Language equivalence of deterministic real-time one-counter automata is NL-complete

Stanislav Böhm[1⋆] and Stefan Göller[2]

[1] Technical University of Ostrava, Department of Computer Science, Czech Republic
[2] Universität Bremen, Institut für Informatik, Germany

Abstract. We prove that deciding language equivalence of deterministic real-time one-counter automata is NL-complete, in stark contrast to the inclusion problem which is known to be undecidable. This yields a subclass of deterministic pushdown automata for which the precise complexity of the equivalence problem can be determined. Moreover, we show that deciding regularity is NL-complete as well.

1 Introduction

In formal language theory two of the most fundamental decision problems are to decide whether two languages are equivalent (*language equivalence*) or whether one language is a subset of another (*language inclusion*). It is well-known that already deciding if a context-free language is universal is undecidable.

In recent years, subclasses of context-free languages have been studied for which equivalence or even inclusion becomes decidable.

The most prominent such subclass is the class of deterministic context-free languages (however inclusion remains undecidable). A groundbreaking result by Sénizergues states the decidability of language equivalence of *deterministic pushdown automata (DPDA)* [20], see also [21]. In 2002 Stirling showed that DPDA language equivalence is in fact primitive recursive [23]. Probably due to its intricacy this fundamental problem has not attracted too much research in the past ten years. We emphasize that for DPDA language equivalence there is still a remarkably huge complexity gap ranging from a primitive recursive upper bound to P-hardness (which straightforwardly follows from P-hardness of the emptiness problem). To the best of the authors' knowledge, the same phenomenon holds if the DPDA are restricted to be *real-time* [18], i.e. ε-transitions are not present. However, for finite-turn DPDA a coNP upper bound is known [22]. For *simple DPDA* (which are single state and real-time DPDA) language equivalence is decidable in polynomial time [9], whereas language inclusion is still undecidable [6]. For *deterministic one-counter automata (DOCA)*, which are DPDA over a singleton stack alphabet plus a bottom stack symbol, language equivalence was shown decidable in time $2^{O(\sqrt{n \log n})}$ [24]. By a simple analysis of the proof in [24] a PSPACE upper bound can be derived for this problem.

⋆ S. Böhm has been supported by the Czech Ministry of Education, project No. 1M0567 and GAČR P202/11/0340.

The goal of this paper is to make a step towards understanding (a special case of) of the equivalence problem of DPDA better. We analyze a syntactic restriction of DPDA, namely *deterministic real-time one-counter automata (ROCA)*, which are real-time DOCA. ROCA satisfy the following points: (i) the automaton model is simple, (ii) it is powerful enough to capture relevant non-regular languages such as e.g. the set of well-matched parenthesis or $\{a^n b^n \mid n \geq 0\}$, (iii) its language is not defined modulo some predetermined stack behavior of the automata (i.e. as it is the case for visibly pushdown automata [1] or more general approaches as in [4, 17]), and (iv) tight complexity bounds can be obtained for the equivalence problem. Although points (i) and (ii) have a subjective touch, the authors are not aware of any subclass of the context-free languages that satisfy all of the four mentioned points. We remark that for ROCA language inclusion remains undecidable.

Contributions. The main result of this paper is that language equivalence of ROCA is NL-complete, hence closing the gap from PSPACE [24] (which holds even for DOCA) to NL (hardness for NL is inherited from the emptiness problem for deterministic finite automata). As a second result we prove that deciding *regularity* of ROCA, i.e. deciding if the language of a given ROCA is regular, is NL-complete as well. The previously best known upper bound for this problem (as for DOCA) is a time bound of $2^{O(\sqrt{n \log n})}$ [24] (from which one can also derive a PSPACE upper bound).

Used techniques. For our NL upper bound for language equivalence of ROCA, we prove that if two ROCA are inequivalent, then they can already be distinguished by a word of polynomial length. To show this, we use an established approach that can be summarized as the "*belt technique*" that has already been used in [14, 12, 11, 3] in the context of (bi)simulation equivalence checking of one-counter automata. More specifically, we use an approach from [11, 3] that can be formulated as follows: There is a small set INC of *incompatible configurations* which two configurations necessarily have to have the same shortest distance to provided they are language equivalent — moreover, in case two configurations both have the same finite distance to INC, they must either both have small counter values or they lie in one of polynomially many so called *belts*. To prove the existence of polynomially long distinguishing words, in case two ROCA are not language equivalent, we carefully investigate how paths through such belts can look like.

Related work. Deterministic one-counter automata (DOCA) were introduced by Valiant and Paterson in [24], where the above-mentioned time upper bound for language equivalence was proven. Polynomial time algorithms for language equivalence and inclusion for strict subclasses of ROCA were provided in [7, 8]. In [2, 5] polynomial time learning algorithms were presented for ROCA. Simulation and bisimulation problems on one-counter automata were studied in [3, 12, 11, 14, 15, 13].

Organization. Our paper is organized as follows. Section 2 contains definitions. Our main result is stated in Section 3 and proven in Section 4. Regularity of ROCA is proven NL-complete in Section 5. We conclude in Section 6.

Remark: In [2, 19] it is stated that language equivalence of DOCA can be decided in polynomial time. Unfortunately, the proofs provided in [2, 19] were not exact enough to be verified and raise several questions which are unanswered to date.

2 Definitions

By $\mathbb{Z}$ we denote the integers and by $\mathbb{N} = \{0, 1, \ldots\}$ we denote the naturals. For two integers $i, j \in \mathbb{Z}$ we define the interval $[i, j] = \{i, i+1, \ldots, j\}$ and $[j] = [1, j]$. The *sign* function sgn : $\mathbb{N} \to \{0, 1\}$ is defined as sgn$(n) = 1$ if $n > 0$ and sgn$(n) = 0$ if $n = 0$, for each $n \in \mathbb{N}$. For a word w over a finite alphabet Σ we denote by $|w|$ the *length* of w. By ε we denote the *empty word*. By Σ^* we denote the set of finite words over Σ, by $\Sigma^+ = \Sigma^* \setminus \{\varepsilon\}$ the set of *non-empty words*, and for each $\ell \geq 0$ we define $\Sigma^{\leq \ell} = \{w \in \Sigma^* : |w| \leq \ell\}$.

