Inclusion Problems for One-Counter Systems

Patrick Totzke

Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2014

Abstract

We study the decidability and complexity of verification problems for infinite-state systems.

A fundamental question in formal verification is if the behaviour of one process is reproducible by another. This inclusion problem can be studied for various models of computation and behavioural preorders. It is generally intractable or even undecidable already for very limited computational models.

The aim of this work is to clarify the status of the decidability and complexity of some well-known inclusion problems for suitably restricted computational models. In particular, we address the problems of checking strong and weak simulation and trace inclusion for processes definable by one-counter automata (OCA), that consist of a finite control and a single counter ranging over the non-negative integers. We take special interest of the subclass of one-counter nets (OCNs), that cannot fully test the counter for zero and which is subsumed both by pushdown automata and Petri nets / vector addition systems.

Our new results include the PSPACE-completeness of strong and weak simulation, and the undecidability of trace inclusion for OCNs. Moreover, we consider semantic preorders between OCA/OCN and finite systems and close some gaps regarding their complexity. Finally, we study deterministic processes, for which simulation and trace inclusion coincide.

Acknowledgements

Above all, I'd like to thank my supervisors Colin Stirling and Richard Mayr for their continuous support, patience and encouragement. They introduced me to a colourful area of research, and always provided detailed and valuable feedback on my work, much of which is based on their earlier contributions.

Thanks to everybody at LFCS for creating such a welcoming and enjoyable working environment, especially Fridays at 4pm. I am particularly grateful for helpful comments from Kousha Etessami, Mary Cryan, Andras Salamon, and the many others I had the pleasure to talk to during social or academic events in the past three years. Cheers to my fellow PhD students, who were simply great fun to be around and surely prevented me from going mad at times. Ben, Lorenzo, Juan, Julian, Chris, Daniel, Karoliina and Fabian: rock on!

I was lucky to get the chance to visit and work with colleagues in Warsaw, Uppsala and Bordeaux. I thank Sławomir Lasota, Mikołaj Bojańczyk, Wojciech Czerwiński, Piotr Hofman, Parosh Abdulla, Mohamed Faouzi Atig, Grégoire Sutre and Jérôme Leroux for inviting me over and for many interesting and inspiring discussions.

A special thanks goes to Piotrek Hofman, who suffered through writing papers with me and never tired of discussing even the silliest problems for long hours via video chat. I thank him and Wojtek for showing me some tanks, proper soviet era restaurants/milk bars and many other lovely spots in Warsaw. I enjoyed the amazing hospitality of Jari & Nilla, who made sure I had a fantastic time in Uppsala. Thanks guys!

I gratefully acknowledge financial support from EPSRC, SICSA and LFCS who jointly funded my stay in Edinburgh and also summer schools, workshops and conferences I attended.

I am incredibly grateful to have found Nicole, my long-term partner, best friend and soulmate. Thank you for following me into the unknown and brightening up my every day!

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise, and that this work has not been submitted for any other degree or professional qualification except as specified.

This thesis also incorporates the outcome of joint research undertaken in collaboration with Piotr Hofman and Richard Mayr. The results of this collaboration are covered in Chapters 4 and 5, and have been published separately in the proceedings of FSTTCS'13 [19] (with Sławomir Lasota) and LICS'13 [20]. The author played a leading role in the development of the key ideas and in the writing of those papers.

(*Patrick Totzke*)

Table of Contents

Chapter 1

Introduction

Verification is the branch of theoretical computer science that studies mathematical formalisms and methods for arguing about the correctness of processes. The fundamental problem is to show that a given system satisfies the requirements imposed by a given specification.

Several formalisms for the description of processes/algorithms have been proposed, including Turing Machines, the λ-calculus, random access machines or counter automata [22, 41] . A major result in the theory of computation is that these models are equivalent in the sense that they are all capable of expressing the same input-output functions, and therefore of solving the same problems. This yields a meaningful definition of what it means for a function to be computable and for a problem to be decidable. Moreover, we can derive solid measures to compare the computational complexity of decidable problems in terms of the time and space requirements on these abstract machines [42, 4]. The downside is that all non-trivial properties of algorithms represented in any of these *universal* models are themselves undecidable. In particular, a fully automatic verification of arbitrary computer programs is theoretically impossible.

However, one can regain decidability of the verification problem by sufficiently restricting the formalisms used to model processes and specifications. This leads to a trade-off between the expressiveness of the model of computation and the specification on the one hand, and the decidability/complexity of the resulting verification problem on the other. A lot of research has been carried out towards charting the border of decidability and the complexity of different verification problems. This dissertation continues this line of research.

Traditionally, there are two approaches to automatic verification: model checking and equivalence checking. In model checking, the specification is given as a temporal logic formula, and verifying a process amounts to checking if it satisfies the formula. Logics of various expressibility have been proposed that highlight different aspects, i.e. are more or less suitable to express certain properties. Popular examples include linear-temporal logic (LTL) [52], that can express safety and liveness properties, computation-tree logic (CTL) [10], that allows

to make assertions about the branching structure of processes and the μ-calculus [32, 8], that subsumes the previous two and introduces fixpoint operators to argue about recursion.

