Determinization of One-Counter Nets

Shaull Almagor
Department of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel

Asaf Yeshurun
Department of Computer Science, Technion – Israel Institute of Technology, Haifa, Israel

Abstract

One-Counter Nets (OCNs) are finite-state automata equipped with a counter that is not allowed to become negative, but does not have zero tests. Their simplicity and close connection to various other models (e.g., VASS, Counter Machines and Pushdown Automata) make them an attractive model for studying the border of decidability for the classical decision problems.

The deterministic fragment of OCNs (DOCNs) typically admits more tractable decision problems, and while these problems and the expressive power of DOCNs have been studied, the determinization problem, namely deciding whether an OCN admits an equivalent DOCN, has not received attention.

We introduce four notions of OCN determinizability, which arise naturally due to intricacies in the model, and specifically, the interpretation of the initial counter value. We show that in general, determinizability is undecidable under most notions, but over a singleton alphabet (i.e., 1 dimensional VASS) one definition becomes decidable, and the rest become trivial, in that there is always an equivalent DOCN.

2012 ACM Subject Classification Theory of computation → Formal languages and automata theory

Keywords and phrases Determinization, One-Counter Net, Vector Addition System, Automata, Semilinear

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.18

Related Version *Previous Version*: https://arxiv.org/abs/2112.13716

1 Introduction

One-Counter Nets (OCNs) are finite-state machines equipped with an integer counter that cannot decrease below zero and cannot be explicitly tested for zero.

OCNs are closely related to several computational models: they are a test-free syntactic restriction of One-Counter Automata – Minsky Machines with only one counter. If counter updates are restricted to ± 1, they are equivalent to Pushdown Automata with a single-letter stack alphabet. In addition, over a singleton alphabet, they are the same as 1-dimensional Vector Addition Systems with States.

An OCN $\mathcal{A}$ over alphabet Σ accepts a word $w \in \Sigma^*$ from initial counter value $c \in \mathbb{N}$ if there is a run of $\mathcal{A}$ on w from an initial state to an accepting state in which the counter, starting from value c, does not become negative. Thus, for every counter value $c \in \mathbb{N}$ the OCN $\mathcal{A}$ defines a language $\mathcal{L}(\mathcal{A}, c) \subseteq \Sigma^*$.

OCNs are an attractive model for studying the border of decidability of classical decision problems. Indeed – several problems for them lie delicately close to the decidability border. For example, OCN universality is decidable [16], whereas parameterized-universality (in which the initial counter is existentially quantified) is undecidable [2].

As is the case with many computational models, certain decision problems for deterministic OCNs (OCNs that admit a single legal transition for each state q and letter σ), denoted DOCNs, are computationally easier than for nondeterministic OCNs (e.g., inclusion is undecidable for OCNs, but is in NL for DOCNs [16]. Universality is Ackermannian for OCNs, but is in NC for DOCNs [2]). While decision problems for DOCNs have received some attention, the *determinization* problem for OCNs, namely deciding whether an OCN admits

33rd International Conference on Concurrency Theory (CONCUR 2022).
Editors: Bartek Klin, Sławomir Lasota, and Anca Muscholl; Article No. 18; pp. 18:1–18:23
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

an equivalent DOCN, has (to our knowledge) not been studied. Apart from the theoretical interest of OCN determinization, which would yield a better understanding of the model, it is also of practical interest: OCNs can be used to model properties of concurrent systems, so when an OCN can be determinized, automatic reasoning about correctness becomes easier.

OCN Determinization

Recall that the language $\mathcal{L}(\mathcal{A}, c)$ of an OCN $\mathcal{A}$ depends on the initial counter c, so $\mathcal{A}$ essentially defines a family of languages. Thus, it is not clear what we mean by "equivalent DOCN". We argue that the definition of determinization depends on the role of the initial counter c. To this end, we identify four notions of determinization for an OCN $\mathcal{A}$, as follows.

- In `0-Det`, we ask whether there is a DOCN $\mathcal{D}$ such that $\mathcal{L}(\mathcal{A}, 0) = \mathcal{L}(\mathcal{D}, 0)$.
- In `∃-Det`, we ask whether there exist $c \in \mathbb{N}$ and a DOCN $\mathcal{D}$ such that $\mathcal{L}(\mathcal{A}, c) = \mathcal{L}(\mathcal{D}, 0)$.
- In `∀-Det`, we ask whether for every $c \in \mathbb{N}$, there is a DOCN $\mathcal{D}$ such that $\mathcal{L}(\mathcal{A}, c) = \mathcal{L}(\mathcal{D}, 0)$.
- In `Uniform-Det`, we ask whether there is a DOCN $\mathcal{D}$ such that for every $c \in \mathbb{N}$ we have $\mathcal{L}(\mathcal{A}, c) = \mathcal{L}(\mathcal{D}, c)$.

The motivation for each of the problems depends, intuitively, on the interpretation of the initial counter, and on the stage at which the equivalent DOCN is computed, as we now demonstrate.

- Consider an OCN modelling an access-control handler, where the counter corresponds to the number of access requests in a queue. Since the controller is deployed with an empty queue, an equivalent DOCN would need to be equivalent only on an initial 0 counter, so we would want to solve `0-Det`.
- Consider an OCN modelling resource handler, where the counter corresponds to the available resources. When searching for a deterministic controller, we may initialize it with some fixed amount of resources to start with, hence `∃-Det` is suitable.
- Now consider the task of devising an OCN for the resource handler above, so that it can be deployed in many different concrete settings as a DOCN, but where each setting has its own amount of available initial resources. In order to design a single OCN that can be determinized to appropriate DOCNs, we would want to solve `∀-Det`.
- Finally, `Uniform-Det` is of interest in any setting that is exactly modelled as an OCN, but needs to be determinized, e.g., when the resource handler above needs to be deployed but the initial resources depend on the system's load.

Paper Organization and Contribution

In this paper, we study the decidability of the determinization problems derived from the four notions. In Section 3 we examine the relation between the notions, and demonstrate that no pair of them coincide. In Section 4 we show that `0-Det`,`∃-Det`, and `∀-Det` are generally undecidable. For `Uniform-Det`, we are not able to resolve decidability, but we do show an Ackermannian lower bound.

In order to recover some decidability, we turn to the fragment of OCNs over a singleton alphabet (1-dimensional VASS). There, we show that `0-Det`,`∃-Det`, and `∀-Det` become trivial (i.e., they always hold), whereas `Uniform-Det` becomes decidable. We conclude with a discussion and future work in Section 6.

Technically, our undecidability results for general alphabets use reductions from two different models – one from the model of Lossy Counter Machines [23, 27], and one from a careful analysis of recent results about OCNs [2]. For the singleton-alphabet case, the decidability of `Uniform-Det` requires some machinery from the theory of low-dimensional

VASS and Presburger Arithmetic, as well as some basic linear algebra and number theory. Our main contribution in this part is the introduction and analysis of the *Minimal Counter Relation (MCR)* – a sequence characterizing the minimal counter needed to accept each length of words. We characterize `Uniform-Det` using this sequence, and we suspect this sequence may prove useful in other contexts.

Related Work

The determinization problem we consider in this work assumes that the deterministic target model is also that of OCNs. An alternative approach to simplifying a nondeterministic OCN is to find an equivalent deterministic finite automaton, if one exists. This amounts to deciding whether the language of an OCN is regular. This problem was shown to be undecidable for OCNs in [32]. Interestingly, the related problem of regular separability was shown to be in `PSPACE` in [10]. A related result in [11] describes a determinization procedure for "unambiguous blind counter automata" over infinite words, to a Muller counter machine.

From a different viewpoint, determinization is a central problem in quantitative models, which can be thought of as counter automata where the counter value is the output, rather than a Boolean language acceptor. The decidability of determinization for Tropical Weighted Automata is famously open [9, 20] with only partial decidable fragments [20, 21]. A slightly less related model is that of discounted-sum automata, whose determinization has intricate connections to number theory [7].

Determinization of computational models is closely related to various notions of semantic equivalence. The three main concepts scrutinized in this regard are, from most restrictive to least restrictive: bisimulation, simulation and trace inclusion. Each of the three notions has strong and weak variants. Strong bisimulation was shown to be `PSPACE`-complete both for OCNs and OCAs [5, 6], while weak bisimulation was shown to be undecidable [23]. Conversely, trace inclusion, both weak and strong, is undecidable both for OCNs and OCAs [15, 31]. Finally, simulation, both weak and strong, is undecidable for OCAs [17], but decidable for OCNs [1, 18, 19, 28, 30].

2 Preliminaries

A *one-counter net* (OCN) is a finite automaton whose transitions are labelled both by letters and by integer weights. Formally, an OCN is a tuple $\mathcal{A} = \langle \Sigma, Q, s_0, \delta, F\rangle$ where Σ is a finite alphabet, Q is a finite set of states, $s_0 \in Q$ is the initial state, $\delta \subseteq Q \times \Sigma \times \mathbb{Z} \times Q$ is the set of transitions, and $F \subseteq Q$ are the accepting states. We say that an OCN is *deterministic* if for every $s \in Q, \sigma \in \Sigma$, there is at most one transition (s, σ, e, s') for some $e \in \mathbb{Z}$ and $s' \in Q$.

For a transition $t = (s, \sigma, e, s') \in \delta$ we define $\mathsf{eff}(t) = e$ to be its (counter) *effect*.

A *path* in the OCN is a sequence $\pi = (s_1, \sigma_1, e_1, s_2)(s_2, \sigma_2, e_2, s_3)\dots(s_k, \sigma_k, e_k, s_{k+1}) \in \delta^*$. Such a path π is a *cycle* if $s_1 = s_{k+1}$, and is a *simple cycle* if no other cycle is a proper infix of it. We say that the path π *reads* the word $\sigma_1\sigma_2\dots\sigma_k \in \Sigma^*$. The *effect* of π is $\mathsf{eff}(\pi) = \sum_{i=1}^{k} e_i$, and its *nadir*, denoted $\mathsf{nadir}(\pi)$, is the minimal effect of any prefix of π (note that the nadir is non-positive, since $\mathsf{eff}(\epsilon) = 0$).

A *configuration* of an OCN is a pair $(s, v) \in Q \times \mathbb{N}$ comprising a state and a non-negative integer. For a letter $\sigma \in \Sigma$ and configurations $(s, v), (s', v')$ we write $(s, v) \xrightarrow{\sigma} (s', v')$ if there exists $d \in \mathbb{Z}$ such that $v' = v + d$ and $(s, \sigma, d, s') \in \delta$.

A *run* of $\mathcal{A}$ from initial counter c on a word $w = \sigma_1 \cdots \sigma_k \in \Sigma^*$ is a sequence of configurations $\rho = (q_0, v_0), (s_1, v_1), \dots, (s_k, v_k)$ such that $v_0 = c$ and for every $1 \le i \le k$ it holds that $(s_{i-1}, v_{i-1}) \xrightarrow{\sigma_i} (s_i, v_i)$. Since configurations may only have a non-negative counter, this enforces that the counter does not become negative.

Note that every run naturally induces a path in the OCN. For the converse, a path π induces a run from initial counter c iff $c \geq -\mathsf{nadir}(\pi)$ (indeed, the minimal initial counter needed for traversing a path π is exactly $-\mathsf{nadir}(\pi)$). We extend the definitions of effect and nadir to runs, by associating them with the corresponding path.

The run ρ is *accepting* if $s_k \in F$, and we say that $\mathcal{A}$ *accepts* w with initial counter c if there exists an accepting run of $\mathcal{A}$ on w from initial counter c. We define $\mathcal{L}(\mathcal{A}, c) = \{w \in \Sigma^* : \mathcal{A}$ accepts w with initial counter $c\}$, and we define the complement of a language $\mathcal{L}(\mathcal{A}, c)$ to be $\overline{\mathcal{L}(\mathcal{A}, c)} = \Sigma^* \setminus \mathcal{L}(\mathcal{A}, c)$. Observe that OCNs are monotonic – if $\mathcal{A}$ accepts w from counter c, it also accepts it from every $c' \geq c$. Thus, $\mathcal{L}(\mathcal{A}, c) \subseteq \mathcal{L}(\mathcal{A}, c')$ for $c' \geq c$.

3 OCN Determinization

In this section we examine the relationship between the four determinization notions. For brevity, we use the same term for the decision problems and the properties they represent, e.g., we say "$\mathcal{A}$ is `0-Det`" if $\mathcal{A}$ has an equivalent DOCN under `0-Det`.