A *deterministic and complete transition system* is a tuple $\mathcal{T} = (S, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\}, F)$, where S is a set of *states*, Σ is a *finite alphabet*, for each $a \in \Sigma$ we have that $\xrightarrow{a} \subseteq S \times S$ is a set of transitions, where for each $s \in S$ there is precisely one $t \in S$ such that $s \xrightarrow{a} t$, and $F \subseteq S$ is a set of *final states*. We extend $\xrightarrow{w}$ to words $w \in \Sigma^*$ inductively as expected, $\xrightarrow{\varepsilon} = \{(s, s) \mid s \in S\}$ and $\xrightarrow{wa} = \{(s, t) \mid \exists u \in S : s \xrightarrow{w} u \xrightarrow{a} t\}$, where $w \in \Sigma^*$ and $a \in \Sigma$. We also write $\longrightarrow$ for $\bigcup_{a \in \Sigma} \xrightarrow{a}$. For each subset $U \subseteq S$ we write $s \xrightarrow{w} U$ (resp. $s \to^* U$) if $s \xrightarrow{w} u$ (resp. $s \to^* u$) for some $u \in U$. For each state $s \in S$ we define the *language up to length* ℓ of s as $L_\ell(s) = \{w \in \Sigma^{\leq \ell} \mid s \xrightarrow{w} t, t \in F\}$ and the *language* of s as $L(s) = \bigcup_{\ell \in \mathbb{N}} L_\ell(s)$. We write $s \equiv_\ell t$ whenever $L_\ell(s) = L_\ell(t)$ and $s \equiv t$ whenever $L(s) = L(t)$. So note that we have $s \equiv_0 t$ if and only if either $s, t \in F$ or $s, t \notin F$. We call a word $w \in \Sigma^*$ a *(distinguishing) witness* for states s and t if $s \xrightarrow{w} s'$ and $t \xrightarrow{w} t'$ with $s' \not\equiv_0 t'$. A *mininal witness* is a witness of minimal length among all witnesses.

A *deterministic real-time one-counter automaton (ROCA)* is a tuple $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$, where Q is a finite set of *control states*, Σ is a *finite alphabet*, $\delta : Q \times \Sigma \times \{0, 1\} \to Q \times \{-1, 0, 1\}$ is a *transition function* that satisfies $\delta(Q \times \Sigma \times \{0\}) \subseteq Q \times \{0, 1\}$ (i.e. no decrement is allowed when the counter is zero), $q_0 \in Q$ is an *initial control state*, $F \subseteq Q$ is a set of *final control states*. If the initial state q_0 of $\mathcal{A}$ is not relevant, we just write $\mathcal{A} = (Q, \Sigma, \delta, F)$. A *configuration* of $\mathcal{A}$ is a pair $(q, n) \in Q \times \mathbb{N}$ that we also abbreviate by $q(n)$. Each ROCA $\mathcal{A}$ defines a deterministic transition system $\mathcal{T}(\mathcal{A}) = (Q \times \mathbb{N}, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\}, F \times \mathbb{N})$, where $q(n) \xrightarrow{a} q'(n + j)$ whenever $\delta(q, a, \text{sgn}(n)) = (q', j)$. We define $L(\mathcal{A}) = L(q_0(0))$. In this paper, we are mainly interested in the following decision problem.

LANGUAGE EQUIVALENCE OF ROCA	
INPUT:	Two ROCA $\mathcal{A}$ and $\mathcal{A}'$.
QUESTION:	$L(\mathcal{A}) = L(\mathcal{A}')$?

Interestingly, inclusion between ROCA is undecidable. Valiant and Paterson were already aware of this without providing a proof [24].

Proposition 1 (Simple consequence of [16]). *Given two ROCA $\mathcal{A}$ and $\mathcal{A}'$, deciding whether $L(\mathcal{A}) \subseteq L(\mathcal{A}')$ holds, is undecidable.*

It is worth noting that it is also a consequence of [16] that language equivalence of *nondeterministic* (real-time) one-counter automata is undecidable.

3 NL-completeness of equivalence of ROCA

Instead of considering language equivalence of *two* ROCA, we can simply take their disjoint union and ask whether two configurations of it are language equivalent. Therefore let us fix for the rest of this and the next section some ROCA $\mathcal{A} = (Q, \Sigma, \delta, F)$ and two control states $p_{\text{init}}, q_{\text{init}} \in Q$ for which we wish to decide if $p_{\text{init}}(0) \equiv q_{\text{init}}(0)$.

Lemma 2. *We have $p_{init}(0) \equiv q_{init}(0)$ if and only if $p_{init}(0) \equiv_\ell q_{init}(0)$, where ℓ is polynomially bounded in $|Q|$.*

In Section 4 we prove Lemma 2. We now use it to derive NL-completeness of language equivalence of ROCA.

Theorem 3. *Language equivalence of ROCA is* NL*-complete.*

Proof. The NL lower bound already holds for the emptiness problem for deterministic finite automata. For the upper bound, we apply Lemma 2 and store in logarithmic space a pair of configurations (the two counter values are stored in binary) for which we check inequivalence in an on-the-fly fashion: We repeatedly guess a symbol $a \in \Sigma$ and update the pair of configurations by applying the transition function on both of them synchronously. If the current pair is not $\equiv_0$-related, then the initial pair of configurations is inequivalent and if such a guessing is not possible then the initial pair of configurations has to be equivalent by Lemma 2. Hence inequivalence is NL. Since NL is closed under complement [10] the theorem follows. □

4 Polynomially long distinguishing witnesses suffice

Before we prove Lemma 2, we introduce some notions that allow us to get a better visual intuition of what minimal distinguishing witnesses can look like.