In the equivalence checking approach, both implementation and specification are given as processes. The idea is that the specification is a simplified or schematic description of the intended behaviour. Verification amounts to checking if these two processes are semantically equivalent. This can be interpreted as asking if two programs are the realization of the same algorithm and equal up to irrelevant implementation details. One can consider different notions of equivalence, corresponding to different degrees of abstraction [14]. Among the semantic equivalences, bisimulation [5, 43, 40] stands out as the most well understood, mathematically elegant and in many cases, practically tractable notion.

A natural variation of the equivalence checking approach is to consider partial orders instead of equivalences: one asks if one process subsumes another process, i.e., can to some extend reproduce its behaviour. Naturally, for any partial order on processes there is a corresponding equivalence defined by mutual inclusion. From the perspective of program verification, this allows to specify a range of tolerable behaviour by independently checking if a process subsumes some under-specification, that captures the necessary "good" behaviour, and moreover is subsumed by an over-specification that limits the acceptable "bad" behaviour. Another motivation for partial order checking is that one can, instead of reasoning about a program that involves a mathematically "unnatural" description or data-structure, replace it with one that is easier to deal with and then show that the replacement subsumes the original. In order to exclude certain bad behaviour in the original process it then suffices to demonstrate the correctness of the replacement.

Most verification problems can be phrased in terms of logic games played between two players. The idea was pioneered by Andrzej Ehrenfeucht in the 1960s, who used games to formalize "elementary equivalence" on logical structures, as introduced earlier by Roland Fraïssé. In model theory, these back-and-forth games are used as method to determine the expressive power of logical theories. Similar games now enjoy widespread popularity as an intuitive characterization of various model checking and equivalence checking problems [49]. In this setting, one player tries to defend some assertion and the goal of the opponent is to convince him otherwise. Different verification problems can be considered by varying the rules of the game, i.e., the winning condition and the way the players can move. In order to find out if a property holds, one can equivalently look for a solution of the corresponding logic game. The question reduces to effectively constructing a winning strategy for one of the players, that prescribes a way of playing that ensures a win, regardless of the opponent's choices. A finite representation of such a winning strategy then serves as a certificate for the truth or falsity of the property in question.

For systems with only finitely many distinguishable configurations, most interesting verification problems are decidable by exhaustive search. One still has to deal with the so called "state-space explosion", the fact that generally, the number of different configurations is exponential in the description of a process. But although arguing about large finite systems might be difficult in practice, it is at least theoretically possible to check all possible configurations individually.

On the other hand, many interesting computational models are capable of describing infinite-state systems: processes with infinitely many possible configurations. This is desirable if one wants to model unbounded recursion or data structures, or use real numbers to model continuous time or probabilities. Some formalisms are even capable of defining infinitely-branching processes, that can jump from one configuration to one of infinitely many different successor configurations. This can be useful for modelling unbounded resource reloading or uncertain environments of a reactive process. Unlike for finite systems, verification by exhaustive search is not an option for these models and one has to study their structural properties in order to find necessary and sufficient criteria for correctness.

In this thesis we study inclusion problems for one-counter automata (OCAs) and their subclass one-counter nets (OCNs). These are very limited computational models for infinite-state systems. As the name suggests, one-counter automata are a restriction of (Minski) counter machines [41]. They consist of a finite-state control and a single counter, that can store an unbounded non-negative integer value. Transitions can only increment or decrement the counter by one and the counter can be tested for zero, which means that a particular step can depend on the current counter value being exactly zero. OCAs correspond to the classical model of pushdown automata from the theory of formal languages, where the stack alphabet is unary. One-counter *nets* are OCAs that cannot fully test their counter for zero in the sense that an empty counter can only restrict possible moves. This model is subsumed not only by OCAs and hence pushdown automata, but also by vector addition systems with states (VASS) and Petri nets [12], which are widely used models of concurrent computation. One-counter nets correspond to 1-dimensional VASS or Petri nets with at most one unbounded place.

We study the computational complexity of semantic preorders between OCAs, OCNs, and finite systems. Specifically, we are interested in the standard notions of strong and weak simulation and trace inclusion. The main technical contributions in this work are the following.

1. Checking strong simulation for OCNs is PSPACE-complete. Moreover, we show that a semilinear representation of the largest simulation is effectively computable for a given pair of OCNs and that the size of this representation is exponential in the number of control states of the input-nets.

2. Weak simulation is PSPACE-complete for OCNs. In particular, one can represent the

largest weak simulation relation for a given pair of OCNs in terms of strong simulation for a pair of only polynomially bigger OCNs. An exponential-sized semilinear representation of weak simulation is therefore also computable.

3. Strong and weak simulation between OCNs and finite systems are P-complete in both directions.

4. Trace inclusion between OCNs is undecidable.

5. Trace inclusion between a OCNs and a deterministic OCNs is NL-complete. This contrasts our undecidability result for trace inclusion between two (nondeterministic) OCNs, as well as the known undecidability of trace inclusion for deterministic OCAs [50].