We first examine how the definitions compare in their strictness:

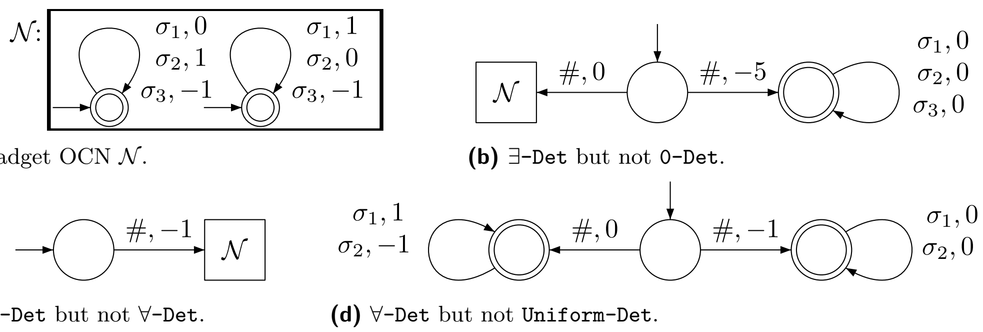
▶ **Observation 1.** *Consider an OCN $\mathcal{A}$. If $\mathcal{A}$ is `Uniform-Det`, then $\mathcal{A}$ is $\forall$-`Det`, if $\mathcal{A}$ is $\forall$-`Det`, then $\mathcal{A}$ is `0-Det`, and if $\mathcal{A}$ is `0-Det`, then $\mathcal{A}$ is $\exists$-`Det`.*

Next, we prove that none of the definitions coincide. Following Observation 1, it suffices to prove the following.

▶ **Lemma 2.** *There exist OCNs $\mathcal{A}, \mathcal{B}, \mathcal{C}$ such that $\mathcal{A}$ is $\exists$-`Det` but not `0-Det`, $\mathcal{B}$ is `0-Det` but not $\forall$-`Det`, and $\mathcal{C}$ is $\forall$-`Det` but not `Uniform-Det`.*

Proof (sketch). The OCNs $\mathcal{A}, \mathcal{B}, \mathcal{C}$ are depicted in Figure 1. We demonstrate the intuition on $\mathcal{C}$, see Appendix A.1 for the complete proof. To show that $\mathcal{C}$ is $\forall$-`Det`, we observe that for initial counter 0, we can omit the $(\#, -1)$ transition, thus obtaining an equivalent DOCN. For initial counter $c \geq 1$ we have that $\mathcal{L}(\mathcal{C}, c) = \# \cdot \{\sigma_1, \sigma_2\}^*$ is regular and thus has a DOCN.

We claim $\mathcal{C}$ is not `Uniform-Det`. An equivalent DOCN $\mathcal{D}$ with k states, starting from initial counter 0, must accept the word $\#\sigma_1^{k+1}\sigma_2^{k+1}$. It is easy to show that upon reading σ_2^{k+1} it must make a negative cycle. This, however, causes some word of the form $\#\sigma_1^{k+1}\sigma_2^m$ not to be accepted even with counter 1, which means $\mathcal{L}(\mathcal{D}, 1) \neq \mathcal{L}(\mathcal{C}, 1)$. ◀



(a) Gadget OCN $\mathcal{N}$. **(b)** $\exists$-`Det` but not `0-Det`. **(c)** `0-Det` but not $\forall$-`Det`. **(d)** $\forall$-`Det` but not `Uniform-Det`.

Figure 1 Examples separating the determinization notions.

4 Lower Bounds for Determinization

In this section we prove lower bounds for the four determinization decision problems. We show that `0-Det`, `∀-Det`, and `∃-Det` are undecidable, while for `Uniform-Det` we show an Ackermannian lower bound, and its decidability remains an open problem.

We start by introducing *Lossy Counter Machines* (LCMs) [23, 27], from which we will obtain some undecidability results. Intuitively, an LCM is a Minsky counter machine, whose semantics are such that counters may arbitrarily decrease at each step. Formally, an LCM is $\mathcal{M} = \langle \mathsf{Loc}, \mathsf{Z}, \Delta \rangle$ where $\mathsf{Loc} = \{\ell_1, \ldots, \ell_m\}$ is a finite set of *locations*, $\mathsf{Z} = (z_1, \ldots, z_n)$ are n counters, and $\Delta \subseteq \mathsf{Loc} \times \mathsf{OP}(\mathsf{Z}) \times \mathsf{Loc}$, where $\mathsf{OP}(\mathsf{Z}) = \mathsf{Z} \times \{++, --, =0?\}$.

A *configuration* of $\mathcal{M}$ is $\langle \ell, \boldsymbol{a} \rangle$ where $\ell \in \mathsf{Loc}$ and $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{N}^{\mathsf{Z}}$. There is a *transition* $\langle \ell, \boldsymbol{a} \rangle \to \langle \ell, \boldsymbol{b} \rangle$ if there exists $\mathsf{op} \in \mathsf{OP}$ and either:

- $\mathsf{op} = c_k{++}$ and $b_k \leq a_k + 1$ and $b_j \leq a_j$ for all $j \neq k$, or
- $\mathsf{op} = c_k{--}$ and $b_k \leq a_k - 1$ and $b_j \leq a_j$ for all $j \neq k$, or
- $\mathsf{op} = c_k{=}0?$ and $b_k = a_k = 0$ and $b_j \leq a_j$ for all $j \neq k$.

Since we only require $\leq$ on the counter updates, the counters nondeterministically decrease at each step.

A *run* of $\mathcal{M}$ is a finite sequence $\langle \ell_1, \boldsymbol{a_1} \rangle \to \langle \ell_2, \boldsymbol{a_2} \rangle \to \ldots \to \langle \ell_r, \boldsymbol{a_r} \rangle$. Given a configuration $\langle \ell, \boldsymbol{a} \rangle$, the *reachability set* of $\langle \ell, \boldsymbol{a} \rangle$ is the set of all configurations reachable from $\langle \ell, \boldsymbol{a} \rangle$ via runs of $\mathcal{M}$. In [27], it is shown that the problem of deciding whether the reachability set of a configuration is finite, is undecidable. A slight modification of this problem (see Appendix A.2) yields the following.

▶ **Lemma 3.** *The following problem, dubbed 0-Finite-Reach, is undecidable: Given an LCM $\mathcal{M}$ and a location ℓ_0, decide whether the reachability set of $\langle \ell_0, (0, \ldots, 0) \rangle$ is finite.*

4.1 Undecidability of `0-Det`

We show that `0-Det` is undecidable using a reduction from 0-Finite-Reach. Intuitively, given an LCM $\mathcal{M}$ and a location ℓ_0, we construct an OCN $\mathcal{A}$ that accepts, from initial counter 0, all the words that do not represent runs of $\mathcal{M}$ from $\langle \ell_0, (0, \ldots, 0) \rangle$.

In order for the OCN $\mathcal{A}$ to verify the illegality of a run, it guesses a violation in it. Control violations, i.e., illegal transitions between locations, are easily checked. In order to capture counter violations, $\mathcal{A}$ must find a counter whose value in the current configuration is *smaller* than in the next iteration (up to ± 1 for $++$ and $--$ commands). This, however, cannot be done by an OCN, since intuitively an OCN can only check that the later number is smaller, by first incrementing the counter and then decrementing it. To overcome this, we encode runs in reverse, as follows.

Consider an LCM $\mathcal{M} = \langle \mathsf{Loc}, \mathsf{Z}, \Delta \rangle$ with $\mathsf{Loc} = \{\ell_1, \ldots, \ell_m\}$ and $\mathsf{Z} = (z_1, \ldots, z_n)$. We encode a configuration $\langle \ell, (a_1, \ldots, a_n) \rangle$ over the alphabet $\Sigma = \mathsf{Loc} \cup \mathsf{Z}$ as $\ell \cdot z_1^{a_1} \cdots z_n^{a_n} \in \Sigma^*$. We then encode a run by concatenating the encoding of its configurations.

For a word $w = \sigma_1 \cdots \sigma_k \in \Sigma^*$, let $w^R = \sigma_k \cdots \sigma_1$ be its *reverse*, and for a language $L \subseteq \Sigma^*$, define its reverse to be $L^R = \{w^R : w \in L\}$.

We now define $L_{\mathcal{M},\ell_0} = \{w \in \Sigma^* : w \text{ encodes a run of } \mathcal{M} \text{ from } \langle \ell_0, (0, \ldots, 0) \rangle\}$.

We are now ready to describe the construction of $\mathcal{A}$.

▶ **Lemma 4.** *Given an LCM $\mathcal{M}$ and a location ℓ_0, we can construct an OCN $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}, 0) = \overline{L^R_{\mathcal{M},\ell_0}}$.*

Proof sketch: We construct $\mathcal{A}$ such that it accepts a word w iff w^R does not describe a run of $\mathcal{M}$ from $\langle \ell_0, (0, \ldots, 0) \rangle$.

As mentioned above, $\mathcal{A}$ reads w and guesses when a violation would occur, where control violations are relatively simple to spot, by directly encoding the structure of $\mathcal{M}$ in $\mathcal{A}$.

In order to spot counter violations, namely two consecutive configurations $\langle \ell, (a_1, \ldots, a_n) \rangle$ and $\langle \ell', (a'_1, \ldots, a'_n) \rangle$ such that some a'_i is too large compared to its counterpart a_i (how much larger is "too large" depends on $\mathcal{M}$'s transitions), $\mathcal{A}$ reads a configuration $\ell \cdot z_1^{a_1} \cdots z_n^{a_n}$ and increments its counter to count up to a_i, if it guesses that z_i is the counter that violates the transition. Assume for simplicity that the command in the transition does not involve counter z_i, then upon reading the next configuration $\ell' \cdot z_1^{b_1} \cdots z_n^{b_n}$, $\mathcal{A}$ decrements its counter while reading z_i, so that the counter value is $a_i - b_i$. Then, $\mathcal{A}$ takes another transition with counter value -1. Since the configuration is reversed, if this is indeed a violation, then $a_i > b_i$ (since the counters are lossy), so $a_i - b_i - 1 \geq 0$, and $\mathcal{A}$ accepts. Otherwise, $a_i \leq b_i$, so this run of $\mathcal{A}$ cannot proceed.

In Appendix A.3 we give the complete details of the construction. ◀

The correctness of the construction is proved in the following lemma.

▶ **Lemma 5.** *Consider an LCM $\mathcal{M}$ and a location ℓ_0, and let $\mathcal{A}$ be the OCN constructed in Lemma 4. Then $(\mathcal{M}, \ell_0)$ is in 0-FINITE-REACH iff $\mathcal{A}$ is `0-Det`.*

Proof sketch: Assume the reachability set of $\langle \ell_0, (0 \ldots 0) \rangle$ is finite under $\mathcal{M}$. Then there exists an upper bound $M \in \mathbb{N}$ of all counter values in all legal runs of $\mathcal{M}$ from $\langle \ell_0, (0 \ldots 0) \rangle$. $\mathcal{A}$'s behaviour can then be fully captured by a DFA $\mathcal{D}$ with the set of all states of the form $\langle \ell, a_1 \ldots a_k, b_1 \ldots b_k \rangle$ such that ℓ is a state in $\mathcal{M}$, k the number of counters, the values of $a_1 \ldots a_k$ represent counter values of the "current" configuration already fully known, and the values of $b_1 \ldots b_k$ represent counter values of the "previous" configuration, that $\mathcal{D}$ is in the process of accumulating. In addition, all values of $a_1 \ldots a_k, b_1 \ldots b_k$ are bounded by M. by assigning the only accepting state of $\mathcal{D}$ as $q_f = \langle \ell_0, 0 \ldots 0, 0 \ldots 0 \rangle$, and addressing several minor technical nuances, we can conclude $\mathcal{L}(\mathcal{D}) = \overline{\mathcal{L}(\mathcal{A}, 0)}$, therefore both $\overline{\mathcal{L}(\mathcal{A}, 0)}$ and $\mathcal{L}(\mathcal{A}, 0)$ are regular. Specifically, $\mathcal{A}$ is `0-Det`.

As for the other direction, assume the reachability set of $\langle \ell_0, (0 \ldots 0) \rangle$ under $\mathcal{M}$ is infinite, and assume by way of contradiction that $\mathcal{A}$ has a deterministic equivalent $\mathcal{D}'$. Note that for every word $u \in \Sigma^*$, the run of $\mathcal{D}'$ does not end due to the counter becoming negative, since we can always concatenate some $\lambda \in \Sigma^*$ such that $u\lambda$ does not correspond to a run, and is hence accepted by $\mathcal{D}'$, so the run on u must be able to continue reading λ.

Since the reachability set of $\langle \ell_0, (0 \ldots 0) \rangle$ is infinite, there exists a counter of $\mathcal{M}$, w.l.o.g z_1, that can take unbounded values (in different runs). Let w be a word corresponding to a run of $\mathcal{M}$ that ends with the value of z_1 being N for some large N. We can then write $w = a_k^* \cdots a_1^N \ell a_k^* \cdots a_1^{N'} \ell' \rho$, such that ρ represents the reverse of a legal prefix of a run of $\mathcal{M}$. $\mathcal{D}'$ necessarily goes through a cycle β when reading a_1^N. We pump the cycle k times until the word obtained, w', represents an illegal run due to the difference between $N + k \cdot |\beta|$ and N'. w' should then be accepted, but is in fact rejected, either due to the run ending successfully in the same non accepting state as w, or halting ahead of time due to a counter violation. Either way, that is a contradiction.

In Appendix A.4 we give the formal construction of $\mathcal{D}$, and a detailed correctness proof. ◀

Combining Lemmas 4 and 5, we conclude the following.

▶ **Theorem 6.** *`0-Det` is undecidable for OCNs over a general alphabet.*

4.2 Undecidability of $\forall$-Det and of $\exists$-Det

The undecidability of $\forall$-Det follows from that of 0-Det.

