Bisimulation equivalence and regularity for real-time one-counter automata

Stanislav Böhm[1]

Technical University of Ostrava, FEI, 17. listopadu 15/2172, 70833 Ostrava, Czech Republic

Stefan Göller

University of Bremen, Fachbereich 03, Postfach 330440, 28334 Bremen, Germany

Petr Jančar[1]

Technical University of Ostrava, FEI, 17. listopadu 15/2172, 70833 Ostrava, Czech Republic

Abstract

A *one-counter automaton* is a pushdown automaton with a singleton stack alphabet, where stack emptiness can be tested; it is a *real-time* automaton if it contains no ε-transitions. We study the computational complexity of the problems of equivalence and regularity (i.e. semantic finiteness) on real-time one-counter automata. The first main result shows PSPACE-completeness of bisimulation equivalence; this closes the complexity gap between decidability (Jančar, 2000) and PSPACE-hardness (Srba, 2006). The second main result shows NL-completeness of language equivalence of *deterministic* real-time one-counter automata; this improves the known PSPACE upper bound (indirectly shown by Valiant and Paterson, 1975). Finally we prove P-completeness of the problem if a given one-counter automaton is bisimulation equivalent to a finite system, and NL-completeness of the problem if the language accepted by a given deterministic real-time one-counter automaton is regular.

Keywords: one-counter automaton, bisimulation equivalence, language equivalence, regularity

1. Introduction

Among the various notions of behavioural equivalence in *concurrency theory* [1], *bisimulation equivalence* (or *bisimilarity* for short) is undoubtedly a central one in formal verification (cf, e.g., [2]). We note that elegant characterizations of the bisimulation-invariant fragments of well-known logics like first-order logic, monadic second-order logic or monadic path logic have been obtained in terms of modal logic [3], the modal μ-calculus [4], and CTL^* [5], respectively. Hence it is natural to formulate the *bisimilarity problem*, asking if two given states of a given system are bisimilar. On *finite transition systems* this problem is P-complete [6] and well understood.

Email addresses: `stanislav.bohm@vsb.cz` (Stanislav Böhm), `goeller@informatik.uni-bremen.de` (Stefan Göller), `petr.jancar@vsb.cz` (Petr Jančar)

[1] S. Böhm and P. Jančar have been supported by the Grant Agency of the Czech Rep. (project GAČR:P202/11/0340)

In the setting of *infinite-state systems* (see, e.g., [7] for Mayr's classification of some of them) the situation is less clear, though a lot of research has been devoted to this area (see [8] for an up-to-date record). On the positive side we mention a very general and involved result by Sénizergues who shows that bisimilarity on equational graphs of finite out-degree (closely related to pushdown graphs) is decidable [9]. Unfortunately, there are various classes of infinite-state systems for which the decidability status of bisimilarity is not clarified so far. As examples we mention bisimilarity of PA (Process Algebra) processes and of ground tree rewrite systems.

When focussing on the *computational complexity* of bisimilarity checking of infinite-state systems for which this problem is decidable, the situation becomes even worse. E.g., the above-mentioned decidability result by Sénizergues only shows two semi-decision procedures, whereas a nonelementary lower bound has been established only recently [10]. To the best of the authors' knowledge, there has been essentially only one established class of infinite-state systems for which bisimilarity is decidable and the "exact" complexity is known, namely the basic parallel processes, where bisimilarity is PSPACE-complete [11].

Language equivalence essentially asks whether the sets of executable sequences of two given systems (often presented by automata) are equal; this is a central decision problem in formal languages and automata theory. It is folklore that already deciding whether a given pushdown automaton is universal is undecidable. We note that bisimilarity is finer than language equivalence, and the two equivalences coincide on *deterministic* systems. Language equivalence for deterministic devices has turned out to have several intricate instances, in particular for various subclasses of context-free languages. The most prominent result in this area is the decidability of equivalence of *deterministic pushdown automata (DPDA)*; this long-standing open decidability question has been answered positively by Sénizergues [12] (see also [13]), to which Stirling [14] established a primitive recursive upper bound. The problem still does not seem completely understood, which was one motivating factor for the recent simplified proof via first-order grammars, given in [15]. Regarding the lower bound for DPDA, language equivalence is only known P-hard (by the P-hardness of emptiness), hence the known complexity gap is very large.

Hence, a lot of research has been devoted to studying bisimulation (resp. language) equivalence of subclasses of (resp. deterministic) pushdown automata. A coNP upper bound for language equivalence was shown for finite-turn DPDA [16]. For simple grammars (real-time DPDA with a single control state), a polynomial algorithm was given in [17] (see [18] for a recent upper bound); the inclusion problem is undecidable even here [19]. For bisimilarity of the subclass BPA (real-time pushdown automata with a single control state) a 2EXPTIME upper bound has been stated by Burkart, Caucal and Steffen [20] (see [21] for an explicit proof), whereas the lower bound has recently been lifted from PSPACE to EXPTIME by Kiefer [22].

Another natural subclass of pushdown automata, the one in which we are interested here, are *one-counter automata*, i.e., pushdown automata with a singleton stack alphabet, where stack-emptiness can be tested. For bisimilarity of one-counter automata, decidability was shown in [23]. An unpublished article [24] analyses the decision procedure of [23] and derives a 3EXPSPACE upper bound. A PSPACE lower bound for bisimilarity is proven by Srba [25], even for a weaker model of visibly one-counter nets (that cannot test for zero). Srba [25] also shows a PSPACE upper bound for bisimilarity of visibly one-counter automata, via a reduction to the model checking problem of the modal μ-calculus over one-counter automata [26]. In the general case of (non-visibly) one-counter automata, the situation is surely more involved.

Deterministic one-counter automata (DOCA), where ε-transitions may occur in a deterministic fashion, were introduced by Valiant and Paterson [27]. In the same paper it was shown that language equivalence is decidable in time $2^{O(\sqrt{n \log n})}$. A simple analysis of the proof in [27] would yield a PSPACE upper bound for the problem. An announcement has been made that DOCA equivalence can be solved in polynomial time [28]; unfortunately, the full proof [29] has to be considered as incomplete. Hence the established complexity of the equivalence of DOCA has remained unsolved between NL and PSPACE. Polynomial time algorithms for language equivalence and inclusion for strict subclasses of real-time DOCA were given in [30, 31].

1.1. Our contribution

We study the computational complexity of deciding bisimilarity over transition systems generated by real-time one-counter automata (with no ε-transitions), denoted *ROCA* for short.[2] In general ROCA are nondeterministic; we also consider the deterministic version, det-ROCA, where bisimilarity essentially coincides with language equivalence.

The *first main result* of this paper closes the complexity gap for bisimilarity on ROCA: the known decidability (or the previously mentioned unpublished 3EXPSPACE upper bound) is improved by establishing PSPACE-completeness. Our *second main result* closes the complexity gap for det-ROCA: the known PSPACE upper bound is improved by establishing NL-completeness.

Another natural problem we consider is deciding *regularity* (semantic finiteness); the problem asks, given a state, if it is equivalent to a state of a finite system. For (nondeterministic) ROCA, the decidability of this problem with respect to bisimilarity was proven in [23]; according to [25], it follows from [6] and [32] that the problem is also P-hard. We show here that this problem is, in fact, P-complete. Besides giving a new upper bound, we also provide a simple direct proof of the lower bound.

We also show NL-completeness of the question if the language of a given deterministic real-time one-counter automaton is regular. The previously best known upper bound for this problem (similarly as for the more general model with ε-transitions) is a time bound of $2^{O(\sqrt{n \log n})}$ [27] (from where one can also derive a PSPACE upper bound).

The next table summarizes our complexity results. The lower bounds (including the folklore undecidability) were already known; here we show the upper bounds.

	Bis-EQUIV	Bis-REG	Lang-EQUIV	Lang-REG
ROCA	PSPACE-complete	P-complete	Undecidable	Undecidable
det-ROCA	NL-complete	NL-complete	NL-complete	NL-complete

As already mentioned, bisimilarity essentially coincides with language equivalence in the deterministic case; the bottom row thus contains only two results, in fact.

For proving these results, we employ an approach that can be called the "*belt technique*"; it was used already in [23] for decidability. Here we refine and enhance the technique, to yield a PSPACE upper bound. The main ideas can be sketched as follows. Given a ROCA $\mathcal{A}$, by $\mathcal{F}_{\mathcal{A}}$ we denote the finite automaton corresponding to the control unit of $\mathcal{A}$ in which we ignore the zero tests. For "large" counter values, $\mathcal{A}$ behaves like $\mathcal{F}_{\mathcal{A}}$ for "long time"; the only chance for $\mathcal{A}$ to show a difference with $\mathcal{F}_{\mathcal{A}}$ is to reach one of specific configurations with zero in the counter, called "incompatible configurations". If two configurations $p(m)$ and $q(n)$, where p, q are control states and m, n are counter values, are equivalent, then they must have the same distance to incompatible configurations; this implies that n is roughly linearly related to m, and

[2] Preliminary versions of the presented research results appeared at conferences Concur 2010 and MFCS 2011.

thus the pairs (m, n) of equivalent configurations lie inside "linear belts" when viewed as points in a 2-dimensional space.

To show that bisimilarity of ROCA belongs to PSPACE, we describe a nondeterministic procedure that is implementable in polynomial space; it constructs (guesses) a bisimulation relation *on-the-fly* while checking the *local consistency* of the guesses. In fact, the guesses are performed only for the pairs in (polynomially many) belts, since for the pairs outside the belts the correct answer can be computed in polynomial time by using the above observation about the distances to incompatible configurations. It is sufficient to perform only exponentially many steps; as if no inconsistency has been found then we are sure that the pigeonhole principle guarantees a repetition in each belt, and this guarantees the correctness of the positive answer.

The ideas in the proof also show that the set of all pairs $(p(m), q(n))$ that are equivalent has a regular structure, with exponential periods, whose natural description can be computed by using polynomial workspace.

For *deterministic* ROCA, our analysis shows that if we follow a shortest distinguishing word for two configurations with small counter values, then we cannot move in a belt for long; and once we leave the belt(s), the rest is short. This shows that two configurations with small counter values are not equivalent if and only if they can be distinguished by a word whose length can be bounded by a polynomial in the size of the input; an NL upper bound is thus immediate. For configurations with large counter values (written in binary), the shortest distinguishing words might be exponential but we can verify in nondeterministic logarithmic space that we can reach a nonequivalent pair outside the belts shortly or that we can reach a nonequivalent pair with small counter values (by moving down in a belt).

Finally the results on regularity follow easily, once we realize that a configuration is not equivalent to any finite state system if and only if its reachability set contains configurations with arbitrarily large distances to incompatible configurations.

1.2. Further related work

Further simulation and bisimulation problems on one-counter automata (with or without the zero tests) were studied in other papers; some of them also used the "belt technique". We can refer to the recent paper [33] and the references therein. Other problems studied for one-counter automata in the verification community can be exemplified by papers [34, 35, 36, 37, 38, 39].

Our NL-completeness result for deterministic real-time one-counter automata has not clarified the complexity of equivalence checking for general deterministic one-counter automata (with ε-transitions), left open in [27]. By using further (nontrivial) notions and ideas, we have shown NL-completeness also for the mentioned general case in [40].

1.3. Organisation of the paper

Section 2 provides general definitions and the statements of the results. Section 3 shows some simple facts, and clarifies the notion of "incompatible configurations". Section 4 contains a description of the main algorithm, deciding bisimilarity of real-time one-counter automata; a "geometrical presentation" of the algorithm is given in Section 5. In Section 6 we show the polynomial-space complexity of the algorithm, its correctness, and we sketch the description of the whole bisimulation equivalence relation for a given real-time one-counter automaton. Section 7 shows that the equivalence problem is in NL for deterministic ROCA. Finally, Section 8 presents the results for regularity problems.

2. Basic definitions and results

By $\mathbb{N}$ and $\mathbb{Z}$ we denote the set of nonnegative integers and the set of all integers, respectively. For $i, j \in \mathbb{Z}$, by $[i, j]$ we denote the set $\{i, i+1, \ldots, j\}$. For a finite set X, by $|X|$ we denote its cardinality. By Σ^* we denote the set of finite sequences of elements of Σ, i.e. of *words* over Σ. If $w \in \Sigma^*$ then $|w|$ denotes its *length*. By ε we denote the *empty word*; thus $|\varepsilon| = 0$. We put $\Sigma^+ = \Sigma^* \smallsetminus \{\varepsilon\}$.

Labelled transition systems (LTSs); deterministic LTSs

A *labelled transition system*, an *LTS* for short, is a tuple $\mathcal{T} = (S, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$, where S is a set of *states*, Σ is a set of *actions*, and $\xrightarrow{a} \subseteq S \times S$ is a set of *transitions* labelled with action a. If S and Σ are finite sets then $\mathcal{T}$ is a *finite LTS*. (In fact, we will only deal with LTSs where the action set Σ is finite while the state set S can be countably infinite.)

We write $s \xrightarrow{a} t$ instead of $(s, t) \in \xrightarrow{a}$, and we extend the relations $\xrightarrow{a}$ to $\xrightarrow{w}$ for words $w \in \Sigma^*$ inductively: $s \xrightarrow{\varepsilon} s$; if $s \xrightarrow{a} s'$ and $s' \xrightarrow{u} s''$ then $s \xrightarrow{au} s''$. By $s \xrightarrow{w}$ we denote that w is *enabled in* s, i.e., $s \xrightarrow{w} t$ for some t. We write $\longrightarrow$ for $\bigcup_{a\in\Sigma} \xrightarrow{a}$, and by $\longrightarrow^*$ we denote the reflexive and transitive closure of $\longrightarrow$. We say that t is *reachable from* s if $s \longrightarrow^* t$ (i.e., $s \xrightarrow{w} t$ for some $w \in \Sigma^*$).

An LTS $\mathcal{T} = (S, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$ is a *deterministic* LTS, a *det-LTS* for short, if for each pair $s \in S$, $a \in \Sigma$ there is at most one t such that $s \xrightarrow{a} t$.

Bisimulation equivalence on LTSs and det-LTSs

Let $\mathcal{T} = (S, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$ be an LTS. We say that $B \subseteq S \times S$ *covers* $(s, t) \in S \times S$ if for any $s \xrightarrow{a} s'$ there is $t \xrightarrow{a} t'$ such that $(s', t') \in B$, and for any $t \xrightarrow{a} t'$ there is $s \xrightarrow{a} s'$ such that $(s', t') \in B$. For $B, B' \subseteq S \times S$ we say that B *covers* B' if B covers each $(s, t) \in B'$. A set $B \subseteq S \times S$ is a *bisimulation* if B covers B. States $s, t \in S$ are *bisimilar*, which is denoted by $s \sim t$, if there is a bisimulation containing the pair (s, t).

The union of bisimulations is obviously a bisimulation. The relation $\sim$ is the greatest bisimulation, i.e., the union of all bisimulations on S; it is obviously an equivalence relation. *Bisimulation equivalence*, also called *bisimilarity*, is defined also between states of different LTSs, referring implicitly to their disjoint union.

We also note that for *deterministic LTSs* bisimulation equivalence coincides with the variant of language equivalence called *trace equivalence*: $s \sim t$ iff for all words $w \in \Sigma^*$ we have $s \xrightarrow{w} \Leftrightarrow t \xrightarrow{w}$ (i.e., s and t enable the same words, also called traces).

One-counter automata, and the generated LTSs

A *real-time one-counter automaton*, a *ROCA* for short, is a tuple $\mathcal{A} = (Q, \Sigma, \delta)$ where Q is a nonempty finite set of *control states*, Σ is a *finite alphabet*, whose elements are called *actions* in our context, and $\delta \subseteq Q \times \Sigma \times \{0, 1\} \times Q \times \{-1, 0, 1\}$ is a *transition relation* for which $(q, a, c, q', -1) \in \delta$ implies $c = 1$. The tuples $(q, a, c, q', j) \in \delta$ are also called *rules*; the *zero rules* have $c = 0$, and the *positive rules* have $c = 1$.

Remark. The word "real-time" refers to the fact that there are no ε-rules $(q, \varepsilon, c, q', j)$.

A *configuration* of $\mathcal{A}$ is a pair $(q, n) \in Q \times \mathbb{N}$ where n is the *value of the counter*; we often write $q(n)$ instead of (q, n). A ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ defines the LTS $\mathcal{T}(\mathcal{A}) = (Q \times \mathbb{N}, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$, where $q(n) \xrightarrow{a} q'(n + j)$ iff $(q, a, \mathrm{sgn}(n), q', j) \in \delta$; we put $\mathrm{sgn}(n) = 1$ if $n > 0$ and $\mathrm{sgn}(n) = 0$

if $n = 0$. The configurations $p(0)$ are called the *zero configurations*. (We note that no counter decrement is allowed in the zero configurations.)

A *ROCA* $\mathcal{A} = (Q, \Sigma, \delta)$ is *deterministic*, a *det-ROCA* for short, if for each triple $q \in Q$, $a \in \Sigma$, $c \in \{0, 1\}$ there is at most one rule of the form (q, a, c, q', j). We note that $\mathcal{T}(\mathcal{A})$ is deterministic iff $\mathcal{A}$ is deterministic.

In Fig. 1 we can see a fragment of $\mathcal{T}(\mathcal{A})$, where $\mathcal{A}$ contains the rules $(p, a, 0, q, 0)$, $(p, a, 1, q, 0)$, $(p, a, 1, p, 0)$, $(p, b, 0, r, 0)$, $(p, b, 1, r, 0)$, $(q, a, 0, q, +1)$, $(q, a, 1, p, -1)$, $(r, b, 0, r, 0)$, $(r, b, 0, q, +1)$, $(r, b, 1, q, +1)$.

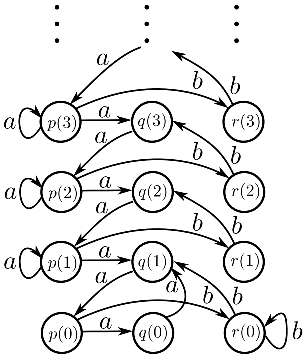

Figure 1: A fragment of the LTS $\mathcal{T}(\mathcal{A})$ generated by a ROCA $\mathcal{A}$

Decision problems, and the results

We recall two standard propositions and then state our results as theorems. We use the notation L (logarithmic space), NL, P (polynomial time), PSPACE, $\mathsf{NPSPACE}$ for the respective standard complexity classes.

The *bisimilarity problem for finite LTSs* asks, given a finite LTS (in a natural graph presentation) and two states s, t, whether $s \sim t$.

