

Simulation Over One-counter Nets is PSPACE-Complete*

Piotr Hofman¹, Sławomir Lasota¹, Richard Mayr², and Patrick Totzke²

1 University of Warsaw, Poland

2 University of Edinburgh, UK

Abstract

One-counter nets (OCN) are Petri nets with exactly one unbounded place. They are equivalent to a subclass of one-counter automata with just a weak test for zero. Unlike many other semantic equivalences, strong and weak simulation preorder are decidable for OCN, but the computational complexity was an open problem. We show that both strong and weak simulation preorder on OCN are PSPACE-complete.

1998 ACM Subject Classification F.1.1; D.2.4

Keywords and phrases Simulation preorder; one-counter nets; complexity

Digital Object Identifier 10.4230/LIPIcs.xxx.yyy.p

1 Introduction

The model. One-counter automata (OCA) are Minsky counter automata with only one counter, and they can also be seen as a subclass of pushdown automata with just one stack symbol (plus a bottom symbol). One-counter nets (OCN) are Petri nets with exactly one unbounded place, and they correspond to a subclass of OCA where the counter cannot be fully tested for zero, because transitions enabled at counter value zero are also enabled at nonzero values. OCN are arguably the simplest model of discrete infinite-state systems, except for those that do not have a global finite control.

Previous results on semantic equivalence checking. Notions of behavioral semantic equivalences have been classified in Van Glabbeek's linear time - branching time spectrum [3]. The most common ones are, in order from finer to coarser, bisimulation, simulation and trace equivalence. Each of these have their standard (called strong) variant, and a weak variant that abstracts from arbitrarily long sequences of internal actions.

For OCA/OCN, strong bisimulation is PSPACE-complete [2], while weak bisimulation is undecidable [9]. Strong trace inclusion is undecidable for OCA [11], and even for OCN [4], and this trivially carries over to weak trace inclusion.

The picture is more complicated for simulation preorders. While strong and weak simulation are undecidable for OCA [7], they are decidable for OCN. Decidability of strong simulation on OCN was first proven in [1], by establishing that the simulation relation follows a certain regular pattern. This idea was made more graphically explicit in later proofs [6, 5], which established the so-called *Belt Theorem*, that states that the simulation preorder relation

* This work was partially supported by Polish NCN grant 2012/05/NST6/03226 and Polish MNiSW grant N N206 567840.



on OCN can be described by finitely many partitionings of the grid $\mathbb{N} \times \mathbb{N}$, each induced by two parallel lines. In particular, this implies that the simulation relation is semilinear. However, the proofs in [1, 6, 5] did not yield any upper complexity bounds, since the first was based on two semi-decision procedures and the later proof of the Belt Theorem was non-constructive. A PSPACE lower bound for strong simulation on OCN follows from [10].

Decidability of weak simulation on OCN was shown in [4], using a converging series of semilinear approximants. This proof used the decidability of strong simulation on OCN as an oracle, and thus did not immediately yield any upper complexity bound.

Our contribution. We provide a new constructive proof of the Belt Theorem and derive a PSPACE algorithm for checking strong simulation preorder on OCN. Together with the lower bound from [10], this shows PSPACE-completeness of the problem.

Via a technical adaption of the algorithm for weak simulation in [4], and the new PSPACE algorithm for strong simulation, we also obtain a PSPACE algorithm for weak simulation preorder on OCN. Thus even weak simulation preorder on OCN is PSPACE-complete.

	simulation	bisimulation	weak sim.	weak bis.	trace inclusion
OCN	PSPACE	PSPACE [2]	PSPACE	undecidable [9]	undecidable [11]
OCA	undecidable [7]	PSPACE [2]	undecidable [7]	undecidable [9]	undecidable [4]

2 Problem Statement

A labelled transition system (LTS) over a finite alphabet A of actions consists of a set of configurations and, for every action $a \in A$, a binary relation \xrightarrow{a} between configurations.

Given two LTS S and S' , a relation R between the configurations of S and S' is a *simulation* if for every pair of configurations $(c, c') \in R$ and every step $c \xrightarrow{a} d$ there exists a step $c' \xrightarrow{a} d'$ such that $(d, d') \in R$. Simulations are closed under union, so there exists a unique maximal simulation. If $S = S'$ then this maximal simulation is a preorder, called *simulation preorder*, and denoted by \preceq . If $c \preceq c'$ then one says that c' *simulates* c .

Simulation preorder can also be characterized by a *Simulation Game* as follows. The *positions* are all pairs (c, c') of configurations of S and S' respectively. The game is played by two players called *Spoiler* and *Duplicator* and proceeds in rounds. In every round, starting in a position (c, c') , Spoiler chooses some $a \in A$ and some configuration d with $c \xrightarrow{a} d$. Then Duplicator responds by choosing a configuration d' with $c' \xrightarrow{a} d'$, and the next round continues from position (d, d') . If one of the players cannot move then the other player wins, and Duplicator wins every infinite play. It is well known that the Simulation Game is determined: for every initial position (c, c') , exactly one of players has a winning strategy. Configuration c' simulates c iff Duplicator has a strategy to win the Simulation Game from position (c, c') .

► **Definition 1 (One-Counter Nets).** A *one-counter net* (OCN) is a triple $\mathcal{N} = (Q, A, \delta)$ given by finite sets of control-states Q , action labels A and transitions $\delta \subseteq Q \times A \times \{-1, 0, 1\} \times Q$. It induces an infinite-state labelled transition system over the state set $Q \times \mathbb{N}$, whose elements will be written as pm , where $pm \xrightarrow{a} qn$ iff $(p, a, d, q) \in \delta$ and $n = m + d \geq 0$.

We study the computational complexity of the following decision problem.

Simulation Checking for OCN

INPUT:	Two OCN \mathcal{N} and \mathcal{N}' together with configurations qn and $q'n'$ of \mathcal{N} and \mathcal{N}' respectively, where n and n' are given in binary.
QUESTION:	$qn \preceq q'n'$?

► **Theorem 2.** *The Simulation Checking Problem for OCN is in PSPACE.*

Combined with the PSPACE-hardness result of [10], this yields PSPACE-completeness of the problem.

► **Remark.** Our construction can also be used to compute the simulation relation as a semilinear set, but its description requires exponential space. However, checking a point instance $qn \preceq q'n'$ of the simulation problem can be done in polynomial space by stepwise guessing and verifying only a polynomially bounded part of the relation; cf. Section 5.

Without restriction (see [1] for a justification) we assume that both OCN are *normalised*:

1. In Spoiler's net \mathcal{N} , every control-state has some outgoing transition with a non-negative change of counter value.
2. Duplicator's net \mathcal{N}' is *complete*, i.e., every control-state has an outgoing transition for every action (though the change in counter value may be negative).

Thus Spoiler cannot get stuck and only loses the game if it is infinite. Moreover, Duplicator can only be stuck (and lose the game) when his counter equals zero.

Outline of the proof. One easily observes that the Simulation Game is monotone for both players. If Duplicator wins the Simulation Game from a position $(qn, q'n')$ then he also wins from $(qn, q'm)$ for $m > n'$. Similarly, if Spoiler wins from $(qn, q'n')$ then she also wins from $(qm, q'n')$ for $m > n$. For a fixed pair (q, q') of control-states, both players winning regions therefore split the grid $\mathbb{N} \times \mathbb{N}$ into two connected subsets. It is known [6, 5] that the *frontier* between these subsets is contained in a *belt*, i.e., it lays between two parallel lines with rational slope.

For the proof of our main result we analyse a symbolic *Slope Game*. This new game is similar to the Simulation Game but necessarily ends after a small number of rounds. We show that given sufficiently high excess of counter-values, both players can re-use winning strategies for the Slope Game also in the Simulation Game. As a by-product of this characterization, we obtain polynomial bounds on widths and slopes of the belts. Once the belt-coefficients are known, one can compute the frontiers exactly because every frontier necessarily adheres to a regular pattern.

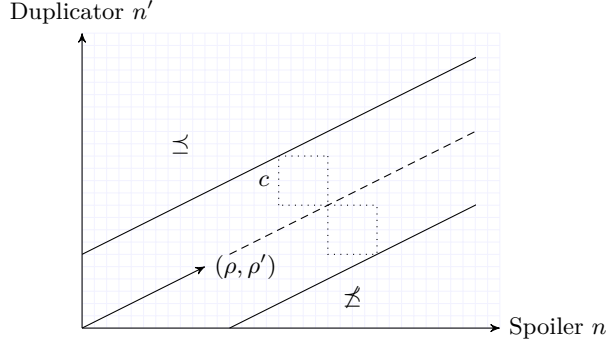
3 Polynomially Bounded Belts

Let us fix two OCN \mathcal{N} and \mathcal{N}' , with sets of control-states Q and Q' , respectively. Following [5], we interpret \preceq as 2-colouring of $K = |Q \times Q'|$ Euclidean planes, one for each pair of control-states $(q, q') \in Q \times Q'$.

The main combinatorial insight of [5] (this was also present in [1], albeit less explicitly) is the so-called *Belt Theorem*, that states that each such plane can be cut into segments by two parallel lines such that the colouring of \preceq in the outer two segments is constant; see Figure 1. We provide a new constructive proof of this theorem, stated as Theorem 4 below, that allows us to derive polynomial bounds on the coefficients of all belts.