6. Trace inclusion between finite systems and OCNs, as well as trace universality of OCNs are decidable and Ackermann-complete. This contrasts the known undecidability of trace universality for OCAs and the NL-upper bound for the deterministic case. In particular, we show that these problems are not primitive recursive, but lie exactly at level ω of the so called fast-growing hierarchy. We also show Ackermannian lower bounds on the length of shortest witnesses for non-universality.

The remainder of this thesis is structured as follows. In **Chapter 2**, we introduce the necessary background on semantic preorder checking. We define counter machines and recall labelled transition systems (LTS), that serve as operational semantics for various computational models and thus formally capture the behaviour of processes. Relative to a given LTS, we define trace inclusion and simulation, as well as their weak variants and discuss their characterizations in terms of approximants and games.

In **Chapter 3**, we discuss one-counter automata and nets in more detail. We provide a complete characterization of shortest paths in OCAs and derive a NL-complexity upper bound for checking reachability. We then introduce some common notation and properties of semantic preorders between processes of two OCAs. In particular, we show that one can always assume w.l.o.g., that the given OCAs are in a certain normal form when checking any of the partial orders considered in this thesis. We discuss the consequences of the monotonicity of steps in OCNs and finally, recall the idea of presenting binary relations on OCA-processes as finitely many black/white colourings of the grid $\mathbb{N}^2$. This allows to study geometric properties of such relations.

Chapter 4 is devoted to strong simulation preorder. It was known before that checking simulation is undecidable for OCAs [30] and decidable for OCNs [2, 29]. We focus here on the complexity of the problem and show a PSPACE upper bound that matches the known lower bound. For this, we provide a new and constructive proof of the so called *Belt Theorem*, a

geometric property of the maximal simulation between two OCNs, that underpins a previous argument for decidability. In our proof, we analyse a finitary abstraction of the usual simulation games, and show how to transfer winning strategies for both players between these games. This yields polynomial bounds for important parameters in the description of the maximal simulation according to the Belt Theorem. Based on this, and the locality of strong simulation, we derive a nondeterministic procedure for checking simulation that requires only polynomial space and an exponential-sized semilinear description of the whole relation.

Chapter 5 is about weak simulation. We show that for OCNs, one can effectively reduce weak to strong simulation. The main difficulty one has to overcome is that the players in the corresponding games now move on infinitely branching systems, so one cannot finitely approximate weak simulation from above using standard weak simulation approximants. This means that there can be pairs of processes that are not in weak simulation but for which no finite winning strategies in the game exist.

To solve this problem, we first reduce weak simulation for OCNs to strong simulation for a generalized model that symbolically captures the infinite branching capability. Then, we show that the resulting strong simulation relation can be finitely approximated using customized approximants that essentially count the number of infinite-branching steps in a play of the game. We show that 1) these approximants always converge to simulation at a polynomially bounded level, and 2) that the individual approximant relations are themselves expressible in terms of strong simulation over suitably modified OCNs. As a byproduct, we get that the standard weak simulation approximants converge at level ω^2, but not earlier in general.

A closer analysis of the reduction reveals that the constructed OCNs are only polynomially larger than the given input nets, which implies that the PSPACE upper bound for checking inclusion as well as the EXPSPACE-bound on the size of a representation of the whole relation can be transferred from strong to weak simulation.

At the end of this chapter we consider strong and weak simulation between OCA/OCN and finite systems.

In **Chapter 6** we study trace inclusion. First, we compare it to the classical notion of language inclusion from formal language theory. The respective problems are inter-reducible for OCAs, regardless of the acceptence criteria. For OCNs, this is only true for languages defined by acceptance with final state, due to the lack of zero-tests. We recall the negative results of Valiant [50] regarding the language inclusion problem of OCAs. In particular, we get that trace inclusion is undecidable between *deterministic* OCAs, and trace universality is undecidable for OCAs. We then continue to show that even for (nondeterministic) OCNs, trace inclusion is undecidable. Next, we turn to the trace universality problem for OCNs, which was already known to be decidable [25] and show that this problem is in fact Ackermannian, i.e., not primitive recursive. We conclude by showing how to construct Ackermannian lower

bounds for the length of shortest witnesses for non-universality.

Finally, in **Chapter 7**, we look at deterministic systems, for which trace inclusion and simulation coincide. Inclusion between DOCA is undecidable in general [50], but it turns out that for deterministic one-counter *nets* (DOCNs), trace inclusion is not only decidable but has surprisingly low complexity. We show that this problem is NL-complete, even with binary encoded initial counter values as part of the input. We then apply our technique to the trace inclusion problems between DOCAs and DOCNs (and vice versa), where one of the given processes is allowed to have zero-testing transitions.

Chapter 2

Preliminaries

When comparing the behaviour of processes we are mainly interested if two given processes are semantically equivalent or if one subsumes the other. These equivalence and inclusion problems allow two degrees of freedom. Firstly, different formalisms can be used to define processes and secondly, different notions of equivalence or inclusion can be used to compare them. In this section we will clarify what exactly is meant by "a process" and introduce the notions of process equivalence and inclusion considered in further sections.