For the rest of the paper, it will sometimes be more convenient to identify each pair of configurations $\langle p(m), q(n)\rangle$ by the point $\langle m, n, (p, q)\rangle$ in the 3D space $\mathbb{N} \times \mathbb{N} \times (Q \times Q)$, where the first two dimensions represent the two counter values and the third dimension $Q \times Q$ corresponds to the pair of control states. We will partition the 3D space into an *initial space*, *belt space* and *background space* as exemplarily depicted in Figure 1. The size of the initial space and the thickness and the number of belts will be polynomially bounded in $|Q|$.

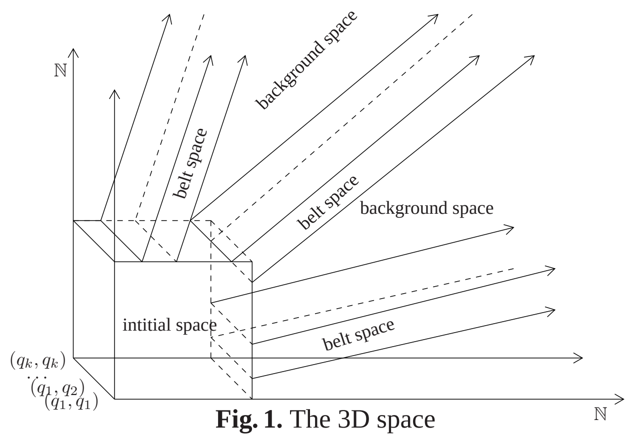

Fig. 1. The 3D space

We remark that the *belt technique* in the context of one-counter automata has already successfully been used in [14, 12, 11, 3]. Moreover, we remark that our concrete way of partitioning the 3D space was already present in [11, 3].

To each pair of configurations $\langle p(m), q(n)\rangle$ and each word $w = a_1 \cdots a_\ell \in \Sigma^*$ we can assign a unique sequence $\mathsf{Comp}(p(m), q(n), w) = \pi_0 \cdots \pi_\ell$ of 3D points that we call the *computation*, formally $\pi_0 = \langle m, n, (p, q)\rangle$ and if $\pi_i = \langle m_i, n_i, (p_i, q_i)\rangle$ for each $i \in [0, \ell]$, then in the transition system $\mathcal{T}(\mathcal{A})$ we have $p_{i-1}(m_{i-1}) \xrightarrow{a_i} p_i(m_i)$ and $q_{i-1}(n_{i-1}) \xrightarrow{a_i} q_i(n_i)$ for each $i \in [1, \ell]$. Hence $\mathsf{Comp}(p(m), q(n), w)$ can be seen as a path through the 3D space. The *counter effect of* π is defined as $(m_\ell - m_0, n_\ell - n_0) \in \mathbb{Z} \times \mathbb{Z}$. A *factor* of π is a sequence $\pi_i \pi_{i+1} \cdots \pi_j$ for some $0 \leq i \leq j \leq \ell$.

The overall proof strategy for Lemma 2 will be to show that for every *minimal (distinguishing) witness* w for $p_{\text{init}}(0)$ and $q_{\text{init}}(0)$ the path $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w)$ has the following property: It can stay in the initial space, it can be inside each belt space but only polynomially many steps consecutively, but once it is in the background space it terminates surely after polynomially many steps. This implies that the overall length of $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w)$ is polynomially bounded.

In Section 4.1 we (re-)investigate an important set INC of configurations and discuss in Section 4.2 that two configurations that have the same finite shortest distance to INC necessarily must lie in the initial space or in the belt space. In Section 4.3 we finally prove that any minimal witness has the above mentioned behavior in the 3D space, thus implying Lemma 2. For the rest of this section, let $k = |Q|$ denote the number of control states of the ROCA $\mathcal{A}$ that we fixed for this section.

4.1 The underlying DFA and incompatible configurations

We start with the observation what happens in the transition system $\mathcal{T}(\mathcal{A})$ if the counter value is very big: It behaves for a long time just like a deterministic finite automaton (DFA). We will call this DFA the *underlying DFA* of $\mathcal{A}$. We can partition the set of configurations of $\mathcal{A}$ into two sets: Those configurations that are not equivalent to all states of the underlying finite system up to words of length at most k and the rest. By analyzing the reachability to the former set, we establish the partition of the 3D space in the next section.

We remark that below the notion *underlying DFA*, the set INC with its useful property stated in Lemma 4 and the distance function dist were already present in [11, 3].

The *underlying DFA* of $\mathcal{A}$ is $\mathcal{F} = (Q, \Sigma, \{\xrightarrow{a} \mid a \in \Sigma\}, F)$, where $q \xrightarrow{a} q'$ if and only if $\delta(q, a, 1) \in \{q'\} \times \{-1, 0, 1\}$.

Observe that on the one hand we write Q to denote the set of control states of $\mathcal{A}$ and on the other hand we denote by Q the states of the DFA $\mathcal{F}$. Recall that $k = |Q|$. For each $q \in Q$ we write $L_k(q)$ to denote the language of $\mathcal{F}$ up to length at most k in case q is the initial state. Also note that in $\mathcal{F}$ we have that $\equiv_k$ coincides with $\equiv_{k-1}$.

Define the set INC as those configurations of $\mathcal{T}(\mathcal{A})$ that are incompatible (not k-equivalent) to *all* states in $\mathcal{F}$, formally

$$\mathsf{INC} \quad = \quad \{p(m) \in Q \times \mathbb{N} \mid \forall q \in Q : L_k(p(m)) \neq L_k(q)\}.$$

Remark 4. If $p(m) \in \mathsf{INC}$, then $m < k$.

The main motivation to study the set INC is due to the following lemma.

Lemma 5. *Assume* $p(m) \not\longrightarrow^* \mathsf{INC}$, *and* $q(n) \not\longrightarrow^* \mathsf{INC}$. *Then* $p(m) \equiv q(n)$ *if and only if* $p(m) \equiv_k q(n)$.