► **Definition 3** (Positive vectors, direction, c-above, c-below). A vector $(\rho, \rho') \in \mathbb{Z} \times \mathbb{Z}$ of integers is called *positive* if $(\rho, \rho') \in \mathbb{N} \times \mathbb{N}$ and $(\rho, \rho') \neq (0, 0)$. Its *direction* is the half-line $\mathbb{R}^+ \cdot (\rho, \rho')$. For a positive vector (ρ, ρ') and a number $c \in \mathbb{N}$ we say that the point $(n, n') \in \mathbb{Z} \times \mathbb{Z}$ is *c-above* (ρ, ρ') iff there exists some point $(r, r') \in \mathbb{R}^+ \cdot (\rho, \rho')$ in the direction of (ρ, ρ') such that

$$n < r - c \quad \text{and} \quad n' > r' + c. \tag{1}$$



■ **Figure 1** A belt with slope $\frac{\rho}{\rho'}$. The dashed half-line is the direction of (ρ, ρ') .

Symmetrically, (n, n') is *c-below* (ρ, ρ') if is a point $(r, r') \in \mathbb{R}^+ \cdot (\rho, \rho')$ with

$$n > r + c \quad \text{and} \quad n' < r' - c. \quad (2)$$

► **Theorem 4** (Belt Theorem). *For every two one-counter nets \mathcal{N} and \mathcal{N}' with sets of control-states Q and Q' respectively, there is a bound $c \in \mathbb{N}$ such that for every pair $(q, q') \in Q \times Q'$ of control-states there is a positive vector (ρ, ρ') such that*

1. *if (n, n') is c-above (ρ, ρ') then $qn \preceq q'n'$, and*
2. *if (n, n') is c-below (ρ, ρ') then $qn \not\preceq q'n'$.*

Moreover, c and all ρ, ρ' are bounded polynomially w.r.t. the sizes of \mathcal{N} and \mathcal{N}' .

4 Proof of the Belt Theorem

We consider OCN \mathcal{N} and \mathcal{N}' with sets of control-states Q and Q' , resp., and define the constant $K = |Q \times Q'|$. Abdulla and Cerans [1] showed that, above a certain level, the simulation relation has a regular structure. An important parameter for this structure is the *ratio* n/n' of the respective counter values n in Spoiler's configuration qn of \mathcal{N} and n' in Duplicator's configuration $q'n'$ of \mathcal{N}' .

We further develop this intuition by defining a new finitary game (called the Slope Game; cf. Section 4.1) that is played directly on the control graphs of the nets, and in which the objective of the players is to minimize (resp. maximize) the ratio of the effects of recently observed minimal cycles. Then we show how to transform winning strategies in the Slope Game into winning strategies in the original simulation game. First we need to define some properties of vectors.

► **Definition 5** (Behind, Steeper). Let (ρ, ρ') be a positive and $(\alpha, \alpha') \in \mathbb{Z}^2$ an arbitrary vector. We place the two on the plane with a common starting point and consider the clockwise oriented angle from (ρ, ρ') to (α, α') . We say that (α, α') is *behind* (ρ, ρ') if the oriented angle is strictly between 0° and 180° . See Figure 2 for an illustration.

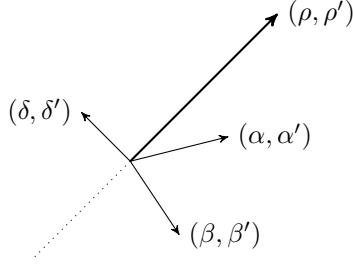
Positive vectors may be naturally ordered: We will call (ρ, ρ') *steeper* than (α, α') , written $(\alpha, \alpha') \prec (\rho, \rho')$, if (α, α') is behind (ρ, ρ') .

Note that the property of one vector being behind another only depends on their directions. The following simple lemma will be useful in the sequel.

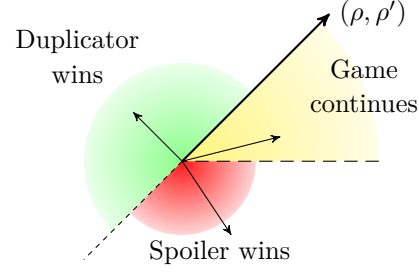
► **Lemma 6.** *Let (ρ, ρ') be a positive vector and $c, n, n' \in \mathbb{N}$.*

1. *If (n, n') is c-below (ρ, ρ') then $(n, n') + (\alpha, \alpha')$ is c-below (ρ, ρ') for any vector (α, α') which is behind (ρ, ρ') .*

2. If (n, n') is c -above (ρ, ρ') then $(n, n') + (\alpha, \alpha')$ is c -above (ρ, ρ') for any vector (α, α') which is not behind (ρ, ρ') .



■ **Figure 2** Vectors (α, α') and (β, β') are behind (ρ, ρ') , but (δ, δ') is not. Also, $(\alpha, \alpha') \prec (\rho, \rho')$.



■ **Figure 3** Evaluating the winning condition in position $(\pi, (\rho, \rho'))$ after a phase of the Slope Game.

4.1 Slope Game

► **Definition 7** (Product Control Graph, Lasso, Effect of a path). Given two OCN $\mathcal{N} = (Q, A, \delta)$ and $\mathcal{N}' = (Q', A, \delta')$, their *product control graph* is the finite, edge-labelled graph with nodes $Q \times Q'$ and $(A \times \mathbb{N} \times \mathbb{N})$ -labelled edges E given by

$$(p, p') \xrightarrow{a, d, d'} (q, q') \in E \text{ iff } p \xrightarrow{a, d} q \in \delta \text{ and } p' \xrightarrow{a, d'} q' \in \delta'. \quad (3)$$

A *path*

$$\pi = (q_0, q'_0) \xrightarrow{a_0, d_0, d'_0} (q_1, q'_1) \xrightarrow{a_1, d_1, d'_1} \dots \xrightarrow{a_{k-1}, d_{k-1}, d'_{k-1}} (q_k, q'_k) \quad (4)$$

from (q_0, q'_0) to (q_k, q'_k) in this graph is called *lasso* if it contains a cycle while none of its strict prefixes does. That is, if there exist $i < k$ such that $(q_k, q'_k) = (q_i, q'_i)$ and for all $0 \leq i < j < k$, $(q_i, q'_i) \neq (q_j, q'_j)$. The lasso π splits into $\text{PREFIX}(\pi) = (q_0, q'_0) \xrightarrow{a_0, d_0, d'_0} \dots \xrightarrow{a_{i-1}, d_{i-1}, d'_{i-1}} (q_i, q'_i)$ and $\text{CYCLE}(\pi) = (q_i, q'_i) \xrightarrow{a_i, d_i, d'_i} \dots \xrightarrow{a_{k-1}, d_{k-1}, d'_{k-1}} (q_k, q'_k)$. The *effect* of a path is the cumulative sum of the effects of its transitions:

$$\Delta(\pi) = \sum_{i=0}^{k-1} (d_i, d'_i) \in \mathbb{Z} \times \mathbb{Z}. \quad (5)$$

The effects of cycles will play a central role in our further construction. The intuition is that if a play of a Simulation Game describes a lasso then the players “agree” on the chosen cycle. Repeating this cycle will change the ratio of the counter values towards its effect.

To formalize this intuition, we define a finitary Slope Game which proceeds in phases. In each phase, the players alternately move on the control graphs of their original nets, ignoring the counter, and thereby determine the next lasso that occurs. After such a phase, a winning condition is evaluated that compares the effect of the chosen lasso’s cycle with that of previous phases. Now either one player immediately wins or the next phase starts, but then the steepness of the observed effect must have strictly decreased. The number of different effects of simple cycles thus bounds the maximal length of a game.

► **Definition 8** (Slope Game). A *Slope Game* is a strictly alternating two player game played on a pair $\mathcal{N}, \mathcal{N}'$ of one-counter nets. The game positions are pairs $(\pi, (\rho, \rho'))$, where π is an acyclic path in the product control graph of \mathcal{N} and \mathcal{N}' , and (ρ, ρ') is a positive vector which we call *slope*.

The game is divided into *phases*, each starting with a path $\pi = (q_0, q'_0)$ of length 0. Until a phase ends, the game proceeds in rounds like a Simulation Game, but the players pick transition rules instead of transitions: in a position $(\pi, (\rho, \rho'))$ where π ends in states (q, q') , Spoiler chooses a transition rule $q \xrightarrow{a,d} p$, then Duplicator responds with a transition rule $q' \xrightarrow{a,d'} p'$. If the extended path $\pi' = \pi \xrightarrow{a,d,d'} (p, p')$ is still not a lasso, the next round continues from the updated position $(\pi', (\rho, \rho'))$; otherwise the phase ends with *outcome* $(\pi', (\rho, \rho'))$. The slope (ρ, ρ') does not restrict the possible moves of either player, nor changes during a phase. We thus speak of *the slope of a phase*.

If a round ends in position $(\pi, (\rho, \rho'))$ where π is a lasso, then the winning condition is evaluated. We distinguish three non-intersecting cases depending on how the effect $\Delta(\text{CYCLE}(\pi)) = (\alpha, \alpha')$ of the lasso's cycle relates to (ρ, ρ') :

1. If (α, α') is not behind (ρ, ρ') , Duplicator wins immediately.
2. If (α, α') is behind (ρ, ρ') but not positive, Spoiler wins immediately.
3. If (α, α') is behind (ρ, ρ') and positive, the game continues with a new phase from position $(\pi', (\alpha, \alpha'))$, where π' is the path of length 0 consisting of the pair of ending states of π .

