

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 [5], 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 [8], 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 [7, 6], 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, 7, 6] 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 [5], 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 [5], 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 [8]	PSPACE [2]	undecidable [8]	undecidable [9]	undecidable [5]

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 [7, 6] 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.

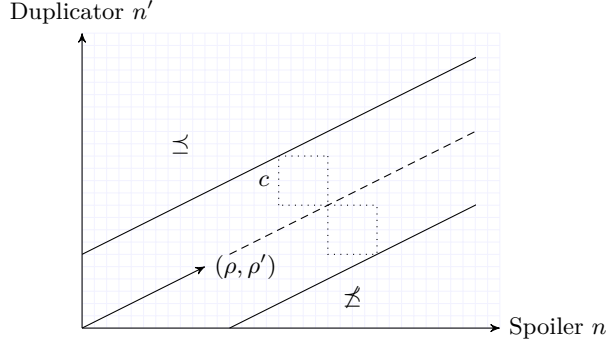
3 Polynomially Bounded Belts

Let us fix two OCN \mathcal{N} and \mathcal{N}' , with sets of control-states Q and Q' , respectively. Following [6], 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 [6] (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.

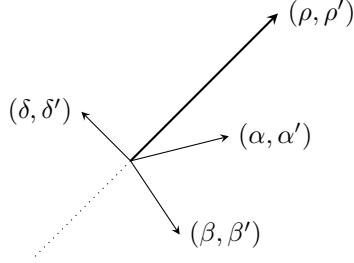
Positive vectors may be naturally ordered: We will call (ρ, ρ') *steeper* than (α, α') , written $(\alpha, \alpha') \prec (\rho, \rho')$, if (α, α') is behind (ρ, ρ') .

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.

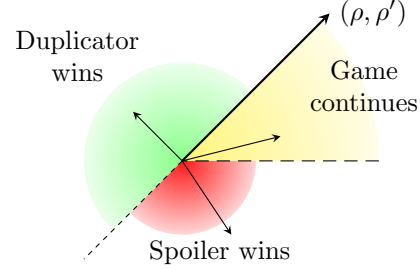
► **Lemma 6.** *Let (ρ, ρ') be a positive vector and $c, n, n' \in \mathbb{N}$.*

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 [4], 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, 7, 6].

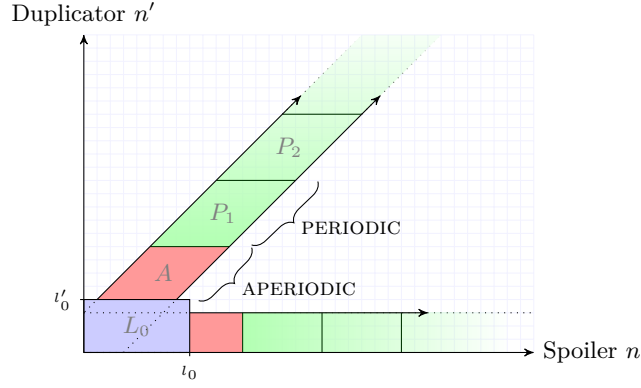
As before we fix two OCN \mathcal{N} and \mathcal{N}' , with sets of control-states Q and Q' , respectively. By Lemma 9, Point 2, we can compute in PSPACE, for every pair $(q, q') \in Q \times Q'$, the positive vector (ρ, ρ') satisfying Theorem 4; we denote this vector by $\text{SLOPE}(q, q')$. We define $\text{BELT}(q, q')$ 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 [6], 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 [5]. 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 [5], 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 [6] 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 [5], 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 [4], 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. Jančar. 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, S. Lasota, R. Mayr, and P. Totzke. Simulation over one-counter nets is PSPACE-complete. Technical Report EDI-INF-RR-1418, University of Edinburgh, 2013.
- 5 P. Hofman, R. Mayr, and P. Totzke. Decidability of weak simulation on one-counter nets. In *Proc. of LICS 2013*. IEEE, 2013.
- 6 P. Jančar, 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.
- 7 P. Jančar and F. Moller. Simulation of one-counter nets via colouring. Technical report, Uppsala Computing Science Research Report 159, February 1999.
- 8 P. Jančar, F. Moller, and Z. Sawa. Simulation problems for one-counter machines. In *SOFSEM*, volume 1725 of *LNCS*, pages 404–413, 1999.
- 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.