Following Milner [40], we interpret processes as nodes of transition graphs generated by some suitable rewriting system. Semantic notions of equivalence or inclusion are then just preorders (reflexive and transitive binary relations) on the set of nodes, i.e. processes. We refer to [14] for a classification of semantic preorders and to [9, 48, 34] for surveys providing a recent and high level view of the status quo of infinite-state verification.

We write $\mathbb{Z}$, $\mathbb{N}$ and $\mathbb{Q}$ for the sets of integers, non-negative integers and rational numbers respectively. For a set A let A^* denote the set of finite sequences (or *words*) over A. We write $\varepsilon \in A^* \setminus A$ for the empty word, $|w| \in \mathbb{N}$ for the length of word $w \in A^*$ and $w^i = ww\ldots w$ for its i-fold concatenation.

2.1 Models of Computation

The unifying perspective on computational models is that they all have operational semantics defined as directed, edge-labelled graphs called *transition systems*. A process is a node in such a transition graph, and its behaviour is interpreted as the tree of possible unfoldings.

Definition 2.1. A *labelled transition system* (LTS) is described by a triple $(V, \mathrm{Act}, \longrightarrow)$ where

- V is a (possibly infinite) set of *states* or *processes*,
- Act is a finite set of *action labels*, or *actions* for short,
- $\longrightarrow \subseteq (V \times \mathrm{Act} \times V)$ is the labelled *transition relation*.

We use the infix notation $\alpha \xrightarrow{a} \beta$ instead of writing $(\alpha,a,\beta) \in \longrightarrow$ and say there is an *a-step* from α to β. The transition relation is inductively extended to words over Act in the natural way: We let $\alpha \xrightarrow{\varepsilon} \alpha$ for all $\alpha \in V$ and $\xrightarrow{wa} = \xrightarrow{w} \circ \xrightarrow{a}$ for any word $w \in \mathrm{Act}^*$ and action $a \in \mathrm{Act}$. For $k \in \mathbb{N}$, we write $\alpha \longrightarrow^k \beta$ if $\alpha \xrightarrow{w} \beta$ for some word w of length $|w| = k$. The transitive and reflexive closure of $\longrightarrow$ is $\longrightarrow^* = \bigcup_{k\in\mathbb{N}} \longrightarrow^k$. We say β is *reachable* from α if $\alpha \longrightarrow^* \beta$, that is, if some $w \in \mathrm{Act}^*$ exists with $\alpha \xrightarrow{w} \beta$.

For a given state $\alpha \in V$ and action $a \in \mathrm{Act}$, the sets of *a-predecessors* and *a-successors* of α are defined as $Pre_a(\alpha) = \{\beta \mid \beta \xrightarrow{a} \alpha\}$ and $Post_a(\alpha) = \{\beta \mid \alpha \xrightarrow{a} \beta\}$, respectively. The *branching degree* of a process α is the maximal cardinality of any set $Post_a(\beta)$, where β is reachable from α. A process with branching degree 1 is called *deterministic*. It is *image-finite* or *finitely-branching* if it has a finite branching degree and *infinitely-branching* otherwise. The branching degree of a LTS is the maximal degree over its processes and it is called deterministic, image-finite or infinitely-branching if its degree is 1, finite or infinite, respectively.

we want to reason about, or simply execute, a computer program, it needs to be finitely represented. Well known universal formalisms include Turing machines, (parallel) random-access machines, and the λ-calculus. In this thesis we focus on (restrictions of) *counter machines* [41], that extend a finite-state control by a number of counters that range over the naturals, and which can be incremented, decremented and tested for zero.

Definition 2.2. A *k-counter machine (k-CM)* consists of a finite-state control and k *counters*, each capable of storing an unbounded non-negative integer.

Formally, a k-CM is a tuple $\mathcal{M} = (Q,\mathrm{Act},\delta)$ where Q is a finite set of control states, Act a finite alphabet of *actions* and δ a transition relation of the form $\delta \subseteq Q \times \mathrm{Act} \times OP \times Q$, where $OP = \{\mathrm{inc_i},\mathrm{dec_i},\mathrm{ifz_i} \mid 0 < i \leq k\}$ are the possible operations on the counters. A *configuration* $(q,c_1,c_2,\ldots,c_k) \in Q \times \mathbb{N}^k$ of $\mathcal{M}$ consists of a state and a valuation of the counters. $\mathcal{M}$ induces a transition system with states $V = Q \times \mathbb{N}^k$ and steps

$$(q,c_1,c_2,\ldots,c_k) \xrightarrow{a} (q',c_1',c_2',\ldots,c_k') \tag{2.1}$$

iff there is a transition $(q,a,\mathrm{op},q') \in \delta$ and the following are satisfied.

- If $\mathrm{op} = \mathrm{inc_i}$ then $c_i' = c_i + 1$ and $c_j' = c_j$ for all $j \neq i$.
- If $\mathrm{op} = \mathrm{dec_i}$ then $c_i' = c_i - 1 \geq 0$ and $c_j' = c_j$ for all $j \neq i$.
- If $\mathrm{op} = \mathrm{ifz_i}$ then $c_i' = c_i = 0$ and $c_j' = c_j$ for all $j \neq i$.