Figure 3 illustrates the winning condition. Note that if there is no immediate winner it is guaranteed that (α, α') is a positive vector.

The fundamental intuition for the connection between the Slope Game and the Simulation Game is as follows. The Slope Game from initial position $((q, q'), (\rho, \rho'))$ determines how the initial slope (ρ, ρ') relates to the belt in the plane for (q, q') in the simulation relation. Roughly speaking, if (ρ, ρ') is less steep than the belt then Spoiler wins; if (ρ, ρ') is steeper then Duplicator wins. Finally, when the initial slope (ρ, ρ') is exactly as steep as the belt, any player may win the Slope Game.

Consider a Simulation Game in which the ratio n/n' of the counter values of Spoiler and Duplicator is the same as the ratio ρ/ρ' , i.e. suppose (n, n') is contained in the direction of (ρ, ρ') . Suppose also that the values (n, n') are sufficiently large. By monotonicity, we know that the steeper the slope (ρ, ρ') , the better for Duplicator. Hence if the effect (α, α') of some cycle is behind (ρ, ρ') and positive, then it is beneficial for Spoiler to repeat this cycle. With more and more repetitions, the ratio of the counter values will get arbitrarily close to (α, α') . On the other hand, if (α, α') is behind (ρ, ρ') but not positive then Spoiler wins by repeating the cycle until the Duplicator's counter decreases to 0. Finally, if the effect of the cycle is not behind (ρ, ρ') then repeating this cycle leads to Duplicator's win.

The next lemma follows from the observation that in Slope Games, the slope of a phase must be strictly less steep than those of all previous phases.

► **Lemma 9.** *For a fixed pair $\mathcal{N}, \mathcal{N}'$ of OCN,*

1. *any Slope Game ends after at most $(K + 1)^2$ phases, and*
2. *Slope Games are effectively solvable in PSPACE.*

Proof. After every phase, the slope (ρ, ρ') is equal to the effect of a simple cycle, which must be a positive vector. Thus the absolute values of both numbers ρ and ρ' are bounded by $K = |Q \times Q'|$. It follows that the total number of different possible values for (ρ, ρ') , and therefore the maximal number of phases played, is at most $(K + 1)^2$. This proves the

first part of the claim. Point 2 is a direct consequence as one can find and verify winning strategies by an exhaustive search. ◀

Strategies in Slope Games. Consider one phase of a Slope Game, starting from a position $(\pi, (\rho, \rho'))$. The phase ends with a lasso whose cycle effect (α, α') satisfies exactly one of three conditions, as examined by the evaluating function. Accordingly, depending on its initial position, every phase falls into exactly one of three disjoint cases:

1. Spoiler has a strategy to win the Slope Game immediately,
2. Duplicator has a strategy to win the Slope Game immediately or
3. neither Spoiler nor Duplicator have a strategy to win immediately.

In case 1. or 2. we call the phase *final*, and in case 3. we call it *non-final*. The non-final phases are the most interesting ones because in those, both players have a strategy that at least prevents an immediate loss.

Strategy Trees. Both in final and non-final phases, a strategy for Spoiler or Duplicator is a tree as described below. For the definition of strategy trees we need to consider, not only Spoiler's positions $(\pi, (\rho, \rho'))$ but also Duplicator's positions, the intermediate positions within a single round. These intermediate positions may be modelled as triples $(\pi, (\rho, \rho'), t)$ where t is a transition rule in \mathcal{N} from the last state of π . Observe that the bipartite directed graph, with positions of a phase as vertices and edges determined by the single-move relation, is actually a tree, call it T . Thus a Spoiler-strategy, i.e. a subgraph of T containing exactly one successor of every Spoiler's position and all successors of every Duplicator's position, is a tree as well; and so is any strategy for Duplicator.

Such a strategy (tree) in the Slope Game naturally splits into *segments*, each segment being a strategy (tree) in one phase. The segments themselves are also arranged into a tree, which we call *segment tree*. Irrespectively which player wins a Slope Game, according to the above observations, this player's winning strategy contains segments of two kinds:

- non-leaf segments are strategies to either win immediately or continue the Slope Game (these are strategies for non-final phases);
- leaf segments are strategies to win the Slope Game immediately (these are strategies in final phases).

By the *segment depth* of a strategy we mean the depth of its segment tree. By Lemma 9, Point 1, we know that a Slope Game ends after at most $d_{\max} = (K+1)^2$ phases. Consequently, the segment depths of strategies are at most d_{\max} as well.

A value of $c = K \cdot d_{\max}$ is sufficient for the claim of Theorem 4. The intuition behind this value is that for a winning player in the Slope Game, an excess of K per phase is sufficient to be able to safely “replay” a winning strategy in the Simulation Game. Formally, this is stated by the following two crucial lemmas, proofs of which can be found in Appendix A.

► **Lemma 10.** *Suppose Spoiler has a winning strategy of segment depth d in the Slope Game from a position $((q, q'), (\rho, \rho'))$. Then Spoiler wins the Simulation Game from every position $(qn, q'n')$ which is $(K \cdot d)$ -below (ρ, ρ') .*

► **Lemma 11.** *Suppose Duplicator has a winning strategy of segment depth d in the Slope Game from a position $((q, q'), (\rho, \rho'))$. Then Duplicator wins the Simulation Game from every position $(qn, q'n')$ which is $(K \cdot d)$ -above (ρ, ρ') .*

4.2 Proof of Theorem 4

Let $c = K \cdot d_{max}$. For any two states $q \in Q$ and $q' \in Q'$ of the nets \mathcal{N} and \mathcal{N}' we will determine the ratio (ρ, ρ') that, together with c , characterises the belt of the plane (q, q') . First observe the following monotonicity property of the Slope Game.

► **Lemma 12.** *If Spoiler wins the Slope Game from a position $((q, q'), (\rho, \rho'))$ and (σ, σ') is less steep than (ρ, ρ') then Spoiler also wins the Slope Game from $((q, q'), (\sigma, \sigma'))$.*

Proof. Assume that Spoiler wins the Slope Game from $((q, q'), (\rho, \rho'))$ while Duplicator wins from $((q, q'), (\sigma, \sigma'))$, for some $(\sigma, \sigma') \prec (\rho, \rho')$. Observe that in both cases, winning strategies of segment depth $\leq d_{max}$ exist. As (σ, σ') is less steep than (ρ, ρ') , there is a point $(n, n') \in \mathbb{N} \times \mathbb{N}$ which is both c -above (σ, σ') and c -below (ρ, ρ') . Applying both Lemma 10 and 11 immediately yields a contradiction. ◀

Equivalently, if Duplicator wins the Slope Game from $((q, q'), (\rho, \rho'))$ and (σ, σ') is steeper than (ρ, ρ') then Duplicator also wins the Slope Game from $((q, q'), (\sigma, \sigma'))$. We conclude that for every pair (q, q') of states, there is a *boundary slope* (β, β') such that

1. Spoiler wins the Slope Game from $((q, q'), (\sigma, \sigma'))$ for every (σ, σ') less steep than (β, β') ;
2. Duplicator wins the Slope Game from $((q, q'), (\sigma, \sigma'))$ for every (σ, σ') steeper than (β, β') .

Note that we claim nothing about the winner from the position $((q, q'), (\beta, \beta'))$ itself. Applying Lemmas 10 and 11 we see that this boundary slope (β, β') satisfies the claims 1 and 2 of Theorem 4. Indeed, consider a pair $(n, n') \in \mathbb{N} \times \mathbb{N}$ of counter values. If (n, n') is c -below (β, β') , then there is certainly a line $(\bar{\beta}, \bar{\beta}')$ less steep than (β, β') such that (n, n') is c -below $(\bar{\beta}, \bar{\beta}')$. By point 1 above, Spoiler wins the Slope Game from $((q, q'), (\bar{\beta}, \bar{\beta}'))$. By Lemma 10, Spoiler wins the Simulation Game from $(qn, q'n')$. Analogously, one can use point 2 above together with Lemma 11 to show Point 2 of Theorem 4.

It remains to show that the boundary slope (β, β') is polynomial in the sizes of \mathcal{N} and \mathcal{N}' . We show that (β, β') must in fact be the effect of a simple cycle. Because such cycles are no longer than $K = |Q \times Q'|$ and because along a path of length K the counter values cannot change by more than K , we conclude that $-K \leq \beta, \beta' \leq K$.