Proof. The "only if"-direction is trivial. For the "if"-direction, assume by contradiction $p(m) \not\equiv q(n)$ but $p(m) \equiv_k q(n)$. Let ℓ be minimal such that $p(m) \not\equiv_\ell q(n)$. Note that $\ell > k$. Thus, there is some word $u \in \Sigma^{\ell-k}$ with $p(m) \xrightarrow{u} p'(m')$ and $q(n) \xrightarrow{u} q'(n')$ where $p'(m') \not\equiv_k q'(n')$ but $p'(m') \equiv_{k-1} q'(n')$. Since by assumption $p'(m'), q'(n') \notin \mathsf{INC}$, there are $s, t \in Q$ such that $s \equiv_k p'(m')$ and $t \equiv_k q'(n')$ and hence $s \equiv_{k-1} t$. Recall that in $\mathcal{F}$ we have that $\equiv_{k-1}$ coincides with $\equiv_k$ and hence $s \equiv_k t$. Altogether we obtain $p'(m') \equiv_k s \equiv_k t \equiv_k q'(n')$, contradicting $p'(m') \not\equiv_k q'(n')$. □

Next, let us define the distance to the set INC for each configuration $p(m)$. We define

$$\mathsf{dist}(p(m)) \quad = \quad \min\left\{|w| : p(m) \xrightarrow{w} \mathsf{INC}\right\}.$$

By convention we put $\min \emptyset = \omega$. Note that $p(m) \equiv q(n)$ implies $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n))$.

4.2 Partitioning the 3D space into initial space, belt space and background space

Let us formally define belts, see also [14, 12, 11, 3]. Let $\alpha, \beta \geq 1$ be relatively prime. The *belt of thickness* d *and slope* $\frac{\alpha}{\beta}$ consists of those pairs $(m, n) \in \mathbb{N} \times \mathbb{N}$ that satisfy $|\alpha \cdot m - \beta \cdot n| \leq d$. An example of a belt is depicted in Figure 2.

Similarly as in [24] we say that two integers m and n are (γ, d)-*rationally related* if there are $\alpha, \beta \in [1, \gamma]$ that are relatively prime such that (m, n) is inside the belt of thickness d and of slope $\frac{\alpha}{\beta}$.

We call $w = a_1 \cdots a_n \in A^+$ ($n \geq 1$) a *simple cycle* from $p(m)$ if the corresponding unique computation $p_0(m_0) \xrightarrow{a_1} p_1(m_1) \cdots \xrightarrow{a_n} p_n(m_n)$ (i.e. $p_0(m_0) = p(m)$) satisfies $p_0 = p_n$ and $p_i \neq p_j$ for all $i, j \in [1, n]$ with $i \neq j$. In case $n_0 > n_m$ we call $n_0 - n_m$ the *counter loss* of w from $p(m)$.

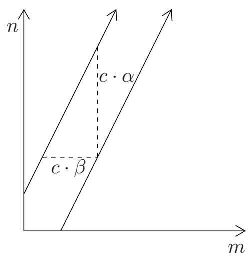

Fig. 2. A belt

The next lemma from [3] states that minimal words from configurations to INC can be chosen in a certain normal form: One first executes a polynomially long prefix, then repeatedly some most effective simple cycle (i.e. a simple cycle where the quotient of counter loss and length is maximal), and finally some polynomially long suffix.

Lemma 6 (Lemma 10 in [3]). *There is some polynomial* poly_0 *such that if* $p(m) \longrightarrow^* \mathsf{INC}$ *then already for some word* $u = u_1(u_2)^r u_3$ *(with* $r \geq 0$*) we have* $p(m) \xrightarrow{u} \mathsf{INC}$, *where (i)* $|u| = \mathsf{dist}(p(m))$, *(ii)* $|u_1 u_3| \leq \mathsf{poly}_0(k)$, *and (iii)* $|u_2| \leq k$, *and (iv) either* $u_2 = \varepsilon$ *or* u_2 *is a simple cycle of counter loss from* $[1, k]$.

The following lemma from [3] allows us to partition the 3D space.

Lemma 7 (Points 3. and 4. of Lemma 11 in [3]). *There are polynomials* poly_1 *and* poly_2 *s.t. if* $\max\{m,n\} > \mathsf{poly}_2(k)$ *and* $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) < \omega$*, then* (m,n)

(1) lies in a unique belt of thickness $\mathsf{poly}_1(k)$ *and slope* $\frac{\alpha}{\beta}$*, where* $\alpha, \beta \in [1, k^2]$ *and*
(2) is not neighbor to any point (m', n') *inside a different belt of thickness* $\mathsf{poly}_1(k)$ *and slope* $\frac{\alpha'}{\beta'}$ *with* $\alpha', \beta' \in [1, k^2]$*, i.e.* $\min\{|m - m'|, |n - n'|\} \geq 2$.

We now partition $\mathbb{N} \times \mathbb{N} \times (Q \times Q)$ into the following three subspaces, cf. Figure 1:

- *initial space*: All points $\langle m, n, (p,q)\rangle$ such that $m, n \leq \mathsf{poly}_2(k)$.
- *belt space*: All points $\langle m, n, (p,q)\rangle$ outside the initial space such that m and n are $(k^2, \mathsf{poly}_1(k))$-rationally related: By Lemma 7 the belt in which (m,n) lies is uniquely determined.
- *background space:* All remaining points.

4.3 Bounding the minimal witness

In this section we demonstrate the core of the proof of Lemma 2: any minimal witness w for $\langle p_{\text{init}}(0), q_{\text{init}}(0)\rangle$ is polynomially bounded in k. For the rest of this section we will assume that $p_{\text{init}}(0) \not\equiv q_{\text{init}}(0)$ and that w is a minimal witness for them.

Recall that $k = |Q|$. Our first lemma tells us once the minimal witness enters the background space at some point $\langle m, n, (p,q)\rangle$ then its remaining suffix is bounded by $k \cdot (\max\{m,n\} + 1) + \mathsf{poly}_0(k)$.