A k-CM is *deterministic* if it induces a deterministic LTS. A path in the LTS induced by $\mathcal{M}$ is sometimes referred to as a *run* of $\mathcal{M}$.

Remark 2.3. According to this definition, each k-CM step that is not due to a zero-testing transition (those with operation ifz_i), changes the value of exactly one counter. It is however not difficult to simulate steps of a more general type of machine, that can change multiple counters in a single step or change control state but not the counter values, with a fixed sequence of ordinary CM steps. We choose to consider the two resulting machines as implementing the same algorithm and therefore essentially equivalent[1].

It is a well-known fact that deterministic 2-CMs are already Turing-complete and have an undecidable halting problem. Most of the undecidability results we later obtain are shown by reduction from this problem.

Theorem 2.4 ([41])**.** *It is undecidable if* $(q_{init},0,0) \longrightarrow^* (q_{halt},0,0)$ *holds for a given deterministic* 2*-CM with two designated initial and halting states* q_{init} *and* q_{halt}*.*

Since counter machines allow explicit zero-testing transitions that make steps dependent on a particular counter being zero, it is also undecidable if there exist values $a_1,a_2,b_1,b_2 \in \mathbb{N}$ such that $(q_{init},a_1,a_2) \longrightarrow^* (q_{halt},b_1,b_2)$ holds. This problem is sometimes referred to as the *control state reachability* problem.

In light of this undecidability result, it becomes clear that one needs to restrict counter machines in order to get less expressive models that still allows for decidable verification problems. Naturally, different restrictions will highlight different features and are therefore more or less suitable to model particular behaviour of systems.

We will consider two natural restrictions. The first one is to allow at most one counter. The resulting model is called *one-counter automata* (OCAs) and corresponds to the classical notion of pushdown automata with the restriction that the stack alphabet is unary. If we use OCAs to define languages, accepting for instance by final state, then the language of a OCA is therefore context-free. This in particular means that OCAs are not Turing-complete. We will see in Section 3.1 that the reachability problem for OCAs is decidable and even NL-complete.

The second restriction we focus on is to disallow explicit zero-testing transitions The resulting model is called *vector addition systems with states*, and is equivalent to vector addition systems (without finite-state control) and Petri nets, which are widely used to model concurrent processes.

Definition 2.5. An *n-dimensional vector addition system with states (VASS)* is represented by a triple (Q,Act,δ). Here, Q is a finite set of control states, Act is a finite set of action labels and $\delta \subseteq Q \times \text{Act} \times Q \times \{-1,0,1\}^n$ is a finite transition relation. It induces an infinite-state labelled transition system over the stateset $Q \times \mathbb{N}^n$ where $(p,C) \xrightarrow{a} (p',C+D)$ iff there is some $(p,a,p',D) \in \delta$ and $C+D \in \mathbb{N}^n$.

[1]We can be more precise and say that for any process of the original machine there is a process in the constructed CM that is *weakly simulation equivalent*, i.e., the pair is in mutural weak simulation (Definition 2.20).

An important property of vector addition systems with states is that the induced steps are monotone with respect to the counter values: If $(p,C) \xrightarrow{a} (p',C')$ then also $(p,C+E) \xrightarrow{a} (p',C'+E)$ for any vector $E \in \mathbb{N}^n$.

We will take special notice of a model called *one-counter nets* (OCNs), which are subsumed both by vector addition systems and one-counter automata. One-counter nets are OCAs without the ability to test the counter for zero or equivalently, 1-dimensional VASSs. One-counter nets are arguably one of the simplest models of discrete infinite-state systems.

2.2 Behavioural Preorders

Various notions of process equivalence and inclusion have been used for verification. The most common and well-understood equivalences are bisimulation, simulation equivalence and trace equality. A popular "catalogue" of semantic equivalences on processes is [14], where the relations mentioned above and variations are compared.

Formally, a semantic equivalence or inclusion relation is a preorder on the states of a given labelled transition system. This thesis focusses on inclusions, specifically on strong and weak simulation preorder and trace inclusion. We will now introduce these relations relative to some global LTS $(V,\mathrm{Act},\longrightarrow)$.

2.2.1 Trace Inclusion

Trace inclusion asks if the set of *traces*, all emittable sequences of actions of one process is included in that of another.

Definition 2.6. The word $w \in \mathrm{Act}^*$ is a *trace* of process $\alpha \in V$ if $\alpha \xrightarrow{w} \beta$ for some $\beta \in V$. We write $\mathcal{T}(\alpha) \subseteq \mathrm{Act}^*$ for the set of traces of α and $\alpha \subseteq \beta$ iff $\mathcal{T}(\alpha) \subseteq \mathcal{T}(\beta)$ for two processes α and β.

Whether or not trace inclusion holds between two processes is a global condition in the sense that it does not depend at all on their branching structure.