▶ **Theorem 7.** *$\forall$-Det is undecidable.*

Proof. We show a reduction from 0-Det. Given an OCN $\mathcal{A} = \langle \Sigma, Q, s_0, \delta, F\rangle$, we construct an OCN $\mathcal{B} = \langle \Sigma', Q', q_0, \delta', F'\rangle$ as illustrated in figure 2. Formally, the states of $\mathcal{B}$ are $Q' = Q \cup \{q_0, q_{\rm All}\}$, the initial state is q_0, its alphabet is $\Sigma' = \Sigma \cup \{\#\}$ such that $\# \notin \Sigma$, its accepting states are $F' = F \cup \{q_{\rm All}\}$, and its transition relation is $\Delta' = \Delta \cup \{(q_0, \#, -1, q_{\rm All}), (q_0, \#, 0, s_0)\} \cup \{(q_{All}, \sigma, 0, q_{\rm All}) : \sigma \in \Sigma'\}$.

We claim that $\mathcal{A}$ is 0-Det iff $\mathcal{B}$ is $\forall$-Det. For the first direction, assume $\mathcal{A}$ is 0-Det. Thus, $\mathcal{L}(\mathcal{B}, 0) = \# \cdot \mathcal{L}(\mathcal{A}, 0)$ has an equivalent DOCN. Since $\mathcal{L}(\mathcal{B}, k) = \#\Sigma'^*$ (which has a DOCN) for all $k \geq 1$, $\mathcal{B}$ is $\forall$-Det.

Conversely, assume $\mathcal{A}$ is not 0-Det. Since the transition $(q_0, \#, -1, q_{All})$ cannot be taken by $\mathcal{B}$ with initial counter value 0, $\mathcal{L}(\mathcal{B}, 0) = \{\#w\}_{w \in \mathcal{L}(\mathcal{A},0)}$, hence $\mathcal{B}$ is not 0-Det (since a DOCN for $\mathcal{L}(\mathcal{B}, 0)$ would easily imply a DOCN for $\mathcal{L}(\mathcal{A}, 0)$). Thus, $\mathcal{B}$ is not $\forall$-Det. ◀

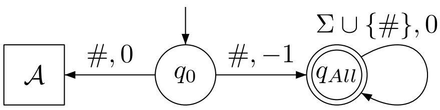


Figure 2 The OCN $\mathcal{B}$ in Theorem 7.

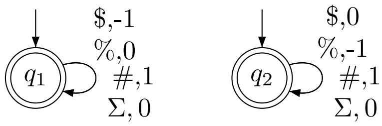


Figure 3 Gadget OCN $\mathcal{G}$ for Theorem 9.

Next, we show the undecidability of $\exists$-Det with a reduction from the halting problem for two-counter (Minsky) machines (2CM). Technically, we rely on a construction from [2], which reduces the latter problem to the "parameterized universality" problem for OCN. For our purpose, the reader need not be familiar with Minsky Machines, as it suffices to know that their halting problem is undecidable [24]. We start the reduction with the following property.

▶ **Theorem 8** ([2]). *Given a 2CM $\mathcal{M}$, we can construct an OCN $\mathcal{B}$ over alphabet $\Sigma \cup \{\#\}$ with $\# \notin \Sigma$ such that the following holds:*

- *If $\mathcal{M}$ halts, there exists $c \in \mathbb{N}$ such that $\mathcal{L}(\mathcal{B}, c) = \Sigma^*$,*
- *If $\mathcal{M}$ does not halt, then for every $c \in \mathbb{N}$ there exists a word $w_c \in (\Sigma \cup \{\#\})^*$ such that every run of $\mathcal{B}$ on w_c enters a state from which reading any word of the form $\#^*$ does not lead to an accepting state.*

We can now proceed with the reduction to $\exists$-Det.

▶ **Theorem 9.** *$\exists$-Det is undecidable.*

Proof. We reduce the halting problem for 2CM to $\exists$-Det. Given a 2CM $\mathcal{M}$, we start by constructing the OCN $\mathcal{B}$ as per Theorem 8. We augment $\mathcal{B}$ to work over the alphabet $\Sigma' = \Sigma \cup \{\#, \$, \%\}$, where $\$, \% \notin \Sigma$, by fixing the behaviour of \$ and % to be identical to #.

Next, consider the gadget OCN $\mathcal{G}$ depicted in Figure 3. A similar argument to the proof of Lemma 2 (specifically, Figure 1a), shows that $\mathcal{G}$ does not have an equivalent DOCN for any initial counter c.

We now obtain a new OCN $\mathcal{A}$ by taking the union of $\mathcal{B}$ and $\mathcal{G}$ (i.e. placing them "side by side"). We claim that $\mathcal{M}$ halts iff $\mathcal{A}$ is $\exists$-Det.

If $\mathcal{M}$ halts, by Theorem 8 there exists an initial counter c such that $\mathcal{L}(\mathcal{B}, c) = \{\Sigma \cup \{\#\}\}^*$. Since in $\mathcal{B}$ the letters \$ and % behave like #, we have that $\mathcal{L}(\mathcal{A}, c) = \Sigma'^*$, so $\mathcal{A}$ is $\exists$-Det.

If M does not halt, then again by Theorem 8, for every $c \in \mathbb{N}$ there exists a word w_c such that $w_c \notin \mathcal{L}(\mathcal{B}, c)$, and such every run of $\mathcal{B}$ on w_c enters a state from which reading $\#^*$ (and hence any word from $\{\#, \%\,\$\}^*$) does not lead to an accepting state. Now assume by way of contradiction that $\mathcal{A}$ has a deterministic equivalent $\mathcal{D}$ with k states for initial counter c. $\mathcal{A}$ accepts w_c with the runs of $\mathcal{G}$, since w_c does not contain \$ or %. Thus, $\mathcal{D}$ accepts w_c with initial counter 0. In addition, $\mathcal{A}$, and therefore $\mathcal{D}$, both accept $w'_c = w_c\#^{k+1-j}\%^{k+1}\$^{k+1}$ where j is the number of occurrences of #'s in w_c. Using the fact that $\mathcal{G}$ does not have an equivalent DOCN, we can now reach a contradiction with similar arguments as the proof of Lemma 2 (Figure 1a). ◀

4.3 A Lower Bound for `Uniform-Det`

Unfortunately, as of yet we are unable to resolve the decidability of `Uniform-Det`. In this section, we show that `Uniform-Det` is Ackermann-hard, and in particular non primitive recursive.

▶ **Theorem 10.** *`Uniform-Det` is Ackermann-hard.*

Proof. We show a reduction from the OCN universality problem with initial counter 0, shown to be Ackermann-hard in [16].

Consider an OCN $\mathcal{A} = \langle \Sigma, Q, s_0, \delta, F\rangle$. We construct an OCN $\mathcal{B} = \langle \Sigma', Q', q_0, \delta', F'\rangle$ as depicted in Figure 4 (for $\#, \$ \notin \Sigma$).

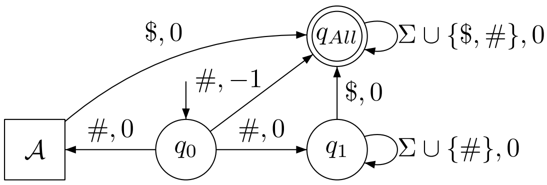


Figure 4 The OCN $\mathcal{B}$ in the proof of Theorem 10.

We claim that $\mathcal{L}(\mathcal{A}, 0) = \Sigma^*$ iff $\mathcal{B}$ is `Uniform-Det`.

Assume $\mathcal{L}(\mathcal{A}, 0) = \Sigma^*$, then $\mathcal{L}(\mathcal{B}, c) = \{\#w : w \in \Sigma'^*\}$ for every counter value c. Indeed, every word starting with # can be accepted by $\mathcal{B}$ with initial counter value 0 either through $\mathcal{A}$, if it does not contain \$, or in q_{All} if it does. However, every word not starting with # cannot be accepted by $\mathcal{B}$ for any initial counter value. Thus, $\mathcal{B}$ is `Uniform-Det`.

Conversely, if $\mathcal{L}(\mathcal{A}, 0) \neq \Sigma^*$, let $w \notin \mathcal{L}(\mathcal{A}, 0)$. Assume by way of contradiction that there exists a deterministic OCN $\mathcal{D}$ that is uniform-equivalent to $\mathcal{B}$.

$\#w \notin \mathcal{L}(\mathcal{B}, 0)$, so $\#w \notin \mathcal{L}(\mathcal{D}, 0)$. Moreover, the run of $\mathcal{D}$ on $\#w$ cannot end in a non-accepting state, since $\#w \in \mathcal{L}(\mathcal{B}, 1) = \mathcal{L}(\mathcal{D}, 1)$. Thus, the run of $\mathcal{D}$ on $\#w$ terminates due to the counter becoming negative. However, this is a contradiction, since $\#w\$ \in \mathcal{L}(\mathcal{B}, 0) = \mathcal{L}(\mathcal{D}, 0)$. We conclude that $\mathcal{B}$ is not `Uniform-Det`. ◀

5 Singleton Alphabet

We now turn to study OCNs over a singleton alphabet denoted $\Sigma = \{\sigma\}$ throughout.

We start by briefly introducing Presburger Arithmetic (PA) [13, 26]. We refer the reader to [13] for a detailed survey. PA is the first-order theory of integers with addition and order $\mathrm{FO}(\mathbb{Z}, 0, 1, +, <)$, and it is a decidable logic.

There is an important connection between PA and semilinear sets: for a *basis vector* $\boldsymbol{b} \in \mathbb{Z}^d$ and a set of *periods* $P = \{\boldsymbol{p_1} \dots \boldsymbol{p_k}\} \subseteq \mathbb{Z}^d$, we define the *linear set* $\mathrm{Lin}(\boldsymbol{b}, P) = \{\boldsymbol{b} + \lambda_1 \boldsymbol{p_1} + \dots + \lambda_k \boldsymbol{p_k} : \lambda_i \in \mathbb{N} \text{ for all } 1 \le i \le k\}$. Then, a *semilinear* set is a finite union of linear sets.

A fundamental theorem about PA [12] shows that that for every PA formula $\varphi(\boldsymbol{x})$ with free variables $\boldsymbol{x}$, the set $[\![\varphi]\!] = \{\boldsymbol{a} : \boldsymbol{a} \models \varphi(\boldsymbol{x})\}$ is semilinear, and the converse also holds – every semilinear set is PA-definable.

Consider an OCN $\mathcal{A}$ over $\Sigma = \{\sigma\}$. For every word σ^n, either σ^n is not accepted by $\mathcal{A}$ for any counter value, or there exists a minimal counter value c such that $\sigma^n \in \mathcal{L}(\mathcal{A}, c')$ iff $c' \ge c$. We can therefore fully characterize the language of $\mathcal{A}$ on any counter value using the *Minimal Counter Relation*[1] (MCR), defined as

$$\mathrm{MCR}(\mathcal{A}) = \left\{(n, c) \subseteq \mathbb{N}^2, c \text{ is the minimal integer such that } \sigma^n \in \mathcal{L}(\mathcal{A}, c)\right\}.$$

We start by showing that $\mathrm{MCR}(\mathcal{A})$ is semilinear.

▶ **Lemma 11.** *Consider an OCN $\mathcal{A}$ over $\Sigma = \{\sigma\}$, then MCR($\mathcal{A}$) is effectively semilinear.*

Proof. We prove the claim using well-known and deep results about low-dimensional VASS. A 2D-VASS is (for our purposes[2]) identical to an OCN over $\Sigma = \{\sigma\}$, but has two counters (both need to be kept non-negative). Formally, a 2D VASS is $\mathcal{V} = \langle Q, s_0, \delta, F\rangle$, where $\delta \subseteq Q \times \mathbb{Z}^2 \times Q$. The semantics are similar to OCNs, acting separately on the two counters, as follows. A *configuration* of $\mathcal{V}$ is $(q, (c_1, c_2))$ where $q \in Q$ and $(c_1, c_2) \in \mathbb{N}^2$ are the counter values, and a *run* is a sequence of configurations $(q_1, (c_1^1, c_2^1)), \dots, (q_k, (c_1^k, c_2^k))$ that follow according to δ, i.e., for every $1 \le i < k$ we have that $(q_i, (c_1^{i+1} - c_1^i, c_2^{i+1} - c_2^i), q_{i+1}) \in \delta$. We denote $(q_1, (c_1^1, c_2^1)) \xrightarrow{\mathcal{V}} (q_k, (c_1^k, c_2^k))$ if such a run exists.

In [22], it is proved that given a 2D-VASS, we can effectively compute a PA formula $\psi_{\mathrm{Reach}}(q, x_1, x_2, q', y_1, y_2)$ such that $[\![\psi_{\mathrm{Reach}}(q, x_1, x_2, q', y_1, y_2)]\!] = \{(q, c_1, c_2, q', d_1, d_2) : (q, c_1, c_2) \xrightarrow{\mathcal{V}} (q', d_1, d_2)\}$ (the states q, q' are encoded as variables taking values in $\{1, \dots, |Q|\}$).