Proposition 1. *The bisimilarity problem is* P*-complete for finite LTSs, and* NL*-complete for deterministic finite LTSs.*

We refer to [6] for P-completeness. For a finite *deterministic* LTS $\mathcal{F}$ and two states s_0, t_0, we note that $s_0 \nsim t_0$ iff in the LTS $\mathcal{F} \times \mathcal{F}$ (where we put $(s, t) \xrightarrow{a} (s', t')$ if $s \xrightarrow{a} s'$ and $t \xrightarrow{a} t'$) we have $(s_0, t_0) \longrightarrow^* (s, t)$ for some (s, t) such that some action a is enabled precisely in one of s, t in $\mathcal{F}$. Hence bisimilarity in finite deterministic LTSs can be presented as digraph reachability, i.e., as a well-known NL-complete problem.

The *bisimilarity problem for ROCA* asks, given a ROCA $\mathcal{A}$ and two configurations $p(m)$ and $q(n)$, whether $p(m) \sim q(n)$ in $\mathcal{T}(\mathcal{A})$. In our complexity results (stated below) we assume a *standard input encoding* where the *counter values* m, n are given *in binary*; in fact, the given complexity bounds are also valid in the case of unary encodings.

We first observe that the bisimilarity problem and the language equivalence problem are log-space reducible to each other in the case of *deterministic* ROCA. The latter problem assumes a given det-ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ with a set of *accepting states* $F \subseteq Q$, and two configurations $p(m)$ and $q(n)$; it asks whether $L(p(m)) = L(q(n))$ where $L(r(k)) = \{w \in \Sigma^* \mid r(k) \xrightarrow{w} r'(k')$ for some $r' \in F$ and $k' \in \mathbb{N}\}$.

Proposition 2. *When restricted to det-ROCA, the bisimilarity problem and the language equivalence problem are log-space reducible to each other.*

Proof. Given a det-ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, for $F = Q$ we have $p(m) \sim q(n)$ iff $L(p(m)) = L(q(n))$. Hence bisimilarity reduces to language equivalence.

Now we assume a det-ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ and $F \subseteq Q$; we construct the det-ROCA $\mathcal{A}' = (Q \cup \{s\}, \Sigma \cup \{h\}, \delta \cup \delta')$ arising from $\mathcal{A}$ as follows. We extend Q with a fresh "sink" control state s and we add the rules $(s, a, c, s, 0)$ for all $a \in \Sigma$ and $c \in \{0, 1\}$; moreover, if for some triple (q, a, c) there is no rule of the form (q, a, c, q', j) then we add the rule $(q, a, c, s, 0)$. Finally we extend Σ with a fresh letter h and add the rules $(q, h, c, q, 0)$ for all $q \in F$ and $c \in \{0, 1\}$.

We can easily check that $p(m) \nsim q(n)$ in $\mathcal{T}(\mathcal{A}')$, for $p, q \in Q$, if and only if there is a word $w \in \Sigma^*$ such that wh is enabled precisely in one of $p(m), q(n)$; it is easy to check that the latter condition holds if and only if $L(p(m)) \neq L(q(n))$ (for $\mathcal{A}$ and F). Hence language equivalence reduces to bisimilarity. □

We will get the following results; recall the previous remark on the encodings of numbers.

Theorem 3. *The bisimilarity problem for ROCA is* PSPACE*-complete.*

Theorem 4. *For a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, the relation $\sim$ on the state set of $\mathcal{T}(\mathcal{A})$, i.e. the set $\{(p(m), q(n)) \mid p(m) \sim q(n)\}$, is effectively semilinear, with the description size exponential in the size of $\mathcal{A}$.*

Theorem 5.

1. *There is a polynomial* POLY *with the following property. For any det-ROCA $\mathcal{A}$ with* N *control states, if $p(0) \nsim q(0)$ then there is a word w that is enabled in precisely one of $p(0), q(0)$ and that satisfies $|w| \leq$* POLY(N).

2. *The bisimilarity problem and the language equivalence problem are* NL*-complete for det-ROCA.*

Recall that the semilinearity of $\sim$ (in Theorem 4) means that the set $\{(m, n) \mid p(m) \sim q(n)\}$ is the union of finitely many linear subsets of $\mathbb{N} \times \mathbb{N}$, for each pair p, q; a set $A \subseteq \mathbb{N}^k$ is linear if there is a base vector $b \in \mathbb{N}^k$ and periods $p_1, p_2, \ldots, p_\ell \in \mathbb{N}^k$ such that $A = \{b + c_1 p_1 + c_2 p_2 + \cdots + c_\ell p_\ell \mid c_1, c_2, \ldots, c_\ell \in \mathbb{N}\}$. Another view is that $\sim$ can be described by a formula in Presburger arithmetic [41]. In fact, our semilinear sets will be rather special, filling the "belts" and the "background" sketched in Fig. 5 periodically, with exponential periods. Polynomial workspace is sufficient for an algorithm generating a corresponding (exponential) description of $\sim$.

PSPACE-hardness in Theorem 3 follows from [25], and NL-hardness in Theorem 5 follows from Proposition 1; hence our contribution consists in showing the upper bounds.

We also consider the regularity problem. We say that a *configuration* $p(m)$ of a ROCA $\mathcal{A}$ is *regular* if $p(m) \sim f$ for some state f in a finite LTS; in other words, $p(m)$ is regular iff the set of states reachable from $p(m)$ is finite up to bisimilarity.

Theorem 6. *The problem asking if a given configuration $p(m)$ of a ROCA $\mathcal{A}$ is regular is* P*-complete. The restriction of the problem to det-ROCA is* NL*-complete.*

For det-ROCA we have an analogue of Proposition 2, i.e., our regularity problem and the language regularity problem are log-space reducible to each other in this case. In contrast, we recall that both language equivalence and language regularity are undecidable for general, i.e. nondeterministic, ROCA.

3. Prerequisites for the main algorithm

In Section 3.1 we observe some useful facts; Section 3.2 then recalls some important notions that already appeared in [23].

3.1. Simple facts about bisimilarity

We assume a fixed LTS $\mathcal{T} = (S, \Sigma, (\xrightarrow{a})_{a \in \Sigma})$.

Proposition 7. *If $R \subseteq S \times S$ is covered by $R \cup R'$ where $R' \subseteq\sim$ then $R \subseteq\sim$.*

Proof. If R is covered by $R \cup \sim$ then $R \cup \sim$ is a bisimulation, and thus $R \cup \sim\ \subseteq\ \sim$. □

For $U \subseteq S$, by $s \xrightarrow{w} U$ we denote that $s \xrightarrow{w} t$ for some $t \in U$; similarly $s \longrightarrow^* U$ means that $s \longrightarrow^* t$ for some $t \in U$. By the *distance* of $s \in S$ to $U \subseteq S$ we mean

$$\mathsf{distance}(s, U) = \min\{\, \ell \in \mathbb{N} \mid \exists w \in \Sigma^* : |w| = \ell \wedge s \xrightarrow{w} U\}\text{, where we put } \min \emptyset\ = \omega.$$

We view ω as the first limit ordinal; hence $n < \omega$ for all $n \in \mathbb{N}$.

We say that $U \subseteq S$ is *bisim-closed* if $\{s \in S \mid s \sim s' \text{ for some } s' \in U\} = U$.

Proposition 8. *If $s \sim t$ and U is bisim-closed then* $\mathsf{distance}(s, U) = \mathsf{distance}(t, U)$.

Proof. If $s \sim t$ and $s \xrightarrow{w} s'$ then there must be some t' such that $t \xrightarrow{w} t'$ and $s' \sim t'$; if, moreover, $s' \in U$ and U is bisim-closed then $t' \in U$. □

We now define the equivalences $\sim_0 \supseteq \sim_1 \supseteq \sim_2 \supseteq \cdots$ by the following inductive definition. We put $\sim_0= S \times S$. For $k \geq 1$, $\sim_k \subseteq S \times S$ is the set of all pairs covered by $\sim_{k-1}$. Note that $s \not\sim_1 t$ iff s and t enable different sets of actions (in which case there is no $B \subseteq S \times S$ that covers (s, t)). We obviously have $\bigcap_{i=0}^{\infty} \sim_i \supseteq \sim$.

Remark. An LTS $\mathcal{T} = (S, \Sigma, (\xrightarrow{a})_{a \in \Sigma})$ is *image-finite* if $\{s' \mid s \xrightarrow{a} s'\}$ is finite for each pair $s \in S$, $a \in \Sigma$; in this case we have $\bigcap_{i=0}^{\infty} \sim_i = \sim$. We note that $\mathcal{T}(\mathcal{A})$ generated by a ROCA $\mathcal{A}$ is image-finite.

The next proposition is also standard.

Proposition 9. *For any LTS $\mathcal{T} = (S, \Sigma, (\xrightarrow{a})_{a \in \Sigma})$ where $|S| = n \in \mathbb{N}$ we have $\sim_{n-1} = \sim_n = \sim$.*

Proof. By a standard partition refinement: when constructing $\sim_0, \sim_1, \sim_2, \ldots$, we must reach a fixpoint within n iterations. □

3.2. *The underlying finite LTS $\mathcal{F}_{\mathcal{A}}$ and the set* INC *of incompatible configurations*

Let us consider a ROCA $\mathcal{A}$. We recall that the counter value can change by at most one in one step and that the transitions do not depend on the concrete counter value when this value is positive. Hence if m is "large" then $p(m)$ behaves "for a long time" like p in the following finite LTS $\mathcal{F}_{\mathcal{A}}$ controlled by the *positive* rules of $\mathcal{A}$ (Fig. 2 shows an example):

Definition 10. For a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, we define the *underlying finite LTS $\mathcal{F}_{\mathcal{A}}$* as

$$\mathcal{F}_{\mathcal{A}} = (Q, \Sigma, (\xrightarrow{a})_{a\in\Sigma})$$

where $q \xrightarrow{a} q'$ iff there is j such that $(q, a, 1, q', j) \in \delta$.

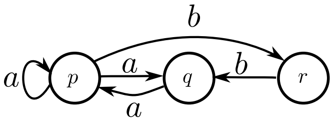

Figure 2: $\mathcal{F}_{\mathcal{A}}$ arising from $\mathcal{T}(\mathcal{A})$ in Fig. 1

We obviously have $p(m) \sim_m p$ (for any $p \in Q$ and any $m \in \mathbb{N}$).

Convention. We will usually leave implicit if a concrete occurrence of p (with no counter value) refers to a control state or to a state in $\mathcal{F}_{\mathcal{A}}$. E.g., in the expression $p(m) \sim_m p$ we view $p(m)$ as a state in $\mathcal{T}(\mathcal{A})$ and p as a state in $\mathcal{F}_{\mathcal{A}}$.

We now define the set INC as the set of configurations of $\mathcal{A}$ which are "INCompatible" with $\mathcal{F}_{\mathcal{A}}$ in the following sense:

Definition 11. Assuming a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, where $|Q| = \text{N}$, we define $\mathsf{INC} \subseteq Q \times \mathbb{N}$ and $\mathsf{dist} : Q \times \mathbb{N} \to \mathbb{N} \cup \{\omega\}$ as follows:

- $\mathsf{INC} = \{p(m) \mid \forall q \in Q : p(m) \not\sim_{\text{N}} q\}$,
- $\mathsf{dist}(p(m)) = \mathsf{distance}(p(m), \mathsf{INC})$.

We note that $p(m) \in \mathsf{INC}$ implies $m < \text{N}$ (since $m \geq \text{N}$ implies $p(m) \sim_{\text{N}} p$). Since INC is bisim-closed (if $p(m) \not\sim_{\text{N}} r$ and $p(m) \sim q(n)$ then $q(n) \not\sim_{\text{N}} r$), the next fact follows from Proposition 8:

Proposition 12. *If* $\mathsf{dist}(p(m)) \neq \mathsf{dist}(q(n))$ *then* $p(m) \not\sim q(n)$.

Comparing the distances of configurations to INC is an important ingredient of our algorithms. Regarding *the* INC*-membership problem*, asking if $p(m) \in \mathsf{INC}$ when given a ROCA $\mathcal{A}$ and $p(m)$, it is sufficient to observe a PSPACE-upper bound for the analysis of Alg-Bisim in Section 4. The more precise complexity bounds captured by the next proposition are useful later.

Proposition 13. *The* INC*-membership problem is* P*-complete; it is* NL*-complete when restricted to deterministic ROCA.*

Proof. We assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, where $|Q| = \mathrm{N}$, and show a polynomial-time algorithm constructing INC. To the underlying finite LTS $\mathcal{F}_{\mathcal{A}}$ we (disjointly) add the restriction of $\mathcal{T}(\mathcal{A})$ to the state set $\{p(m) \mid p \in Q, m \in [0, \mathrm{N}{-}1]\}$; each original transition $p(\mathrm{N}{-}1) \xrightarrow{a} q(\mathrm{N})$ is replaced with $p(\mathrm{N}{-}1) \xrightarrow{a} q$ (recall that $q(\mathrm{N}) \sim_{\mathrm{N}} q$). In the resulting finite LTS with $\mathrm{N} + \mathrm{N}^2$ states we construct the state-set partition corresponding to $\sim_{\mathrm{N}}$, by standard partition-refinement techniques (constructing $\sim_0, \sim_1, \ldots, \sim_{\mathrm{N}}$). Now $p(m)$ belongs to INC iff it has no q in its partition class. Hence the INC-membership problem is in P.

We now show that the INC-membership problem is in NL for det-ROCA. The respective nondeterministic algorithm, given a det-ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ and $p_0(m_0)$, first compares m_0 and $\mathrm{N} = |Q|$; if $m_0 \geq \mathrm{N}$, then it returns NO (since $p_0(m_0) \sim_{\mathrm{N}} p_0$ and thus $p_0(m_0) \notin \mathsf{INC}$). If $m_0 < \mathrm{N}$ then the algorithm tries to show $p_0(m_0) \not\sim_{\mathrm{N}} q$, successively for each $q \in \mathcal{F}_{\mathcal{A}}$. Since the LTSs $\mathcal{T}(\mathcal{A})$ and $\mathcal{F}_{\mathcal{A}}$ are deterministic, we have $p(m) \not\sim_k q$ iff $p(m) \not\sim_1 q$ or there is $a \in \Sigma$ such that $p(m) \xrightarrow{a} p'(m')$, $q \xrightarrow{a} q'$, and $p'(m') \not\sim_{k-1} q'$. It is thus sufficient that the workspace of the algorithm can store a pair $(p(m), q)$ and a number $k \leq \mathrm{N}$, where $m < 2\mathrm{N}$; since the numbers m, k can be stored in binary, a logarithmic bound for the workspace size is obvious.

We show the hardness results by a (log-space) reduction from the non-bisimilarity problem for finite LTSs (recall Proposition 1, and the fact that both P and NL are closed under complement). Assume a finite LTS $\mathcal{T} = (S, \Sigma, (\xrightarrow{a})_{a \in \Sigma})$ and two states $p_0, q_0 \in S$. We construct the ROCA $\mathcal{A} = (S \cup \{p'_0, q'_0\}, \Sigma \cup \{a'\}, \delta)$ where $p'_0, q'_0 \notin S$, $p'_0 \neq q'_0$, and $a' \notin \Sigma$; the rules in δ are defined inductively as follows: for any $p, q \in S$ and $a \in \Sigma$, if $p \xrightarrow{a} q$ (in $\mathcal{T}$) then $(p, a, 1, q, 0)$ is in δ; we also put $(p'_0, a', 0, p_0, 1)$ and $(q'_0, a', 1, q_0, 0)$ in δ. We note that $\mathcal{A}$ is a det-ROCA if $\mathcal{T}$ is a det-LTS. We observe that $r(1) \sim r$ for all states r of $\mathcal{F}_{\mathcal{A}}$; moreover, if $r \neq q'_0$ then $p'_0(0) \not\sim_1 r$. It is also clear that $p_0 \sim_k q_0$ in $\mathcal{T}$ iff $p'_0(0) \sim_{k+1} q'_0$. Hence if $p_0 \sim q_0$ in $\mathcal{T}$ then $p'_0(0) \sim q'_0$, in which case $p'_0(0) \notin \mathsf{INC}$. If $p_0 \not\sim q_0$ in $\mathcal{T}$, hence $p_0 \not\sim_k q_0$ for $k = |S| - 1$ (by Proposition 9), then $p'_0(0) \not\sim_{k+1} q'_0$, and thus $p'_0(0) \in \mathsf{INC}$. □

The distance of $p(m)$ to INC is given by a shortest appropriate path in $\mathcal{T}(\mathcal{A})$ (if it exists). A possible shortest path from $p(m)$ to INC is depicted in Fig. 3. Since the counter can drop by at most one in one step, and $r(k) \in \mathsf{INC}$ implies $k < \mathrm{N}$, we have $\mathsf{dist}(p(m)) > m - \mathrm{N}$; hence $\mathsf{dist}(p(m)) < \omega$ implies that the set $\{q(n) \mid \mathsf{dist}(q(n)) = \mathsf{dist}(p(m))\}$ is finite. We can also anticipate that the constraint $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) < \omega$ yields a certain linear relation between m and n, as made more precise later. The complexity questions of computing $\mathsf{dist}(p(m))$ will be also addressed later.

Now we note an important property of the configurations from which INC is unreachable:

Lemma 14. *Assume a ROCA $\mathcal{A}$. If $\mathsf{dist}(p(m)) = \omega$ then $p(m) \sim r$ for some state r of $\mathcal{F}_{\mathcal{A}}$.*

Proof. Let us assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, where $|Q| = \mathrm{N}$. We verify that the set

$$R \quad = \quad \{\, (p(m), q) \mid p(m) \not\longrightarrow^{*} \mathsf{INC}, p(m) \sim_{\mathrm{N}} q \,\}$$

is a bisimulation; the proof will be finished, since $p(m) \not\longrightarrow^{*} \mathsf{INC}$ implies $p(m) \notin \mathsf{INC}$, and thus $p(m) \sim_{\mathrm{N}} q$ for some q.