► **Definition 13** (Equivalent vectors). Consider all the non-zero effects (α, α') of all cycles together with their opposite vectors $(-\alpha, -\alpha')$ and denote the set of all these vectors by V . Call two positive vectors (ρ, ρ') and (σ, σ') *equivalent* if for all $(\alpha, \alpha') \in V$,

$$(\alpha, \alpha') \text{ is behind } (\rho, \rho') \iff (\alpha, \alpha') \text{ is behind } (\sigma, \sigma'). \quad (6)$$

In other words, equivalent vectors lie in the same angle determined by a pair of vectors from V that are neighbours angle-wise. We claim that equivalent slopes have the same winner in the Slope Game:

► **Lemma 14.** *If (ρ, ρ') and (σ, σ') are equivalent then the same player wins the Slope Game from $((q, q'), (\rho, \rho'))$ and $((q, q'), (\sigma, \sigma'))$.*

Proof. A winning strategy in the Slope Game from $((q, q'), (\rho, \rho'))$ may be literally used in the Slope Game from $((q, q'), (\sigma, \sigma'))$. This holds because the assumption that (ρ, ρ') and (σ, σ') are equivalent implies that all possible outcomes of the initial phase of the Slope Game are evaluated equally. ◀

Lemma 14 implies that the boundary slope is in V . This concludes the proof of Theorem 4. ◀

4.3 A Sharper Estimation

Theorem 4 provides a polynomial bound on the constant c and the slopes of all belts, with respect to the sizes of \mathcal{N} and \mathcal{N}' . However, the proof of Theorem 4 reveals that a slightly stronger result actually holds, which will be useful in proving the complexity bound for weak simulation in Section 6. We can estimate a bound on c in terms of the following two parameters of the product control graph $\mathcal{N} \times \mathcal{N}'$:

- SCC, the size of the largest strongly connected component, and
- ACYC, the length of the longest acyclic path.

In particular, we claim that Theorem 4 still holds with the constant c bounded by

$$c \leq \text{poly}(\text{SCC}) + \text{ACYC}. \quad (7)$$

Intuitively, c is the excess of counter value needed to replay a Slope Game strategy in the Simulation Game. This directly corresponds to the maximal number of alternations in a play of the Slope Game. Every phase ends in a cycle, which must be contained in some strongly connected component and is thus no longer than SCC. So the segment depth of Slope Game strategies is bounded by $(\text{SCC} + 1)^2$.

We can decompose plays of the Slope Game by separating subpaths that contain at least one cycle and stay in one strongly connected component, and the remaining subpaths. One can now show that in fact, a counter value of SCC suffices to enable subpaths of the first kind. The segment depth bounds the number of such subpaths in any play. Secondly, by definition, the subpaths of the second kind cannot share any points. The sum of their lengths is hence bounded by ACYC. We conclude that a value of $c = (\text{SCC} + 1)^2 \cdot \text{SCC} + \text{ACYC}$ is sufficient.

5 Strong Simulation is PSPACE-complete

Using our stronger version of the Belt Theorem from Section 4, we derive an algorithm for checking simulation preorder, similarly as in [1, 6, 5].

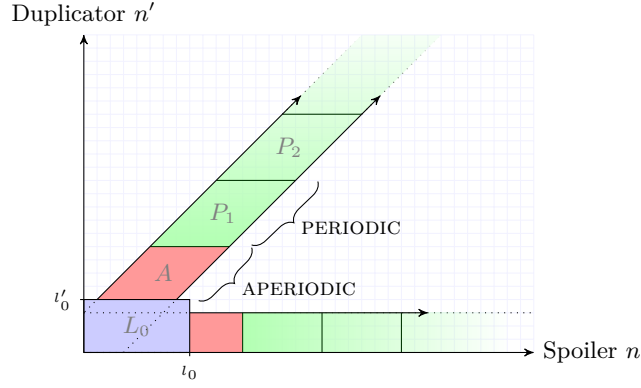
As before we fix two OCN \mathcal{N} and \mathcal{N}' , with sets of control-states Q and Q' , respectively. By Lemma 9, Point 2, we can compute in PSPACE, for every pair $(q, q') \in Q \times Q'$, the positive vector (ρ, ρ') satisfying Theorem 4; we denote this vector by $\text{SLOPE}(q, q')$. We define $\text{BELT}(q, q')$ to be the set of points $(n, n') \in \mathbb{N}^2$ that are neither c -above nor c -below $\text{SLOPE}(q, q')$. As all vectors $\text{SLOPE}(q, q')$ and the widths of all belts are polynomially bounded (by Theorem 4), we observe that every two non-parallel belts are disjoint outside a polynomially bounded *initial rectangle*, denoted L_0 , between corners $(0, 0)$ and (l_0, l'_0) (see Figure 4).

Recall that the simulation preorder on the configurations with the pair of control-states (q, q') is trivial outside of $\text{BELT}(q, q')$: it contains all pairs $(qn, q'n')$ s.t. (n, n') is c -above $\text{SLOPE}(q, q')$, and contains no pairs $(qn, q'n')$ s.t. (n, n') is c -below $\text{SLOPE}(q, q')$. We show that inside a belt, the points corresponding to configurations in simulation are ultimately periodic in the sense defined below.

By the definition of belts, $(n, n') \in \text{BELT}(q, q') \iff (n, n') + \text{SLOPE}(q, q') \in \text{BELT}(q, q')$, i.e., translation via the vector $\text{SLOPE}(q, q')$ preserves membership in $\text{BELT}(q, q')$. This is why we restrict our focus to multiples of vectors $\text{SLOPE}(q, q')$. We write $\text{RECT}(q, q', j)$ for the rectangle between corners $(0, 0)$ and $(l_0, l'_0) + j \cdot \text{SLOPE}(q, q')$.

► **Definition 15** (ultimately-periodic). For a fixed pair $(q, q') \in Q \times Q'$ and $j, k \in \mathbb{N}$, a subset $R \subseteq \text{BELT}(q, q')$ is called (j, k) -ultimately-periodic if for all $(n, n') \in \mathbb{N}^2 \setminus \text{RECT}(q, q', j)$,

$$(n, n') \in R \iff (n, n') + k \cdot \text{SLOPE}(q, q') \in R. \quad (8)$$



■ **Figure 4** The initial rectangle L_0 (blue) and two belts. Outside L_0 , the colouring of a belt consists of some exponentially bounded block (red), and another exponentially bounded non-trivial block (green) which repeats ad infinitum along the rest of the belt.

► **Remark.** Observe that for fixed q and q' , every (j, k) -ultimately-periodic set R can be represented by the numbers j and k , and two sets

$$R \cap \text{RECT}(q, q', j) \quad \text{and} \quad (R \setminus \text{RECT}(q, q', j)) \cap \text{RECT}(q, q', j + k).$$

The following lemma states a property which is crucial for our algorithm. It is actually a sharpening of the result of [5], with additional effective bounds on periods inside belts.

► **Lemma 16.** *For every pair $(q, q') \in Q \times Q'$, the set*

$$\preceq_{q, q'} = \{(n, n') \in \text{BELT}(q, q') : qn \preceq q'n'\}$$

is (j, k) -ultimately periodic for some $j, k \in \mathbb{N}$ exponentially bounded w.r.t. the sizes of $\mathcal{N}, \mathcal{N}'$.

Thus, when searching for a simulation relation inside belts, we may safely restrict ourselves to (j, k) -ultimately-periodic relations, for exponentially bounded j and k . According to the remark above, every such simulation admits the EXPSPACE description that consists, for every pair of states (q, q') , of:

- a polynomially bounded vector $(\rho, \rho') = \text{SLOPE}(q, q')$;
- a polynomially bounded relation $\text{INIT}(q, q') \subseteq L_0$ inside the initial rectangle L_0 ;
- exponentially bounded natural numbers $j_{q, q'}, k_{q, q'} \in \mathbb{N}$; and
- two exponentially bounded relations:

$$\begin{aligned} \text{APERIODIC}(q, q') &\subseteq \text{BELT}(q, q') \cap \text{RECT}(q, q', j_{q, q'}) \\ \text{PERIODIC}(q, q') &\subseteq (\text{BELT}(q, q') \setminus \text{RECT}(q, q', j_{q, q'})) \cap \text{RECT}(q, q', j_{q, q'} + k_{q, q'}). \end{aligned}$$

The above characterization leads to the following naive decision procedure, which works in EXPSPACE: Guess the description of a candidate relation R for the simulation relation, verify that it is a simulation and check if it contains the input pair of configurations.

Checking whether the input pair is in the (semilinear) relation R is trivial. To verify that the relation R is a simulation, one needs to check the *simulation condition* for every pair of configurations $(qn, q'n')$ in R , i.e., Duplicator can ensure that after playing one round of the Simulation Game, the resulting pair of configurations is still in R .

The simulation condition is local in the sense that it refers only to positions with neighbouring counter values (plus/minus 1). This, together with the fact that belts are

disjoint outside L_0 , implies that the complete one-neighbourhoods of points in the periodic part repeats along the belt. It therefore suffices to examine those elements which are in the EXPSPACE description to check if the simulation condition holds.

A PSPACE procedure. The naive algorithm outlined above may easily be turned into a PSPACE algorithm by a standard shifting window trick. Instead of guessing the complete exponential-size description upfront, we start by guessing the polynomially bounded relation inside L_0 and verifying it locally. Next, the procedure stepwise guesses parts of the relations $\text{APERIODIC}(q, q')$ and later $\text{PERIODIC}(q, q')$, inside a polynomially bounded rectangle window through the belt and shifts this window along the belt, checking the simulation condition for all contained points on the way. Since the simulation condition is local, everything outside this window may be forgotten, save for the first repetitive window that is used as a certificate for successfully having guessed a consistent periodic set, once it repeats. Because this repetition needs to occur after an exponentially bounded number of shifts, polynomial space is sufficient to store a binary counter that counts the number of shifts and allows to terminate unsuccessfully once the limit is reached. ◀

6 Application to Weak Simulation Checking

A natural extension of simulation is *weak simulation*, that abstracts from internal steps.