Observe that ψ_{Reach} does not encode information about the length of the run, whereas MCR does require it. On the other hand, ψ_{Reach} works for 2D-VASS, whereas we only need an OCN (i.e., 1D-VASS). We therefore proceed by first introducing the notion of Linear Path Schemes [22, 4]. Consider the transitions of $\mathcal{A}$ as an alphabet (i.e., each transition $(q, \sigma, v, p) \in \delta$ is a letter). A *Linear Path Scheme* is a regular expression of the form $\rho = \alpha_0 \beta_1^* \alpha_1 \cdots \beta_k^* \alpha_k$ where the α_i and β_i are words in δ^*, such that each α_i represents a path in $\mathcal{A}$, and each β_i represents a cycle. The *length* of ρ is defined as $|\alpha_1| + |\beta_1| + \dots + |\alpha_k| + |\beta_k|$, i.e., the length of the underlying path, excluding repetitions of the β_i.

The following result can be obtained from [4] by using 2D-VASS as a proxy, as we do in Lemma 11, or directly from [2].

▶ **Lemma 12.** *Let $\mathcal{A}$ be an OCN over singleton alphabet, then there exists a finite set S of linear path schemes such that the following holds:*

1. *Every $\rho \in S$ has length at most $2|Q|^2$.*
2. *For every two configurations $(p, c_1), (q, c_2) \in Q \times \mathbb{N}$ and every $n \in \mathbb{N}$, if there is a run of $\mathcal{A}$ on σ^n from (p, c_1) to (q, c_2), then there is such a run of the form $\rho \in S$.*

[1] We remark that $\mathrm{MCR}(\mathcal{A})$ is in fact the graph of a partial function. For convenience of working with PA, we stick with the relation notation.

[2] Usually, OCNs are defined as 1D-VASS, not the other way around.

As a consequence of Lemma 12, in order to decide if σ^n is accepted in $\mathcal{A}$, it is enough to consider runs that are linear path schemes of length at most $2|Q|^2$.

Consider a linear path scheme $\rho = \alpha_0\beta_1^*\alpha_1\beta_2^*\alpha_2\cdots\alpha_k^*$. We now construct a formula $\varphi_\rho(n,c)$ which intuitively states that the word σ^n has a run of the form ρ starting with initial counter value c. This is defined as follows.

$$\varphi_\rho(n,c) := \exists e_1\cdots e_k,\ \textsc{correct-length}_\rho(n,e_1\cdots e_k) \wedge \textsc{sufficient-counter}_\rho(c,e_1\cdots e_k)$$

Intuitively, $\varphi_\rho(n,c)$ states that there exist numbers $e_1\cdots e_k$ such that the concrete run $\alpha_0\beta_1^{e_1}\alpha_1\beta_2^{e_2}\alpha_2\cdots\alpha_k$ takes exactly n transitions, and that starting the run with initial counter c is sufficient to complete the run.

Formally, we define the sub-formulas as follows:

- $\textsc{correct-length}_\rho(n,e_1\cdots e_k) := |\alpha_0| + e_1\cdot|\beta_1| + |\alpha_1| + \cdots + |\alpha_k| = n$.
- $$\begin{aligned}&\textsc{sufficient-counter}_\rho(c,e_1\cdots e_k) :=\\ &\bigwedge_{i=0}^{k} c + \mathsf{eff}(\alpha_0) + e_1\cdot\mathsf{eff}(\beta_1) + \cdots + e_i\cdot\mathsf{eff}(\beta_i) \geq \mathsf{nadir}(\alpha_i)\\ \wedge &\bigwedge_{i=1}^{k} (c + \mathsf{eff}(\alpha_0\beta_1^{e_1}\cdots\alpha_{i-1}) \geq \mathsf{nadir}(\beta_i) \wedge c + \mathsf{eff}(\alpha_0\beta_1^{e_1}\cdots\alpha_{i-1}) + (e_i-1)\mathsf{eff}(\beta_i) \geq \mathsf{nadir}(\beta_i^{e_i}))\end{aligned}$$

The correctness of $\textsc{correct-length}_\rho$ is obvious. The correctness of the formula $\textsc{sufficient-counter}_\rho(c,e_1\cdots e_k)$ is based on the observation that in order to traverse the cycle β for e times, the counter c must be enough to traverse β once, and must be enough so that $c + (e-1)\mathsf{eff}(\beta) \geq \mathsf{nadir}(\beta)$, so that the "last" time can be traversed [3]. Indeed, if the counter becomes negative during some iteration of the cycle, it will be even "more" negative at the last iteration. See [4] for an analogous proof.

We can now readily obtain the formula $\theta(n,c)$ which captures $\mathrm{MCR}(\mathcal{A})$ as follows: define $P \subseteq S$ to be the set of linear path schemes that start in q_0 and end in an accepting state, then

$$\theta(n,c) := \bigvee_{\rho\in P} \varphi_\rho(n,c) \wedge \forall c' < c,\ \bigwedge_{\rho\in P} \neg\varphi_\rho(n,c').$$

Indeed, $\theta(n,c)$ is satisfied iff there exists some linear path scheme $\rho \in P$ that can be traversed with length n and counter value c, and there is no smaller counter for which this holds.

Note that we can obtain $\theta(n,c)$ from $\mathcal{A}$ in polynomial space, by generating all possible linear path schemes of length $2|Q|^2$ and constructing the respective subformulas. In particular, the length of $\theta(n,c)$ is single exponential in the description of $\mathcal{A}$. Moreover, $\theta(n,c)$ has two quantifier alternations – the disjunction is an existential formula, and the conjunction of negations can be viewed as a universal formula. Since quantifier alternation counting assumes starting with an existential quantifier, the universal formula is counted as two alternations. ◀

[3] This argument assumes strictly positive exponents. This assumption is safe, since we can define a set S' that contains all linear path schemes obtained by possibly omitting any number of cycles in any of the linear path schemes in S. Every legal path in S can then be represented by a path in S' whose exponents are all strictly positive. By working with S' we then circumvent this issue. Note that $|S'|$ is still single-exponential in $|\mathcal{A}|$.

5.1 Decidability of `Uniform-Det` over Singleton Alphabet

In this subsection we prove that `Uniform-Det` is decidable for OCN over a singleton alphabet, and we can construct an equivalent DOCN, if one exists. Our characterization of `Uniform-Det` is based on its MCR, and specifically on two notions for subsets of $\mathbb{N}^2$ (applied to MCR). Consider a set $S \subseteq \mathbb{N}^2$. We say that S is *increasing* if it is the graph of an increasing partial function. That is, for every $(n_1, c_1), (n_2, c_2) \in S$, if $n_1 \leq n_2$ then $c_1 \leq c_2$, and if $n_1 = n_2$ then $c_1 = c_2$. Next, we say that S is (N, k, d)-*Ultimately Periodic* for $N, k, d \in \mathbb{N}$ if for every $n \geq N$, $(n, x) \in S$ iff $(n + k, x + d) \in S$. We say that S is (effectively) ultimately periodic if it is (N, k, d)-ultimately periodic for some (effectively computable) parameters $N, k, d \in \mathbb{N}$.

The main technical result of this section is the following.

▶ **Theorem 13.** *Consider an OCN $\mathcal{A}$ over $\Sigma = \{\sigma\}$, then the following are equivalent:*

1. *MCR($\mathcal{A}$) is increasing.*
2. *MCR($\mathcal{A}$) is increasing and effectively ultimately periodic.*
3. *$\mathcal{A}$ is `Uniform-Det`, and we can effectively compute an equivalent DOCN.*

We prove Theorem 13 in the remainder of this section. We start with a technical lemma concerning the implication $1 \implies 2$.

▶ **Lemma 14.** *Consider an effectively semilinear set $S \subseteq \mathbb{N}^2$. If S is increasing, then S is effectively periodic.*

Proof. Since S is effectively semilinear, then by [12] we can write $S = \bigcup_{i=1}^{M} \mathrm{Lin}(\boldsymbol{b_i}, P_i)$ where $\boldsymbol{b_i} \in \mathbb{N}^2$ and $P_i \subseteq \mathbb{N}^2$ for every $1 \leq i \leq M$. Moreover, by [12, 13], we can assume that each P_i is a linearly-independent set of vectors.

All periods are singletons. We show that since S is increasing, then $|P_i| \leq 1$ for every $1 \leq i \leq M$. Assume $(n_1, c_1), (n_2, c_2) \in P_i$, and denote $\boldsymbol{b_i} = (a, b)$, then by the definition of a linear set, for every $\lambda_1, \lambda_2 \in \mathbb{N}$ we have that $(a, b) + \lambda_1(n_1, c_1) + \lambda_2(n_2, c_2) \in S$. Setting $\lambda_1 = 0$ and $\lambda_2 = n_1$, we have that $(a + n_1 n_2, b + n_1 c_2) \in S$, and setting $\lambda_1 = n_2$ and $\lambda_2 = 0$, we have that $(a + n_2 n_1, b + n_2 c_1) \in S$. Observe that $a + n_1 n_2 = a + n_2 n_1$, and since S is increasing, this implies $b + n_1 c_2 = b + n_2 c_1$, that is $n_1 c_2 = n_2 c_1$. It follows that $n_2(n_1, c_1) = n_1(n_2, c_2)$, but P_i is linearly independent, so it must hold that $(n_1, c_1) = (n_2, c_2)$, so $|P_i| \leq 1$.

Thus, we can in fact write $S = \bigcup_{i=1}^{M} \mathrm{Lin}(\boldsymbol{b_i}, \{\boldsymbol{p_i}\})$ where $\boldsymbol{b_i}, \boldsymbol{p_i} \in \mathbb{N}^2$ (note that if $P_i = \emptyset$ we now take $\boldsymbol{p_i} = (0, 0)$). For every $1 \leq i \leq M$, denote $\boldsymbol{b_i} = (a_i, b_i)$ and $\boldsymbol{p_i} = (p_i, r_i)$.

All Periods have the same first component. We now claim that we can restrict all periods to have the same first component. That is, we can compute $\gamma \in \mathbb{N}$ and write $S = \bigcup_{j=1}^{K} \mathrm{Lin}((\alpha_j, \beta_j), \{(\gamma, \eta_j)\})$.

Indeed, take $\gamma = \mathrm{lcm}(\{p_i\}_{i=1}^{M})$, we now "spread" each linear component $\mathrm{Lin}((a_i, b_i), \{(p_i, r_i)\})$ by changing the period to $(\gamma, \frac{\gamma}{p_i} r_i)$, and compensating by adding additional linear sets with the same period and offset basis, to capture the "skipped" elements. In Appendix A.5 we describe the construction in general, and illustrate with an example.

All Periods are the same. Finally, we claim that we now have $\eta_i = \eta_j$ for every $1 \leq i, j \leq K$, so that in fact all the periods are the same vector (γ, η). Indeed, Assume by way of contradiction that $\eta_j < \eta_i$ for some $1 \leq i, j \leq K$. Now, let $y \in \mathbb{N}$ be large enough so that $\alpha_i \leq \alpha_j + y \cdot \gamma$, and let $x \in \mathbb{N}$ be large enough so that (given y): $\beta_i + x \cdot \eta_i > \beta_j + y \cdot \eta_j + x \cdot \eta_j$.

We now have that $(\alpha_i, \beta_i) + x \cdot (\gamma, \eta_i) \in S$ and $(\alpha_j, \beta_j) + (x + y) \cdot (\gamma, \eta_j) \in S$, which contradicts S being increasing, since $\alpha_i \leq \alpha_j + y \cdot \gamma$ and therefore $\alpha_i + x \cdot \gamma \leq \alpha_j + (y + x) \cdot \gamma$, but also $\beta_i + x \cdot \eta_i > \beta_j + (y + x) \cdot \beta_j$.

Thus, we can now write $S = \bigcup_{j=1}^{K} \text{Lin}((\alpha_j, \beta_j), \{(\gamma, \eta)\})$

S is effectively ultimately periodic. Let $\alpha_{\max} = \max\{\alpha_j\}_{j=1}^{K}$, we claim that S is $(\alpha_{\max}, \gamma, \eta)$-ultimately periodic. Let $n \geq \alpha_{\max}$, then $(n, c) \in S$ for some $c \in \mathbb{N}$ iff $(n, c) = (\alpha_i + \gamma \cdot m, \beta_i + \eta \cdot m)$ for some $1 \leq i \leq K$ and $m \in \mathbb{N}$. This happens iff $(n + \gamma, c + \eta) \in S$, since $(n + \gamma, c + \eta) = (\alpha_i + \gamma \cdot (m + 1), \beta_i + \eta \cdot (m + 1))$.

Finally, observe that all the constants in the proof are effectively computable. ◀

We now turn to the implication 2 $\implies$ 3 of Theorem 13.

