l}}\#\widetilde{a^{M_{l+1}}}\#\cdots\#\widetilde{a^{M_{k+1}}}\#\widehat{b^{m^j_b}}c^{m^j_c}$$

We now pump arbitrary cycles in the c segment to construct a $\mathbb{Z}$-run ρ''_j on:

$$\widetilde{a^{M_1}}\#\cdots\#\widetilde{a^{m^j_l}}\#\widetilde{a^{M_{l+1}}}\#\cdots\#\widetilde{a^{M_{k+1}}}\#\widehat{b^{m^j_b}}c^{M_c}$$

Next, we compensate for possible negative cycles in the c segment by pumping a > 0 cycle in the b segment. Thus, we construct an $\mathbb{N}$-run ρ'''_j on

$$\widetilde{a^{M_1}}\#\cdots\#\widetilde{a^{m^j_l}}\#\widetilde{a^{M_{l+1}}}\#\cdots\#\widetilde{a^{M_{k+1}}}\#\widehat{b^{B}}c^{M_c}$$

where B is chosen to be large enough such that ρ'''_j is indeed an $\mathbb{N}$-run for all $1 \leq j \leq k$. Note that it remains to fix L.

We now turn to Form 1 with a similar process. We start with an accepting $\mathbb{N}$-run ρ_j on

$$\widetilde{a^{m^j_1}}\#\cdots\#\widehat{a^{m^j_l}}\#a^{m^j_{l+1}}\#\cdots\#a^{m^j_{k+1}}\#b^{m^j_b}c^{m^j_c}$$

Pump ≥ 0 cycles to obtain an accepting $\mathbb{N}$-run ρ'_j on

$$\widetilde{a^{M_1}}\#\cdots\#\widehat{a^{m^j_l}}\#a^{m^j_{l+1}}\#\cdots\#a^{m^j_{k+1}}\#b^{m^j_b}c^{m^j_c}$$

obtain a $\mathbb{Z}$-run ρ''_j by pumping arbitrary cycles in the remaining segments, including the b segment:

$$\widetilde{a^{M_1}}\#\cdots\#\widehat{a^{m^j_l}}\#a^{M_{l+1}}\#\cdots\#a^{M_{k+1}}\#b^{B}c^{M_c}$$

and compensate for negative cycles by taking L large enough so that pumping > 0 cycles in segment l yields an accepting $\mathbb{N}$-run ρ'''_j on

$$\widetilde{a^{M_1}}\#\cdots\#\widehat{a^{L}}\#a^{M_{l+1}}\#\cdots\#a^{M_{k+1}}\#b^{B}c^{M_c}$$

We now return to Form 3 and fix the l-th segment by pumping ≥ 0 cycles to construct an accepting $\mathbb{N}$-run on

$$\widetilde{a^{M_1}}\#\cdots\#\widetilde{a^{L}}\#\widetilde{a^{M_{l+1}}}\#\cdots\#\widetilde{a^{M_{k+1}}}\#\widehat{b^{B}}c^{M_c}$$

We are left with Form 2, which are the easiest to handle. We simply pump ≥ 0 cycles in all segments to construct an accepting $\mathbb{N}$-run ρ'_j on

$$\widetilde{a^{M_1}}\#\cdots\#\widetilde{a^{L}}\#\widetilde{a^{M_{l+1}}}\#\cdots\#\widetilde{a^{M_{k+1}}}\#\widetilde{b^{B}}\widetilde{c^{M_c}}$$

Note that the requirement for all segments leading up to l to be ≥ 0 is crucial, otherwise we would not have been able to pump all the Forms simultaneously.

We now have that w is accepted by every $\mathcal{V}_j$, and segment l is bad for all $\mathcal{V}_j$ for it.

By applying Lemma 26 for each of the $\mathcal{V}_j$ and taking global constants to be the products of the respective constants x, y, z for each $\mathcal{V}_j$, We can now find $X, Y, Z \in \mathbb{N}$ multiples of α such that for every $n \in \mathbb{N}$ the word

$$w_n = a^{M_1}\# \cdots \# a^{L+Xn}\# a^{M_{l+1}}\# \cdots \# a^{M_{k+1}}\# b^{B+Yn}c^{M_c+Zn}$$

is accepted by every $\mathcal{V}_j$. We then choose n large enough to satisfy $\sum_{i\in[k+1]\setminus\{l\}} M_i < \min\{B+Yn, M_c+Zn\}$, so that $w_n \notin \mathcal{L}(\mathcal{P})$, since segment l can only pay for b or for c, and the remaining segments cannot pay for the remaining segment.

This contradicts the assumption that $\mathcal{L}(\mathcal{P}) = \bigcap_{j\in[k]} \mathcal{L}(\mathcal{V}_j)$, concluding the proof of Theorem 8. ◀

▶ Remark 28 (Unbounded Compositeness). The proof of Theorem 8 shows that if words with $k+1$ segments are allowed, then the language is not $(1,k)$-composite (and we use it to establish primality). By intersecting $\mathcal{L}(\mathcal{P})$ with words that allow at most $k+1$ segments, we obtain a language that is not $(1,k)$-composite, but it is not hard to show that it is $(1,2^{k+1})$-composite. This shows that a 2-CN can be composite, but require unboundedly many factors.