► **Definition 17.** For a LTS over actions $A \cup \{\tau\}$ define *weak* step relations by $\xrightarrow{\tau} = \xrightarrow{\tau}^*$ and $\xrightarrow{a} = \xrightarrow{\tau}^* \xrightarrow{a} \xrightarrow{\tau}^*$ for $a \neq \tau$. Weak simulation (\preceq) is now defined just like \preceq , using Simulation Games, in which Duplicator moves along weak steps.

For systems without τ -labelled transitions, $\xrightarrow{a} = \xrightarrow{a}$ and therefore strong and weak simulation coincide. The PSPACE lower bound from [10] for checking strong simulation thus also holds for weak simulation checking over OCN.

Weak simulation has recently been shown to be decidable for OCN [4]. The main obstacle was that Duplicator's system is infinitely branching w.r.t. the weak \xrightarrow{a} steps, which implies that non-simulation does not necessarily manifest itself locally.

In [4], this problem is resolved by constructing a monotone decreasing sequence of semilinear *approximant relations* that converges to weak simulation at a finite index. The approximant relations are derived from a symbolic characterization of Duplicator's infinitely-branching system. They can be computed inductively by characterizing them in terms of strong simulation over suitably modified OCN. The fact that one can effectively compute semilinear descriptions of \preceq over OCN [5] allows to successively compute the approximant relations and to detect convergence of the sequence.

Here we show that the polynomial bounds from Theorem 4, together with the technique from [4], imply a PSPACE upper bound even for checking *weak* simulation on OCN. In particular, we claim that the sizes of the “suitably modified OCN” mentioned above, which characterize the approximants, are in fact polynomial for every index $i \in \mathbb{N}$ in the sequence. A more detailed analysis can be found in Appendix B.

► **Theorem 18.** *Checking weak simulation preorder on OCN is PSPACE-complete.*

7 Conclusion

We have shown that both strong and weak simulation preorder checking between two given OCN processes is PSPACE-complete. Moreover, it is possible to compute representations of

the entire simulation relations as semilinear sets, but these require exponential space. One cannot expect polynomial-size representations of the relations as semilinear sets, because otherwise one could first guess the representation and then verify in $coNP^{NP}$ (for strong simulation) that there are no counterexamples to the local simulation condition. This would yield an algorithm in Σ_p^3 in the polynomial hierarchy, which (under standard assumptions in complexity theory) contradicts the PSPACE-hardness of the problem.

References

- 1 P.A. Abdulla and K. Cerans. Simulation is decidable for one-counter nets (extended abstract). In *CONCUR*, volume 1466 of *LNCS*, pages 253–268, 1998.
- 2 S. Böhm, S. Göller, and P. Jančár. Bisimilarity of one-counter processes is PSPACE-complete. In *CONCUR*, volume 6269 of *LNCS*, 2010.
- 3 R.J. van Glabbeek. The linear time – branching time spectrum I; the semantics of concrete, sequential processes. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001.
- 4 P. Hofman, R. Mayr, and P. Totzke. Decidability of weak simulation on one-counter nets. In *Proc. of LICS 2013*. IEEE, 2013.
- 5 P. Jančár, A. Kučera, and F. Moller. Simulation and bisimulation over one-counter processes. In *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science*, volume 1770 of *LNCS*, pages 334–345, London, UK, 2000. Springer-Verlag.
- 6 P. Jančár and F. Moller. Simulation of one-counter nets via colouring. Technical report, Uppsala Computing Science Research Report 159, February 1999.
- 7 P. Jančár, F. Moller, and Z. Sawa. Simulation problems for one-counter machines. In *SOFSEM*, volume 1725 of *LNCS*, pages 404–413, 1999.
- 8 A. Kučera and P. Jančár. Equivalence-checking on infinite-state systems: Techniques and results. *TPLP*, 6(3):227–264, 2006.
- 9 R. Mayr. Undecidability of weak bisimulation equivalence for 1-counter processes. In *ICALP*, volume 2719 of *LNCS*, pages 570–583, 2003.
- 10 J. Srba. Beyond language equivalence on visibly pushdown automata. *Logical Methods in Computer Science*, 5(1):1–22, 2009.
- 11 L.G. Valiant. *Decision procedures for families of deterministic pushdown automata*. PhD thesis, Department of Computer Science, University of Warwick, Coventry, July 1973.

A

 Missing Proofs from Sections 4 and 5

A.1 Proof of Lemma 10

Suppose Spoiler wins the Slope Game from $((q, q'), (\rho, \rho'))$ using a strategy of segment depth d . A position in the Slope Game contains a positive vector (ρ, ρ') , while a position in the Simulation Game contains a pair $(n, n') \in \mathbb{N} \times \mathbb{N}$ of counter values, that can also be interpreted as a positive vector. We will derive a strategy for Spoiler in the Simulation Game that is winning from all positions $(qn, q'n')$ where (n, n') is $(K \cdot d)$ -below (ρ, ρ') . The crucial idea of the proof is to consider the segments of the supposed winning strategy in the Slope Game separately. Each such segment is a strategy for one phase and as such, describes how to move in the Simulation Game until the next lasso is observed. Afterwards, Spoiler can chose to continue playing according to the next lower segment, or “roll back” the cycle and continue playing according to the current segment. By the rules of the Slope Game we observe that after sufficiently many such rollbacks the difference between the ratio n/n' of the actual counters and the slope of the next lower segment is negligible, i.e., these vectors are equivalent in the sense of Definition 13 in Section 4.2. Then, Spoiler can safely continue to play according to the next lower segment at level $d - 1$.

To safely play such a strategy in the Simulation Game, Spoiler needs to ensure that her own counter does not decrease too much as that could restrict her ability to move. We observe however, that any partial play that “stays in some segment” at height d , can be decomposed into a single acyclic prefix plus a number of cycles. Such a play therefore preserves the invariant that all visited points are $K \cdot (d - 1)$ -below the slope of the phase. In particular, this means that Spoiler’s counter is always $\geq K \cdot (d - 1)$.

Formally, the proof of Lemma 10 proceeds by induction on the segment depth d .

Case $d = 1$. This means that Spoiler has a strategy to win the Slope Game in the first phase, and hence to enforce that the effect of all cycles is behind (ρ, ρ') but not positive. Denote this strategy by T . In the Simulation Game, Spoiler will re-use this strategy as we describe below. At every position $(qn, q'n')$ in the Simulation Game, Spoiler keeps a record of the *corresponding position* $(\pi, (\rho, \rho'))$ in the Slope Game, enforcing the invariant that (q, q') are the ending states of the path π .

From the initial position $(qn, q'n')$ with corresponding position $((q, q'), (\rho, \rho'))$, Spoiler starts playing the Simulation Game according to T , until the path in the corresponding position of the Slope Game, say π_1 , describes a lasso (this must happen after at most K rounds). Thus π_1 splits into:

$$\pi_1 = \tilde{\pi}_1 \bar{\pi}_1 \tag{9}$$

where the suffix $\bar{\pi}_1$ is a cycle. Denote by $(\tilde{\alpha}_1, \tilde{\alpha}'_1)$ and $(\bar{\alpha}_1, \bar{\alpha}'_1)$ the effects of $\tilde{\pi}_1$ and $\bar{\pi}_1$, respectively. The current values of counters are clearly

$$n + \tilde{\alpha}_1 + \bar{\alpha}_1 \quad \text{and} \quad n' + \tilde{\alpha}'_1 + \bar{\alpha}'_1 \tag{10}$$

assuming that the play did not end by now with Spoiler’s win. As the length of path π_1 is at most K and (n, n') is assumed to be K -below (ρ, ρ') , we know that all positions visited by now in the Simulation Game were below (ρ, ρ') . In particular, Spoiler’s counter value was surely non-negative by now.

Now Spoiler *rolls back* the cycle $\bar{\pi}_1$, namely changes the corresponding position in the Slope Game from $(\pi_1, (\rho, \rho'))$ to $(\tilde{\pi}_1, (\rho, \rho'))$ and continues playing according to T . The play

continues until Spoiler wins or the path in the corresponding position of the Slope Game, say π_2 , is a lasso again. Again, we split the path into an acyclic prefix and a cycle:

$$\pi_2 = \tilde{\pi}_2 \bar{\pi}_2. \quad (11)$$

Denote the respective effects by $(\tilde{\alpha}_2, \tilde{\alpha}'_2)$ and $(\bar{\alpha}_2, \bar{\alpha}'_2)$. A crucial but simple observation is that, assuming that the play did not end by now with Spoiler's win, the current values of counters are now

$$n + \tilde{\alpha}_2 + \bar{\alpha}_1 + \bar{\alpha}_2 \quad \text{and} \quad n' + \tilde{\alpha}'_2 + \bar{\alpha}'_1 + \bar{\alpha}'_2, \quad (12)$$

i.e. the effect $(\tilde{\alpha}_1, \tilde{\alpha}'_1)$ of $\tilde{\pi}_1$ does not contribute any more. As $(\bar{\alpha}_1, \bar{\alpha}'_1)$ is behind (ρ, ρ') we may apply Lemma 6 to $(\bar{\alpha}_1, \bar{\alpha}'_1)$ and $c = 0$ in order to deduce, similarly as before, that all positions by now were below (ρ, ρ') . Now Spoiler rolls back $\bar{\pi}_2$ by establishing $(\tilde{\pi}_2, (\rho, \rho'))$ as the new corresponding position in the Slope Game. Continuing in this way, after k rollbacks the counter values are:

$$\begin{aligned} n + \tilde{\alpha}_k + (\bar{\alpha}_1 + \bar{\alpha}_2 + \dots + \bar{\alpha}_{k-1}) + \bar{\alpha}_k & \quad \text{and} \\ n' + \tilde{\alpha}'_k + (\bar{\alpha}'_1 + \bar{\alpha}'_2 + \dots + \bar{\alpha}'_{k-1}) + \bar{\alpha}'_k, \end{aligned} \quad (13)$$

assuming that Spoiler did not win earlier. All the vectors $(\bar{\alpha}_i, \bar{\alpha}'_i)$, and thus also the sum

$$(\bar{\alpha}_1 + \bar{\alpha}_2 + \dots + \bar{\alpha}_{k-1}, \bar{\alpha}'_1 + \bar{\alpha}'_2 + \dots + \bar{\alpha}'_{k-1}) \quad (14)$$

are behind (ρ, ρ') , hence similarly as before all positions by now have been below (ρ, ρ') , by Lemma 6 applied to the vector (14) above.