▶ **Lemma 15.** *Consider an OCN $\mathcal{A}$ over $\Sigma = \{a\}$. If MCR($\mathcal{A}$) is increasing and ultimately periodic, then $\mathcal{A}$ is* `Uniform-Det`*, and we can effectively compute it.*

Proof. Assume MCR($\mathcal{A}$) is (N, k, d)-ultimately periodic. We start by completing MCR($\mathcal{A}$) to a (full) function $f : \mathbb{N} \to \mathbb{N}$ as follows: set $f(0) = 0$, and for $n > 0$ inductively define $f(n) = c$ if $(n, c) \in \text{MCR}(\mathcal{A})$, or $f(n) = f(n-1)$ otherwise. That is, f matches MCR($\mathcal{A}$) on its domain, and remains fixed between defined values. Observe that there is no violation in defining $f(0) = 0$, since if $(0, c) \in \text{MCR}(\mathcal{A})$, then $c = 0$, as the empty word requires a minimal counter of 0 to be accepted.

We now use f to obtain a DOCN $\mathcal{D}$ as depicted in Figure 5. Formally, we construct $\mathcal{D} = \langle \{\sigma\}, Q, q_0, \delta, F \rangle$ as follows.

- $Q = \{q_i\}_{i=1}^{N+k-1}$.
- $\delta = \{(q_i, a, f(i) - f(i+1), q_{i+1})\}_{i=1}^{N+k-2} \cup \{(q_{N+k-1}, a, f(N) + d - f(N+k-1), q_N)\}$.
- $F = \{q_i : (i, f(i)) \in \text{MCR}(\mathcal{A}),\ 1 \leq i \leq N + k - 1\}$.

Observe that since f is increasing (as MCR($\mathcal{A}$) is increasing), the weight of all transitions in $\mathcal{D}$ is non-positive.

We claim that for every c, $\mathcal{L}(\mathcal{A}, c) = \mathcal{L}(\mathcal{D}, c)$. To show this, observe that for every $n \in \mathbb{N}$ we have that the sum of weights along n consecutive transitions of $\mathcal{D}$ (ignoring the OCN semantics) is exactly $-f(n)$. In particular, if $\sigma^n \in \mathcal{L}(\mathcal{A}, c)$, then $(n, c') \in \text{MCR}(\mathcal{A})$ for some $c' \leq c$ and $f(n) = c'$. Indeed, this is trivial for $n \leq N + k - 1$, and for $n > N + k - 1$ this follows immediately from (N, k, d)-ultimate periodicity.

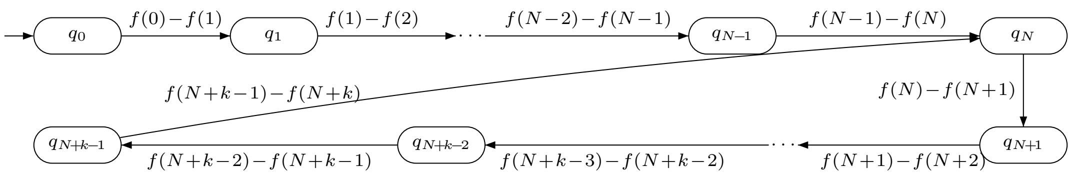


Figure 5 An illustration of the construction method for a uniform-deterministic-equivalent of an OCN $\mathcal{A}$, given f. Accepting states are not mentioned in the illustration.

Thus, if $\sigma^n \in \mathcal{L}(\mathcal{A}, c)$ then there exists $c' \leq c$ such that $(n, c') \in \text{MCR}(\mathcal{A})$ it follows that with initial counter c, $\mathcal{D}$ can traverse n transitions. Moreover, the state reached is accepting, since $(n, c') \in \text{MCR}(\mathcal{A})$, so $\sigma^n \in \mathcal{L}(\mathcal{D}, c)$.

Conversely, if $\sigma^n \in \mathcal{L}(\mathcal{D}, c)$ then $c \geq f(n)$ and $(n, f(n)) \in \text{MCR}(\mathcal{A})$, thus, $\sigma^n \in \mathcal{L}(\mathcal{A}, c)$.

Finally, observe that the construction is computable given the parameters of ultimate periodicity. ◀

We now address the implication $3 \implies 1$.

▶ **Lemma 16.** *Consider an OCN $\mathcal{A}$ over $\Sigma = \{\sigma\}$. If $\mathcal{A}$ is* `Uniform-Det`*, then MCR($\mathcal{A}$) is increasing.*

Proof. Let $\mathcal{D}$ be a DOCN such that $\mathcal{L}(\mathcal{A}, c) = \mathcal{L}(\mathcal{D}, c)$ for every c, and let $(n_1, c_1), (n_2, c_2) \in$ MCR($\mathcal{A}$) with $n_1 \leq n_2$. Assume by way of contradiction that $c_1 > c_2$, then $\sigma^{n_2} \in \mathcal{L}(\mathcal{D}, c_2)$, but $\sigma^{n_1} \notin \mathcal{L}(\mathcal{D}, c_2)$. It follows that the run of $\mathcal{D}$ on σ^{n_1} must end in a non-accepting state starting from counter value c_2 (i.e., the counter does not become negative). But then the same run is taken from counter value c_1, so $\sigma^{n_1} \notin \mathcal{L}(\mathcal{D}, c_1)$, which is a contradiction. ◀

By Lemma 11, MCR($\mathcal{A}$) is semilinear. Thus, if MCR($\mathcal{A}$) is increasing, then by Lemma 14 it is also effectively ultimately periodic. This completes the implication $1 \implies 2$, and the implications $2 \implies 3$ and $3 \implies 1$ are immediate from Lemmas 15 and 16, respectively. This completes the proof of Theorem 13.

Finally, we can show the decidability of `Uniform-Det` by combining the characterization of Theorem 13 with the procedure of Lemma 11 and the decidability of PA [3].

▶ **Theorem 17.** *For OCNs over singleton alphabet,* `Uniform-Det` *is decidable. Moreover, it is in $3-$* `EXPSPACE`*.*

Proof. We start by showing the decidability of `Uniform-Det`. Consider an OCN $\mathcal{A}$. By Theorem 13, it suffices to show that it is decidable whether MCR($\mathcal{A}$) is increasing. By Lemma 11, we can compute a PA formula $\theta(n, c)$ such that $[\![\theta]\!] =$ MCR($\mathcal{A}$). We now state the assertion that MCR($\mathcal{A}$) is not increasing in PA as follows: $\chi = \exists n_1, n_2, c_1, c_2,\ n_1 < n_2 \wedge c_1 > c_2 \wedge \theta(n_1, c_1) \wedge \theta(n_2, c_2)$. Since PA is decidable, we can decide whether this sentence holds.

It remains to analyze the complexity of `Uniform-Det`. To this end, observe that in the proof of Lemma 11 we show that the length of $\theta(n, c)$ is single-exponential in $|\mathcal{A}|$ (and that we can obtain $\theta(n, c)$ from $\mathcal{A}$ in polynomial space). Since PA is decidable in $2-$`EXPSPACE` [3], we conclude that `Uniform-Det` is decidable in $3-$`EXPSPACE`. ◀

▶ Remark 18 (On the $3-$`EXPSPACE` upper bound). It is easy to show that in fact $\theta(n, c)$ has at most 3 quantifier alternations. Therefore, the upper bound can be somewhat lowered using bounds for PA with fixed quantifier alternations [14]. However, applied to the exponential-length formula, these bounds do not get us as low as the next "major" complexity classes (e.g., $3-$`NEXPTIME`, or $2-$`EXPSPACE`), so Theorem 17 is stated with $3-$`EXPSPACE`.

While we suspect this upper bound can be lowered, deciding whether MCR($\mathcal{A}$) is increasing seems to be a hard problem. Indeed, MCR($\mathcal{A}$) intuitively corresponds to the reachability relation of the OCN with two additional constraints: the length of the path is fixed, and the counter value is required to be minimal. The former constraint can be circumvented using 2D-VASS, as we do in Lemma 11, but the latter introduces a flavour of universal quantification. In particular, this poses a barrier to techniques attempting to reduce the behaviour of MCR($\mathcal{A}$) to a reachability relation.

We proceed to give a lower bound on `Uniform-Det` (albeit far from the upper bound).

▶ **Theorem 19.** *For OCNs over singleton alphabet,* `Uniform-Det` *is* `coNP`*-hard.*

Proof. We show a reduction from the universality problem for NFAs over a singleton alphabet, which is `coNP`-hard [29].

We start by describing a gadget OCN $\mathcal{B}$ as depicted in Figure 6. Note that $\mathcal{B}$ is not `Uniform-Det`, since MCR($\mathcal{B}$) is not increasing. Indeed, σ is only accepted with counter 1, whereas $\sigma\sigma$ is accepted with counter 0. Thus, $(1, 1), (2, 0) \in$ MCR($\mathcal{B}$).

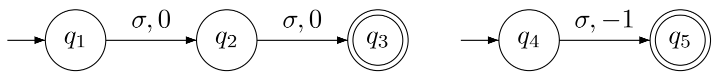


Figure 6 Gadget OCN $\mathcal{B}$ for the reduction in Theorem 19.

We now describe the reduction. Consider an NFA $\mathcal{A} = \langle \{\sigma\}, Q, S_0, \delta, F\rangle$. We assume that $\mathcal{A}$ is complete, i.e. that $\mathcal{A}$ has a (not necessarily accepting) run on every word (if $\mathcal{A}$ is not complete, we add a rejecting sink state as an initial state to $\mathcal{A}$).

We start by obtaining a new NFA $\mathcal{A}' = \langle \{\sigma\}, Q', S_0, \delta', F\rangle$ by "stretching" $\mathcal{A}$ threefold: we define $Q' = \bigcup_{q\in Q} \{q, q', q''\}$ and the transition relation $\delta' = \{(q_1, \sigma, q_1'), (q_1', \sigma, q_1''), (q_1'', \sigma, q_2) : (q_1, \sigma, q_2) \in \delta\}$. We then connect every non-accepting state q originally in $\mathcal{A}$ to the initial states of the gadget OCN $\mathcal{B}$, and we connect every accepting state of $\mathcal{A}'$ to a gadget NFA $\mathcal{C}$ that accepts exactly $\{\sigma, \sigma\sigma\}$.

We now obtain from $\mathcal{A}'$ an OCN $\mathcal{A}''$ by assigning counter updates of 0 on all transitions except those of $\mathcal{B}$.

It remains to prove that $L(\mathcal{A}) = \{\sigma\}^*$ iff $\mathcal{A}''$ is `Uniform-Det`.

Indeed, assume $L(\mathcal{A}) = \{\sigma\}^*$, then for every $n \in \mathbb{N}$, there is an accepting run of $\mathcal{A}''$ on σ^{3n} from counter 0. Since we have connected every accepting state in $\mathcal{A}$ to the gadget $\mathcal{C}$ that accepts both σ and $\sigma\sigma$, we have that σ^{3n+1} and σ^{3n+2} are accepted from counter 0 as well. Therefore, $\mathcal{A}''$ is universal for initial counter 0, hence it is universal for all initial counter values, and in particular $\mathcal{A}''$ is `Uniform-Det`.

Conversely, if $\mathcal{A}$ is not universal, then there exists a word $w = \sigma^n$ such that all runs of $\mathcal{A}$ on w end in non-accepting states (and at least one such run exists, by completeness). We then have that all successful runs of $\mathcal{A}''$ on σ^{3n} end in non-accepting states. The words $\sigma^{3n+1}, \sigma^{3n+2}$ can therefore only be accepted through $\mathcal{B}$. By the structure of $\mathcal{B}$ we then have that $(3n+1, 1), (3n+2, 0) \in \mathrm{MCR}(\mathcal{A}'')$, so $\mathrm{MCR}(\mathcal{A}'')$ is not increasing, and $\mathcal{A}''$ is not `Uniform-Det`. ◀

5.2 `Uniform-Det`– Properties and Fragments

The wide complexity gap between the bounds of Theorems 17 and 19 suggest that `Uniform-Det` is an intricate problem. We now turn to present several results shed some light on the behaviour of `Uniform-Det`.

We start by showing that the first witness to the fact that $\mathrm{MCR}(\mathcal{A})$ is non-increasing may be exponential in $|\mathcal{A}|$. This holds when $\mathcal{A}$ has weights encoded in unary, and if the weights are encoded in binary this holds already for OCNs with 3 states.

▶ **Example 20.** Consider the OCN $\mathcal{A}$ depicted in Figure 7, where k is encoded in binary. It is not hard to verify that for $0 \leq n \leq k$ it holds that $(n, \min(n, k-n+1)) \in \mathrm{MCR}(\mathcal{A})$, but $(k+1, 0) \in \mathrm{MCR}(\mathcal{A})$, since σ^{k+1} is accepted with counter value 0 in the left component. Thus, already for 3-state OCNs, the minimal witness for decreasing MCR can be exponential.

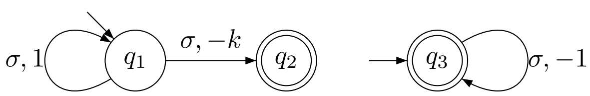


Figure 7 Binary encoded OCN $\mathcal{A}$ in Example 20.

▶ **Example 21.** We now describe a unary-encoded OCN whose minimal witness for decreasing MCR is exponential. Let $p_1, \dots, p_m$ be the first m prime numbers. We construct an OCN $\mathcal{A}$ as a disjoint union of m cycles of lengths $p_1, \dots, p_m$, where on each cycles all transitions have counter update -1, and all states are accepting except the initial state on each cycle. In addition, $\mathcal{A}$ has another initial and accepting self loop with counter update -2.