For the purpose of verification, where one wants to check the consistency of a concrete implementation with respect to some abstract specification, this is a desirable property. After all, in this scenario we are only interested if the implementation is correct and not in the actual implementation details.

However, it turns out that for many computational models, checking trace inclusion is expensive if at all decidable. For instance, classic results from the theory of formal languages are that language/trace inclusion are PSPACE-complete for nondeterministic finite automata (NFA) and undecidable for pushdown automata. We will later show (in Chapter 6), that trace inclusion is already undecidable for one-counter nets.

2.2.2 Simulation Preorder

A more fine-grained notion of inclusion is simulation preorder [39, 43], which also takes the branching structure of the two given processes into account. For a process to simulate another, it is not sufficient to have at least the same set of traces. Rather, it must be able to stepwise preserve being simulation larger. The standard co-inductive definition of simulation is the following.

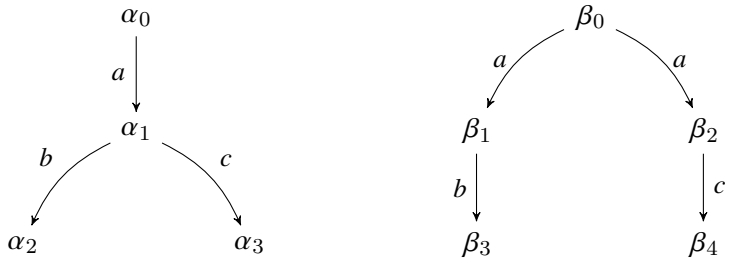
Definition 2.7. A *simulation* is a relation $R \subseteq V \times V$ over the states of a LTS $(V, \mathrm{Act}, \longrightarrow)$, such that for every pair $(\alpha, \alpha') \in R$,

$$\forall(\alpha \xrightarrow{a} \beta) \quad \exists(\alpha' \xrightarrow{a} \beta') \quad (\beta, \beta') \in R. \tag{2.2}$$

Simulations are closed under union, so there exists a unique maximal relation that satisfies the *simulation condition* (2.2). This maximal simulation is called *simulation preorder* or simply *the* simulation and written as $\preceq$.

By induction on the length of possible traces we see that simulation implies trace inclusion. The following classical example shows that the converse is not always true.

Example 2.8. The processes α_0 and β_0 depicted below satisfy $\mathcal{T}(\alpha_0) = \mathcal{T}(\beta_0) = \{\varepsilon, a, ab, ac\}$ and therefore $\alpha_0 \subseteq \beta_0 \subseteq \alpha_0$.



Further, $\beta_0 \preceq \alpha_0$ can be witnessed by the relation $\{(\beta_0, \alpha_0), (\beta_1, \alpha_1), (\beta_2, \alpha_1), (\beta_3, \alpha_2), (\beta_4, \alpha_3)\}$, which is a simulation (but not the maximal one). In the other direction we see that neither β_1 nor β_2 simulate α_1 and therefore also $\alpha_0 \not\preceq \beta_0$.

In the absence of nondeterministic choice one can easily verify that trace inclusion $\subseteq$ satisfies the simulation condition, so simulation preorder and trace inclusion coincide. This is already the case if only the supposedly larger process on the right is deterministic. Moreover, checking non-inclusion $\alpha \not\subseteq \beta$ for an arbitrary process α and a *deterministic* process β can be seen as a reachability problem in the synchronous product of α and β.

Approximants

There is an alternative inductive characterization of simulation by refinement, as the (transfinite) limit of a strictly decreasing sequence of *approximant relations*.

We write *Ord* for the set of ordinal numbers, which is well-ordered by $\leq$ and write ω for the first limit ordinal. In particular it holds that $n < \omega < \omega + 1$ for all $n \in \mathbb{N}$. We refer to [36] for more background on ordinal arithmetic.

Definition 2.9. Consider the function $\mathcal{F} : 2^{V\times V} \to 2^{V\times V}$ on relations of processes where

$$(\alpha,\alpha') \in \mathcal{F}(R) \iff \forall(\alpha \xrightarrow{a} \beta) \quad \exists(\alpha' \xrightarrow{a} \beta') \quad (\beta,\beta') \in R. \tag{2.3}$$

Based on this refinement function $\mathcal{F}$, we inductively define a sequence of ordinal-indexed *simulation approximants:* The approximant at level 0 is $\preceq_0 = V \times V$, the full relation. For successors, let $\preceq_{\iota+1} = \mathcal{F}(\preceq_\iota)$ and for limit ordinals λ, define $\preceq_\lambda = \bigcap_{\iota<\lambda} \preceq_\iota$.

Note that every simulation $R \in V^2$ is a post-fixed point of $\mathcal{F}$: It holds that $\mathcal{F}(R) \supseteq R$. Moreover, $\mathcal{F}$ satisfies $R \subseteq R' \implies \mathcal{F}(R) \subseteq \mathcal{F}(R')$ for any two relations $R,R' \subseteq V \times V$, i.e., $\mathcal{F}$ is an order-preserving function on the powerset lattice $(2^{V\times V},\subseteq)$. A straightforward application of a fixed point theorem due to Knaster and Tarski yields that the maximal simulation is exactly the limit of the sequence of simulation approximants.