Let $(p(m), q) \in R$. Since $p(m) \sim_{\mathrm{N}} q$, for any $p(m) \xrightarrow{a} p'(m')$ there is $q \xrightarrow{a} q'$ such that $p'(m') \sim_{\mathrm{N}-1} q'$; similarly for any $q \xrightarrow{a} q'$ there is $p(m) \xrightarrow{a} p'(m')$ such that $p'(m') \sim_{\mathrm{N}-1} q'$. Since $p(m) \not\longrightarrow^{*} \mathsf{INC}$, we have $p'(m') \not\longrightarrow^{*} \mathsf{INC}$, and thus also $p'(m') \notin \mathsf{INC}$; let r satisfy $r \sim_{\mathrm{N}} p'(m')$. Since $\sim_{\mathrm{N}-1}$ coincides with $\sim_{\mathrm{N}}$ in $\mathcal{F}_{\mathcal{A}}$ (by Proposition 9), we have $r \sim_{\mathrm{N}} q'$, and thus $p'(m') \sim_{\mathrm{N}} q'$; this implies $(p'(m'), q') \in R$. □

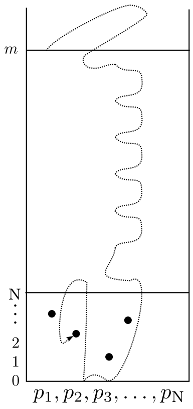

Figure 3: A path from $p(m)$ to INC

Corollary 15. *Assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, where $|Q| = \mathrm{N}$. If $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) = \omega$ then $p(m) \sim q(n)$ iff $p(m) \sim_{\mathrm{N}} q(n)$.*

Proof. Assume $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) = \omega$. The “only-if”-direction of the claim is trivial. For proving the “if”-direction, we recall that $p(m) \sim r_1$ and $q(n) \sim r_2$ for some r_1, r_2 in $\mathcal{F}_{\mathcal{A}}$ (by Lemma 14); if $r_1 \sim_{\mathrm{N}} r_2$ then $r_1 \sim r_2$ (by Proposition 9). □

4. Algorithm Alg-Bisim deciding bisimilarity for ROCA

After introducing some further notation we will present our main algorithm, deciding the bisimilarity problem for ROCA in polynomial space.

Definition 16. Assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = \mathrm{N}$. We partition $(Q \times \mathbb{N}) \times (Q \times \mathbb{N})$ into three parts: $(Q \times \mathbb{N}) \times (Q \times \mathbb{N}) = \mathsf{ClearYes} \cup \mathsf{ClearNo} \cup \mathsf{Unclear}$ where

- $\mathsf{ClearYes} = \{(p(m), q(n)) \mid \mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) = \omega$ and $p(m) \sim_{\mathrm{N}} q(n)\}$,
- $\mathsf{ClearNo} = \{(p(m), q(n)) \mid \mathsf{dist}(p(m)) \neq \mathsf{dist}(q(n))$ or $p(m) \not\sim_{\mathrm{N}} q(n)\}$,
- $\mathsf{Unclear} = \{(p(m), q(n)) \mid \mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) < \omega$ and $p(m) \sim_{\mathrm{N}} q(n)\}$.

We also put

$$\mathsf{Unclear} = \mathsf{EFD}_0 \cup \mathsf{EFD}_1 \cup \mathsf{EFD}_2 \cup \cdots$$

where $\mathsf{EFD}_i = \mathsf{Unclear} \cap \{(p(i), q(n)) \mid p, q \in Q, n \in \mathbb{N}\}$. ($\mathsf{EFD}$ can be read as “Equal Finite Distances”.)

We note that $\mathsf{ClearYes} \subseteq \sim$ and $\mathsf{ClearNo} \subseteq \not\sim$ (by the previously established facts). We have already observed that $\mathsf{dist}(p(m)) < \omega$ implies that the set $\{q(n) \mid \mathsf{dist}(q(n)) = \mathsf{dist}(p(m)\}$ is finite; hence EFD_i is finite for each $i \in \mathbb{N}$.

The nondeterministic algorithm Alg-Bisim:

Input: a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, and two configurations $p_0(m_0), q_0(n_0)$.

1. If $(p_0(m_0), q_0(n_0))$ is in $\mathsf{ClearYes}$ then return YES; if in $\mathsf{ClearNo}$ then return NO.
2. (This point applies when $(p_0(m_0), q_0(n_0)) \in \mathsf{EFD}_{m_0}$.)
   (a) Compute a bound ExpB (to be clarified later), exponential in the size of $\mathcal{A}$.
   (b) Put $R_{-2} = R_{-1} = \emptyset$.
   (c) *For* $i = 0, 1, 2, \ldots, m_0, m_0{+}1, m_0{+}2, \ldots, m_0{+}\text{ExpB}$ *do*
      i. Choose $R_i \subseteq \mathsf{EFD}_i$; if $i = m_0$ then R_i must contain $(p_0(m_0)), (q_0(n_0))$.
      ii. If R_{i-1} is not covered by $R_{i-2} \cup R_{i-1} \cup R_i \cup \mathsf{ClearYes}$ then FAIL.
   (d) Return YES.

It will turn out that this algorithm can be implemented to run in polynomial space, and that there is a computation returning YES if and only if $p_0(m_0) \sim q_0(n_0)$. Since PSPACE = NPSPACE, the upper bound in Theorem 3 will be thus established.

We perform the respective analysis of Alg-Bisim in Section 6, after we "visualize" some related notions in Section 5.

Now we just remark that $p_0(m_0) \sim q_0(n_0)$ implies that the computation that always chooses $R_i = \mathsf{EFD}_i \cap \sim$ in 2(c)i returns YES. On the other hand, if the for-loop in 2(c) had no upper bound then for any infinite (i.e., non-failing) computation we would have $(R_0 \cup R_1 \cup R_2 \cup \cdots) \subseteq\sim$, by Proposition 7; this would imply $p_0(m_0) \sim q_0(n_0)$. The bound ExpB in 2(a) will be chosen so that a successful run up to $m_0 + \text{ExpB}$ guarantees a certain periodicity that in turn guarantees the existence of some infinite successful run if the for-loop had no upper bound.

5. Geometrical presentation of Alg-Bisim computations

Let us assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$. For any fixed $p, q \in Q$, a subset X of $\{(p(m), q(n)) \mid m, n \in \mathbb{N}\}$ can be naturally represented by black points in the 2-dimensional grid $\mathbb{N} \times \mathbb{N}$: point (m, n) is black if $(p(m), q(n)) \in X$, and white if $(p(m), q(n)) \notin X$. This is depicted in Fig. 4.

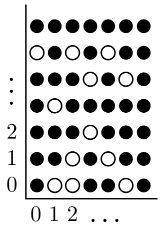

Figure 4: A black-white colouring representing a subset of $\{(p(m), q(n)) \mid m, n \in \mathbb{N}\}$, for fixed p, q.

For representing subsets X of $\{(p(m), q(n)) \mid p, q \in Q, m, n \in \mathbb{N}\}$, we can put the respective $|Q|^2$ 2-dimensional grids together, creating the 3-dimensional grid $\mathbb{N} \times \mathbb{N} \times (Q \times Q)$; we have only $|Q|^2$ values in the third dimension. (Figures 5 and 7 should make this clear.) Here the point with coordinates $(m, n, (p, q))$ is black iff $(p(m), q(n)) \in X$.

Fig. 5 indicates an over-approximation of $\mathsf{Unclear}$, as the later analysis will establish. The set $\mathsf{Unclear}$ resides in the "belt space", consisting of polynomially many linear belts with a polynomial (vertical) thickness. There is a polynomially bounded "initial space" covering all intersections of different belts; moreover, ClearYes will turn out to be periodic outside the initial space, with an exponentially bounded period.

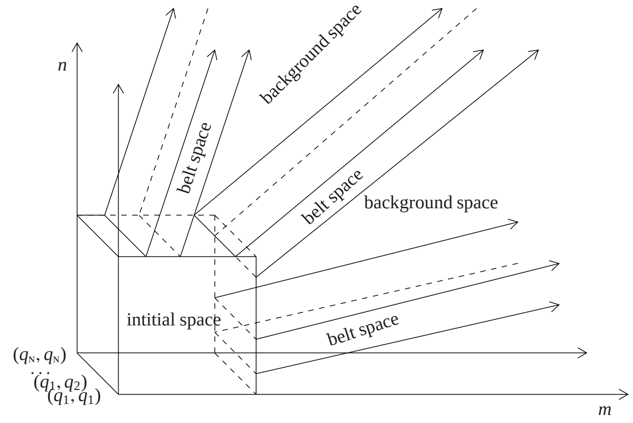

Figure 5: Partition of our 3-dimensional grid

A computation of Alg-Bisim can be viewed as moving a width-3 vertical window, depicted in Fig. 6. Each chosen set R_i is contained in the i-th "vertical cut" of the belts.

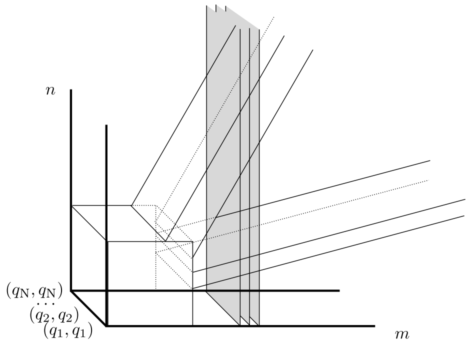

Figure 6: Vertical window of width 3, moved by Alg-Bisim

Fig. 7 illustrates a "repeat" of the cut in a belt, at positions i and i'; here each depicted black point corresponds to an element of either R_j ($j \in \{i, i'\}$) or ClearYes. The exponential bound ExpB in 2(a) of Alg-Bisim (and the pigeonhole principle) will guarantee a repeat in which the difference of positions is a multiple of the (exponentially bounded) period of ClearYes; this will provide the announced guarantee of the existence of an infinite computation when no fail is encountered in 2(c)ii till m_0 + ExpB. To be more precise, we will need a repeat of a width-2 belt-cut, not just of a width-1 belt-cut depicted in Fig. 7.

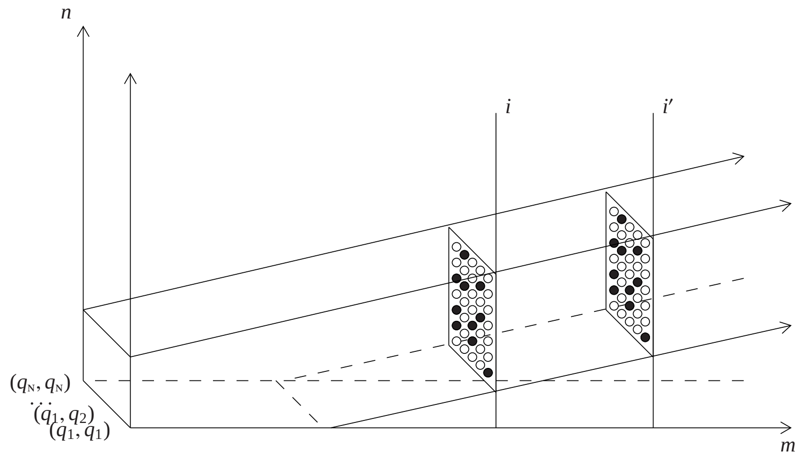

Figure 7: Repeat of a belt-cut

6. Analysis of Alg-Bisim, and the effective semilinearity of ∼

In Section 6.1 we note some facts about the shortest paths in $\mathcal{T}(\mathcal{A})$, in particular a normal form based on a lemma given already in [27]. In Section 6.2 we note some consequences of these facts for computing distances in $\mathcal{T}(\mathcal{A})$, and for the membership problems for ClearYes, ClearNo, and Unclear. We then look at the shortest paths to INC, yielding the function $\mathsf{dist}(p(m))$ (the distance to INC), in Section 6.3. In Section 6.4 we make precise the periodicity of ClearYes, and we show the linear belts in which Unclear resides. In Section 6.5 we confirm that Alg-Bisim works in polynomial space, and in Section 6.6 we demonstrate that Alg-Bisim indeed decides the bisimilarity problem for ROCA. In Section 6.7 we derive the semilinear description of ∼ stated in Theorem 4.

6.1. Normal forms of shortest paths in $\mathcal{T}(\mathcal{A})$

If we have $p(m) \longrightarrow^* q(n)$ in the LTS $\mathcal{T}(\mathcal{A})$ for a ROCA $\mathcal{A}$, then a shortest path from $p(m)$ to $q(n)$ might be long even if $|m - n|$ is small; in this case $q(n)$ is not reachable from $p(m)$ by using positive rules only. We now want to show a normal form of shortest paths; it is sketched in Fig. 8 for the case when using zero rules is necessary.

The paths induced solely by positive rules are called *positive paths*; we formalize the positive reachability relation as follows:

Definition 17. For a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, we define the relations $\xrightarrow{w}_+$ for all $w \in \Sigma^*$ inductively: $p(m) \xrightarrow{\varepsilon}_+ p(m)$; if $m > 0$, $p(m) \xrightarrow{a} p'(m')$ for $a \in \Sigma$, and $p'(m') \xrightarrow{u}_+ q(n)$ then $p(m) \xrightarrow{au}_+ q(n)$. By $p(m) \longrightarrow^*_+ q(n)$ we denote that $p(m) \xrightarrow{w}_+ q(n)$ for some $w \in \Sigma^*$.

We note that only the last node of a positive path might be a zero configuration.

The following proposition, illustrated in Fig. 9, captures a standard simple fact: if a path from $p(m)$ to $q(n)$ makes a "high hill" then there is a shorter path from $p(m)$ to $q(n)$. The bounds in the proposition are not the best possible, but they are easy to show.

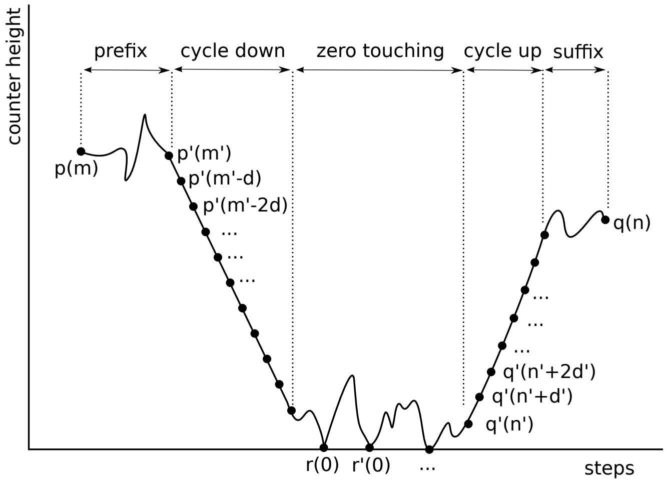

Figure 8: A shortest path from $p(m)$ to $q(n)$

Proposition 18. *Assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, where $|Q| = \textsc{n}$, and a path*

$$p_0(m_0) \xrightarrow{a_1} p_1(m_1) \xrightarrow{a_2} \cdots \xrightarrow{a_\ell} p_\ell(m_\ell) \tag{1}$$

where $a_i \in \Sigma$ and $w = a_1 a_2 \ldots a_\ell$ is a shortest word such that $p_0(m_0) \xrightarrow{w} p_\ell(m_\ell)$. Then for each $j \in [0, \ell]$ we have $m_j \leq \textsc{n}^2$ in the case $m_0 = m_\ell = 0$, and $m_j < \max\{m_0, m_\ell\} + \textsc{n}^2$ otherwise.

Moreover, if (1) is a shortest positive *path from $p_0(m_0)$ to $p_\ell(m_\ell)$ then for each $j \in [0, \ell]$ we have* $\min\{m_0, m_\ell\} - \textsc{n}^2 < m_j < \max\{m_0, m_\ell\} + \textsc{n}^2$.

Proof. If there is a counterexample (1) with $m_0 = m_\ell = 0$ then for the smallest i such that $m_i > 0$, i.e. $m_i = 1$, we have that $p_i(m_i) \xrightarrow{a_{i+1}} p_{i+1}(m_{i+1}) \xrightarrow{a_{i+2}} \cdots \xrightarrow{a_\ell} p_\ell(m_\ell)$ is also a counterexample.

Suppose now that (1) is a counterexample where $m_x = \max\{m_0, m_\ell\} \geq 1$. Let us fix some $j \in [1, \ell - 1]$ such that $m_j = m_x + \textsc{n}^2$. For each $h \in [0, \textsc{n}^2]$ we now define

$$f(h) = \max\{i \in [0, j] \mid m_i = m_x + h\} \qquad \text{and} \qquad g(h) = \min\{i \in [j, \ell] \mid m_i = m_x + h\}.$$

We note that $f(h), g(h)$ are well defined, and $f(0) < f(1) < \cdots < f(\textsc{n}^2) = j = g(\textsc{n}^2) < g(\textsc{n}^2{-}1) < \cdots < g(0)$; moreover, $m_i \geq m_x + h$ for all $i \in [f(h), g(h)]$. This also implies that the path $p_{f(0)} \xrightarrow{a_{f(0)+1}} p_{f(0)+1} \xrightarrow{a_{f(0)+2}} \cdots \xrightarrow{a_{g(0)}} p_{g(0)}$ is positive. By the pigeonhole principle we get some h, h', where $0 \leq h < h' \leq \textsc{n}^2$ and $p_{f(h)} = p_{f(h')}$, $p_{g(h)} = p_{g(h')}$ (we have $\textsc{n}^2 + 1$ values h in $[0, \textsc{n}^2]$, and only $\textsc{n}^2$ pairs of control states). But then we could remove $a_{f(h)+1} \ldots a_{f(h')}$ and $a_{g(h')+1} \ldots a_{g(h)}$ since $p_{f(h)}(m_{f(h)}) = p_{f(h)}(m_x + h) \xrightarrow{u}_+ p_{g(h)}(m_x + h) = p_{g(h)}(m_{g(h)})$ for $u = a_{f(h')+1} \ldots a_{g(h')}$; this contradicts the assumption that w is a shortest word such that $p_0(m_0) \xrightarrow{w} p_\ell(m_\ell)$.

The final claim for positive paths is derived analogously. □

Given a shortest path from $p(m)$ to $q(n)$, it is trivial that any subpath is a shortest path from its start to its end. Proposition 18 thus bounds the maximum counter value in the "zero-touching" part in Fig. 8, as well as the maxima of the "going-down" part and of the "going-up" part. We

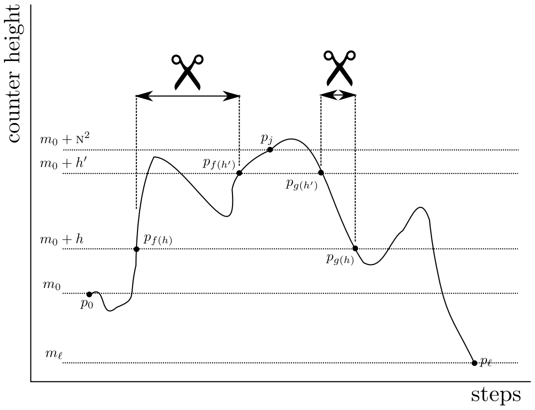

Figure 9: "Cutting a hill"

also have a lower bound for the overall minimum when there is no zero touching. Now we clarify the cycles; we concentrate just on the "going-down" part, since the "going-up" part is almost analogous when we reverse the positive ROCA-rules (i.e., replace each rule $(p, a, 1, q, j)$ with $(q, a, 1, p, -j)$).