Let $M = \prod_{i=1}^{m} p_i$, then for every $0 \le n < M$ we have that $(n, n) \in \text{MCR}(\mathcal{A})$, since upon reading σ^n at least one cycle of length p_j does not divide n and is therefore not back at its initial state. Similarly, $(M+1, M+1) \in \text{MCR}(\mathcal{A})$. However, $(M, 2M) \in \text{MCR}(\mathcal{A})$ since σ^M is only accepted in the -2 self loop. Thus, the first witness for the non-increasing MCR is $M+1$, which is exponential in $|\mathcal{A}| = O(\sum_{i=1}^{n} p_i)$.

The next property shows that when all states are accepting, `Uniform-Det` becomes trivial.

▶ **Theorem 22.** *Consider an OCN $\mathcal{A}$ over a singleton alphabet such that all states in $\mathcal{A}$ are accepting. Then $\mathcal{A}$ is `Uniform-Det`.*

Proof. We show that $\text{MCR}(\mathcal{A})$ is increasing, and therefore $\mathcal{A}$ is `Uniform-Det`. Let $n_1, c_1 \in \mathbb{N}$ such that $(n_1, c_1) \in \text{MCR}(\mathcal{A})$, then initial counter c_1 is sufficient for $\mathcal{A}$ to read (and hence accept) σ^{n_1} via some run ρ. Let $n_2 < n_1$, then $\mathcal{A}$ reads σ^{n_2} along a prefix of ρ with initial counter value c_1, and since all states are accepting, c_1 is sufficient to accept σ^{n_2}. Thus, if $(n_2, c_2) \in \text{MCR}(\mathcal{A})$, we have $c_2 \le c_1$, so $\text{MCR}(\mathcal{A})$ is increasing. ◀

Our final property concerns unambiguous OCNs. An OCN $\mathcal{A}$ over alphabet $\{\sigma\}$ is *unambiguous* if for every $n \in \mathbb{N}$ there exists at most one accepting run of $\mathcal{A}$ on σ^n, for any counter value c. Technically, this means that the OCN is structurally unambiguous, in that its underlying NFA is unambiguous.

▶ **Theorem 23.** *For unambiguous OCNs over a singleton alphabet, deciding `Uniform-Det` is in PSPACE.*

Proof. Let $\mathcal{A}$ be an unambiguous OCNs over a singleton alphabet. In Appendix A.6 we show that by careful analysis of the PA formula obtained as per Lemma 11, we can represent the notion of $\text{MCR}(\mathcal{A})$ being non-increasing using a PA formula ν that is a disjunction of exponentially many existential formulas – each polynomial in the size of $\mathcal{A}$. By traversing these fragments in polynomial space, and since existential PA is decidable in `NP` [8], we conclude the `PSPACE` bound. ◀

5.3 Triviality of `0-Det`, ∀`-Det`, ∃`-Det`

We now turn to study the remaining notions of determinization for singleton alphabet.

▶ **Theorem 24.** *Consider an OCN $\mathcal{A}$ over $\Sigma = \{\sigma\}$, then $\mathcal{A}$ is ∀`-Det`, `0-Det`, and ∃`-Det`.*

Proof. By Observation 1, it is enough to prove that $\mathcal{A}$ is ∀`-Det`. To this end, recall that by Lemma 11, $\text{MCR}(\mathcal{A})$ is PA definable by a formula $\varphi(n, c)$.

For every initial counter value c, define $\varphi_{\le c}(n) = \bigvee_{i=0}^{c} \varphi(n, i)$, then $[\![\varphi_{\le c}(n)]\!] = \{n : \mathcal{A} \text{ accepts } \sigma^n \text{ with initial counter } c\}$. Then, we can write $\mathcal{L}(\mathcal{A}, c) = \{\sigma^m : m \in [\![\varphi_{\le c}(n)]\!]\}$.

It is folklore that a singleton-alphabet language whose set of lengths is semilinear, is regular. We bring a short proof of this for completeness: Let $S = \bigcup_{i=1}^{k} \text{Lin}(c_i, p_i) \subseteq \mathbb{N}$ be a semilinear set (by assuming that the periods are linearly independent, it follows each has a single number), and let $L_S = \{\sigma^k : k \in S\}$. For every i, the language $\{a^k | k \in \text{Lin}(c_i, p_i)\}$ can be defined by the regular expression $r_i = \sigma^{c_i}(\sigma^{p_i})^*$. So L_S is defined by the regular expression $r = r_1 + \cdots + r_k$.

Thus, for every $c \in \mathbb{N}$, we have that $\mathcal{L}(\mathcal{A}, c)$ is regular, and in particular is recognized by a DOCN, so $\mathcal{A}$ is ∀`-Det`, and we are done. ◀

6 Discussion and Future Work

In this work, we introduce and study notions of determinization for OCNs. We demonstrate that the notions, while comparable in strictness, are distinct both from a conceptual perspective, having different motivations, as well as from a technical perspective: the mathematical tools needed to analyze them vary.

The most pressing direction for future work is resolving the decidability status of `Uniform-Det`. Note that `Uniform-Det` bears some similarities to the determinization problem for tropical automata, in that both models essentially follow the $(\min, +)$ semantics. The differences between the models are that (1) in OCNs we only care about Boolean acceptance, whereas in weighted automata we need to match the function exactly, and (2) in OCNs we have the restriction that the counter is nonnegative, unlike in weighted automata.

The determinization problem of weighted automata is famously open, and thus it could well be that `Uniform-Det` is similarly difficult. It is worth noting that techniques for handling the determinization of weighted automata in some fragments (namely unambiguous [25], or polynomially ambiguous [20]) can be easily shown not carry over to determinization of OCNs, meaning that besides the semantic differences, there are also technical differences in reasoning about these models.

Another important direction of future work is tightening the complexity gap of `Uniform-Det` over singleton alphabet. Our preliminary analysis in Lemma 11 suggests that this may require a more ad-hoc technique than using Presburger Arithmetic.

References

1 Parosh Aziz Abdulla and Karlis Čerāns. Simulation is decidable for one-counter nets. In *International Conference on Concurrency Theory*, pages 253–268. Springer, 1998.

2 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.

3 Leonard Berman. The complexity of logical theories. *Theoretical Computer Science*, 11(1):71–77, 1980.

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 Stanislav Böhm, Stefan Göller, and Petr Jančar. Bisimilarity of one-counter processes is pspace-complete. In *International Conference on Concurrency Theory*, pages 177–191. Springer, 2010.

6 Stanislav Böhm, Stefan Göller, and Petr Jančar. Bisimulation equivalence and regularity for real-time one-counter automata. *Journal of Computer and System Sciences*, 80(4):720–743, 2014.

7 Udi Boker and Thomas A. Henzinger. Exact and approximate determinization of discounted-sum automata. *Log. Methods Comput. Sci.*, 10(1), 2014. doi:10.2168/LMCS-10(1:10)2014.

8 Itshak Borosh and Leon Bruce Treybig. Bounds on positive integral solutions of linear diophantine equations. *Proceedings of the American Mathematical Society*, 55(2):299–304, 1976.

9 Adam L Buchsbaum, Raffaele Giancarlo, and Jeffery R Westbrook. On the determinization of weighted finite automata. *SIAM Journal on Computing*, 30(5):1502–1531, 2000.

10 Wojciech Czerwiński and Slawomir Lasota. Regular separability of one counter automata. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

11 Alain Finkel, Gilles Geeraerts, J-F Raskin, and Laurent Van Begin. On the ω-language expressive power of extended petri nets. *Theoretical computer science*, 356(3):374–386, 2006.

12 Seymour Ginsburg and Edwin H Spanier. Bounded algol-like languages. *Transactions of the American Mathematical Society*, 113(2):333–368, 1964.

13 C. Haase. A survival guide to presburger arithmetic. *ACM SIGLOG News*, 5(3):67–82, 2018. URL: `https://dl.acm.org/citation.cfm?id=3242964`, `doi:10.1145/3242953.3242964`.

14 Christoph Haase. Subclasses of presburger arithmetic and the weak exp hierarchy. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10, 2014.

15 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.

16 Piotr Hofman and Patrick Totzke. Trace inclusion for one-counter nets revisited. *Theoretical Computer Science*, 735:50–63, 2018.

17 Petr Jančar, Antonín Kučera, and Faron Moller. Simulation and bisimulation over one-counter processes. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 334–345. Springer, 2000.

18 Petr Jancar and Faron Moller. Simulation of one-counter nets via colouring. In *Proceedings of Workshop Journées Systemes Infinis*, pages 1–6, 1999.

19 Petr Jančar, Faron Moller, et al. Simulation problems for one-counter machine. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 404–413. Springer, 1999.

20 Daniel Kirsten and Sylvain Lombardy. Deciding unambiguity and sequentiality of polynomially ambiguous min-plus automata. In *26th International Symposium on Theoretical Aspects of Computer Science*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2009.

21 Ines Klimann, Sylvain Lombardy, Jean Mairesse, and Christophe Prieur. Deciding unambiguity and sequentiality from a finitely ambiguous max-plus automaton. *Theoretical Computer Science*, 327(3):349–373, 2004.

22 Jérôme Leroux and Grégoire Sutre. On flatness for 2-dimensional vector addition systems with states. In *International Conference on Concurrency Theory*, pages 402–416. Springer, 2004.

23 Richard Mayr. Undecidable problems in unreliable computations. *Theoretical Computer Science*, 297(1-3):337–354, 2003.

24 Marvin Lee Minsky. *Computation: Finite and Infinite Machines*. Prentice-Hall Englewood Cliffs, 1967.

25 M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23(2):269–311, 1997.

26 Mojzesz Presburger. Uber die vollstandigkeiteines gewissen systems der arithmetik ganzer zahlen, in welchen die addition als einzige operation hervortritt. In *Comptes-Rendus du ler Congres des Mathematiciens des Pays Slavs*, 1929.

27 Philippe Schnoebelen. Lossy counter machines decidability cheat sheet. In *International Workshop on Reachability Problems*, pages 51–75. Springer, 2010.

28 Jiří Srba. Beyond language equivalence on visibly pushdown automata. *Logical Methods in Computer Science*, 5, 2009.

29 Larry J Stockmeyer and Albert R Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 1–9, 1973.

30 Patrick Totzke, Richard Mayr, Slawomir Lasota, and Piotr Hofman. Simulation problems over one-counter nets. *Logical Methods in Computer Science*, 12, 2016.

31 Leslie Valiant. *Decision procedures for families of deterministic pushdown automata*. PhD thesis, University of Warwick, 1973.

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

A Proofs

A.1 Proof of Lemma 2

A.1.1 $\mathcal{A}$ is $\exists$-Det, but not 0-Det

We define formally $\mathcal{A} = \langle \{a, b, c, \#\}, \{q_0, q', q'', q_5\}, q_0, \delta_{\mathcal{A}}, \{q', q'', q_5\}\rangle$, for:

$\delta_{\mathcal{A}} = \langle (q_0, \#. - 5, q_5), (q_0, \#.0, q'), (q_0, \#.0, q''), (q', a, 1, q'), (q', b, 0, q')\rangle \cup$
$\{(q', c, -1, q'), (q'', a, 0, q''), (q'', b, 1, q''), (q'', c, -1, q''), (q_5, a, 0, q_5), (q_5, b, 0, q_5), (q_5, c, 0, q_5)\}$.

$\mathcal{A}$ is $\exists$-Det, since $\mathcal{L}(\mathcal{A}, k) = \Sigma^*$ for $k \geq 5$. Now, assume by way of contradiction that $\mathcal{A}$ is 0-Det, and let $\mathcal{D}$ be a deterministic OCN with $n \in \mathbb{N}$ states that satisifies $\mathcal{L}(\mathcal{A}, 0) = \mathcal{L}(\mathcal{D}, 0)$. We now define $w = \#c^{n+1}a^{n+1}b^{n+1}$. throughout the run of $\mathcal{D}$ on w, $\mathcal{D}$ travels through a cycle β_1 when reading a^{n+1}, and a cycle β_2 when reading b^{n+1}. If the cumulative costs of both β_1 and β_2 are non-negative, then $\mathcal{D}$ accepts $w' = \#c^{n+1}a^N b^N$ for arbitrarily large $N \in \mathbb{N}$, which contradicts $\mathcal{L}(\mathcal{A}, 0) = \mathcal{L}(\mathcal{D}, 0)$. Otherwise, the cumulative cost of either β_1 or β_2 is negative, w.l.o.g β_1. In this case, $w'' = \#c^{n+1}a^N$ is not accepted by $\mathcal{D}$ for sufficiently large $N \in \mathbb{N}$, which again contradicts $\mathcal{L}(\mathcal{A}, 0) = \mathcal{L}(\mathcal{D}, 0)$. ◀

A.1.2 $\mathcal{B}$ is 0-Det, but not $\forall$-Det

We define formally $\mathcal{B} = \langle \{a, b, c, \#\}, \{q_0, q', q''\}, q_0, \delta_{\mathcal{B}}, \{q', q''\}\rangle$, for:

$\delta_{\mathcal{B}} = \{(q_0, \#. - 1, q'), (q_0, \#. - 1, q''), (q', a, 1, q'), (q', b, 0, q'), (q', c, -1, q')\} \cup$
$\{(q'', a, 0, q''), (q'', b, 1, q''), (q'', c, -1, q'')\}$.