Theorem 2.10. $\bigcap_{\iota\in Ord} \preceq_\iota = \preceq$.

This means there is some smallest ordinal $\kappa \in Ord$ called *convergence index* that satisfies $\preceq_\kappa = \preceq$ and therefore, $\preceq_\iota \supsetneq \preceq_{\iota+1}$ for all $\iota < \kappa$ and $\preceq_\kappa = \preceq_\iota$ for all $\kappa \leq \iota$. Further, for any two processes $\alpha \not\preceq \alpha'$, there is some index $\iota < \kappa$ with $\alpha \not\preceq_\iota \alpha'$. We recall a standard result that for image-finite systems, the sequence of approximants converges at level ω, the first limit ordinal.

Theorem 2.11. *For image-finite LTS it holds that* $\preceq_\omega = \preceq$.

Proof. It suffices to show that $\preceq_\omega \subseteq \preceq$ because $\preceq \subseteq \preceq_\alpha$ holds for every $\alpha \in Ord$ by Theorem 2.10. Assume the converse. Then there are processes $\alpha,\alpha' \in V$ such that $\alpha \preceq_\omega \alpha'$ but $\alpha \not\preceq_{\omega+1} \alpha'$. This means in particular that $\alpha \preceq_{n+1} \alpha'$ holds for every $n \in \mathbb{N}$ and so, for every $\alpha \xrightarrow{a} \beta$ and $n \in \mathbb{N}$, some step $\alpha' \xrightarrow{a} \beta'_n$ exists with $\alpha' \preceq_n \beta'_n$. Since we assumed an image-finite LTS, there are only finitely many different a-successors β'_n of α' and therefore some β' exists with $\beta' = \beta_n$ for infinitely many indices n. Therefore, $\alpha' \preceq_n \beta'$ holds for every $n \in \mathbb{N}$ which means that $\alpha' \preceq_\omega \beta'$. Since the initial step $\alpha \xrightarrow{a} \beta$ was chosen arbitrarily we conclude $\alpha \preceq_{\omega+1} \alpha'$, which contradicts our initial assumption. □

Remark 2.12. The argument above only uses that the supposedly bigger process α' on the right has finite branching degree. Therefore, $\alpha \not\preceq \alpha'$ immediately implies $\alpha \not\preceq_n \alpha'$ for some $n \in \mathbb{N}$ if α' is finitely branching. Also, in this case non-simulation at any level $i \in \mathbb{N}$, i.e., the relations $\not\preceq_i$ are at least semi-decidable because $\alpha \not\preceq_i \alpha'$ can be witnessed by a finite tree of height i. Consequently, non-simulation is semi-decidable if the supposedly bigger process is finitely branching.

Remark 2.13. Refinement functions other than $\mathcal{F}$ may be used to approximate simulation preorder. Similar to Milner's original definition of bisimulation approximants, one can for instance define the refinement function $\mathcal{F}'$ such that $(\alpha,\alpha') \in \mathcal{F}'$ iff Equation (2.5) holds for all action *sequences* $a \in \mathrm{Act}^*$. The resulting approximants $\preceq'_\iota$ then satisfy $\preceq \subseteq \preceq'_\iota \subseteq \preceq_\iota$ for every ordinal ι, and thus still converge to $\preceq$, but faster, i.e. at some lower level. In this case, the first approximant $\preceq'_1$ coincides with trace inclusion $\subseteq$.

2.2.2.1 Simulation Games

Simulation can be interpreted as an interactive, two-player game played between *Spoiler*, who wants to establish non-simulation and *Duplicator*, who wants to frustrate this.

Definition 2.14. A *simulation game* is played in rounds between the two players Spoiler and Duplicator, where the latter tries to stepwise match the moves of the former.

A *play* is a finite or infinite sequence of game *positions*, which are pairs of processes. If a finite play $(\alpha_0,\alpha'_0),(\alpha_1,\alpha'_1),\ldots,(\alpha_i,\alpha'_i)$ is not already winning for one of the players, the next pair $(\alpha_{i+1},\alpha'_{i+1})$ is determined by a round of choices:

1. Spoiler chooses a step $\alpha_i \xrightarrow{a} \alpha_{i+1}$.
2. Duplicator responds by picking an equally labelled transition $\alpha'_i \xrightarrow{a} \alpha'_{i+1}$.

If one of the players cannot move then the other wins, and Duplicator wins every infinite play.

A *strategy* is a set of rules that tells a player how to move. A player plays according to a strategy if all his moves obey the rules of the strategy. A strategy is *winning* from (α,α') if every play that starts in (α,α') and which is played according to that strategy is winning. Finally, we say that a player wins the simulation game from (α,α') if there is some winning strategy for this player from position (α,α').

In game-theoretic terms, simulation games are zero-sum games of perfect information. This means that there are no draws and that both players are aware of the global configuration of the game as well as their opponent's choices.