Definition 19. Let $\mathcal{A} = (Q, \Sigma, \delta)$ be a ROCA. By a *cycle* we mean a nonempty sequence of positive rules $(q_1, a_1, 1, q_2, j_1), (q_2, a_2, 1, q_3, j_2), (q_3, a_3, 1, q_4, j_3), \dots, (q_k, a_k, 1, q_{k+1}, j_k)$ where $q_{k+1} = q_1$; the number $k \geq 1$ is the *length of the cycle*. The above *cycle* is *simple* if $1 \leq i < j \leq k$ implies $q_i \neq q_j$. The number $e = \sum_{i=1}^{k} j_i$ is called the *effect* of the cycle; if $e < 0$, then $d = -e$ is called the *drop* of the cycle.

We note that the effect of a cycle is the change of the counter value that the cycle causes when performed. If the length of a cycle is k, then its effect is in $[-k, k]$. If $|Q| = \textsc{n}$, then the length of any simple cycle is in $[1, \textsc{n}]$ (and its effect is in $[-\textsc{n}, \textsc{n}]$).

We refer to [27] for a proof of the next proposition; intuitively, if $|m - n| \geq \textsc{n}^2$ and $p(m) \longrightarrow_{+}^{*} q(n)$, then there is a shortest positive path from $p(m)$ to $q(n)$ in a certain normal form: the path starts with a "short" prefix, then uses repeatedly a simple cycle (at least once), and finishes with a "short" suffix (where the sum of lengths of the prefix and the suffix is less than $\textsc{n}^2$).

In fact, only *deterministic* one-counter automata are considered in [27]. Nevertheless, the actions labelling the transitions are irrelevant for the reachability questions. In the proposition we can thus conveniently assume a bijection between Σ and δ: each action a has a corresponding rule (q, a, c, q', j). We then say that $v \in \Sigma^+$ is a cycle if the corresponding sequence of (positive) rules is a cycle.

Proposition 20. *(***Lemma** 2 *in [27].) Let $\mathcal{A} = (Q, \Sigma, \delta)$ be a ROCA where $|Q| = \textsc{n}$. Assume $p(m) \longrightarrow_{+}^{*} q(n)$ and $m \geq n + \textsc{n}^2$. Then there are words w, v_1, v_2, v_3 such that w is a shortest word satisfying $p(m) \xrightarrow{w}_{+} q(n)$, and*

- *$w = v_1(v_2)^i v_3$ for some $i > 0$,*
- *$|v_1 v_3| < \textsc{n}^2$, and v_2 is a cycle with $|v_2| \leq \textsc{n}$ and with a drop $d \in [1, \textsc{n}]$,*

- $p(m) \xrightarrow{v_1}_+ p'(m') \xrightarrow{v_2}_+ p'(m'-d) \xrightarrow{v_2}_+ p'(m'-2d) \xrightarrow{v_2}_+ \cdots \xrightarrow{v_2}_+ p'(m'-id) \xrightarrow{v_3}_+ q(n)$ *for some* $p' \in Q$ *and* $m' \in \mathbb{N}$ *(where* v_2 *is repeated* i *times).*

In later applications of Proposition 20 we will also implicitly use the fact that in the described case we can cut off and pump the cycle in the following sense:

$$p(m+(j-i)d) \xrightarrow{v_1(v_2)^j v_3}_+ q(n) \text{ for all } j > 0 \text{ such that } m+(j-i)d \geq n+\text{N}^2, \text{ and}$$

$$p(m) \xrightarrow{v_1(v_2)^j v_3}_+ q(n+(i-j)d) \text{ for all } j \in [0,i].$$

There is an analogous claim for $p(m) \longrightarrow^*_+ q(n)$ where $m + \text{N}^2 \leq n$; here v_2 is a cycle with a positive effect. The claim follows from Proposition 20 by reversing the positive rules (i.e. replacing $(p,a,1,q,j)$ with $(q,a,1,p,-j)$) and considering $q(n) \longrightarrow^*_+ p(m)$. We can also analogously cut off and pump the cycle.

In the next section we use Propositions 18 and 20 for noting a fact about the complexity of computing distances. This fact will help us later to clarify the membership problems for ClearYes, ClearNo and Unclear. In fact, just polynomial-space algorithms would suffice for our analysis of Alg-Bisim; the better complexity bounds in Section 6.2 are substantial for the deterministic case.

6.2. Computing (representations of) distances for ROCA

We first recall a standard simple fact regarding space-efficient implementations of (integer) arithmetic operations:

Proposition 21. *There is a procedure that, given* $\text{OP} \in \{+,-,\cdot,\div, \bmod\}$ *and* $m,n,j \in \mathbb{N}$ *in binary, returns the* j*-th bit of* $(m \text{ OP } n)$*, while using workspace* $O(\log\log\max\{m,n\})$ *when* $\text{OP} \in \{+,-\}$ *and* $O(\max\{\log\log\max\{m,n\}, \log\min\{m,n\}\})$ *when* $\text{OP} \in \{\cdot,\div,\bmod\}$.

Informally speaking, in the case $\text{OP} \in \{+,-\}$ just two pointers moving in the binary presentations of m and n are sufficient (when performing the standard algorithm); if $\text{OP} \in \{\cdot,\div,\bmod\}$ then we also use a piece of workspace that can store the smaller of m,n (while realizing a standard textbook algorithm).

Given a ROCA $\mathcal{A}$ and two configurations $p(m)$, $q(n)$, the value $\mathsf{distance}(p(m),\{q(n)\})$ can be obviously written in linear space (in binary); this follows easily from Propositions 18 and 20 (recall also Fig. 8). The next proposition shows that each specific bit of $\mathsf{distance}(p(m),\{q(n)\})$ can be computed in nondeterministic logarithmic space (and thus also in polynomial time).

Remark. We thus also get NL-completeness of the reachability problem for ROCA, when the initial and final counter values are given in binary. The proposition is derived from Prop. 20 (i.e. Lemma 2 in [27]) by using standard means (like Prop. 21); we provide a proof to be self-contained.

Proposition 22. *The following decision problem is* NL*-complete.*

Input: *A ROCA* $\mathcal{A}$*, two configurations* $p(m)$*,* $q(n)$*,* $j \in \mathbb{N}$*,* $c \in \{0,1\}$ *(*m,n,j *written in binary).*

Question: *Is* $\mathsf{distance}(p(m),\{q(n)\})$ *finite and is the* j*-th bit of its binary presentation* c *?*

Proof. NL-hardness follows from digraph reachability; we will show that the problem is in NL.

Assume a given ROCA $\mathcal{A} = (Q,\Sigma,\delta)$, where $|Q| = \text{N}$, and two configurations $p(m), q(n)$. We first show a nondeterministic procedure deciding if $p(m) \longrightarrow^*_+ q(n)$.

1. If $|m - n| < \text{N}^2$, then we just stepwise guess a respective positive path from $p(m)$ to $q(n)$; we always remember just the current configuration $p'(m')$, where m' is represented by the difference $d = m' - m$ in the workspace. By Proposition 18 we can restrict ourselves to $d \in [-(m-\min\{m, n\}+\text{N}^2-1), \max\{m, n\}-m+\text{N}^2-1]$; this guarantees $d \in [-2\text{N}^2+2, 2\text{N}^2-2]$, and thus d can be written in $4 \log \text{N}$ bits. At the same time we can count the length ℓ of the guessed path (presenting ℓ in binary).

2. If $|m-n| \geq \text{N}^2$ then we base the procedure on the normal-form path guaranteed by Prop. 20; w.l.o.g. we assume $m > n$ since otherwise we could just reverse the positive rules. We guess a tuple $(d_1, \ell_1, d_2, \ell_2, d_3, \ell_3, p')$ where $0 \leq \ell_1 + \ell_2 < \text{N}^2$, $|d_1| + |d_2| < \text{N}^2$, $d_3, \ell_3 \in [1, \text{N}]$, and $p' \in Q$. We verify that

   - from $p(m)$ we can reach $p'(m+d_1)$ in ℓ_1 moves,
   - from $p'(n+d_2)$ we can reach $q(n)$ in ℓ_2 moves,
   - from $p'(n+d_2+d_3)$ we can reach $p'(n+d_2)$ in ℓ_3 moves, and
   - d_3 divides $(m + d_1) - (n + d_2)$.

   Each configuration $r(k)$ stored in the workspace during this process is represented by $(r, k-m)$ or by $(r, k-n)$ (i.e., we put only small differences in the workspace, as in 1.).

The above nondeterministic procedure obviously runs in logarithmic space; moreover, any successful run also yields a (small) presentation of the length of some path from $p(m)$ to $q(n)$ (i.e. of an upper bound for $\mathsf{distance}(p(m), \{q(n)\})$): either ℓ in 1., or the tuple $(\ell_1, \ell_2, \ell_3, d_1, d_2, d_3)$ in 2.; in the latter case, the represented (big) number is

$$\ell_1 + \ell_2 + \ell_3 \cdot ((m + d_1) - (n + d_2)) \div d_3.$$

For deciding if $p(m) \longrightarrow^* q(n)$ (when the zero rules are allowed), we add the possibility to guess some $r, r' \in Q$ and to verify that $p(m) \longrightarrow^*_+ r(0)$, $r(0) \longrightarrow^* r'(0)$, and that $q(n) \longrightarrow^*_+ r'(0)$ when the (positive) rules are reversed. It is clear that this variant also runs in logarithmic space, and any successful run provides a (small) presentation of the length of a path from $p(m)$ to $q(n)$.

For a concrete presentation of an upper bound for $\mathsf{distance}(p(m), \{q(n)\})$, we can decide in nondeterministic logarithmic space if the bound can be strengthened; this follows from the fact that we can compare two (small) presentations by using the procedures captured by Proposition 21.

Since NL is closed under complement, we can thus construct a nondeterministic procedure working in logarithmic space where each successful run finishes with a presentation of $\mathsf{distance}(p(m), \{q(n)\})$. Extracting the j-th bit of $\mathsf{distance}(p(m), \{q(n)\})$ from the presentation can be done in logarithmic space (by invoking Proposition 21 again). □

Proposition 22 will be particularly helpful later, for clarifying the complexity of the membership problems for ClearYes, ClearNo, and Unclear. We can now note that it implies that $\mathsf{dist}(p(m)) = \mathsf{distance}(p(m), \mathsf{INC})$ can be computed in polynomial time (once we recall the efficient constructability of INC, shown in Proposition 13 and its proof).

6.3. *Distance to* INC*, and the period* $\Delta_{\textsc{n}}$

Our previous reasoning allows us to derive further useful consequences for the function $\mathsf{dist}(p(m))$, including the exponentially bounded periodicity of the set $\{m \mid \mathsf{dist}(p(m)) = \omega\}$ (for any fixed p).

Convention. In the rest of the paper we will derive the existence of several polynomials $\textsc{poly}_i : \mathbb{N} \to \mathbb{N}$, in particular

$\textsc{poly}_0(n) \in O(n^3)$, $\textsc{poly}'_0(n) \in O(n^2)$ in Proposition 23,
$\textsc{poly}_1(n) \in O(n^4)$ in Proposition 26,
$\textsc{poly}_2(n) \in O(n^8)$ in Proposition 31,
$\textsc{poly}_3$ in Proposition 38.

Their concrete form will be left implicit (as well as the degree of $\textsc{poly}_3$) but we will assume that such polynomials are fixed, and whenever we refer to one of them, we mean the respective fixed polynomial. We will later relate $\textsc{poly}_1(\textsc{n})$ and $\textsc{poly}_2(\textsc{n})$ to the belt-thickness and to the initial space in Fig. 5.

We now show a set of linear equations $x = \sigma_1 m + \sigma_2$ (where σ_1, σ_2 are rational constants) such that any finite $\mathsf{dist}(p(m))$ must satisfy one of them. (Recall the shortest path to INC sketched in Fig. 3.)

Proposition 23. *There are polynomials* $\textsc{poly}_0(n) \in O(n^3)$ *and* $\textsc{poly}'_0(n) \in O(n^2)$ *such that the following holds. Given a ROCA* $\mathcal{A} = (Q, \Sigma, \delta)$*, with* $|Q| = \textsc{n}$*, if* $p(m) \longrightarrow^* \mathsf{INC}$ *then*

$$\mathsf{dist}(p(m)) = c_1 + d_1 \frac{m + c_2}{d_2} \tag{2}$$

where $d_1 \in [0, \textsc{n}]$*,* $d_2 \in [1, \textsc{n}]$*,* $c_1 \in [0, \textsc{poly}_0(\textsc{n})]$*,* $c_2 \in [-\textsc{poly}'_0(\textsc{n}), \textsc{poly}'_0(\textsc{n})]$*.*

Proof. Suppose that $p_0(m_0) \xrightarrow{a_1} p_1(m_1) \xrightarrow{a_2} \cdots \xrightarrow{a_\ell} p_\ell(m_\ell)$ is a shortest path from $p_0(m_0)$ to INC; hence $p_\ell(m_\ell) \in \mathsf{INC}$ and thus $m_\ell < \textsc{n}$. The path obviously never visits a configuration twice, and each subpath of this path is a shortest path from its start to its end. By using Proposition 18 we derive that $m_j < \max\{m_0, \textsc{n}\} + \textsc{n}^2$ for all $j \in [0, \ell]$.

If $m_0 < \textsc{n} + \textsc{n}^2$ then $p_i(m_i) \in Q \times [0, \textsc{n}{+}2\textsc{n}^2{-}1]$ for all $i \in [0, \ell]$, and thus $\ell < \textsc{n} \cdot (\textsc{n} + 2\textsc{n}^2)$. We can put $c_1 = \ell$ and $d_1 = 0$ in (2); here d_2, c_2 are irrelevant, and we can consider $d_2 = 1$, $c_2 = 0$.

Assume now $m_0 \geq \textsc{n} + \textsc{n}^2$, and let i_0 be the smallest such that $m_{i_0} = \textsc{n} - 1$; we note that $p_i(m_i) \in Q \times [0, \textsc{n}{+}\textsc{n}^2{-}1]$ for all $i \in [i_0, \ell]$, and thus $\ell - i_0 < \textsc{n} \cdot (\textsc{n} + \textsc{n}^2)$. The (positive) path $p_0(m_0) \xrightarrow{a_1} p_1(m_1) \xrightarrow{a_2} \cdots \xrightarrow{a_{i_0}} p_{i_0}(m_{i_0})$ can be assumed to be in the form guaranteed by Proposition 20, where $a_1 a_2 \ldots a_{i_0} = v_1 (v_2)^i v_3$ for the appropriate v_1, v_2, v_3 and $i > 0$. Hence $i_0 = |v_1 v_3| + |v_2| \cdot \frac{m_0 + c - (\textsc{n}-1)}{d}$ where d is the drop of the cycle v_2 and c is the counter change caused by $v_1 v_3$. Since $|v_1 v_3| < \textsc{n}^2$, and thus $c \in [-(\textsc{n}^2{-}1), \textsc{n}^2{-}1]$, and $|v_2| \leq \textsc{n}$, $d \in [1, \textsc{n}]$, we are done: in (2) we put $c_1 = |v_1 v_3| + (\ell - i_0)$, $d_1 = |v_2|$, $d_2 = d$, $c_2 = c - (\textsc{n} - 1)$. We thus have $c_1 \in [0, \textsc{n}^2{-}1 + \textsc{n} \cdot (\textsc{n}{+}\textsc{n}^2) - 1]$, $d_1 \in [1, \textsc{n}]$, $d_2 \in [1, \textsc{n}]$, $c_2 \in [-(\textsc{n}^2{-}1) - (\textsc{n}{-}1), (\textsc{n}^2{-}1) - (\textsc{n}{-}1)]$. □

The reasoning in the proof of Proposition 23 has further consequences. Informally speaking, the next proposition shows that the set $\{\, m \mid p(m) \longrightarrow^* \mathsf{INC} \,\}$ is "dense" if it is not a small finite set. The set $\{\, m \mid p(m) \not\longrightarrow^* \mathsf{INC} \,\}$ might be not "dense", but it is "periodic". Any number that is a multiple of drops of simple cycles of the relevant ROCA $\mathcal{A}$ can serve as a period but we use

$$\Delta_{\textsc{n}} \text{ defined as } \Delta_{\textsc{n}} = \textsc{n}! = \textsc{n} \cdot (\textsc{n}{-}1) \cdot (\textsc{n}{-}2) \cdot \cdots \cdot 2 \cdot 1$$

at our level of analysis. (See also Remark after Proposition 24.)

Proposition 24. *Assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = \textsc{n}$, and a configuration $p(m)$ such that $m \geq \textsc{n} + \textsc{n}^2$.*

1. *If $\mathsf{dist}(p(m)) < \omega$ then there is $d \in [1, \textsc{n}]$ such that $\mathsf{dist}(p(m{+}jd)) < \omega$ for all $j \in \mathbb{Z}$ satisfying $m + jd \geq \textsc{n} + \textsc{n}^2$.*
2. *We have $\mathsf{dist}(p(m)) = \omega$ iff $\mathsf{dist}(p(m{+}\Delta_{\textsc{n}})) = \omega$ (for $m \geq \textsc{n} + \textsc{n}^2$).*

Proof. Point 1. A shortest path from $p(m)$ to INC, where $m \geq \textsc{n} + \textsc{n}^2$, starts with a positive path $p(m) \xrightarrow{v_1}_{+} p'(m') \xrightarrow{v_2}_{+} p'(m' - d) \xrightarrow{v_2}_{+} p'(m' - 2d) \xrightarrow{v_2}_{+} \cdots \xrightarrow{v_2}_{+} p'(m' - id) \xrightarrow{v_3}_{+} r(\textsc{n}{-}1)$ (for some $p', r \in Q$ and $m' \in \mathbb{N}$), as discussed in the proof of Proposition 23; here d is the drop of the cycle v_2. It is clear that $p(m{+}jd) \xrightarrow{v_1(v_2)^{i+j}v_3} r(\textsc{n}{-}1)$ whenever $i + j > 0$. Since $r(\textsc{n}{-}1) \longrightarrow^{*} \mathsf{INC}$, we are done.