This in particular means that Spoiler's counter remains above value c . However, as by assumption all observed cycles come from a final segment in her Slope Game strategy, the vector (14) cannot be positive for any k . Thus, every rollback strictly decreases Duplicator's counter value. We conclude that after sufficiently many rollbacks, Duplicator's counter must eventually drop below 0 and hence Spoiler eventually wins.

Case $d > 1$. By assumption, Spoiler has a strategy T for the Slope Game, which has segment depth $d > 0$. As before, Spoiler's strategy in the Simulation Game will re-use the strategy T from the Slope Game, using rollbacks.

Spoiler plays according to the initial segment of this strategy, that allows her to win or at least guarantee that the effect of the first observed lasso's circle is less steep than (ρ, ρ') . After l rollbacks, the counter values will be of the form:

$$\begin{aligned} n + \tilde{\alpha} + (\bar{\alpha}_1 + \dots + \bar{\alpha}_m) + (\tilde{\gamma}_1 + \dots + \tilde{\gamma}_l) & \quad \text{and} \\ n' + \tilde{\alpha}' + (\bar{\alpha}'_1 + \dots + \bar{\alpha}'_m) + (\tilde{\gamma}'_1 + \dots + \tilde{\gamma}'_l), \end{aligned} \quad (15)$$

where the absolute values of $\tilde{\alpha}$ and $\tilde{\alpha}'$ are at most K , the vectors $(\tilde{\gamma}_i, \tilde{\gamma}'_i)$ are behind (ρ, ρ') and positive, and the vectors $(\bar{\alpha}_i, \bar{\alpha}'_i)$ are behind (ρ, ρ') and non-positive. We apply Lemma 6 to $c = K \cdot (d - 1)$ and learn that all the positions by now have been $(K \cdot (d - 1))$ -below (ρ, ρ') .

In general Spoiler has no power to choose whether a effect of a cycle at a next rollback is positive or not. However, if from some point on all effects are non-positive then Duplicator's counter eventually drops below 0 and Spoiler wins. Thus w.l.o.g; we focus on positions in the Simulation Game immediately after a rollback of a cycle with positive effect. Using the notation from (15), suppose (γ_l, γ'_l) is the effect of the last rolled back cycle. We need the following claim in order to apply the induction assumption:

► **Claim 1.** After sufficiently many rollbacks the vector (\bar{n}, \bar{n}') of current counter values in the Simulation Game is $(K \cdot (d - 1))$ -below some vector (γ, γ') which is equivalent to the positive effect (γ_l, γ'_l) of the last rolled back cycle.

Proof. By an easy geometric argument. Ignore vectors (α_i, α'_i) as they preserve being $(K \cdot (d - 1))$ -below all positive vectors that are less steep than (ρ, ρ') . If Duplicator wants to falsify the condition, he would need to increase the steepness of the rolled back cycle infinitely often, which is clearly impossible as there are only finitely many simple cycles. ◀

Let $(\bar{q}\bar{n}, \bar{q}'\bar{n}')$ be a position of the Simulation Game satisfying the claim. We know that Spoiler has a winning strategy in the Slope Game from $((\bar{q}, \bar{q}'), (\gamma_l, \gamma'_l))$, of segment depth at most $d - 1$. Because (γ_l, γ'_l) is equivalent to (γ, γ') , we can apply Lemma 14 and know that the same strategy is winning in the Slope Game from $((\bar{q}, \bar{q}'), (\gamma, \gamma'))$. By the induction assumption we conclude that Spoiler wins the Simulation Game from $(\bar{q}\bar{n}, \bar{q}'\bar{n}')$, which completes the proof of Lemma 10. ◀

A.2 Proof of Lemma 11

Suppose Duplicator wins the Slope Game from $((q, q'), (\rho, \rho'))$ using a strategy of segment depth d . We will show that Duplicator wins the Simulation Game from every position $(qn, q'n')$ where (n, n') is $(K \cdot d)$ -above (ρ, ρ') . We will again build on the concept of rollbacks and proceed by induction on d .

Case $d = 1$. In this case, Duplicator has a strategy to win the Slope Game immediately after the first phase. This means he can enforce that the effects of the cycles of all observed lassos are not behind (ρ, ρ') . By a straightforward induction using part 2 of Lemma 6 one can show that Duplicator can preserve the invariant that all visited points are K -above (ρ, ρ') . This in particular means that his counter value stays positive and he wins by enforcing an infinite play.

Case $d > 1$. Let T denote the initial segment of Duplicator's strategy in the Slope Game. Every effect of a cycle in T is either not behind (ρ, ρ') or behind (ρ, ρ') , but positive.

In the Simulation Game, Duplicator will play according to this initial segment T , using rollbacks, as long as the effect of the rolled back cycle is not behind (ρ, ρ') . Just as in the previous case, we can apply part 2 of Lemma 6 for $c = K \cdot d$ and derive that in this way, Duplicator is able to keep the current counter values $(K \cdot d)$ -above (ρ, ρ') .

Suppose that after a few iterations, the effect (α, α') of the last cycle is behind (ρ, ρ') and let $(\bar{q}\bar{n}, \bar{q}'\bar{n}')$ be the position in the Simulation Game directly afterwards. In this case, (α, α') is clearly positive and less steep than (ρ, ρ') . Now the point described by the counter values before this last cycle was $(K \cdot d)$ -above (ρ, ρ') and because the cycle is no longer than K , we know that the point (\bar{n}, \bar{n}') of current counter values (after the cycle) is still $(K \cdot (d - 1))$ -above (ρ, ρ') . This means, as $(\alpha, \alpha') \prec (\rho, \rho')$, that (\bar{n}, \bar{n}') is also $(K \cdot (d - 1))$ -above (α, α') .

Knowing that Duplicator has a winning strategy in the Slope Game from $((\bar{q}, \bar{q}'), (\alpha, \alpha'))$ of segment depth at most $d - 1$, by induction assumption we obtain a winning strategy for Duplicator in the Simulation Game from $(\bar{q}\bar{n}, \bar{q}'\bar{n}')$. This completes the description of Duplicator's winning strategy from $(qn, q'n')$ and hence also the proof of Lemma 11. ◀

A.3 Proof of Lemma 16

For technical convenience we assume w.l.o.g. that no belt contains the upper right corner of L_0 (this can always be achieved by minimally extending L_0 , if necessary.) Thus every belt intersects either the horizontal, or the vertical border of L_0 , but not both.

Recall that the non-parallel belts do not overlap/interfere with each other outside L_0 , hence we can consider them separately. For the rest of the proof fix states q, q' and let $(\rho, \rho') = \text{SLOPE}(q, q')$. W.l.o.g. suppose that $\text{BELT}(q, q')$ intersects the horizontal border of L_0 (if it intersects the vertical border of L_0 the proof is analogous).

For simplicity we assume that no other belt is parallel to $\text{BELT}(q, q')$. The proof below may be easily adapted to the general case by considering a bunch of parallel belts jointly, instead of just the single one $\text{BELT}(q, q')$.

By a *cross-section* at level n' we mean the intersection of $\preceq_{q, q'}$ with two consecutive horizontal lines at that level, i.e. with $\{(n, n'), (n, n'+1) : n \in \mathbb{N}\}$. We may assume that cross-sections are always non-empty (this can always be ensured by slightly widening $\text{BELT}(q, q')$ if necessary). We say that two cross-sections s_1 and s_2 are *equal* if one of them is obtained by a shift of the other by a multiple of (ρ, ρ') ; formally, we require for some $k \in \mathbb{N}$,

$$s_1 + k \cdot (\rho, \rho') = s_2. \quad (16)$$

Choose two cross-sections s_1, s_2 at levels n'_1 and n'_2 respectively, and $k > 0$ that satisfies (16). Let P be the restriction of $\preceq_{q, q'}$ to the area between s_1 and s_2 , and A be the restriction of $\preceq_{q, q'}$ to the area below s_1 :

$$A = \{(n, n') \in \preceq_{q, q'} : n' < n'_1\} \quad P = \{(n, n') \in \preceq_{q, q'} : n'_1 \leq n' < n'_2\}.$$

Recall that A and P , similarly as $\preceq_{q, q'}$, are subsets of $\text{BELT}(q, q')$. We claim:

► **Lemma 19.** *For every s_1, s_2 and $k > 0$ satisfying (16),*

$$\preceq_{q, q'} = A \cup P^*, \quad \text{where } P^* = \bigcup_{i \in \mathbb{N}} (P + i \cdot k \cdot (\rho, \rho')).$$

Before proving this lemma note that it implies Lemma 16. Indeed, by Theorem 4, a cross-section contains polynomially many points, and therefore there are at most exponentially many non-equal cross sections. Thus, by the pigeonhole principle, there are surely two equal cross-sections at exponentially bounded levels n'_1 and n'_2 .