Lemma 8. *For each point* $\langle m, n, (p,q)\rangle$ *in the background space we have* $p(m) \equiv q(n)$ *if and only if* $p(m) \equiv_\ell q(n)$*, where* $\ell \leq k \cdot (\max\{m,n\} + 1) + \mathsf{poly}_0(k)$.

Proof. The "only if"-direction is trivial. For the "if"-direction assume $p(m) \not\equiv q(n)$. Since $\langle m, n, (p,q)\rangle$ is in the background space we cannot have $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) < \omega$ by Point (1) of Lemma 7. In case $\mathsf{dist}(p(n)) = \mathsf{dist}(q(n)) = \omega$, then already for $\ell = k$ we have $p(m) \not\equiv_\ell q(n)$ by Lemma 5. So it remains to consider the case $\mathsf{dist}(p(m)) < \mathsf{dist}(q(n))$ without loss of generality, in particular $\mathsf{dist}(p(m)) < \omega$. Let u be a minimal word such that $p(m) \xrightarrow{u} p'(m')$ for some $p'(m') \in \mathsf{INC}$, note that if $q(n) \xrightarrow{u} q'(n')$, then $p'(m') \not\equiv_k q'(n')$. By applying Lemma 6, we can choose $u = u_1(u_2)^r u_3$ for some $r \geq 0$ such that (i) $|u| = \mathsf{dist}(p(m))$, (ii) $|u_1 u_3| \leq \mathsf{poly}_0(k)$, (iii) $|u_2| \leq k$ and (iv) either $u_2 = \varepsilon$ or u_2 is a simple cycle of counter loss from $[1,k]$. This implies that already for $\ell = |u| + k \leq k \cdot m + \mathsf{poly}_0(k) + k$ we have $p(m) \not\equiv_\ell q(n)$. □

With this lemma one now observes that in case $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w)$ enters the background space after polynomially many steps, then the whole computation is polynomially bounded (the two counters are initialized with zero and by a polynomially bounded computation we can only obtain polynomially large counter values).

Thus, it suffices to focus on the longest prefix w_1 of w such that $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$ enters the background space for at most one point (i.e. if at all, then the last one). Thus, $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$ entirely stays inside the initial space or the belt space (except for the last point possibly). For the rest of this section will show that the length of w_1 is polynomially bounded in k.

First observe that if $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$ does not leave the initial space, then $|w_1|$ is trivially polynomially bounded since the size of the initial space is polynomially bounded by definition. So for the rest of this section assume that $\mathsf{Comp}(p(0), q(0), w_1)$ enters at least one belt.

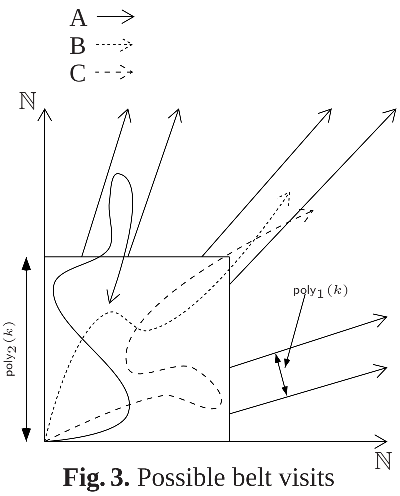

Fig. 3. Possible belt visits

In the following, whenever we talk about a belt we mean its points *outside* the 2D projection of the initial space. Recall that we made the initial space sufficiently large such that there are no intersections between belts and one cannot switch from one belt to another in one step (recall Point (2) of Lemma 7). Let us fix a computation π. A *belt visit* (with respect to some belt B) is a maximal factor of π whose points are all entirely in B. It is clear that $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$ can contain at most polynomially many belt visits. The following cases for belt visits can now be distinguished (a 2D projection of these cases is depicted in Figure 3):

- **Case A:** The initial space is visited immediately after the belt visit.
- **Case B:** The belt visit ends in the belt.
- **Case C:** The background space is visited immediately after the belt visit.

The goal of this section is to prove the following lemma.

Lemma 9. *Every belt visit of* $\mathsf{Comp}(p_{\mathit{init}}(0), q_{\mathit{init}}(0), w_1)$ *is polynomially bounded in* k.

First, we need some more notation. Let $\alpha, \beta \in [1, k^2]$ be relatively prime. We assume $\alpha \geq \beta$, i.e. $\frac{\alpha}{\beta} \geq 1$. The case when $\alpha < \beta$ can be proven analogously.

Points $\langle p(m), q(n)\rangle$ and $\langle p'(m'), q'(n')\rangle$ are $\frac{\alpha}{\beta}$*-related* if $p = p'$, $q = q'$, and $\alpha \cdot m - \beta \cdot n = \alpha \cdot m' - \beta \cdot n'$. Roughly speaking, they are $\frac{\alpha}{\beta}$-related if their control states coincide and they lie on a line with slope $\frac{\alpha}{\beta}$. An $\frac{\alpha}{\beta}$*-repetition* is a computation $\pi_0\pi_1 \cdots \pi_\ell$ such that π_0 and π_ℓ are $\frac{\alpha}{\beta}$-related. Figure 4 shows an example of an $\frac{\alpha}{\beta}$-repetition that lies inside some belt (these are the $\frac{\alpha}{\beta}$-repetitions we will be interested in).

Before we handle the cases **A**, **B**, and **C**, let us fix a belt B with slope $\frac{\alpha}{\beta}$. We will make use of the following claim.

Claim*: There is a polynomial poly_3 such that for each sequence of points $\langle p_0(m_0), q_0(n_0)\rangle \cdots \langle p_h(m_h), q_h(n_h)\rangle$ in B with $h = \mathsf{poly}_3(k)$ and $m_i = m_{i-1} + 1$ for each $i \in [h]$, there are two indices $0 \leq i < i' \leq h$ such that $\langle p_i(m_i), q_i(n_i)\rangle$ and $\langle p_{i'}(m_{i'}), q_{i'}(n_{i'})\rangle$ are $\frac{\alpha}{\beta}$-related.