Point 2. If $m \geq \textsc{n} + \textsc{n}^2$ then Point 1 implies that $p(m) \longrightarrow^{*} \mathsf{INC}$ iff $p(m{+}\Delta_{\textsc{n}}) \longrightarrow^{*} \mathsf{INC}$; this follows from the fact that $m = (m + \Delta_{\textsc{n}}) - \frac{\Delta_{\textsc{n}}}{d} d$ and $\Delta_{\textsc{n}}$ is divisible by any $d \in [1, \textsc{n}]$. □

Remark. We have chosen $\Delta_{\textsc{n}} = \textsc{n}! \leq \textsc{n}^{\textsc{n}} = 2^{\textsc{n} \log \textsc{n}}$; though $\Delta_{\textsc{n}}$ is exponential in $\textsc{n}$, it can be written in $O(\textsc{n} \log \textsc{n})$ bits. In more detail, we could specify $\Delta_{\mathcal{A}}$ as the least common multiple of simple cycle drops in $\mathcal{A}$. But this number is also exponential in the worst case (as shown by creating separate cycles whose drops are pairwise different primes); therefore we use simply $\Delta_{\textsc{n}} = \textsc{n}!$ at our level of complexity analysis. We note that an upper bound finer than $\textsc{n}!$ is recalled from number theory in Lemma 1 in [27].

6.4. ClearYes *is periodic and* Unclear *is inside belts*

We aim to make precise the periodicity of $\mathsf{ClearYes}$; recall that for a ROCA with $\textsc{n}$ control states we have $\mathsf{ClearYes} = \{(p(m), q(n)) \mid \mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) = \omega$ and $p(m)) \sim_{\textsc{n}} q(n)\}$.

Proposition 25. *Assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = \textsc{n}$. If $m, n \geq \textsc{n} + \textsc{n}^2$ then $(p(m), q(n)) \in \mathsf{ClearYes}$ iff $(p(m{+}i\Delta_{\textsc{n}}), q(n{+}j\Delta_{\textsc{n}})) \in \mathsf{ClearYes}$ for all $i, j \in \mathbb{N}$.*

Proof. If $m, n \geq \textsc{n}$ then $p(m) \sim_{\textsc{n}} q(n)$ iff $p \sim_{\textsc{n}} q$ (since $p(m) \sim_{\textsc{n}} p$ and $q(n) \sim_{\textsc{n}} q$). For $m, n \geq \textsc{n}{+}\textsc{n}^2$ we have $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) = \omega$ iff $\mathsf{dist}(p(m{+}i\Delta)) = \mathsf{dist}(q(n{+}j\Delta)) = \omega$ (for all $i, j \in \mathbb{N}$), by Proposition 24(2). □

When discussing Fig. 3, we mentioned informally that a constraint $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) < \omega$ imposes a linear relation between m and n. This is formalized in the next proposition, which implies that $\mathsf{Unclear}$ resides in polynomially belts with polynomial (vertical) thickness.

Proposition 26. *There is a polynomial $\textsc{poly}_1(n) \in O(n^4)$ such that the following holds. If, for a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ with $|Q| = \textsc{n}$, we have $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) < \omega$ then for some $\alpha, \beta \in [1, \textsc{n}^2]$ we have*

$$|n - \tfrac{\alpha}{\beta} m| < \textsc{poly}_1(\textsc{n}).$$

Proof. Assume $\mathsf{dist}(p(m)) = \mathsf{dist}(q(n)) < \omega$. When expressing $\mathsf{dist}(p(m)) = c_1 + d_1 \frac{m + c_2}{d_2}$ and $\mathsf{dist}(q(n)) = c'_1 + d'_1 \frac{n + c'_2}{d'_2}$ as in (2) in Proposition 23, we get $c_1 + d_1 \frac{m + c_2}{d_2} = c'_1 + d'_1 \frac{n + c'_2}{d'_2}$. If $d_1 > 0$ and $d'_1 > 0$ then we (multiply both sides by $\frac{d'_2}{d'_1}$ and) derive

$$n = \frac{d'_2 d_1}{d'_1 d_2} m + \left(\frac{d'_2 c_1}{d'_1} + \frac{d'_2 d_1 c_2}{d'_1 d_2} - \frac{d'_2 c'_1}{d'_1} - c'_2 \right) = \frac{\alpha}{\beta} m + \rho$$

where $\alpha, \beta \in [1, \textsc{n}^2]$ and $|\rho| \leq \textsc{n} \cdot \textsc{poly}_0(\textsc{n}) + \textsc{n}^2 \cdot \textsc{poly}'_0(\textsc{n}) + \textsc{poly}'_0(\textsc{n})$, and thus $|\rho| = |n - \frac{\alpha}{\beta} m| \in O(\textsc{n}^4)$ (since $\textsc{poly}_0(\textsc{n}) \in O(\textsc{n}^3)$ and $\textsc{poly}'_0(\textsc{n}) \in O(\textsc{n}^2)$); we note that ρ is a rational number such that $\beta\rho$ is an integer.

If $d_1 = 0$ or $d'_1 = 0$ then $\mathsf{dist}(p(m) = \mathsf{dist}(q(n)) \leq \textsc{poly}_0(\textsc{n})$, and thus $m < \textsc{n} + \textsc{poly}_0(\textsc{n})$ and $n < \textsc{n} + \textsc{poly}_0(\textsc{n})$ (since $\mathsf{dist}(r(k)) = \mathsf{distance}(r(k), \mathsf{INC}) > r - \textsc{n}$). We can put $\alpha = \beta = 1$ and note that $|n - \frac{\alpha}{\beta} m| < \textsc{n} + \textsc{poly}_0(\textsc{n})$. □

Definition 27. Assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ where $|Q| = \textsc{n}$. By a *belt B* given by its slope $\frac{\alpha}{\beta}$ where $\alpha, \beta \in [1, \textsc{n}^2]$ we mean the set $\{(p(m), q(n)) \mid p, q \in Q, m, n \in \mathbb{N}, |n - \frac{\alpha}{\beta} m| < \textsc{poly}_1(\textsc{n})\}$. By BeltSpace we mean the union of all belts.

Hence Proposition 26 implies that the set $\mathsf{Unclear} = \bigcup_{i=0}^{\infty} \mathsf{EFD}_i$ is a subset of BeltSpace. We can now also note that the vertical thickness of the belts in Fig. 5 is $2 \cdot \textsc{poly}_1(\textsc{n})$.

The next fact is not needed for the analysis of Alg-Bisim but we note it for later use; as expected, the BeltSpace-*membership problem* asks if $(p(m), q(n)) \in$ BeltSpace when given a ROCA $\mathcal{A}$ and $p(m)$, $q(n)$ (where m, n are presented in binary).

Proposition 28. *The* BeltSpace-*membership problem is in* L.

Proof. The membership is determined by m, n (the control states are irrelevant). We have to check if there are $\alpha, \beta \in [1, \textsc{n}^2]$ such that $|n - \frac{\alpha}{\beta} m| < \textsc{poly}_1(\textsc{n})$, i.e., either $\beta n \geq \alpha m$ and $\beta n - \alpha m < \beta \cdot \textsc{poly}_1(\textsc{n})$, or $\beta n < \alpha m$ and $\alpha m - \beta n < \beta \cdot \textsc{poly}_1(\textsc{n})$. It is a routine to show that this can be done in logarithmic space (recalling Proposition 21). □

6.5. Alg-Bisim *works in polynomial space*

As the first step of our complexity analysis, we explicitly recall the *locality* of checking the bisimulation conditions in $\mathcal{T}(\mathcal{A})$, where $\mathcal{A} = (Q, \Sigma, \delta)$ is a ROCA; the locality follows from the fact that the counter value can change by at most one in one step. For $p, q \in Q$ and $m, n \in \mathbb{N}$ we define the *neighbourhood*

$$\mathsf{Neigh}(p(m), q(n)) = \{(p'(m'), q'(n')) \mid p', q' \in Q, |m' - m| \leq 1, |n' - n| \leq 1\}.$$

Proposition 29. *For a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$, a pair $(p(m), q(n))$ is covered by $R \subseteq (Q \times \mathbb{N}) \times (Q \times \mathbb{N})$ in $\mathcal{T}(\mathcal{A})$ iff it is covered by $R \cap \mathsf{Neigh}(p(m), q(n))$.*

It is this locality which allows us to restrict to $R_{i-2} \cup R_{i-1} \cup R_i$ in 2(c)ii in Alg-Bisim.

We now recall that Alg-Bisim also uses procedures for solving the membership problems for ClearYes, ClearNo, and $\mathsf{Unclear} = \bigcup_{i=0}^{\infty} \mathsf{EFD}_i$; though polynomial-space upper bounds would suffice here, we show better bounds in the next proposition; we also include the deterministic case for later use. An instance of the membership problem for ClearYes is a ROCA and two configurations $p(m)$, $q(n)$ (where m, n are presented in binary); similarly for ClearNo and Unclear.

Proposition 30.

1. *The membership problems for* ClearYes, ClearNo, *and* Unclear *are in* P. *When restricted to det-ROCA, the problems are* NL-*complete.*

2. *Given a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ and $i \in \mathbb{N}$ (in binary), the set* EFD_i *can be computed in polynomial time.*

Proof. We consider a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ where $|Q| = \textsc{n}$. First we note that deciding if $p(m) \sim_{k+1} q(n)$ is straightforward once we construct the set $\mathsf{Neigh}(p(m), q(n)) \cap \sim_k$ (due to the locality). This makes clear that deciding if $p(m) \sim_{\textsc{n}} q(n)$ can be done in time bounded by a polynomial (in the size of $\mathcal{A}$). In the deterministic case, deciding $p(m) \nsim_{\textsc{n}} q(n)$ is obviously in NL (we just stepwise guess a word no longer than $\textsc{n}$ that is enabled in precisely one of $p(m), q(n)$), and we recall that NL =co-NL.

Since we can construct INC in polynomial time (recall the proof of Proposition 13), $\mathsf{dist}(p(m))$ is computable in polynomial time (as follows from Proposition 22). It is thus clear that there is a polynomial-time procedure deciding to which of the sets $\mathsf{ClearYes}$, $\mathsf{ClearNo}$, and $\mathsf{Unclear}$ a given pair $(p(m), q(n))$ belongs. Propositions 13 and 22 also show that the membership problems for $\mathsf{ClearYes}$, $\mathsf{ClearNo}$, and $\mathsf{Unclear}$ are NL-complete in the deterministic case.

Since all elements of EFD_i, for any fixed i, are in BeltSpace, their number is bounded by a polynomial in $\textsc{n}$, and EFD_i can be constructed in polynomial time, w.r.t. the size of $\mathcal{A}$ and the length of the binary presentation of i (recall Proposition 26). □

To finish the description of Alg-Bisim, we need to specify the exponential bound ExpB (computed in 2(a)). To this end we introduce a polynomial $\textsc{poly}_2$; the value $\textsc{poly}_2(\textsc{n})$ will bound the initial space in Fig. 5. It is chosen so that it guarantees that the neighbourhood of any "point" in a belt to the right of the initial space does not intersect any other belt, and the background in the neighbourhood guarantees the periodicity of $\mathsf{ClearYes}$ as captured by Proposition 25. Technically, we recall Proposition 26, yielding the polynomial $\textsc{poly}_1(n) \in O(n^4)$, and we fix $\textsc{poly}_2$ by the next proposition:

Proposition 31. *There is a polynomial $\textsc{poly}_2(n) \in O(n^8)$ satisfying the following conditions for any $n \in \mathbb{N}$ and $\alpha, \beta, \alpha', \beta' \in [1, n^2]$, where we write X instead of $\textsc{poly}_2(n)$:*

1. $\frac{\alpha}{\beta} X - \textsc{poly}_1(n) - 1 > n + n^2$;
2. *if* $\frac{\alpha'}{\beta'} < \frac{\alpha}{\beta}$ *then* $\frac{\alpha}{\beta} X - \textsc{poly}_1(n) - 2 > \frac{\alpha'}{\beta'} X + \textsc{poly}_1(n)$.

Proof. We can rewrite 1. as $X > \frac{\beta}{\alpha} \cdot (\textsc{poly}_1(n) + 1 + n + n^2)$, and 2. as $X > \frac{\beta\beta'}{\alpha\beta' - \alpha'\beta} \cdot (2 \cdot \textsc{poly}_1(n) + 2)$. Since $\frac{\beta\beta'}{\alpha\beta' - \alpha'\beta} \leq n^4$ and $\textsc{poly}_1(n) \in O(n^4)$, the claim is clear. □

Corollary 32. *Assume a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ where $|Q| = \textsc{n}$. If $m > \textsc{poly}_2(\textsc{n})$, $\alpha, \beta \in [1, \textsc{n}^2]$, and $|n - \frac{\alpha}{\beta} m| < \textsc{poly}_1(\textsc{n})$ then for any $(p'(m'), q'(n')) \in \mathsf{Neigh}(p(m), q(n))$ we have*

1. $m' \geq \textsc{n} + \textsc{n}^2$ *and* $n' \geq \textsc{n} + \textsc{n}^2$;
2. *if* $|n' - \frac{\alpha'}{\beta'} m'| < \textsc{poly}_1(\textsc{n})$ *for* $\alpha', \beta' \in [1, \textsc{n}^2]$ *then* $\frac{\alpha'}{\beta'} = \frac{\alpha}{\beta}$.

For each ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ where $|Q| = \textsc{n}$ we put

$$\textsc{ExpB} = \textsc{poly}_2(\textsc{n}) + 1 + (\Delta_{\textsc{n}})^3 \cdot 2^{4\textsc{n}^2 \cdot \textsc{poly}_1(\textsc{n})}. \tag{3}$$

Remark. It would suffice to replace $\textsc{poly}_2(\textsc{n})$ in (3) with $\max\{0, \textsc{poly}_2(\textsc{n}) - m_0\}$. We simply want to guarantee that the "window" in Fig. 6 moves far enough to the right of the initial space to ensure a convenient repeat in each belt (whose simplified version is sketched in Fig. 7).

We note that it suffices for Alg-Bisim to always have just current R_{i-2}, R_{i-1}, R_i in memory (a subset of the vertical belt-cuts of the "window" in Fig. 6, where the numbers are presented in binary). Hence the next lemma is now clear.

Lemma 33. Alg-Bisim *can be implemented to run in polynomial space.*

6.6. *Correctness of* Alg-Bisim

We now show that Alg-Bisim indeed decides the bisimilarity problem for ROCA. One direction is easy:

Proposition 34. *If the input satisfies* $p_0(m_0) \sim q_0(n_0)$ *then there is a computation of* Alg-Bisim *that returns YES.*

Proof. If $p_0(m_0) \sim q_0(n_0)$ then either $(p_0(m_0), q_0(n_0)) \in \mathsf{ClearYes}$ or $(p_0(m_0), q_0(n_0)) \in \mathsf{EFD}_{m_0}$. The former case is clear, so we assume the latter. If we always choose $R_i = \mathsf{EFD}_i \cap \sim$ in 2(c)i then we cannot fail in 2(c)ii: it is sufficient to consider just R_{i-2}, R_{i-1}, R_i since any $(p(m), q(n)) \in\sim$ is covered by $\mathsf{Neigh}(p(m), q(n)) \cap \sim$ (due to the locality captured by Proposition 29). □

For the other direction we also use another aspect of the locality, following from the fact that transitions $p(m) \xrightarrow{a} p'(m+j)$ are independent of the concrete value m when the value is positive. Informally, if $(p(m), q(n))$ is covered by R and the "shift" $(m', n') \rightsquigarrow (m'+z_1, n'+z_2)$ (by a "shift-vector" (z_1, z_2)) maps each element of R in $\mathsf{Neigh}(p(m), q(n))$ to an element of R (in $\mathsf{Neigh}(p(m+z_1), q(n+z_2))$) then the assumption that R covers $(p(m), q(n))$ implies that R covers $(p(m+z_1), q(n+z_2))$.

Proposition 35. *Assume a ROCA* $\mathcal{A} = (Q, \Sigma, \delta)$ *and a set* $R \subseteq (Q\times\mathbb{N})\times(Q\times\mathbb{N})$*. Let all* $m, n, m+z_1, n+z_2$ *be positive, where* $m, n \in \mathbb{N}$ *and* $z_1, z_2 \in \mathbb{Z}$*, and assume that for each* $(p'(m'), q'(n')) \in \mathsf{Neigh}(p(m), q(n))$ *we have that* $(p'(m'), q'(n')) \in R$ *implies* $(p'(m'+z_1), q'(n'+z_2)) \in R$*. If* R *covers* $(p(m), q(n))$ *then* R *also covers* $(p(m+z_1), q(n+z_2))$*.*

Proof. Let the assumptions hold and let R cover $(p(m), q(n))$. Consider a transition $p(m+z_1) \xrightarrow{a} p'(m+z_1+j)$. Since there is also the transition $p(m) \xrightarrow{a} p'(m+j)$ we must have $q(n) \xrightarrow{a} q'(n+j')$ such that $(p'(m+j), q'(n+j')) \in R$. Since $(p'(m+j), q'(n+j')) \in \mathsf{Neigh}(p(m), q(n)))$, we have $(p'(m+z_1+j), q'(n+z_2+j')) \in R$, and $p(m+z_1) \xrightarrow{a} p'(m+z_1+j)$ can be thus "matched" by $q(n+z_2) \xrightarrow{a} q'(n+z_2+j')$. For any transition $q(n+z_2) \xrightarrow{a} q'(n+z_2+j')$ we deduce a matching transition $p(m+z_1) \xrightarrow{a} p'(m+z_1+j)$ analogously. □

Lemma 36. *Given a ROCA* $\mathcal{A}$ *and two configurations* $p_0(m_0)$*,* $q_0(n_0)$*, there is a computation of* Alg-Bisim *returning YES (for the input* $\mathcal{A}$*,* $p_0(m_0)$*,* $q_0(n_0)$*) if and only if* $p_0(m_0) \sim q_0(n_0)$*.*

Proof. The "if" part was shown by Proposition 34. To show the "only if" part, let us consider a computation returning YES, for the input $\mathcal{A}$, $p_0(m_0)$, $q_0(n_0)$. If $(p_0(m_0), q_0(n_0)) \in \mathsf{ClearYes}$ then we are done, since $\mathsf{ClearYes} \subseteq\sim$; we thus assume $(p_0(m_0), q_0(n_0)) \in \mathsf{EFD}_{m_0}$. Let $R_0, R_1, \ldots, R_{m_0+\textsc{ExpB}}$ be the sets chosen by the computation in 2(c)i; hence $(p_0(m_0), q_0(n_0)) \in R_{m_0}$.