Now we prove Lemma 19. The proof strongly relies on the locality of the simulation condition. We first claim one inclusion of Lemma 19, namely:

► **Claim 2.** $A \cup P^* \subseteq \preceq_{q, q'}$.

Proof. We show that the following relation is a simulation:

$$R = \preceq \setminus \{(qn, q'n') : (n, n') \in \text{BELT}(q, q')\} \cup \{(qn, q'n') : (n, n') \in A \cup P^*\}.$$

(Roughly speaking, R is obtained from \preceq by replacing $\preceq_{q, q'}$ with $A \cup P^*$.) We claim that R is a simulation, relying on the locality of the simulation condition. Formally, we define the *relative R -neighborhood* of a point (n, n') as

$$\{(pl, p'l') : (p(n+l), p'(n'+l')) \in R, (p, p') \in Q \times Q', l, l' \in \{-1, 0, 1\}\}.$$

Note that the simulation condition for a pair of configurations $(qn, q'n')$ with respect to the relation R only depends on the relative R -neighborhood of (n, n') . Similarly, one defines the relative \preceq -neighborhood of a point (n, n') .

By the definition of cross-section and of the sets A and P , the relative R -neighborhood of a point $(n, n') \in R$ equals the relative \preceq -neighborhood of some (possibly other) point in $\preceq_{q, q'}$. Thus we deduce that every pair in R satisfies the simulation condition wrt. R , i.e. R is a simulation. As \preceq is the largest simulation, the claim follows. \blacktriangleleft

In order to show the other inclusion of Lemma 19, extend n'_1 and n'_2 to an infinite arithmetic progression

$$n'_1, n'_2, n'_3, \dots,$$

i.e. $n_{i+1} = n'_i + k \cdot \rho'$ for $i \geq 1$, and consider the “segments” P_i of $\preceq_{q, q'}$ defined by the corresponding cross-sections:

$$P_i = \{(n, n') \in \preceq_{q, q'} : n'_i \leq n' < n'_{i+1}\} \quad \text{for } i \geq 1.$$

Clearly, $P = P_1$ and $\preceq_{q, q'} = A \cup \bigcup_{i \geq 1} P_i$. By Claim 2 it follows that $P_1 + k \cdot (\rho, \rho') \subseteq P_2$, or equivalently $P_1 \subseteq P_2 - k \cdot (\rho, \rho')$. Analogously one shows:

$$P_i \subseteq P_{i+1} - k \cdot (\rho, \rho') \quad \text{for every } i \geq 1. \quad (17)$$

We claim that the inclusions are actually equalities:

► **Claim 3.** $P_i = P_{i+1} - k \cdot (\rho, \rho')$, for every $i \geq 1$.

Proof. Due to Equation (17), it suffices to show the inclusions $P_{i+1} - k \cdot (\rho, \rho') \subseteq P_i$. The inclusions follow, similarly as in the proof of Claim 2, from the observation that the following relation is a simulation:

$$R = \preceq \setminus \{(qn, q'n') : (n, n') \in \text{BELT}(q, q')\} \cup \{(qn, q'n') : (n, n') \in A \cup \bigcup_{i \geq 2} P_i - k \cdot (\rho, \rho')\}.$$

The relation R is obtained from \preceq , roughly speaking, by removing the first segment P_1 and shifting all other segments P_i by vector $-k \cdot (\rho, \rho')$. To prove that R is a simulation, we exploit locality of the simulation condition exactly as before. Additionally, we use the observation that the simulation condition is monotonic with respect to inclusion of relative neighborhoods, together with the inclusions (17). \blacktriangleleft

Claim 3 immediately implies Lemma 19 and thus Lemma 16.

B Weak Simulation Checking

We show that the bounds on the coefficients of the Belt Theorem, as derived in Section 4, imply that the construction from [4] for checking *weak* simulation uses only polynomial space.

In order to avoid repeating the involved construction from [4], we refer the reader to the original paper for technical details and recover here only those notions and properties which suffice to provide some intuition and derive the claimed PSPACE bound.

We aim to compute a description of \preceq , the largest weak simulation over a given pair of OCN. First we reduce this weak simulation game to a strong simulation game between two modified systems.

► **Definition 20** (ω -Nets). An ω -net $\mathcal{M} = (Q, A, \delta)$ is given by a finite set of control-states Q , a finite set of actions A and transitions $\delta \subseteq Q \times A \times \{-1, 0, 1, \omega\} \times Q$. It induces a transition system over the stateset $Q \times \mathbb{N}$ that allows a step $pm \xrightarrow{a} p'm'$ if either $(p, a, d, p') \in \delta$ and $m' = m + d \in \mathbb{N}$ or if $(p, a, \omega, p') \in \delta$ and $m' > m$.

► **Lemma 21** ([4]). For two OCN \mathcal{N} and \mathcal{N}' with sets of control-states Q and Q' resp., one can construct a OCN \mathcal{M} with states $Q_{\mathcal{M}} \supseteq Q$ and an ω -net \mathcal{M}' with states $Q_{\mathcal{M}'} \supseteq Q'$, such that for each pair $(q, q') \in Q \times Q'$ of original control states,

$$qn \preceq q'n' \text{ w.r.t. } \mathcal{N}, \mathcal{N}' \text{ iff } qn \preceq q'n' \text{ w.r.t. } \mathcal{M}, \mathcal{M}'. \quad (18)$$

Moreover, the sizes of \mathcal{M} and \mathcal{M}' are polynomial in the size of \mathcal{N} and \mathcal{N}' .

Thus, it suffices to compute a description of the strong simulation relation relative to a given OCN $\mathcal{M} = (Q, A, \delta)$ and an ω -net $\mathcal{M}' = (Q', A, \delta')$. To do that, we construct a sequence of successively decreasing (w.r.t. set inclusion) approximant relations \preceq^i and show that 1) for all $i \in \mathbb{N}$, \preceq^i is effectively semilinear and 2) there is some $k \in \mathbb{N}$ with $\preceq^k = \preceq^{k+1} = \preceq$, i.e., the sequence converges to simulation preorder at some finite level k .

Intuitively, \preceq^i is given by a *parameterized simulation game* that keeps track of how often Duplicator uses ω -labelled transitions and in which Duplicator immediately wins if he plays such a step the i th time. It is easy to see that this game favours Duplicator due to the additional winning condition. With growing index i , this advantage becomes less important and the game increasingly resembles a standard simulation game. Hence, $\forall i \in \mathbb{N}$, $\preceq^i \supseteq \preceq^{i+1}$.

In [4], it is shown that these approximants \preceq^i can in fact be characterized by equivalent (in the sense of Lemma 22 below) ordinary strong simulation relations between suitably extended OCN.

► **Lemma 22.** *There is a sequence $(\mathcal{S}_i, \mathcal{S}'_i)$ of pairs of OCN such that for all indices $i \in \mathbb{N}$:*

1. \mathcal{S}_i and \mathcal{S}'_i contain all states of \mathcal{M} and \mathcal{M}' respectively.
2. For all configurations $qn \in (Q \times \mathbb{N})$ and $q'n' \in (Q' \times \mathbb{N})$ of \mathcal{M} and \mathcal{M}' it holds that $qn \preceq^i q'n'$ w.r.t. $\mathcal{M}, \mathcal{M}'$ iff $qn \preceq q'n'$ w.r.t. $\mathcal{S}_i, \mathcal{S}'_i$.
3. \mathcal{S}_{i+1} and \mathcal{S}'_{i+1} can be computed from \mathcal{S}_i and \mathcal{S}'_i alone.

The above conditions ensure decidability of weak simulation as they allow to iteratively compute the approximants and detect convergence, by the effective semilinearity of strong simulation over OCN [5].

To obtain an upper bound for the complexity of this procedure, we will bound the sizes of all $(\mathcal{S}_i, \mathcal{S}'_i)$ polynomially in the sizes of \mathcal{M} and \mathcal{M}' . To do that, we recall some more properties of the construction, starting by describing how the nets \mathcal{S}_i and \mathcal{S}'_i look like.

The nets \mathcal{S}_i and \mathcal{S}'_i

These nets are constructed using the notion of *minimal sufficient values*:

► **Definition 23.** Consider the approximant \preceq^i for some parameter i , which is characterised by nets $\mathcal{S}_i, \mathcal{S}'_i$ (cf. point 2 of Lemma 22 above) and let $(q, q') \in (Q \times Q')$ be a pair of states. By monotonicity, there is a minimal value $\text{suf}(q, q', i) \in \mathbb{N} \cup \{\omega\}$ satisfying

$$\forall n' \in \mathbb{N}. q(\text{suf}(q, q', i)) \not\preceq^i q'n'. \quad (19)$$

Let $\text{suf}(q, q', i)$ be ω if no finite value satisfies this condition.