Since $\mathcal{L}(\mathcal{B}, 0) = \emptyset$, $\mathcal{B}$ is 0-Det trivially. However, since with initial counter 0, both $(q_0, \#. - 1, q')$ and $(q_0, \#. - 1, q'')$ cannot be traversed, we have that $\mathcal{L}(\mathcal{B}, 1) = \mathcal{L}(\mathcal{A}, 0)$. therefore, as can be shown by an identical analysis to the one presented in Appendix A.1.1, there is no deterministic OCN $\mathcal{D}$ that satisfies $\mathcal{L}(\mathcal{B}, 1) = \mathcal{L}(\mathcal{D}, 0)$, and $\mathcal{B}$ is not $\forall$-Det. ◀

A.1.3 $\mathcal{C}$ is $\forall$-Det, but not Uniform-Det

We define formally $\mathcal{C} = \langle \{a, b, \#\}, \{q_0, q_1, q_2\}, q_0, \delta_{\mathcal{C}}, \{q_1, q_2\}\rangle$, for:

$\delta_{\mathcal{C}} = \{(q_0, \#.0, q_1), (q_0, \#. - 1, q_2), (q_1, a.1, q_1), (q_1, b, -1, q_1)\} \cup$
$\{(q_2, a, 0, q_2), (q_2, b, 0, q_2)\}$.

For initial counter 0, the transition $(q_0, \#. - 1, q_2)$ cannot be traversed, therefore $\mathcal{C}$ is 0-Det, since $\mathcal{D} = \langle \{a, b, \#\}, \{q_0, q_1\}, q_0, \{(q_0, \#.0, q_1), (q_1, a.1, q_1), (q_1, b, -1, q_1)\}, \{q_1\}\rangle$ satisfies $\mathcal{L}(\mathcal{D}, 0) = \mathcal{L}(\mathcal{C}, 0)$. In addition, $\mathcal{L}(\mathcal{C}, k) = \#\{a, b\}^*$ for all $k \geq 1$. Hence $\mathcal{C}$ is $\forall$-Det.

Now assume by way of contradiction that $\mathcal{C}$ is Uniform-Det, and let $\mathcal{D}$ be a deterministic OCN with $n \in \mathbb{N}$ states that satisfies $\mathcal{L}(\mathcal{D}, k) = \mathcal{L}(\mathcal{C}, k)$ for all $k \in \mathbb{N}$, and let $w = \#a^{n+1}b^{n+1} \in \mathcal{L}(\mathcal{D}, k)$ for all $k \in \mathbb{N}$. $\mathcal{D}$ travels through a cycle β when reading b^{n+1}. If the cumulative weight of β is non-negative, then $w' = \#a^{n+1}b^N \in \mathcal{L}(\mathcal{D}, 0)$ for arbitrarily large $N \in \mathbb{N}$, which contradicts $\mathcal{L}(\mathcal{D}, 0) = \mathcal{L}(\mathcal{C}, 0)$. If, however, the cumulative weight of β is negative, then $w' = \#a^{n+1}b^N \notin \mathcal{L}(\mathcal{D}, 1)$ for large enough $N \in \mathbb{N}$, which in turn contradicts $\mathcal{L}(\mathcal{D}, 1) = \mathcal{L}(\mathcal{C}, 1)$. ◀

A.2 Proof of Lemma 3

We prove undecidability of 0-Finite-Reach using a straightforward reduction from Finite-Reach. Given an LCM $\mathcal{M} = (\mathsf{Loc}, \mathcal{C}, \Delta)$ and a configuration $\sigma_0 = \langle q, (a_1, a_2 \ldots a_n)\rangle$, we define an LCM $\mathcal{M}'$ with a new initial state q_0 that leads to q with a single path that increments z_1 a_1 times, z_2 a_2 times, etc.

Formally, If $a_i = 0$ for all $0 \le i \le n$, we define $\mathcal{M}' = \mathcal{M}$ and the reduction is trivial. Otherwise, we define $\mathcal{M}' = (\mathsf{Loc}', \mathcal{C}, \Delta')$ such that:

- $\mathsf{Loc}' = \mathsf{Loc} \cup \{q_0\} \cup \left\{\{q_i\}_{i=1}^{\Sigma_{j=1}^n a_j - 1}\right\}$. Note that if $\Sigma_{j=1}^n a_j = 1$, the only new state added is q_0.
- $\Delta' = \Delta \cup \left\{(q_{\Sigma_{j=1}^n a_j - 1}, (z_y, ++), q)\right\} \cup \{(q_i, (z_x, ++), q_{i+1})\}$ such that y is the largest integer $0 \le y \le n$ for which $a_y \ne 0$, and the parameter x varies such that throughout the $\Sigma_{j=1}^n a_j$ transitions, each counter z_i is incremented exactly a_i times.

The reachability set of $\sigma_0 = \langle q, (a_1, a_2 \dots a_n)\rangle$ under $\mathcal{M}$ is finite iff the reachability sets of all configurations $\sigma_0' = \langle q, (a_1', a_2' \dots a_n')\rangle$ such that $a_i' \le a_i$ for all i are finite, due to monotonicity of LCMs. This, in turn, is satisfied iff the reachability set of $\langle q_0, (0, 0 \dots 0)\rangle$ under $\mathcal{M}'$ is finite. ◀

A.3 Proof of Lemma 4

We start by describing several gadgets used in the construction.

A.3.1 Gadgets

Let $\mathcal{M} = \langle \mathsf{Loc}, \mathsf{Z}, \Delta\rangle$ be an LCM, let $z_i \in \mathsf{Z}$, and let $(\ell_1, \mathsf{op}, \ell_2) \in \Delta$. Our goal is to construct an OCN $\mathcal{A}$ that reads two consecutive configuration encodings - an encoding that corresponds to a visit in ℓ_2 and then an encoding that corresponds to a visit in ℓ_1, such that $w \in \mathcal{L}(\mathcal{A}, 0)$ iff w admits a violation for counter z_i.

The structure of $\mathcal{A}$ depends on the value of op, which can any of the following:

1. $z_i{++}$, i.e., increment z_i,
2. $z_i{--}$, i.e., decrement z_i,
3. $z_j{++}$ or $z_j{--}$ for $j \ne i$, which does not affect z_i,
4. $z_i{=}\,0?$, i.e., test z_i for 0.

In addition, we have a special gadget to capture violations in the initial configuration, namely if the counter values is not 0 (recall that the initial configuration is read last, since the encoding is reversed).

Thus, $\mathcal{A}$ can be any of the gadgets presented in figure 8 (depending on op).

Formally, we define $\mathcal{A} = \langle \Sigma, \{q_0, q_1, q_2\}, q_0, \delta, \{q_2\}\rangle$ such that:

- $\Sigma = \mathsf{Loc} \cup \left\{\{a_i\}_{z_i \in \mathsf{Z}}\right\}$.
- $\delta = \{(q_0, a_j, 0, q_0)\}_{j \ne i} \cup \{(q_0, a_i, 1, q_0)\} \cup \{(q_0, \ell_2, \nu, q_1)\} \cup \{(q_1, a_j, 0, q_1)\}_{j \ne i} \cup \{(q_1, a_i, -1, q_1)\} \cup \{(q_1, \ell_1, 0, q_2)\} \cup \{(q_2, \sigma, 0, q_2)\}_{\sigma \in \Sigma}$.

For the initial configuration checker, we define $\mathcal{A} = \langle \Sigma, \{q_0, q_1\}, q_0, \delta, \{q_1\}\rangle$ such that:

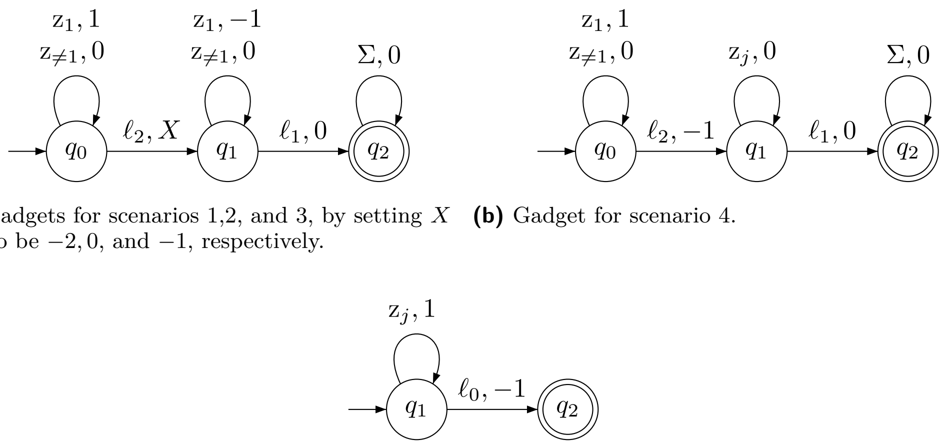
- $\Sigma = \mathsf{Loc} \cup \{a_i\}_{z_i \in \mathsf{Z}}$.
- $\delta = \{(q_0, a_j, 1, q_0)\}_{z_j \in \mathsf{Z}} \cup \{(q_0, \ell_0, -1, q_1)\}$.

Our last gadget captures ill-formed words, regardless of counter values.

Let $\mathcal{M} = \langle \mathsf{Loc}, \mathsf{Z}, \Delta\rangle$ be an LCM. we say that a word w is *well formed* if the following conditions are satisfied:

1. w is of the form $w = a_n^* \cdots a_1^* \ell_{iN} \cdots a_n^* \cdots a_1^* \ell_{i0}$ for $\{\ell_{ij} \in \mathsf{Loc}\}_{0 \le j \le N}$.
2. $\ell_{i0} = \ell_0$.
3. for every $0 \le j \le N-1$, there is at least one transition in $\mathcal{M}$ that leads from ℓ_{ij} to $\ell_{i,j+1}$.

It is easy to see that well formed words are a regular language, and in particular its complement is the desired OCN.



(a) Gadgets for scenarios 1,2, and 3, by setting X to be $-2, 0$, and -1, respectively. **(b)** Gadget for scenario 4.

(c) Gadget for initial configuration (last one in the reverse encoding).

Figure 8 The violation-check gadgets for z_1. By $z_{\neq 1}$ we mean z_j for all $j \neq 1$, and by z_j we mean every counter.

A.3.2 The Main Construction

Let $\mathcal{M} = \langle \mathsf{Loc}, \mathsf{Z}, \Delta \rangle$. We wish to construct an OCN $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}, 0)$ is the set of all words that do not represent legal runs of $\mathcal{M}$.

Intuitively, we construct $\mathcal{A}$ through the following process:

1. Construct a flow violation checker (with regards to $\mathcal{M}$), which will be part of $\mathcal{A}$ as a separate component.
2. for every location $\ell \in \mathsf{Loc}$, add a corresponding state ℓ' in $\mathcal{A}$. all such ℓ''s are initial states in $\mathcal{A}$, and they all have self loops with weight 0 when reading all counter accumulators $\{a_i\}_{z_i \in \mathsf{Z}}$. Intuitively, when $\mathcal{A}$ visits a state ℓ', it means that $\mathcal{A}$ is currently in the process of reading a configuration in which $\mathcal{M}$ is in location ℓ.
3. for every transition $(\ell_1, \mathsf{op}, \ell_2) \in \Delta$, add the transition $(\ell'_2, \ell_2, 0, \ell'_1)$ to $\mathcal{A}$. Intuitively, traveling this transition means that $\mathcal{A}$ has finished reading a configuration of location ℓ_2, and is now starting to read a configuration of location ℓ_1.
4. connect an initial configuration violation checker to ℓ'_0.
5. for every transition $(\ell_1, \mathsf{op}, \ell_2) \in \Delta$, add from ℓ'_2 transitions to all relevant violation checkers for all counters $\{z_i\}_{1 \leq i \leq n}$.

Now let us define the construction formally. Let $V(\ell_i \to \ell_j, z_m)$ be the violation checker that matches the transition $(\ell_i, \mathsf{op}, \ell_j)$ for counter z_m, as detailed in Appendix A.3.1. Let $Q(\ell_i \to \ell_j, z_m)$ be its states, let $F(\ell_i \to \ell_j, z_m)$ be its accepting states, let $\delta(\ell_i \to \ell_j, z_m)$ be its transitions, and $\lambda(\ell_i \to \ell_j, z_m) \subseteq \delta(\ell_i \to \ell_j, z_m)$ be the transitions from its initial state. In that spirit we also define, with regards to the flow control violation checker, and the initial configuration violation checker: $Q(\text{initial})$, $\delta(\text{initial})$, $\lambda(\text{initial})$, $Q(\text{flow})$, $\delta(\text{flow})$, $\lambda(\text{flow})$. Lastly, for convenience' sake alone we define $\mathcal{A}$ as having multiple initial states. this has been done for readability, and can easily be formally circumvented by defining a single initial state α_0, along with an outgoing transition (α_0, σ, z, q) for each $(s_0, \sigma, z, q) \in \delta$.