We now show that there is a bisimulation containing the sets $R_0, R_1, \ldots, R_{m_0}$ (while it might not contain all R_{m_0+j} for $j > 0$); the proof will be thus finished.

We assume that $\mathcal{A} = (Q, \Sigma, \delta)$ where $|Q| = \textsc{n}$, and consider the periodic sequence $i_0 < i_1 < \cdots < i_\ell$ where $i_0 = 1+\max\{m_0, \textsc{poly}_2(\textsc{n})\}$, $i_{j+1} = i_j + (\Delta_{\textsc{n}})^3$ for all $j \in [0, \ell-1]$, and $\ell = 2^{4\textsc{n}^2\cdot\textsc{poly}_1(\textsc{n})}$. The definition (3) guarantees that $i_\ell \leq m_0 + \textsc{ExpB}$, and thus R_i is defined for all $i \leq i_\ell$.

Let us now consider a concrete belt B, given by its slope $\frac{\alpha}{\beta}$ where $\alpha, \beta \in [1, \textsc{n}^2]$. Recall Fig. 7 for the idea of a "width-1 cut" repeat; we now derive a "width-2 cut" repeat (which is needed for a consistent periodic filling of B described later). We say that *a pair* (i, i'), where $i = i_{j_1}$ and $i' = i_{j_2}$ for $1 \leq j_1 < j_2 \leq \ell$, is a *repeat* (of a width-2 B-cut) if the following holds:

> for any $p, q \in Q$, any $m \in \{i, i+1\}$, and any n such that $|n - \frac{\alpha}{\beta}m| < \textsc{poly}_1(\textsc{n})$, if we put $m' = m + (i' - i)(\Delta_{\textsc{n}})^3$ and $n' = n + \frac{\alpha}{\beta}(i' - i)(\Delta_{\textsc{n}})^3$ then $(p(m), q(n)) \in R_m$ iff $(p(m'), q(n')) \in R_{m'}$.

We note that $\frac{\alpha}{\beta}(\Delta_{\textsc{n}})^3$ is a multiple of $\Delta_{\textsc{n}}$ since $\beta \in [1, \textsc{n}^2]$ and $\Delta_{\textsc{n}} = \textsc{n}!$. Thus also $(p(m), q(n)) \in \mathsf{ClearYes}$ iff $(p(m'), q(n')) \in \mathsf{ClearYes}$ (for m, n, m', n' as above); here we use Proposition 25 and Corollary 32(1).

For each $i \geq i_0$, the sets $\{(p, q, m, n) \mid p, q \in Q, m \in \{i, i+1\}, |n - \frac{\alpha}{\beta}m| < \textsc{poly}_1(\textsc{n})\}$ and $\{(p, q, m, n) \mid p, q \in Q, m \in \{i + (\Delta_{\textsc{n}})^3, i + (\Delta_{\textsc{n}})^3 + 1\}, |n - \frac{\alpha}{\beta}m| < \textsc{poly}_1(\textsc{n})\}$ have the same number of elements that is bounded by $\textsc{n}^2 \cdot 2 \cdot 2 \cdot \textsc{poly}_1(\textsc{n})$. We thus easily deduce that our choice $\ell = 2^{4\textsc{n}^2 \cdot \textsc{poly}_1(\textsc{n})}$ and the pigeonhole principle guarantee that there is a repeat (i, i'), where $i = i_{j_1} < i_{j_2} = i'$ for some $j_1, j_2 \in [0, \ell]$; let us fix such a repeat (i, i').

Informally speaking, we now "fill the belt B after i'" periodically, with the period $i' - i = (j_2 - j_1) \cdot (\Delta_{\textsc{n}})^3$. Formally we define the sets R^B_j for $j = \textsc{poly}_2(\textsc{n}) + 1, \textsc{poly}_2(\textsc{n}) + 2, \ldots$ inductively as follows:

1. If $j \in [\textsc{poly}_2(\textsc{n}){+}1, i']$, and n satisfies $|n - \frac{\alpha}{\beta}j| < \textsc{poly}_1(\textsc{n})$, and $(p(j), q(n)) \in R_j$ then $(p(j), q(n)) \in R^B_j$. (Here R^B_j is the intersection of R_j with the belt B.)
2. If $j > i'$, and n satisfies $|n - \frac{\alpha}{\beta}j| < \textsc{poly}_1(\textsc{n})$, and $(p(j - (i' - i)), q(n - \frac{\alpha}{\beta}(i' - i))) \in R^B_{j-(i'-i)}$ then $(p(j), q(n)) \in R^B_j$. (Here R^B_j can be viewed as the "shift" of $R^B_{j-(i'-i)}$ by the vector $(i' - i, \frac{\alpha}{\beta}(i' - i))$.)

We now show inductively that R^B_j is covered by $R_{\textsc{poly}_2(\textsc{n})} \cup R^B_j \cup R^B_{j+1} \cup \mathsf{ClearYes}$ when $j = \textsc{poly}_2(\textsc{n}) + 1$, and that R^B_j is covered by $R^B_{j-1} \cup R^B_j \cup R^B_{j+1} \cup \mathsf{ClearYes}$ for each $j > \textsc{poly}_2(\textsc{n}) + 1$. For each $j \in [\textsc{poly}_2(\textsc{n}) + 1, i' - 1]$ the claim is true since the considered run of Alg-Bisim is successful: by Corollary 32(2) the neighbourhoods of the "points" in the belt B outside the initial space do not intersect other belts, hence covering of R^B_j by $R_{j-1} \cup R_j \cup R_{j+1} \cup \mathsf{ClearYes}$ implies covering of R^B_j by $R^B_{j-1} \cup R^B_j \cup R^B_{j+1} \cup \mathsf{ClearYes}$ (using the locality captured in Proposition 29).

For each $j \geq i'$ the claim follows from the validity of the claim for $j' = j - (i' - i)$: by Corollary 32(1) we can use the periodicity of ClearYes captured in Proposition 25 ($(p(m), q(n)) \in \mathsf{ClearYes}$ implies $(p(m + (i' - i)), q(n + \frac{\alpha}{\beta}(i' - i))) \in \mathsf{ClearYes}$ since both $(i' - i)$ and $\frac{\alpha}{\beta}(i' - i)$ are multiples of $\Delta_{\textsc{n}}$), and we also use the periodicity of our belt filling, and the "shifted" locality captured by Proposition 35.

We put $R_{belt-B} = \bigcup_{j=\textsc{poly}_2(\textsc{n})+1}^{\infty} R^B_j$, and note that R_{belt-B} is covered by $R_{\textsc{poly}_2(\textsc{n})} \cup R_{belt-B} \cup \mathsf{ClearYes}$.

We proceed similarly for all belts (i.e., for all slopes $\frac{\alpha}{\beta}$ where $\alpha, \beta \in [1, \textsc{n}^2]$), and define R_{belts} as the union of the sets R_{belt-B} for all belts B.

Now we deduce that $R = R_0 \cup R_1 \cup \cdots \cup R_{\textsc{poly}_2(\textsc{n})} \cup R_{belts}$ is covered by $R \cup \sim$, and we invoke Proposition 7. Since $R_{belts} \cap \{(p(j), q(n)) \mid p, q \in Q, n \in \mathbb{N}\}$ coincides with R_j for all $j \in [\textsc{poly}_2(\textsc{n}) + 1, m_0]$ (when $m_0 > \textsc{poly}_2(\textsc{n})$), there is a bisimulation containing $R_0, R_1, \ldots, R_{m_0}$, and thus $p_0(m_0) \sim q_0(n_0)$. □

Lemmas 33 and 36 prove the upper bound in Theorem 3 (stated in Section 2).

6.7. *Effective semilinearity of $\sim$ (Theorem 4)*

Theorem 4 can be now verified in a straightforward way. We do not give all tedious technical details but we give the main ideas, based on the previous analysis of Alg-Bisim. First we note that we can now assume that Alg-Bisim is adjusted so that it always chooses $R_i = \mathsf{EFD}_i \cap \sim$; we have shown that the membership in $\sim$ can be decided in polynomial space. In this case, for $R = R_0 \cup R_1 \cup \cdots \cup R_{\text{POLY}_2(\text{N})} \cup R_{belts}$ (defined as in the proof of Lemma 36) we have $R \cup \mathsf{ClearYes} = \sim$, as we now show. Suppose it is not the case. Then for a belt B, given by its slope $\frac{\alpha}{\beta}$, and for the (first) respective repeat (i, i') we would have $p'(m') \sim q'(n')$ for some $m' > i'$ where $(p'(m'), q'(n')) \in \mathsf{EFD}_{m'} \cap B$ though $(p'(m'), q'(n')) \notin R_{belt-B}$; suppose m' is the smallest possible. We now derive a contradiction by using a "shift of $\sim$" by the vector $(-(i'-i)), -\frac{\alpha}{\beta}(i'-i))$ (that is opposite to the vector used for the inductive construction of R_{belt-B}). Let us define $R' = R'_{i+1} \cup R'_{i+2} \cup R'_{i+3} \cup \cdots \subseteq B$ such that $(p(j), q(n)) \in B$ belongs to R'_j (for $j \in \{i{+}1, i{+}2, i{+}3, \ldots\}$) iff $p(j{+}(i'{-}i)) \sim q(n{+}\frac{\alpha}{\beta}(i'{-}i))$. We can now easily check that R' is covered by $R^B_i \cup R' \cup \mathsf{ClearYes}$; hence $R' \subseteq \sim$. But $p'(m' - (i'-i)), q'(n' - \frac{\alpha}{\beta}(i'-i))$ is in $R'_{m'-(i'-i)}$ though it is not in $R^B_{m'-(i'-i)}$; we must surely have $m' - (i'-i) > i'$, and we have thus contradicted that m' was the smallest.

There is surely a procedure producing a formula describing the whole set $\mathsf{ClearYes}$ (based on Proposition 24(2)). We have thus shown that Alg-Bisim can be enhanced to produce a (Presburger) formula describing the whole set $\sim$ if it can remember all constructed $R_0, R_1, R_2, \ldots$, and thus works in exponential space.

It is now a routine to note that the resulting exponential formula can be produced by using only polynomial workspace. The main trick is that the belt-cut repeats (i, i') do not need to be looked for in fully remembered $R_0, R_1, R_2, \ldots$ but they can be nondeterministically guessed and then verified: when processing i, Alg-Bisim guesses that there will be the appropriate i' later (within an exponentially bounded number of steps to be now counted), remembers just the width-2 cut at i, continues with producing the description of the belt-filling until i' where it verifies that (i, i') is indeed a repeat.

7. Bisimilarity is in NL for deterministic ROCA

We recall that $\mathsf{ClearNo} = \{(p(m), q(n)) \mid \mathsf{dist}(p(m)) \neq \mathsf{dist}(q(n)) \text{ or } p(m)) \not\sim_{\text{N}} q(n)\}$, for a (general) ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ where $|Q| = \text{N}$. The fact $\mathsf{ClearNo} \subseteq \not\sim$ can be made more precise:

Proposition 37. *If* $\mathsf{dist}(p(m)) < \mathsf{dist}(q(n))$, *then* $p(m) \not\sim_k q(n)$ *for* $k = \mathsf{dist}(p(m)) + \text{N}$.

Proof. If $\mathsf{dist}(p(m)) < \mathsf{dist}(q(n))$ then $p(m) \xrightarrow{w} p'(m')$ where $|w| = \mathsf{dist}(p(m))$ and $p'(m') \in \mathsf{INC}$. If $p(m) \sim_k q(n)$ for $k = \mathsf{dist}(p(m)) + \text{N}$ then there must be $q'(n')$ such that $q(n) \xrightarrow{w} q'(n')$ and $p'(m') \sim_{\text{N}} q'(n')$. Since $q'(n') \notin \mathsf{INC}$, there is $r \in Q$ (a state in $\mathcal{F}_{\mathcal{A}}$) such that $q'(n') \sim_{\text{N}} r \not\sim_{\text{N}} p'(m')$; this contradicts with $p'(m') \sim_{\text{N}} q'(n')$. □

Let us now consider a *deterministic* ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ generating the deterministic LTS $\mathcal{T}(\mathcal{A}) = (Q \times \mathbb{N}, \Sigma, (\xrightarrow{a})_{a \in \Sigma})$. We note that the LTS $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$, where $(p(m), q(n)) \xrightarrow{a} (p'(m'), q'(n'))$ iff $p(m) \xrightarrow{a} p'(m')$ and $q(n) \xrightarrow{a} q'(n')$, is also deterministic. We observe that $p(m) \not\sim_{k+1} q(n)$ iff there is $w \in \Sigma^*$ of length at most k such that $(p(m), q(n)) \xrightarrow{w} (p'(m'), q'(n'))$ where $p'(m') \not\sim_1 q'(n')$. Hence the question of equivalence in $\mathcal{T}(\mathcal{A})$ reduces to a (specific)

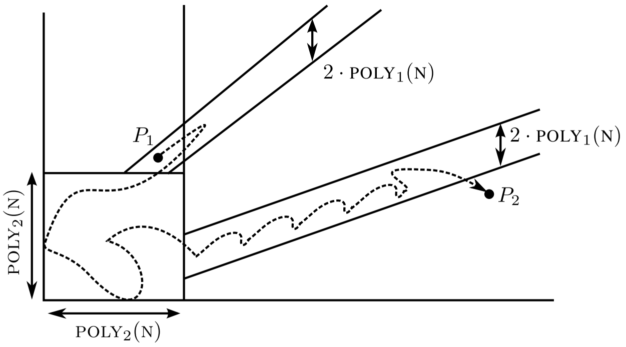
$2\cdot\textsc{poly}_1(\textsc{n})$
P_1
$2\cdot\textsc{poly}_1(\textsc{n})$
P_2
$\textsc{poly}_2(\textsc{n})$
$\textsc{poly}_2(\textsc{n})$

Figure 10: Projection of a path in $\mathcal{T}(\mathcal{A})\times\mathcal{T}(\mathcal{A})$; the start-node is projected to P_1 and the end-node to P_2

reachability question in the deterministic LTS $\mathcal{T}(\mathcal{A})\times\mathcal{T}(\mathcal{A})$. Figure 10 sketches the projection of a path in $\mathcal{T}(\mathcal{A})\times\mathcal{T}(\mathcal{A})$ to $\mathbb{N}\times\mathbb{N}$; here the start-node $(p_1(m_1), q(n_1))$ of the path is projected to the point $P_1 = (m_1, n_1)$, while the end-node $(p_2(m_2), q_2(n_2))$ is projected to the point $P_2 = (m_2, n_2)$. (The figure does not show the third dimension, i.e., the respective pairs of control states are not depicted.)

Remark. We note that the reachability problem in the deterministic LTS $\mathcal{T}(\mathcal{A})\times\mathcal{T}(\mathcal{A})$ is undecidable in general. This follows from the standard fact that the trace inclusion problem, asking if $\forall w\in\Sigma^* : (p(m)\xrightarrow{w}) \Rightarrow (q(n)\xrightarrow{w})$ for a given det-ROCA $\mathcal{A} = (Q,\Sigma,\delta)$ and $p(m), q(n)$, is undecidable; hence the question if $(p_0(m_0), q_0(n_0)) \longrightarrow^* \{(p(m), q(n)) \mid \exists a\in\Sigma : p(m)\xrightarrow{a} \wedge\neg(q(n)\xrightarrow{a})\}$ is undecidable. In contrast, our question if $(p_0(m_0), q_0(n_0)) \longrightarrow^* \{(p(m), q(n)) \mid \exists a\in\Sigma : (p(m)\xrightarrow{a} \wedge\neg(q(n)\xrightarrow{a}))\vee(\neg(p(m)\xrightarrow{a})\wedge q(n)\xrightarrow{a})\}$ is decidable, and even in NL.

The next lemma proves Point 1. in Theorem 5. It shows that if $p_0(m_0)\nsim q_0(n_0)$ for a det-ROCA, where m_0, n_0 are "small" (i.e., bounded by a polynomial) then the "equivalence level", i.e. the maximal k such that $p_0(m_0)\sim_k q_0(n_0)$, is "small".

Remark. This is not true in the case of nondeterministic ROCA. We could use disjoint cycles whose lengths are pairwise different prime numbers to construct a simple example where $p(0)\nsim q(0)$ but $p(0)\sim_k q(0)$ for k being the least common multiple of the cycle lengths.

In a more elegant version of the next lemma we would have $m_0 = n_0 = 0$ but we use a form that is technically convenient later.

Lemma 38. *There is a polynomial* $\textsc{poly}_3$ *with the following property. For any det-ROCA* $\mathcal{A} = (Q,\Sigma,\delta)$ *with* $|Q| = \textsc{n}$, *if* $p_0(m_0)\nsim q_0(n_0)$, *and* $m_0, n_0 \le \textsc{poly}_2(\textsc{n})$ *or* $m_0 \le \textsc{poly}_2(\textsc{n})$ *and* $(p_0(m_0), q_0(n_0)) \in \textsc{BeltSpace}$, *then* $p_0(m_0)\nsim_k q_0(n_0)$ *for* $k = \textsc{poly}_3(\textsc{n})$.

Proof. Let us consider a det-ROCA $\mathcal{A} = (Q,\Sigma,\delta)$ with $|Q| = \textsc{n}$, and suppose $p_0(m_0)\nsim q_0(n_0)$, $m_0 \le \textsc{poly}_2(\textsc{n})$, and $n_0 \le \textsc{poly}_2(\textsc{n})$ or $(p_0(m_0), q_0(n_0)) \in \textsc{BeltSpace}$. It is convenient first to show the existence of a polynomial $\textsc{poly}'_3$ such that $\mathsf{distance}((p_0(m_0), q_0(n_0)), \textsc{Target}) \le \textsc{poly}'_3(\textsc{n})$ where

$$\textsc{Target} =\nsim_{\textsc{n}} \cup (\mathsf{ClearNo} \smallsetminus \textsc{BeltSpace});$$

we will then derive $\textsc{poly}_3$ by using $\textsc{poly}_3'$. Let us thus assume that

$$(p_0(m_0), q_0(n_0)) \xrightarrow{a_1} (p_1(m_1), q_1(n_1)) \xrightarrow{a_2} \cdots \xrightarrow{a_\ell} (p_\ell(m_\ell), q_\ell(n_\ell)) \tag{4}$$

is a shortest path in $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$ such that $(p_\ell(m_\ell), q_\ell(n_\ell)) \in \textsc{Target}$, i.e., $p_\ell(m_\ell) \not\sim_{\textsc{n}} q_\ell(n_\ell)$, or $\mathsf{dist}(p_\ell(m_\ell)) \neq \mathsf{dist}(q_\ell(n_\ell))$ and $|\, n_\ell - \frac{\alpha}{\beta} m_\ell \,| \geq \textsc{poly}_1(\textsc{n})$ for all $\alpha, \beta \in [1, \textsc{n}^2]$. There surely must be such a path, since $p_0(m_0) \not\sim q_0(n_0)$. Fig. 10 might depict such a path, when $(p_0(m_0), q_0(n_0))$ is projected to P_1 and $(p_\ell(m_\ell), q_\ell(n_\ell))$ is projected to P_2.