Proof. Define $d_j = \alpha \cdot m_j - \beta \cdot n_j$ for each $j \in [0, h]$. Since the thickness of B is $\mathsf{poly}_1(k)$, there are at most polynomially many different values for d_j. Hence (for sufficiently large h) by the pigeonhole principle we can find two points $\langle p_i(m_i), q_i(m_i)\rangle$ and $\langle p_{i'}(m_{i'}), q_{i'}(n_{i'})\rangle$ such that $p_i = p_{i'}$, $q_i = q_{i'}$, and $d_i = d_{i'}$. □

Let us now analyze the possible belt visit cases **A**, **B**, and **C**. Note that cases **B** and **C** can occur only in the last belt visit of $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$. For the rest of this section let us fix some B-belt visit $\pi = \pi_0\pi_1 \cdots \pi_z$ of $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$, where $\pi_i = \langle p_i(m_i), q_i(n_i)\rangle$ for each $i \in [0, z]$.

Case A: The intuition behind this case is the following: Consider a long belt visit returning to the initial space. Then we can find two $\frac{\alpha}{\beta}$-repetitions that are factors of π, one going up and one going down with inverse counter effects. We can cut them out and obtain a shorter computation.

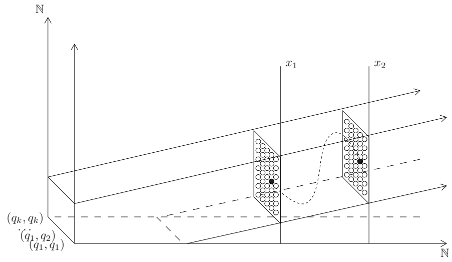

Fig. 4. $\frac{\alpha}{\beta}$-repetition inside a belt

Let us assume that the length of π is sufficiently large such that there is some point π_h on π for the following arguments to work. Define for each suitable $m \in \mathbb{N}$

$$\mathcal{L}(m) = \max\{i \mid m = m_i, i \in [0, h]\} \quad \text{and} \quad \mathcal{R}(m) = \min\{i \mid m = m_i, i \in [h, z]\}.$$

Recall that $\mathsf{poly}_2(k)$ was the height and width of the initial space. By a similar pigeonhole argument as the proof of Claim* there are H and J (since m_h is sufficiently large) such that (i) $\mathsf{poly}_2(k) < H < J < m_h$, (ii) Points π_c and $\pi_{c'}$ are $\frac{\alpha}{\beta}$-related where $c = \mathcal{L}(H)$ and $c' = \mathcal{L}(J)$ and (iii) Points π_d and $\pi_{d'}$ are $\frac{\alpha}{\beta}$-related where $d = \mathcal{R}(H)$ and $d' = \mathcal{R}(J)$. Note that the pair of counter effects from π_c to $\pi_{c'}$ and from $\pi_{d'}$ to π_d add up to $(0, 0)$ componentwise. One can now split up the computation π into $\pi_0 \xrightarrow{\gamma_1} \pi_c \xrightarrow{\gamma_2} \pi_{c'} \xrightarrow{\gamma_3} \pi_{d'} \xrightarrow{\gamma_4} \pi_d \xrightarrow{\gamma_5} \pi_z$. Note that by construction we have $m_i \geq J$ for each $i \in [c', d']$. Since $\alpha \geq \beta$ we can safely cut out the computations $\pi_c \xrightarrow{\gamma_2} \pi_{c'}$ and $\pi_{d'} \xrightarrow{\gamma_4} \pi_d$ and obtain the computation $\pi_0 \xrightarrow{\gamma_1} \pi_c \xrightarrow{\gamma_3} \pi_d \xrightarrow{\gamma_5} \pi_z$. In $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$ we can replace π by this computation and can hence obtain a shorter witness. However, this contradicts minimality of w.

Case B: Let us assume that the belt visit ends in the belt. Since we are considering a computation of a witness we have $p_z(m_z) \not\equiv_0 q_z(n_z)$ for some $m_z, n_z \geq 1$. Thus, $p_z(m) \not\equiv_0 q_z(n)$ for each $m, n \geq 1$. Let us assume π stays in the belt sufficiently long for the following argument to work. By the pigeonhole principle there are i and j with $0 \leq i < j \leq z$ and $j - i \leq k^2$ such that $p_i = p_j$ and $q_i = q_j$. We can assume that m_i, n_i, m_j, and n_j are sufficiently large that we can cut out the computation between π_i and π_j without reaching zero in the rest of the computation. We obtain a shorter computation ending in a point with pair of control states (p_z, q_z), hence contradicting minimality of w.

Case C: Let us assume that the first point after executing π lies in the background space, say in some point $\langle \widehat{p}(\widehat{m}), \widehat{q}(\widehat{n})\rangle$. In other words $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$ ends in $\langle \widehat{p}(\widehat{m}), \widehat{q}(\widehat{n})\rangle$ and π is the last belt visit of $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w_1)$.

First let us consider the case when there is a factor of π that goes "leftward" (and hence necessarily "downward") in the belt for too long. Formally we mean that there is some sufficiently large polynomial poly_4 such that π contains a factor whose counter effect (d_1, d_2) satisfies $d_1 \leq -\mathsf{poly}_4(k)$ or $d_2 \leq -\mathsf{poly}_4(k)$ and the following argument can be realized: There is some point π_h whose counter values are both sufficiently large to which we can apply the same arguments as in Case A and thus obtain a shorter computation, contradicting minimality of w.

Thus, we can assume that for every factor of π with the counter effect (d_1, d_2) we have $d_1, d_2 > -\mathsf{poly}_4(k)$. One can now prove the existence of some polynomial $\mathsf{poly}_5(k)$ for the following arguments to work. In case $\widehat{m} \leq \mathsf{poly}_5(k)$, then $\widehat{n}$ is polynomially bounded and hence π is polynomially bounded.