The idea behind the construction of nets for parameter $i + 1$ is as follows. A Simulation Game played on the arena $\mathcal{S}_{i+1}, \mathcal{S}'_{i+1}$ mimics the $(i + 1)$ -parameterized simulation game played on $\mathcal{M}, \mathcal{M}'$ until Duplicator uses an ω -labelled transition, leading to some game position qn vs. $q'n'$. Afterwards, the parameterized game would continue with the next lower parameter i .

By induction assumption, we can compute a representation of \preceq^i and hence $\text{suf}(q, q', i)$ for every pair (q, q') . Given these values, the nets \mathcal{S}_{i+1} and \mathcal{S}'_{i+1} are constructed so that instead of making steps that are due to ω -labelled transitions, Duplicator can enforce the play to continue in some subgame that he wins iff Spoiler's counter is smaller than the hard-wired value $\text{suf}(q, q', i)$.

This “forcing” of the play can be implemented for OCN simulation using a standard technique called *defender's forcing* (see e.g. [8]). So, the nets \mathcal{S}_i and \mathcal{S}'_i consist of the original nets $\mathcal{M}, \mathcal{M}'$ where all ω -transitions in Duplicator's net \mathcal{M}' are replaced by a small constant defenders-forcing script, leading to the corresponding testing gadgets that test if Spoiler's counter is at least as large as the pre-computed sufficient value and let Spoiler win only if that is the case.

The actual test-gadgets are not very complicated: On Duplicator's side, all gadgets are the same simple loop over a newly introduced symbol, say e . Hence, $\mathcal{S}'_i = \mathcal{S}'_1$ for every i and this new net is polynomial in the size of \mathcal{M}' and \mathcal{M} .

In Spoiler's net \mathcal{S}_i , the gadgets $G(q, q', i)$ for states (q, q') and index i solely depend on the value $\text{suf}(q, q', i)$: If $\text{suf}(q, q', i)$ is finite, it suffices to have a counter-decreasing chain of e -steps of length $\text{suf}(q, q', i)$, leading to some state which enables an action that cannot be replied to by Duplicator. Otherwise, if $\text{suf}(q, q', i) = \omega$ (no counter finite value satisfies Equation 19), Spoiler should always lose, so a simple e -labelled loop can be used as gadget. To conclude, each \mathcal{S}_i essentially consists of \mathcal{M} plus chains $G(q, q', i)$, one for every pair of states (q, q') . We summarize the crucial properties of this construction below.

► **Lemma 24.**

1. $\text{suf}(q, q', 1) = \omega$ for every pair $(q, q') \in Q \times Q'$.
2. $\text{suf}(q, q', i) \geq \text{suf}(q, q', i + 1)$.
3. $(\mathcal{S}_i, \mathcal{S}'_i)$ contains precisely $|Q \times Q'|$ many gadgets, each.
4. If $\text{suf}(q, q', i) \in \mathbb{N}$ then the size of gadget $G(q, q', i)$ is $\text{suf}(q, q', i) + 2$.
5. No chain $G(q, q', i)$ contains transitions leading back to \mathcal{M} .

Using properties 2 and 3 we derive that indeed $(\mathcal{S}_k, \mathcal{S}'_k) = (\mathcal{S}_{k+1}, \mathcal{S}'_{k+1})$, and hence $\preceq^k = \preceq^{k+1} = \preceq$ for some finite $k \in \mathbb{N}$.

Our goal is to bound the sizes of the nets $\mathcal{S}_i, \mathcal{S}'_i$ polynomially in the sizes of $\mathcal{M}, \mathcal{M}'$ and to show that they can indeed be constructed in polynomial space. From point 1 and the fact whenever $\text{suf}(q, q', i) = \omega$, the gadget $G(q, q', i)$ is a trivial loop, we already know that the sizes of \mathcal{S}_1 and \mathcal{S}'_1 are polynomial in $\mathcal{M}, \mathcal{M}'$. Due to the particular shape of the nets $(\mathcal{S}_{i+1}, \mathcal{S}'_{i+1})$, it suffices to bound the values $\text{suf}(q, q', i)$.

Bounding $\text{suf}(q, q', i)$

Observe that $\text{suf}(q, q', i)$ is defined in terms of the approximant \preceq^i , which is characterized as the strong simulation \preceq relative to the nets $\mathcal{S}_i, \mathcal{S}'_i$ by Lemma 22, Point 2. In fact, if we consider the colouring of \preceq w.r.t. $\mathcal{S}_i, \mathcal{S}'_i$, the value $\text{suf}(q, q', i)$ is the width of the belt for (q, q') if this belt is vertical and ω otherwise. Therefore, the value c in the Belt Theorem applied to this colouring bounds all finite $\text{suf}(q, q', i)$.

We show how to bound c using the sharper estimation as formulated in Section 4.3 in terms scc , the maximal size of any strongly connected component and acyc , the length of the longest acyclic path in the product $\mathcal{S}_i \times \mathcal{S}'_i$:

$$c \leq \text{poly}(\text{scc}) + \text{acyc}. \quad (20)$$

This allows us to bound all values $\text{suf}(q, q', i)$ and hence the size of the nets for index $i + 1$.

First, observe that the shape of all $\mathcal{S}_i, \mathcal{S}'_i$ (particularly Point 5 of Lemma 24, and the fact that $\forall_{i \in \mathbb{N}} \mathcal{S}'_i = \mathcal{S}'_1$) implies that the strongly connected components are unchanged from index $i = 1$ onward. Thus, scc is in fact polynomial in $\mathcal{M}, \mathcal{M}'$. Secondly, any path in the product $\mathcal{S}_i \times \mathcal{S}'_i$ can be split into two (possibly empty) parts: the part that remains in $\mathcal{M} \times \mathcal{M}'$ and a suffix that moves into at most one gadget $G(q, q', i)$. Since the maximal length of paths in $G(q, q', i)$ is bounded by $\text{suf}(q, q', i)$, we can bound acyc as follows.

$$\text{acyc} \leq |\mathcal{M} \times \mathcal{M}'| + \max\{\text{suf}(q, q', i) \in \mathbb{N} \mid (q, q') \in Q \times Q'\}. \quad (21)$$

Let W_i denote the maximal width of all vertical belts at level i , i.e., the largest finite value $\text{suf}(q, q', i)$ over all (q, q') . By the argument above, we get for all indices $i \in \mathbb{N}$,

$$W_{i+1} \leq \text{poly}(\mathcal{M}, \mathcal{M}') + W_i. \quad (22)$$

Now, from properties 2 and 4 of Lemma 24 we can deduce that there are no more than $K = |Q \times Q'|$ indices i such that $W_{i+1} \geq W_i$. This is because the size of the value $\text{suf}(q, q', i)$ for a particular pair (q, q') can only increase once, going from index i to $i + 1$ if $\text{suf}(q, q', i) = \omega > \text{suf}(q, q', i + 1)$. Therefore, we can bound W_i , and thus values $\text{suf}(q, q', i)$, for all indices $i \in \mathbb{N}$ by

$$W_i \leq K \cdot (\text{poly}(|\mathcal{M} \times \mathcal{M}'|) + 1). \quad (23)$$

We conclude that the sizes of all $\mathcal{S}_i, \mathcal{S}'_i$ are polynomial in the sizes of \mathcal{N} and \mathcal{N}' . It remains to show that we can compute these values in polynomial space, because this allows us to effectively construct the nets for the next parameter $i + 1$.

Computing $\text{suf}(q, q', i)$

We analyse the colouring of the simulation \preceq relative to the one-counter nets \mathcal{S}_i and \mathcal{S}'_i . In particular we need to answer the following questions, for each given pair of states $(q, q') \in Q \times Q'$,

1. Is the belt for (q, q') vertical? And if yes,
2. What is its exact width?

By Theorem 4, we can bound all ratios (ρ, ρ') , which are the slopes of belts polynomially. Let (ρ, ρ') be the ratio of the steepest belt with $\rho' > 0$. Recall that c bounds the width of all vertical belts. To answer the first question, it suffices to check the colour of some point

(n, n') that is both c -above (ρ, ρ') and c -below of $(0, 1)$, i.e., $n > c$. For instance, $n = c + 1$ and $n' = 2(c + 1)(\frac{\rho}{\rho'})$ is surely such a point.

If the belt for (q, q') is vertical, then by Theorem 4, Point 2, we have $qn, \not\approx q'n'$. Otherwise, if the belt is not vertical, then by point 1 of Theorem 4, we must have $qn, \approx q'n'$.

To answer the second question, we consider the periodicity description of the colouring (cf. Section 5). Although this description is of exponential size and we thus cannot fully keep it in memory, we can, in polynomial space, compute point queries. Moreover, we know that the colouring in any belt is described by some non-trivial initial colouring and is repetitive from some exponentially bounded level onwards. Thus, if we consider the vertical belt for states (q, q') , from some level n'_0 , the colouring stabilizes so that for all $n' \geq n'_0$, we have $qn \approx q'n'$ iff $n < \text{suf}(q, q', i)$.

We can now iteratively check the colour of the point (n, n'_0) for decreasing values $n = c$ to 0 and some fixed, but sufficiently high n'_0 . By Theorem 2, this can surely be done in polynomial space. $\text{suf}(q, q', i)$ must be the largest considered $n < c$ where $qn \not\approx q'n'_0$ still holds.