We note that the path (4) cannot enter $\mathsf{ClearYes}$, so $(p_j(m_j), q_j(n_j)) \in \textsc{BeltSpace}$ for all $j \in [0, \ell{-}1]$. Let us now fix arbitrary $\alpha, \beta \in [1, \textsc{n}^2]$, and consider a maximal "$\frac{\alpha}{\beta}$-segment to the right of $\textsc{poly}_2(\textsc{n})$"; i.e., we consider a subpath of (4) of the form

$$(p_{i_0}(m_{i_0}), q_{i_0}(n_{i_0})) \xrightarrow{a_{i_0+1}} (p_{i_0+1}(m_{i_0+1}), q_{i_0+1}(n_{i_0+1})) \xrightarrow{a_{i_0+2}} \cdots \xrightarrow{a_{i_1}} (p_{i_1}(m_{i_1}), q_{i_1}(n_{i_1})) \tag{5}$$

where $m_{i_0} = \textsc{poly}_2(\textsc{n})$, $m_j > \textsc{poly}_2(\textsc{n})$ for all $j \in [i_0 + 1, i_1]$, and $|\, n_j - \frac{\alpha}{\beta} m_j \,| < \textsc{poly}_1(\textsc{n})$ for all $j \in [i_0, i_1]$; the maximality means that one of the following conditions holds:

1. $i_1 = \ell$, in which case necessarily $p_{i_1}(m_{i_1}) \not\sim_{\textsc{n}} q_{i_1}(n_{i_1})$;
2. $m_{i_1+1} = \textsc{poly}_2(\textsc{n})$, in which case $m_{i_1} = \textsc{poly}_2(\textsc{n}){+}1$ (the segment returns to $\textsc{poly}_2(\textsc{n})$);
3. $(p_{i_1+1}(m_{i_1+1}), q_{i_1+1}(n_{i_1+1}))$ is in $\mathsf{ClearNo} \smallsetminus \textsc{BeltSpace}$.

(In Fig. 10 we can see two such maximal segments, for two different slopes $\frac{\alpha_1}{\beta_1}$, $\frac{\alpha_2}{\beta_2}$.) Since $p(m) \sim_{\textsc{n}} p$ if $m \geq \textsc{n}$, Condition 1 can be rephrased as $p_{i_1} \not\sim_{\textsc{n}} q_{i_1}$ (i.e., $p_{i_1} \not\sim q_{i_1}$). Condition 3 is here equivalent to $|\, n_{i_1+1} - \frac{\alpha}{\beta} m_{i_1+1} \,| \geq \textsc{poly}_1(\textsc{n})$; we have either that $p_{i_1+1} \not\sim_{\textsc{n}} q_{i_1+1}$ or that at least one of the values $\mathsf{dist}(p_{i_1+1}(m_{i_1+1}))$, $\mathsf{dist}(q_{i_1+1}(n_{i_1+1}))$ is finite (which implies that the values differ, since $(p_{i_1+1}(m_{i_1+1}), q_{i_1+1}(n_{i_1+1}))$ is outside $\textsc{BeltSpace}$ and we recall Proposition 26 and Definition 27).

The $\frac{\alpha}{\beta}$-segment (5) can be viewed as a computation of a single ROCA $\mathcal{A}'$, with only positive rules; we can imagine that this ROCA has m_j in the counter, and remembers p_j, q_j and the (rational) offset $(n_j - \frac{\alpha}{\beta} m_j)$ in the control unit. Formally we define $\mathcal{A}' = (Q', \Sigma, \delta')$ where

$$Q' = \{\, (p, q, \rho) \mid p, q \in Q, \text{ and } \rho = n - \tfrac{\alpha}{\beta} m \text{ for some } m, n \in \mathbb{N} \text{ such that } |\, n - \tfrac{\alpha}{\beta} m \,| < \textsc{poly}_1(\textsc{n}) \,\}.$$

We note that there are no more than $\beta \cdot 2 \cdot \textsc{poly}_1(\textsc{n})$ possible values for the rational component ρ; thus the number $|Q'|$ of the control states of $\mathcal{A}'$ is no greater than

$$K = 2 \cdot \textsc{n}^4 \cdot \textsc{poly}_1(\textsc{n}). \tag{6}$$

The rules in δ' are induced by δ as follows:

if $(p, a, 1, p', j_1) \in \delta$ and $(q, a, 1, q', j_2) \in \delta$ then for any possible ρ such that $\rho' = \rho - \frac{\alpha}{\beta} j_1 + j_2$ satisfies $|\rho'| < \textsc{poly}_1(\textsc{n})$ we put $((p, q, \rho), a, 1, (p, q, \rho'), j_1) \in \delta'$.

(Note that $(n + j_2) - \frac{\alpha}{\beta}(m + j_1) = (n - \frac{\alpha}{\beta} m) - \frac{\alpha}{\beta} j_1 + j_2$.)

For technical convenience we also consider $\mathcal{A}'_{\mathsf{rev}} = (Q', \Sigma, \delta'_{\mathsf{rev}})$ working in the opposite direction (simulating (5) from right to left); here δ'_{rev} is induced by δ' as follows:

$$\text{if } ((p, q, \rho), a, 1, (p', q', \rho'), j) \in \delta' \text{ then } ((p', q', \rho'), a, 1, (p, q, \rho), -j) \in \delta'_{\mathsf{rev}}.$$

We can now easily check that the path (5) in $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$ gives rise to the following path in $\mathcal{T}(\mathcal{A}'_{\mathsf{rev}})$:

$$r_{i_1}(m'_{i_1}) \xrightarrow{a_{i_1}} r_{i_1-1}(m'_{i_1-1}) \xrightarrow{a_{i_1-1}} \cdots \xrightarrow{a_{i_0+1}} r_{i_0}(m'_{i_0}) \tag{7}$$

where $m'_j = m_j - m_{i_0}$ and $r_j = (p_j, q_j, n_j - \frac{\alpha}{\beta} m_j)$ for all $j \in [i_0, i_1]$; we have conveniently chosen $m'_{i_0} = m_{i_0} - m_{i_0} = 0$, which is possible since $m_{i_0} = \textsc{poly}_2(\textsc{n})$ and $m_j > \textsc{poly}_2(\textsc{n})$ for all $j \in [i_0+1, i_1]$. (We note that $\frac{\alpha}{\beta} m'_j + \rho_j$, where $\rho_j = n_j - \frac{\alpha}{\beta} m_j$, might be not an integer, but it is convenient that the positive path (7) finishes in a zero configuration.)

We can also easily check that any path from $r_{i_1}(m'_{i_1})$ to $r_{i_0}(0)$ in $\mathcal{T}(\mathcal{A}'_{\mathsf{rev}})$ gives rise to a path (with the same length) from $(p_{i_0}(m_{i_0}), q_{i_0}(n_{i_0}))$ to $(p_{i_1}(m_{i_1}), q_{i_1}(n_{i_1}))$ in $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$.

This implies that (7) is a shortest path from $r_{i_1}(m'_{i_1})$ to $r_{i_0}(0)$ in $\mathcal{T}(\mathcal{A}'_{\mathsf{rev}})$, and that it can be assumed to be in the normal form captured by Proposition 20 if $m'_{i_1} \geq K^2$. (In the lower belt in Fig. 10 we have hinted at this normal form by depicting a repeated "cycle" in the path-segment.)

If $m'_{i_1} < K^2 + \textsc{n} \cdot K$ then the maximal m'_j, $j \in [i_0, i_1]$, is no greater than $2 \cdot K^2 + \textsc{n} \cdot K$ (by Proposition 18). We now assume that $m'_{i_1} \geq K^2 + \textsc{n} \cdot K$, which will be contradicted. The normal form of (7) allows us to assume that the path (5) is of the form

$$(p_{i_0}(m_{i_0}), q_{i_0}(n_{i_0})) \xrightarrow{v_1} (p(m), q(n)) \xrightarrow{v_2} (p(m+D), q(n+\tfrac{\alpha}{\beta}D)) \xrightarrow{v_2} (p(m+2D), q(n+2\tfrac{\alpha}{\beta}D)) \xrightarrow{v_2} \cdots$$
$$\cdots \xrightarrow{v_2} (p(m+xD), q(n+x\tfrac{\alpha}{\beta}D)) \xrightarrow{v_3} (p_{i_1}(m_{i_1}), q_{i_1}(n_{i_1}))$$

where $D \in [1, K]$ and $x \geq \textsc{n}$. We cannot have $p_{i_1}(m_{i_1}) \not\sim_{\textsc{n}} q_{i_1}(n_{i_1})$ (i.e., $p_{i_1} \not\sim q_{i_1}$), since cutting off a cycle leads to a contradiction: our assumptions would yield $p_{i_1} \sim_{\textsc{n}} p_{i_1}(m_{i_1}-D) \sim_{\textsc{n}} p_{i_1}(m_{i_1}) \not\sim_{\textsc{n}} q_{i_1}(n_{i_1}) \sim_{\textsc{n}} q_{i_1}(n_{i_1}-D) \sim_{\textsc{n}} q_{i_1}$, and thus by $(p(m+(x-1)D), q(n+(x-1)\frac{\alpha}{\beta}D)) \xrightarrow{v_3} (p_{i_1}(m_{i_1}-D), q_{i_1}(n_{i_1}-\frac{\alpha}{\beta}D))$ we would reach Target earlier.

Therefore $(p_{i_1+1}(m_{i_1+1}), q_{i_1+1}(n_{i_1+1}))$ is in $\mathsf{ClearNo} \setminus \textsc{BeltSpace}$; we have $|n_{i_1+1} - \frac{\alpha}{\beta} m_{i_1+1}| \geq \textsc{poly}_1(\textsc{n})$ and at least one of $\mathsf{dist}(p_{i_1+1}(m_{i_1+1}))$, $\mathsf{dist}(q_{i_1+1}(n_{i_1+1}))$ is finite (and they are necessarily different). By Proposition 24(1) we deduce that there is $d \in [1, \textsc{n}]$ such that by cutting off d cycles we would keep at least one distance finite and reach Target earlier: we have

$$(p(m+(x-d)D), q(n+(x-d)\tfrac{\alpha}{\beta}D)) \xrightarrow{v_3 a_{i_1+1}} (p_{i_1+1}(m_{i_1+1}-dD), q_{i_1+1}(n_{i_1+1}-\tfrac{\alpha}{\beta}dD))$$

and $m_{i_1+1}-dD > \textsc{poly}_2(\textsc{n})$, $|\,(n_{i_1+1} - \frac{\alpha}{\beta}dD) - \frac{\alpha}{\beta}(m_{i_1+1} - dD)\,| = |\,n_{i_1+1} - \frac{\alpha}{\beta}m_{i_1+1}\,| \geq \textsc{poly}_1(\textsc{n})$, and at least one of $\mathsf{dist}(p_{i_1+1}(m_{i_1+1}-dD))$, $\mathsf{dist}(q_{i_1+1}(n_{i_1+1}-\frac{\alpha}{\beta}dD))$ is finite (and they are different).

We can thus conclude that in the path (4) we have $m_j \leq \textsc{poly}_2(\textsc{n}) + 2 \cdot K^2 + \textsc{n} \cdot K$ and $(p_j(m_j), q_j(n_j)) \in \textsc{BeltSpace}$ for all $j \in [0, \ell-1]$. Since (4) cannot visit a node twice, we surely have

$$\ell \leq (1 + \textsc{poly}_2(\textsc{n}) + 2 \cdot K^2 + \textsc{n} \cdot K) \cdot \textsc{n}^6 \cdot 2 \cdot \textsc{poly}_1(\textsc{n})$$

(where $\textsc{n}^6 = \textsc{n} \cdot \textsc{n} \cdot \textsc{n}^2 \cdot \textsc{n}^2$ accounts for the tuples (p, q, α, β)). We thus get $\textsc{poly}'_3$ such that $\textsc{poly}'_3(\textsc{n})$ bounds the length ℓ of the path (4).

We have $m_\ell \leq m_0 + \textsc{poly}'_3(\textsc{n})$, and $n_\ell \leq n_0 + \textsc{poly}'_3(\textsc{n})$, and we recall that $m_0 \leq \textsc{poly}_2(\textsc{n})$ and $n_0 < \textsc{n}^2 \cdot \textsc{poly}_2(\textsc{n}) + \textsc{poly}_1(\textsc{n})$. If $\mathsf{dist}(p_\ell(m_\ell)) \neq \mathsf{dist}(q_\ell(n_\ell))$ then Proposition 23 implies that $\min\{\mathsf{dist}(p_\ell(m_\ell)), \mathsf{dist}(q_\ell(n_\ell))\} \leq \textsc{n} \cdot (\textsc{n}^2 \cdot \textsc{poly}_2(\textsc{n}) + \textsc{poly}_1(\textsc{n}) + \textsc{poly}'_3(\textsc{n})) + O(\textsc{n}^3)$. By Proposition 37 we thus deduce that $p_0(m_0) \not\sim_k q_0(n_0)$ for $k = \textsc{poly}'_3(\textsc{n}) + \textsc{n} \cdot (\textsc{n}^2 \cdot \textsc{poly}_2(\textsc{n}) + \textsc{poly}_1(\textsc{n}) + \textsc{poly}'_3(\textsc{n})) + O(\textsc{n}^3) + \textsc{n}$. Hence k is indeed bounded by $\textsc{poly}_3(\textsc{n})$ for a polynomial $\textsc{poly}_3$. □

We will now prove the next lemma, recalling that the bisimilarity problem has the instances $\mathcal{A}, p_0(m_0), q_0(n_0)$ where m_0, n_0 are given in binary. The lemma finishes a proof of Theorem 5. (It also applies to language equivalence, by Proposition 2.)

Lemma 39. *The bisimilarity problem is in* NL *for deterministic ROCA.*

Proof. It is sufficient to show that the complement of the trace equivalence problem for det-ROCA is in NL, since NL =co-NL. Let us consider an instance $\mathcal{A} = (Q, \Sigma, \delta)$, $p_0(m_0)$, $q_0(n_0)$ where $|Q| = \textsc{n}$, and assume $p_0(m_0) \nsim q_0(n_0)$.

We recall that the membership for $\textsc{BeltSpace}$ is in L (by Proposition 28) and that the membership problem for $\mathsf{ClearNo}$ is in NL (in our deterministic case, by Proposition 30). It is thus sufficient to explore the case where

$$(p_0(m_0), q_0(n_0)) \in \textsc{BeltSpace} \text{ and } p_0(m_0) \sim_{\textsc{n}} q_0(n_0).$$

The subcase where $m_0 \leq \textsc{poly}_2(\textsc{n})$ is clear by Lemma 38: a (nondeterministic) algorithm can just follow a path $(p_0(m_0), q_0(n_0)) \xrightarrow{a_1} (p_1(m_1), q_1(n_1))) \xrightarrow{a_2} (p_2(m_2), q_2(n_2)) \xrightarrow{a_3} \ldots$ in $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$, where a_i is always guessed and only the current pair $(p_i(m_i), q_i(n_i))$ is kept in memory; at most $\textsc{poly}_3(\textsc{n})$ moves are performed, until some $p_i(m_i) \nsim_1 q_i(n_i)$ is encountered. Here we can present m_i, n_i in the workspace plainly in binary; there is no need to use differences $m_i - m_0$, $n_i - n_0$ since m_0, n_0 are “small”.

We thus further assume that $m_0 > \textsc{poly}_2(\textsc{n})$. Hence m_0 can be “big” and $(p_0(m_0), q_0(n_0))$ can be projected “far to the right” in a belt (recall Fig. 10); let us denote the respective belt by B and its slope by $\frac{\alpha}{\beta}$. Since $p_0(m_0) \nsim q_0(n_0)$, there must be a shortest path from $(p_0(m_0), q_0(n_0))$ to $\textsc{Target}'$ defined as

$$\textsc{Target}' = \nsim_1 \cup (\mathsf{ClearNo} \smallsetminus \textsc{BeltSpace}) \cup (\{(p(\textsc{poly}_2(\textsc{n})), q(n)) \mid p, q \in Q, n \in \mathbb{N}\} \cap B \cap \nsim).$$

Such a path

$$(p_0(m_0), q_0(n_0)) \xrightarrow{a_1} (p_1(m_1), q_1(n_1)) \xrightarrow{a_2} \cdots \xrightarrow{a_{\ell+1}} (p_{\ell+1}(m_{\ell+1}), q_{\ell+1}(n_{\ell+1})) \tag{8}$$

cannot enter $\sim$, and we thus have $m_j > \textsc{poly}_2(\textsc{n})$ and $(p_j(m_j), q_j(n_j)) \in B$ for all $j \in [0, \ell]$; in other words, if $m_j = \textsc{poly}_2(\textsc{n})$ or $|n_j - \frac{\alpha}{\beta} m_j| \geq \textsc{poly}_1(\textsc{n})$ then $j = \ell+1$. The path (8), possibly except of the last move $(p_\ell(m_\ell), q_\ell(n_\ell)) \xrightarrow{a_{\ell+1}} (p_{\ell+1}(m_{\ell+1}), q_{\ell+1}(n_{\ell+1}))$, can be naturally viewed as a path in $\mathcal{T}(\mathcal{A}')$ where the ROCA $\mathcal{A}'$ is defined as in the proof of Lemma 38, with K control states as given in (6).

We have shown, in fact, that the membership problem for $\textsc{Target}'$ is in NL. Informally speaking, other established facts allow us to deduce that either the path (8) is short or $m_{\ell+1} = \textsc{poly}_2(\textsc{n})$. The former case can be easily verified in (nondeterministic) logarithmic space (since ℓ is small, and the differences $m_j - m_0$, $n_j - n_0$ are thus small). The latter case reduces, in fact, to an instance of the reachability problem for $\mathcal{A}'$, which can be solved in (nondeterministic) logarithmic space (recall Proposition 22). We now formalize this idea.