We now define $\mathcal{A} = \langle \Sigma, Q, S_0, \delta, F \rangle$ such that:

- $\Sigma = \mathsf{Loc} \cup \{a_i\}_{z_i \in \mathsf{Z}}$
- $Q = \{\ell'_i\}_{\ell_i \in \mathsf{Loc}} \cup Q(\text{initial}) \cup Q(\text{flow}) \cup \{Q(\ell_i \to \ell_j, z_m)\}$ for all $\ell_i, \ell_j \in \mathsf{Loc}$ such that there is a transition from ℓ_i to ℓ_j in Δ, and for all $1 \leq i \leq m$.

- $S_0 = \{\ell'_i\}_{\ell_i \in \mathsf{Loc}} \cup \{s_{0,\text{flow}}\}$ such that $s_{0,\text{flow}}$ is the initial state of the flow violation checker. $\delta_1 = \{(\ell'_i, a_j, 0, \ell'_i)\}$ for all $\ell_i \in \mathsf{Loc}$ and for all $1 \leq j \leq n$.
- $\delta_2 = \{(\ell'_i, \ell_i, 0, \ell'_j)\}$ for all $\ell_i, \ell_j \in \mathsf{Loc}$ such that there is a transition from ℓ_j to ℓ_i in Δ.
- $\delta_3 = \{(\ell'_i, \sigma, \nu, q')\}$ for all $\ell_i, \ell_j \in \mathsf{Loc}$ such there is a transition from ℓ_j to ℓ_i in Δ, and $(q, \sigma, \nu, q') \in \lambda(\ell_j \to \ell_i, z_m)$ for some $1 \leq i \leq m$, or otherwise $(q, \sigma, \nu, q') \in \lambda(\text{initial})$.
- $\delta_V = \bigcup_{\text{all violations}} \delta(\text{violation})$.
- $\delta = \delta_1 \cup \delta_2 \cup \delta_3 \cup \delta_V$
- $F = \bigcup_{\text{all violations}} F(\text{violation})$.

We turn to prove the correctness of the construction. Consider a word w that represents a legal run of $\mathcal{M}$. Then, first of all, w is well formed, and therefore not accepted by the flow violation checker. second, there is no transition from one configuration to the next that involves a violation, and therefore w cannot be accepted through any of the violation checkers in $\mathcal{A}$. Since all accepting states of $\mathcal{A}$ are inside violation checkers, $w \notin \mathcal{L}(\mathcal{A}, 0)$.

Conversely, assume a word w does not represent a legal run of $\mathcal{M}$. If w is not well formed, then it is accepted through the flow violation checker. Otherwise - a transition from a state $\ell_i \in \mathsf{Loc}$ to a state $\ell_j \in \mathsf{Loc}$ represents a violation for counter z_m such that $1 \leq m \leq n$. $\mathcal{A}$ then accepts w by branching from ℓ'_j to $V(\ell_i \to \ell_j, z_m)$ at the right moment. It is also possible that the violation occurs in the first configuration (last one to be read), and in this case w will be accepted through the initial configuration violation checker.

A.4 Details for the proof of Lemma 5

The following is a formal construction of DFA $\mathcal{D} = \langle \Sigma, Q', s'_0, \delta', F' \rangle$:

- $Q' = \{\langle \ell, a_1 \dots a_k, b_1 \dots b_k \rangle | \ell \in \mathsf{Loc}, 0 \leq a_i, b_i \leq m \text{ for all } 1 \leq i \leq k\} \cup \{\langle \bot, \bot \dots \bot, b_1 \dots b_k \rangle | 0 \leq b_i \leq m \text{ for all } 1 \leq i \leq k\}$.
- $s'_0 = \langle \bot, \bot \dots \bot, 0 \dots 0 \rangle$.
- $\delta'(\langle \ell, a_1 \dots a_k, b_1 \dots b_k \rangle, \ell') = \langle \ell', b_1 \dots b_k, 0 \dots 0 \rangle$ if the configuration $\langle \ell, a_1 \dots a_k \rangle$ can be obtained from the configuration $\langle \ell', b_1 \dots b_k \rangle$ through a single transition in $\mathcal{M}$.
- $\delta'(\langle \bot, \bot \dots \bot, b_1 \dots b_k \rangle, \ell) = \langle \ell, b_1 \dots b_k, 0 \dots 0 \rangle$ for all $\ell \in \mathsf{Loc}, 0 \leq b_1 \dots b_k \leq m$.
- $\delta'(\langle \ell, a_1 \dots a_k, 0 \dots 0, b_j \dots b_k \rangle, z_j) = \langle \ell, a_1 \dots a_k, 0 \dots 0, b_j + 1 \dots b_k \rangle$ for all $0 \leq j \leq k$, $b_j < m$.
- $\delta'(\langle \ell, a_1 \dots a_k, 0 \dots 0, b_j \dots b_k \rangle, z_{j-x}) = \langle \ell, a_1 \dots a_k, 0 \dots 1, 0 \dots b_j \dots b_k \rangle$ for all $1 \leq j \leq k$, $1 \leq x \leq j$.
- $F = \{\langle \ell_0, 0 \dots 0, 0 \dots 0 \rangle\}$.

Correctness stems directly from the construction.

As for the other direction, assume the reachability set of $\langle \ell_0, (0 \dots 0) \rangle$ under $\mathcal{M}$ is infinite, and assume by way of contradiction that $\mathcal{A}$ has a deterministic equivalent $\mathcal{D}$ with d states. Observe that for every word $u \in \Sigma^*$, the run of $\mathcal{D}$ does not end due to the counter becoming negative. Indeed, we can always concatenate some $\lambda \in \Sigma^*$ such that $u\lambda$ does not correspond to a run, and is hence accepted by $\mathcal{D}$, so the run on u must be able to continue reading λ. We call this property of $\mathcal{D}$ *positivity*.

Since the reachability set of $\langle \ell_0, (0 \dots 0) \rangle$ is infinite, there exists a counter of $\mathcal{M}$, w.l.o.g z_1, that can take unbounded values (in different runs). Let w be a word corresponding to a run of $\mathcal{M}$ that ends with the value of z_1 being N for some $N > d$. We can then write $w = a_k^* \cdots a_1^N \ell a_k^* \cdots a_1^{N'} \ell' \rho$ such that ρ represents the reverse of a legal prefix of a run of $\mathcal{M}$, and N' satisfies $N' \geq N - 1$, since no single transition of $\mathcal{M}$ can increase a counter by more than one (but N' can be arbitrarily large).

Since w corresponds to a legal run of $\mathcal{M}$, $\mathcal{A}$ (and therefore $\mathcal{D}$) does not accept w. By the positiviy of $\mathcal{D}$, its run on w ends in a non-accepting state.

Since $N > d$, $\mathcal{D}$ goes through a cycle β when reading a_1^N. We pump the cycle β to obtain a run of $\mathcal{D}$ on a word $w'' = a_k^* \cdots a_1^{N+t} q a_k^* \cdots a_1^{N'} q' \rho$ for some $t \in \mathbb{N}$ that satisfies $N + t > N' + 1$. Again, by the positivity of $\mathcal{D}$, the run cannot end due to the counter becoming negative, so it ends in the same non accepting state as the run on w. However, w'' does not represent a legal run of M, since $N + t > N' + 1$, therefore $w'' \in \mathcal{L}(\mathcal{A}, 0)$, which contradicts $\mathcal{L}(\mathcal{A}, 0) = \mathcal{L}(\mathcal{D}, 0)$.

A.5 Details for the Proof of Lemma 14

We start by demonstrating our method, followed by the general construction. Consider, for example, $S = \text{Lin}((1,0),(4,8)) \cup \text{Lin}((2,1),(6,12))$. In this case $\gamma = 12$. We split $\text{Lin}((1,0),(4,8))$ to $\text{Lin}((1,0),(12,24)) \cup \text{Lin}((5,8),(12,24)) \cup \text{Lin}((9,16),(12,24))$, the intuition being that instead of a $(4,8)$ period, we have a $(12,24)$ period, and we add different basis vectors to fill the gaps, so the new basis vectors are $(5,8)$ and $(9,16)$, where the next basis vector $(13,24)$ is already captured by $(1,0) + (12,24)$. Similarly, we split $\text{Lin}((2,1),(6,12))$ to $\text{Lin}((2,1),(12,24)) \cup \text{Lin}((8,13),(12,24))$. Overall we get $S = \bigcup_{v \in V} \text{Lin}(v,(12,24))$ for $V = \{(1,0),(5,8),(9,16),(2,1),(8,13)\}$.

Generally, let $\gamma = \text{lcm}(\{p_i\}_{i=1}^M)$. We split each linear component $\text{Lin}((a_i,b_i),\{(p_i,r_i)\})$ to $\frac{\gamma}{p_i}$ parts, by defining the *γ-split* of $\text{Lin}((a_i,b_i),(p_i,r_i))$ (defined only for $p_i|\gamma$) to be $\bigcup_{i=0}^{\frac{\gamma}{p_i}-1} \text{Lin}((a_i,b_i) + i \cdot (p_i,r_i),(\gamma,r_i) \cdot \frac{\gamma}{r_i})$. each such split is semilinear by definition, and it is straightforward to show that $S = \bigcup_{i=1}^{k} \text{l-split}(\text{Lin}((a_i,b_i),(p_i,r_i))$.

A.6 Unambiguous OCNs

We now consider the case where $\mathcal{A}$ is unambiguous. Observe that in order to construct $\theta(n,c)$ above, we explicitly placed the requirement that the counter is minimal. As we now show, if $\mathcal{A}$ is unambiguous, we can modify the formula such that no universal quantification is required.

Recall that in the construction of the formula $\varphi_\rho(n,c)$, we define the subformula $\textsc{sufficient-counter}_\rho(c, e_1, \ldots, e_k)$, stating that the counter c is sufficient for traversing the run $\alpha_0 \beta_1^{e_1} \alpha_1 \beta_2^{e_2} \alpha_2 \cdots \alpha_k$. The structure of $\textsc{sufficient-counter}_\rho(c, e_1, \ldots, e_k)$ can be viewed as a conjunction of inequalities $\bigwedge_j \tau_j \geq 0$ where each τ_j is a linear expression containing c. We observe that c is a minimal counter that satisfies these equations iff one of them is satisfied as an equality.

In addition, for unambiguous OCNs, if $\textsc{sufficient-counter}_\rho(c, e_1, \ldots, e_k)$ is satisfied, then all alternative values $e'_1, \ldots, e'_k$ for which this formula is satisfied represent the same run. Therefore, if an initial counter value c is minimal for words of length n and certain $e_1, \cdots e_k$, then it is minimal for all alternative $e'_1, \ldots, e'_k$. We can then construct the following formula

$$\begin{aligned} \psi_\rho(n,c) :=& \exists e_1 \cdots e_k,\ \textsc{correct-length}_\rho(n, e_1 \cdots e_k) \\ & \wedge \textsc{sufficient-counter}_\rho(c, e_1 \cdots e_k) \\ & \wedge \textsc{minimal-counter}_\rho(c, e_1 \cdots e_k) \end{aligned}$$

where $\textsc{minimal-counter}_\rho(c, e_1 \cdots e_k) := \bigvee_j \tau_j = 0$ where τ_j are the inequalities that appear in $\textsc{sufficient-counter}_\rho(c, e_1 \cdots e_k)$.

By the above, we have that $\psi_\rho(n,c)$ is satisfied iff c is the minimal counter value such that there exists a run of length n that is of the shape ρ starting from counter value c.

Defining $P \subseteq S$ to be the set of linear path schemes from the initial state to an accepting state, as above, we can rewrite θ more compactly, as follows: $\theta(n,c) = \bigvee_{\rho \in P} \varphi_\rho(n,c)$.

As for the bigger picture, we remind the reader that `Uniform-Det` can be decided using $\nu = \exists n_1, n_2, c_1, c_2,\ n_1 < n_2 \wedge c_1 > c_2 \wedge \theta(n_1,c_1) \wedge \theta(n_2,c_2)$. In the unambiguous case, we can rewrite ν as follows:

$$\nu = \bigvee_{\rho_1,\rho_2 \in P} \exists n_1, c_1, n_2, c_2, e_{11}, \cdots e_{1k_1}, e_{21}, \cdots e_{2k_2}, n_1 < n_2 \wedge c_1 > c_2 \wedge$$

$$\textsc{correct-length}_{\rho_1}(n_1, e_{11} \cdots e_{1k_1}) \wedge \textsc{sufficient-counter}_{\rho_1}(c_1, e_{11} \cdots e_{1k_1})$$

$$\wedge \textsc{minimal-counter}_{\rho_1}(c_1, e_{11} \cdots e_{1k_1}) \wedge \textsc{correct-length}_{\rho_2}(n_2, e_{21} \cdots e_{2k_2})$$

$$\wedge \textsc{sufficient-counter}_{\rho_2}(c_2, e_{21} \cdots e_{2k_2}) \wedge \textsc{minimal-counter}_{\rho_2}(c_2, e_{21} \cdots e_{2k_2}).$$

This representation of ν is a disjunction of existential fragments, all of which are polynomial in the size of $\mathcal{A}$.