In case $\widehat{m} > \mathsf{poly}_5(k)$, we do not directly contradict minimality of w but we show the existence of some polynomially bounded computation π' that distinguishes $p_{\text{init}}(0)$ and $q_{\text{init}}(0)$. We distinguish the following subcases: **C1:** $\mathsf{dist}(\widehat{p}(\widehat{m})) < \omega$, **C2:** $\mathsf{dist}(\widehat{q}(\widehat{n})) < \omega$ and **C3:** $\mathsf{dist}(\widehat{p}(\widehat{m})) = \mathsf{dist}(\widehat{q}(\widehat{n})) = \omega$.

C1: We note that from $\langle \widehat{p}(\widehat{m}), \widehat{q}(\widehat{n}) \rangle$ we do not care how π exactly looks like. However, we will prove that one can obtain such a polynomially bounded π' by repeatedly cutting out (polynomially long) $\frac{\alpha}{\beta}$-repetitions from π with the invariant that after each cutting-out the resulting computation can be extended in one step to a background point whose first configuration *still* has finite distance to INC.[1]
By assumption $\mathsf{dist}(\widehat{p}(\widehat{m})) < \omega$, so let u be a minimal word such that $\widehat{p}(\widehat{m}) \xrightarrow{u} \mathsf{INC}$. By Lemma 6 and since $\widehat{m}$ is sufficiently large, we can choose u as $u = u_1(u_2)^r u_3$ for some $r \geq 0$, where $|u_1 u_3| \leq \mathsf{poly}_0(k)$, $|u_2| \leq k$, and u_2 is a simple cycle of counter loss $d \in [1, k]$. This implies $\widehat{p}(\widehat{m} - jd) \xrightarrow{u_1(u_2)^{r-j}u_3} \mathsf{INC}$ for each $j \in [r]$.

Define $\lambda(m) = \max\{i \mid m_i = m, i \in [0, z]\}$ for each $m \in [\mathsf{poly}_2(k) + 1, \widehat{m} - 1]$. We note that $\lambda(m + 1) - \lambda(m)$ is polynomially bounded for each $m, m + 1 \in [\mathsf{poly}_2(k) + 1, \widehat{m} - 1]$ since the negative counter effect of each factor of π is polynomially bounded by assumption. Since $\widehat{m}$ assumed to be sufficiently large we can apply Claim* on polynomially many disjoint factors (each of length $\mathsf{poly}_3(k)$) of $\varphi = \pi_{\lambda(\mathsf{poly}_2(k)+1)} \cdots \pi_{\lambda(\widehat{m}-1)}$ and find an $\frac{\alpha}{\beta}$-repetition on each such factor. Each of these disjoint factors of length $\mathsf{poly}_3(k)$ of φ corresponds to a factor of π that also has only polynomial length, and so do the $\frac{\alpha}{\beta}$-repetitions of these factors. Among these $\frac{\alpha}{\beta}$-repetitions (interpreted as factors of π) we can pick out d all having the same counter effect, say (f, g); in particular $\frac{f}{g} = \frac{\alpha}{\beta}$. When cutting out precisely these d factors from $\mathsf{Comp}(p_{\text{init}}(0), q_{\text{init}}(0), w)$ it enters the background space at point $\langle \widehat{p}(\widehat{m} - df), \widehat{q}(\widehat{n} - dg) \rangle$ for the first time. By $\widehat{p}(\widehat{m} - jd) \xrightarrow{u_1(u_2)^{r-j}u_3} \mathsf{INC}$ for each $j \in [r]$ we have that $\widehat{p}(\widehat{m} - df)$ can reach INC. We can apply this cutting-out process repeatedly until the first point that enters the background space, say $\langle \widehat{p}(\widehat{m} - \Delta), \widehat{q}(\widehat{n} - \Delta') \rangle$, satisfies $\widehat{m} - \Delta \leq \mathsf{poly}_5(k)$.

C2: This case is symmetric to case C1.

[1] We note that we have to require that after the cutting-out the first configuration of the earliest point that is in the background space *must* still have finite distance to INC, for otherwise both configurations could have infinite distance to INC and could be language equivalent.

C3: Since $\mathsf{dist}(\widehat{p}(\widehat{m})) = \mathsf{dist}(\widehat{q}(\widehat{n})) = \omega$ and $\widehat{p}(\widehat{m}) \not\equiv \widehat{q}(\widehat{n})$ we know from Lemma 5 that already some $u \in \Sigma^{\leq k}$ distinguishes $\widehat{p}(\widehat{m})$ and $\widehat{q}(\widehat{n})$. So as in Case B, if π is sufficiently long inside the belt, we can cut out a factor of repeated control state pairs and obtain a shorter witness for $p_{\text{init}}(0)$ and $q_{\text{init}}(0)$, thus contradicting minimality of w.

5 Regularity is NL-complete

Theorem 10. *Regularity of ROCA, i.e. given a ROCA $\mathcal{A}$ deciding if $L(\mathcal{A})$ is regular, is* NL-*complete.*

Proof. For the *upper bound*, let us fix a ROCA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ with $k = |Q|$. Recall the definition of the set INC and for each configuration $p(m)$ its shortest distance $\mathsf{dist}(p(m))$ to INC. We make use of the following characterizations, in analogy to [3]. The following statements are equivalent: (1) $L(q_0(0))$ is *not* regular, (2) for all $d \in \mathbb{N}$ there is some configuration $q(n)$ with $q_0(0) \longrightarrow^* q(n) \longrightarrow^* \mathsf{INC}$ and $d \leq \mathsf{dist}(q(n)) < \omega$, (3) there exists some $q \in Q$ such that $q_0(0) \longrightarrow^* q(2k) \longrightarrow^* \mathsf{INC}$.

For an NL upper bound note that, given a configuration $q(n)$ where n is in unary, deciding if $q(n) \in \mathsf{INC}$ can be done in NL, since $q(n) \in \mathsf{INC}$ if and only if for all $r \in Q$ there is some $w_r \in \Sigma^{\leq k}$ that distinguishes $q(n)$ and the state r of $\mathcal{A}$'s underlying DFA. Second, deciding condition (3) is in NL as well, since the length of such a witnessing path is polynomially bounded. Hence deciding regularity of $\mathcal{A}$ is in NL.