Suppose $|m_\ell - m_0| < K^2 + \textsc{n} \cdot K$; then by Proposition 18 we have $m_j \in [m_0 - \textsc{n} \cdot K - 2K^2, m_0 + \textsc{n} \cdot K + 2K^2]$ for all $j \in [0, \ell]$. In this case the algorithm can just guess a pair $(p_{\ell+1}(m_{\ell+1}), q_{\ell+1}(n_{\ell+1}))$ (presenting $m_{\ell+1}$, $n_{\ell+1}$ by the differences $m_{\ell+1} - m_0$, $n_{\ell+1} - n_0$ in the workspace), verify its membership in $\textsc{Target}'$ and its reachability from $(p_0(m_0), q_0(n_0))$ by using logarithmic space only.

If $|m_\ell - m_0| \geq K^2 + \textsc{n} \cdot K$ then the correspondence of the path (8) in $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$ with the respective (shortest) path in $\mathcal{T}(\mathcal{A}')$ allows us to assume that (8) is in the form

$$(p_0(m_0),q_0(n_0)) \xrightarrow{v_1} (p(m),q(n)) \xrightarrow{v_2} (p(m{-}D),q(n{-}\tfrac{\alpha}{\beta}D)) \xrightarrow{v_2} (p(m{-}2D),q(n{-}2\tfrac{\alpha}{\beta}D)) \xrightarrow{v_2} \cdots$$
$$\cdots \xrightarrow{v_2} (p(m{-}xD),q(n{-}x\tfrac{\alpha}{\beta}D)) \xrightarrow{v_3} (p_\ell(m_\ell),q_\ell(n_\ell)) \xrightarrow{a_{\ell+1}} (p_{\ell+1}(m_{\ell+1}),q_{\ell+1}(n_{\ell+1}))$$

where $|v_1v_3| < K^2$, $|v_2| \leq K$, $x \geq \textsc{n}$, and $D \in [1,K]$ or $D \in [-K,-1]$.

The case $(p_{\ell+1}(m_{\ell+1}),q_{\ell+1}(n_{\ell+1})) \in \not\sim_1 \cup (\mathsf{ClearNo} \smallsetminus \textsc{BeltSpace})$ can be excluded by "cutting off the cycles" (i.e., by decreasing the number of v_2-segments), similarly as in the proof of Lemma 38. If we had $p_{\ell+1}(m_{\ell+1}) \not\sim_1 q_{\ell+1}(n_{\ell+1})$, then by cutting off one v_2-segment we would reach $\textsc{Target}'$ earlier. If $(p_{\ell+1}(m_{\ell+1}),q_{\ell+1}(n_{\ell+1})) \in \mathsf{ClearNo} \smallsetminus \textsc{BeltSpace}$, and $p_{\ell+1}(m_{\ell+1}) \not\sim_{\textsc{n}} q_{\ell+1}(n_{\ell+1})$, then by cutting off one v_2-segment we would again reach $\textsc{Target}'$ earlier. In the remaining subcase, when $(p_{\ell+1}(m_{\ell+1}),q_{\ell+1}(n_{\ell+1})) \in \mathsf{ClearNo} \smallsetminus \textsc{BeltSpace}$ and at least one of $\mathsf{dist}(p_{\ell+1}(m_{\ell+1}))$, $\mathsf{dist}(q_{\ell+1}(n_{\ell+1}))$ is finite (and thus $\mathsf{dist}(p_{\ell+1}(m_{\ell+1})) \neq \mathsf{dist}(q_{\ell+1}(n_{\ell+1}))$), there is $d \in [1,\textsc{n}]$ (derived from Proposition 24(1)) such that cutting off d "cycle-segments" v_2 gives rise to a shorter path to $\textsc{Target}'$ (namely to a pair outside $\textsc{BeltSpace}$ for which the distances to INC are different).

We thus have $m_{\ell+1} = \textsc{poly}_2(\textsc{n})$, $(p_{\ell+1}(\textsc{poly}_2(\textsc{n})),q_{\ell+1}(n_{\ell+1})) \in B \cap \not\sim$, (and $m_0 \geq \textsc{poly}_2(\textsc{n}) + K^2 + \textsc{n} \cdot K$). To handle this possibility, our algorithm can guess a pair $(p'(\textsc{poly}_2(\textsc{n})),q'(n)) \in B$, verify that $p'(\textsc{poly}_2(\textsc{n})) \not\sim q'(n)$, and then verify the reachability of $(p'(\textsc{poly}_2(\textsc{n})),q'(n))$ from $(p_0(m_0),q_0(n_0))$ in $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$. Verifying the reachability can be handled by an explicit logspace reduction to the reachability problem for $\mathcal{A}'$. A direct procedure can work as follows: it guesses $p,q \in Q$, $d_{11},d_{12},d_{21},d_{22} \in [-K^2,K^2]$, and $D \in [1,K]$ such that β divides D, and it verifies that in $\mathcal{T}(\mathcal{A}) \times \mathcal{T}(\mathcal{A})$ we have:

- from $(p_0(m_0),q_0(n_0))$ we can reach $(p(m_0{+}d_{11}),q(n_0{+}d_{12}))$ within K^2 moves,
- from $(p(\textsc{poly}_2(\textsc{n}){+}d_{21}),q(n{+}d_{22}))$ we can reach $(p'(\textsc{poly}_2(\textsc{n})),q'(n))$ within K^2 moves,
- from $(p(\textsc{poly}_2(\textsc{n}){+}d_{21}{+}D),q(n{+}d_{22}{+}\frac{\alpha}{\beta}D))$ we can reach $(p(\textsc{poly}_2(\textsc{n}){+}d_{21}),q(n{+}d_{22}))$ within K (positive) moves,
- $((m_0{+}d_{11}){-}(\textsc{poly}_2(\textsc{n}){+}d_{21}))$ mod $D = 0$, $((n_0{+}d_{12}){-}(n{+}d_{22}))$ mod $\frac{\alpha}{\beta}D = 0$, and
- $((m_0{+}d_{11}){-}(\textsc{poly}_2(\textsc{n}){+}d_{21})) \div D = ((n_0{+}d_{12}){-}(n{+}d_{22})) \div (\frac{\alpha}{\beta}D)$.

Recalling Proposition 21, we can easily check that the overall (nondeterministic) algorithm verifying that $p_0(m_0) \not\sim q_0(n_0)$ can be implemented to run in logarithmic space. □

8. Regularity problems

We now prove Theorem 6, which states that the regularity problem (is a given configuration $p(m)$ bisimilar to a state in a finite LTS?) is P-complete for general ROCA, and NL-complete for det-ROCA. We assume a fixed ROCA $\mathcal{A} = (Q,\Sigma,\delta)$ with $\textsc{n}$ control states. The next proposition is a variant of saying that $p(m)$ is nonregular iff the set $\{q(n) \mid p(m) \longrightarrow^* q(n) \longrightarrow^* \mathsf{INC}\}$ is infinite.

Proposition 40. *A configuration $p(m)$ is not regular if and only if there is q such that $p(m) \longrightarrow^* q(m{+}2\textsc{n}) \longrightarrow^* \mathsf{INC}$.*

Proof. We recall that $q(n) \not\longrightarrow^* \mathsf{INC}$ implies that $q(n) \sim r$ for some r in $\mathcal{F}_{\mathcal{A}}$ (by Lemma 14). Hence if $p(m) \longrightarrow^* q(m{+}2\textsc{n})$ implies $q(m{+}2\textsc{n}) \not\longrightarrow^* \mathsf{INC}$ (for all q) then from $p(m)$ we can reach only finitely many configurations up to bisimilarity, since each of them is bisimilar either to some r in $\mathcal{F}_{\mathcal{A}}$ or to $q(n)$ where $n < m{+}2\textsc{n}$. The "only if" part is thus clear.
For the "if" part we note that a path $p(m) \xrightarrow{u_1} q(m+2\textsc{n}) \xrightarrow{u_2} \mathsf{INC}$ can be written in the form

$$p(m) \xrightarrow{u_{11}} q_1(m+\textsc{n}) \xrightarrow{u_{12}} q(m+2\textsc{n}) \xrightarrow{u_{21}} q_2(m+\textsc{n}) \xrightarrow{u_{22}} \mathsf{INC}$$

where the subpath $q_1(m{+}\textsc{n}) \xrightarrow{u_{12}} q(m{+}2\textsc{n}) \xrightarrow{u_{21}} q_2(m{+}\textsc{n})$ is positive. By the pigeonhole principle, this subpath can be written

$$q_1(m+\textsc{n}) \xrightarrow{v_1} r(h) \xrightarrow{v_2} r(h{+}d) \xrightarrow{v_3} q(m+2\textsc{n}) \xrightarrow{w_1} r'(h'{+}d') \xrightarrow{w_2} r'(h') \xrightarrow{w_3} q_2(m+\textsc{n})$$

where $d, d' > 0$. For every $i \geq 1$ we thus have

$$p(m) \xrightarrow{u_{11}v_1} r(h) \xrightarrow{v_2(v_2)^{id'}} r(h{+}d+idd') \xrightarrow{v_3w_1} r'(h'{+}d'+idd') \xrightarrow{w_2(w_2)^{id}} r'(h') \xrightarrow{w_3u_{22}} \mathsf{INC}.$$

Hence for every $\ell \in \mathbb{N}$ there is a configuration that is reachable from $p(m)$ and its distance to INC is finite but larger than ℓ. Therefore $p(m)$ is non-regular. □

We recall that the INC-membership problem is P-complete for (general) ROCA, and NL-complete for deterministic ROCA (Proposition 13); we also recall NL-completeness of the reachability problem (Proposition 22). From Proposition 40 we thus deduce that the regularity problem for ROCA (w.r.t. bisimilarity) is in P in general, and in NL in the case of det-ROCA. The latter problem is obviously NL-hard (by digraph reachability); hence the next lemma finishes a proof of Theorem 6.

In the lemma we only use ROCA with *weak zero-tests* (like in Petri nets): we say that a ROCA $\mathcal{A} = (Q, \Sigma, \delta)$ is a *one-counter net* if $(q, a, 0, q', j) \in \delta$ implies $(q, a, 1, q', j) \in \delta$.

Lemma 41. *Regularity for ROCA is* P*-hard, even when restricted to one-counter nets.*

Proof. We use a log-space reduction from bisimilarity on finite LTSs (recall Prop. 1). Given a finite LTS $\mathcal{F} = (S, \Sigma, \{\xrightarrow{a}\}_{a\in\Sigma})$ and $p_0, q_0 \in S$, we construct a one counter net $\mathcal{A} = (S \cup \{s_0\}, \Sigma, \delta)$, $s_0 \notin S$, as shown below; we will have $p_0 \sim q_0$ in $\mathcal{F}$ iff $s_0(0)$ is regular in $\mathcal{T}(\mathcal{A})$.

For every $p \xrightarrow{a} q$ in $\mathcal{F}$ we put $(p, a, c, q, 0)$ into δ for both $c \in \{0, 1\}$; any $p(n)$ in $\mathcal{T}(\mathcal{A})$ just mimics the behaviour of p in $\mathcal{F}$. We then complete δ by $(s_0, a, c, s_0, +1)$ and $(s_0, b, c, p_0, 0)$ for $c \in \{0, 1\}$, and by $(s_0, a, 1, s_0, -1)$, $(s_0, b, 1, q_0, -1)$.

If $p_0 \sim q_0$ then obviously $s_0(m) \sim s_0(m')$ for any m, m'; hence $s_0(0)$ is regular. If $p_0 \nsim q_0$ then $s_0(0) \nsim s_0(m)$ for any $m > 0$, and thus $s_0(m) \nsim s_0(m')$ for any $m \neq m'$; there are thus infinitely many pairwise nonbisimilar states reachable from $s_0(0)$. □

References

[1] R. van Glabbeek, The linear time - branching time spectrum, in: J. Bergstra, A. Ponse, S. Smolka (Eds.), Handbook of Process Algebra, North-Holland, 2001, pp. 3–99.

[2] R. Milner, Communication and Concurrency, International Series in Computer Science, Prentice Hall, 1989.

[3] J. van Benthem, Modal Correspondence Theory, Ph.D. thesis, University of Amsterdam, 1976.

[4] D. Janin, I. Walukiewicz, On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic, in: Proc. of CONCUR, volume 1119 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 263–277.

[5] F. Moller, A. M. Rabinovich, Counting on CTL*: on the expressive power of monadic path logic, Inf. Comput. 184 (2003) 147–159.

[6] J. L. Balcázar, J. Gabarró, M. Santha, Deciding bisimilarity is P-complete, Formal Asp. Comput. 4 (1992) 638–648.

[7] R. Mayr, Process rewrite systems, Information and Computation 156 (2000) 264–286.

[8] J. Srba, Roadmap of Infinite Results, volume Vol 2: Formal Models and Semantics, World Scientific Publishing Co., 2004, pp. 337–350. http://www.brics.dk/˜srba/roadmap.

[9] G. Sénizergues, The bisimulation problem for equational graphs of finite out-degree, SIAM J. Comput. 34 (2005) 1025–1106.

[10] M. Benedikt, S. Göller, A. Murawski, S. Kiefer, Bisimilarity of pushdown automata is nonelementary, in: Proc. of LICS, IEEE Computer Society, 2013, pp. 488–498.

[11] P. Jančar, Strong bisimilarity on basic parallel processes is PSPACE-complete, in: Proc. of LICS, IEEE Computer Society, 2003, pp. 218–227.

[12] G. Sénizergues, L(A)=L(B)? decidability results from complete formal systems, Theor. Comput. Sci. 251 (2001) 1–166.

[13] G. Sénizergues, L(A)=L(B)? A simplified decidability proof, Theor. Comput. Sci. 281 (2002) 555–608.

[14] C. Stirling, Deciding DPDA equivalence is primitive recursive, in: Proc. of ICALP, volume 2380 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 821–832.

[15] P. Jančar, Decidability of DPDA language equivalence via first-order grammars, in: Proc. of LICS, IEEE Computer Society, 2012, pp. 415–424.

[16] G. Sénizergues, The equivalence problem for t-turn DPDA is co-NP, in: Proc. of ICALP, volume 2719 of *Lecture Notes in Computer Science*, Springer, 2003, pp. 478–489.

[17] Y. Hirshfeld, M. Jerrum, F. Moller, A polynomial algorithm for deciding bisimilarity of normed context-free processes, Theor. Comput. Sci. 158 (1996) 143–159.

[18] W. Czerwinski, S. Lasota, Fast equivalence-checking for normed context-free processes, in: Proc. of FSTTCS, volume 8 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010, pp. 260–271.

[19] E. P. Friedman, The inclusion problem for simple languages, Theor. Comput. Sci. 1 (1976) 297–316.

[20] O. Burkart, D. Caucal, B. Steffen, An elementary bisimulation decision procedure for arbitrary context-free processes, in: Proc. of MFCS, volume 969 of *Lecture Notes in Computer Science*, Springer, 1995, pp. 423–433.

[21] P. Jančar, Bisimilarity on basic process algebra is in 2-ExpTime (an explicit proof), Logical Methods in Computer Science 9 (2013).

[22] S. Kiefer, BPA bisimilarity is EXPTIME-hard, Inf. Process. Lett. 113 (2013) 101–106.

[23] P. Jančar, Decidability of bisimilarity for one-counter processes, Information Computation 158 (2000) 1–17.

[24] H.-C. Yen, Complexity analysis of some verification problems for one-counter machines, 2003. Unpublished manuscript.

[25] J. Srba, Beyond language equivalence on visibly pushdown automata, Logical Methods in Computer Science 5 (2009). (A preliminary version appeared at CSL 2006).

[26] O. Serre, Parity games played on transition graphs of one-counter processes, in: Proc. of FOSSACS, volume 3921 of *Lecture Notes in Computer Science*, Springer, 2006, pp. 337–351.

[27] L. G. Valiant, M. Paterson, Deterministic one-counter automata, J. Comput. Syst. Sci. 10 (1975) 340–350.

[28] P. Berman, R. Roos, Learning one-counter languages in polynomial time (extended abstract), in: Proc. of FOCS, IEEE, 1987, pp. 61–67.

[29] R. Roos, Deciding Equivalence of Deterministic One-Counter Automata in Polynomial Time with Applications to Learning, Ph.D. thesis, The Pennsylvania State University, 1988.

[30] K. Higuchi, M. Wakatsuki, 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 (1995) 939–950.

[31] K. Higuchi, M. Wakatsuki, 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 (1998) 1–11.

[32] J. Srba, Strong bisimilarity and regularity of basic process algebra is PSPACE-hard, in: Proc. of ICALP, volume 2380 of *Lecture Notes in Computer Science*, Springer, 2002, pp. 716–727.

[33] P. Hofman, S. Lasota, R. Mayr, P. Totzke, Simulation over one-counter nets is PSPACE-complete, in: Proc. of FSTTCS, LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013. To appear.

[34] T. Brázdil, V. Brožek, K. Etessami, A. Kučera, D. Wojtczak, One-counter Markov decision processes, in: Proc. of SODA, IEEE, 2010, pp. 863–874.

[35] S. Göller, R. Mayr, A. W. To, On the computational complexity of verifying one-counter processes, in: Proc. of LICS, IEEE Computer Society Press, 2009, pp. 235–244.

[36] C. Haase, S. Kreutzer, J. Ouaknine, J. Worrell, Reachability in succinct and parametric one-counter automata, in: Proc. of CONCUR, volume 5710 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 369–383.

[37] S. Demri, A. Sangnier, When model-checking freeze LTL over counter machines becomes decidable, in: Proc. of FOSSACS, volume 6014 of *Lecture Notes in Computer Science*, Springer, 2010, pp. 176–190.

[38] A. W. To, Model checking FO(R) over one-counter processes and beyond, in: Proc. of CSL, volume 5771 of *Lecture Notes in Computer Science*, Springer, 2009, pp. 485–499.

[39] S. Göller, M. Lohrey, Branching-time model checking of one-counter processes, in: STACS, volume 5 of *LIPIcs*, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2010, pp. 405–416.

[40] S. Böhm, S. Göller, P. Jančar, Equivalence of deterministic one-counter automata is NL-complete, in: Proc. of STOC, ACM, 2013, pp. 131–140.

[41] S. Ginsburg, E. Spanier, Semigroups, Presburger Formulas, and Languages, Pacific Journal of Mathematics 16 (1966) 285–296.