For the *lower bound*, we give a logspace reduction from the emptiness problem for DFA. One can compute in logspace from a given DFA $\mathcal{F}$ a ROCA $\mathcal{A}$ such that $L(\mathcal{A}) = \{a^n \$ w \$ b^n \mid w \in L(\mathcal{F})\}$. Hence $L(\mathcal{A})$ is regular (in particular empty) if and only if $L(\mathcal{F}) = \emptyset$. ◻

6 Conclusion

In this paper we have shown that language equivalence and regularity of ROCA is NL-complete. Using the idea of considering the reachability status of configurations to INC, we can extend our result to prove that it is NL-complete to decide language equivalence of a ROCA and a simple DOCA or to decide regularity of a simple DOCA. A *simple DOCA* is a ROCA that allows spontaneous counter resets (ε-moves) from $p(m)$ to $q(0)$ for some control state q but necessarily *for all* $m \geq 1$: In such configurations $p(m)$ with $m \geq 1$ one can only reset the counter and not read any symbols. We note that simple DOCA and DOCA are equi-expressive but DOCA are exponentially more succinct. The precise complexity of equivalence of DOCA is left for future work.

Acknowledgments: We thank Géraud Sénizergues and Etsuji Tomita for discussions.

References

1. R. Alur and P. Madhusudan. Visibly pushdown languages. In *Proc. of STOC*, pages 202–211. ACM, 2004.
2. P. Berman and R. Roos. Learning One-Counter Languages in Polynomial Time (Extended Abstract). In *Proc. of FOCS*, pages 61–67. IEEE, 1987.

3. S. Böhm, S. Göller, and P. Jančar. Bisimilarity of one-counter processes is PSPACE-complete. In *Proc. of CONCUR*, volume 6269 of *Lecture Notes in Computer Science*, pages 177–191. Springer, 2010.
4. D. Caucal. Synchronization of Pushdown Automata. In *Proc. of DLT*, volume 4036 of *Lecture Notes in Computer Science*, pages 120–132. Springer, 2006.
5. A. F. Fahmy and R. S. Roos. Efficient Learning of Real Time One-Counter Automata. In *Proc. of ALT*, volume 997 of *Lecture Notes in Computer Science*, pages 25–40. Springer, 1995.
6. E. P. Friedman. The inclusion problem for simple languages. *Theor. Comput. Sci.*, 1(4):297–316, 1976.
7. K. Higuchi, M. Wakatsuki, and E. Tomita. A polynomial-time algorithm for checking the inclusion for real-time deterministic restricted one-counter automata which accept by final state. *IEICE Trans. Information and Systems*, E78-D:939–950, 1995.
8. K. Higuchi, M. Wakatsuki, and E. Tomita. A polynomial-time algorithm for checking the inclusion for real-time deterministic restricted one-counter automata which accept by accept mode. *IEICE Trans. Information and Systems*, E81-D:1–11, 1998.
9. Y. Hirshfeld, M. Jerrum, and F. Moller. A Polynomial Algorithm for Deciding Bisimilarity of Normed Context-Free Processes. *Theor. Comput. Sci.*, 158(1&2):143–159, 1996.
10. N. Immerman. Nondeterministic Space is Closed Under Complementation. *SIAM J. Comput.*, 17(5):935–938, 1988.
11. P. Jančar. Decidability of bisimilarity for one-counter processes. *Information Computation*, 158(1):1–17, 2000.
12. P. Jančar, A. Kučera, and F. Moller. Simulation and bisimulation over one-counter processes. In *Proc. of STACS*, volume 1770 of *Lecture Notes in Computer Science*, pages 334–345, 2000.
13. P. Jančar, A. Kučera, F. Moller, and Z. Sawa. DP lower bounds for equivalence-checking and model-checking of one-counter automata. *Inf. Comput.*, 188(1):1–19, 2004.
14. P. Jančar, F. Moller, and Z. Sawa. Simulation Problems for One-Counter Machines. In *Proc. of SOFSEM*, volume 1725 of *Lecture Notes in Computer Science*, pages 404–413. Springer, 1999.
15. R. Mayr. Undecidability of Weak Bisimulation Equivalence for 1-Counter Processes. In *Proc. of ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 570–583, 2003.
16. M. L. Minsky. Recursive unsolvability of Post's problem of "tag" and other topics in theory of Turing machines. *Annals of Mathematics. Second Series*, 74:437–455, 1961.
17. D. Nowotka and J. Srba. Height-Deterministic Pushdown Automata. In *Proc. of MFCS*, volume 4708 of *Lecture Notes in Computer Science*, pages 125–134. Springer, 2007.
18. M. Oyamaguchi. The equivalence problem for real-time DPDAs. *J. ACM*, 34:731–760, 1987.
19. R. Roos. *Deciding Equivalence of Deterministic One-Counter Automata in Polynomial Time with Applications to Learning*. PhD thesis, The Pennsylvania State University, 1988.
20. G. Sénizergues. L(A)=L(B)? decidability results from complete formal systems. *Theor. Comput. Sci.*, 251(1-2):1–166, 2001.
21. G. Sénizergues. L(A)=L(B)? A simplified decidability proof. *Theor. Comput. Sci.*, 281(1-2):555–608, 2002.
22. G. Sénizergues. The Equivalence Problem for t-Turn DPDA Is Co-NP. In *Proc. of ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 478–489. Springer, 2003.
23. C. Stirling. Deciding DPDA Equivalence Is Primitive Recursive. In *Proc. of ICALP*, volume 2380 of *Lecture Notes in Computer Science*, pages 821–832. Springer, 2002.
24. L. G. Valiant and M. Paterson. Deterministic one-counter automata. *J. Comput. Syst. Sci.*, 10(3):340–350, 1975.