In any game, the set of plays is a prefix-closed set of finite or infinite sequences of game positions and a strategy is a function from this set to the set of possible moves. For certain types of games, including the simulation games considered here, a player only needs to know the last position of a play: If there is any winning strategy at all, then there is also one that depends only on the last position. These strategies are called *history-free* or *memoryless*. A strategy for Spoiler in the simulation game can therefore be seen as a partial function $\sigma : (V \times V) \to (\longrightarrow)$, from pairs of processes to $\longrightarrow$, the steps of the underlying transition system. Correspondingly, a strategy for Duplicator is a partial function $\sigma : (V \times V \times \longrightarrow) \to (\longrightarrow)$, that prescribes a response for the current position and Spoiler's move. Strategies can equivalently be represented

as trees, where each node represents a position of the simulation game and encodes a particular move that is to be used. The descendants of a node then correspond to all possible next moves of the opponent. Notice that each branch in a tree that represents a winning strategy for Duplicator must either be infinite or end in a position where Spoiler is deadlocked. Conversely, every winning strategy for Spoiler corresponds to a tree in which every branch is finite. However, strategy trees for Spoiler may still have infinite degree because some game position could allow infinitely many different moves of Duplicator if the underlying system is infinitely-branching.

We see that one round of the simulation game directly corresponds to the simulation condition of Equation (2.2). Spoiler (∀) can stepwise demonstrate that the condition is not an invariant if the initial pair of processes is indeed not in simulation. Conversely, any simulation that contains the initial pair of processes prescribes a winning strategy for Duplicator in the simulation game.

Proposition 2.15. *For any two processes $\alpha,\alpha' \in V$, Duplicator has a winning strategy in the simulation game from position (α,α') if and only if $\alpha \preceq \alpha'$.*

Since each round of a simulation game corresponds to one application of the refinement function $\mathcal{F}$ in the definition of simulation approximants, Spoiler can win the simulation game in at most i rounds iff the initial pair (α,α') fails at approximant level i, that is, if $\alpha \not\preceq_i \alpha'$. An alternative formulation of this is provided by the *approximant games* below, in which ordinals are built explicitly into positions. Such games will prove useful in Chapter 5.

Recall that if Spoiler wins, then she can enforce that every round brings her strictly closer to some directly winning position. If the current pair fails to be in simulation for some limit ordinal, then Spoiler can enforce that her opponent picks some arbitrary high, but still strictly smaller level.

Definition 2.16. A *simulation approximant game* is played in rounds between Spoiler and Duplicator. Game positions are triples $(\alpha,\alpha',\iota) \in (V \times V \times Ord)$, where α and α' are configurations of Spoiler and Duplicator, respectively, and ι is some ordinal.

A *play* is a finite sequence $(\alpha_0,\alpha'_0,\iota_0),(\alpha_1,\alpha'_1,\iota_1),\ldots,(\alpha_k,\alpha'_k,\iota_k)$ of positions with $\iota_{i+1} < \iota_i$ for all $0 \leq i < k$. The next position is determined by a round of choices: First Spoiler chooses a strictly smaller next ordinal $\iota_{i+1} < \iota_i$ and a step $\alpha_i \xrightarrow{a} \alpha_{i+1}$. Then, Duplicator responds by picking a step $\alpha'_i \xrightarrow{a} \alpha'_{i+1}$ with the same action label a. If one of the players cannot move, the other wins.

As in simulation games, we say a player wins the approximant game from (α,α',ι) if he has a winning strategy from this position, i.e., can enforce a win.

The intuition is that whenever Spoiler chooses a next ordinal ι and moves to some configuration α, she claims that she can win the remaining game from position (α,α',ι) regardless of

her opponents response α'. Observe that Spoiler loses every approximant game from a position with ordinal counter equal to 0 because she cannot pick a strictly smaller ordinal.

Proposition 2.17. *For any two processes $\alpha,\alpha' \in V$ and any $\iota \in Ord$, Duplicator wins the approximant game from (α,α',ι) if and only if $\alpha \preceq_\iota \alpha'$.*

2.2.3 Weak Variants

The idea of a weak semantics is to distinguish *observable* from *internal*, non-observable behaviour of processes. In weak trace inclusion or simulation, one process subsumes another already if it can mimic its observable behaviour. Because of this relaxed condition, the respective weak preorders are less discriminative and thus larger relations than their ordinary, *strong* counterparts.

Formally, we assume a dedicated action label $\tau \in \mathrm{Act}$ that is used to model internal steps. Based on this, we define a *weak step* relation $\Longrightarrow$ that abstracts from τ-labelled actions.

Definition 2.18. Let $L = (V,\mathrm{Act},\longrightarrow)$ be a LTS and $\tau \in \mathrm{Act}$ a special "silent" action. The *weak step* relation $\Longrightarrow \subseteq (V \times \mathrm{Act} \times V)$ is defined by $\overset{\tau}{\Longrightarrow} = \overset{\tau}{\longrightarrow}^*$ and for $a \neq \tau$,

$$\overset{a}{\Longrightarrow} = \overset{\tau}{\longrightarrow}^* \circ \overset{a}{\longrightarrow} \circ \overset{\tau}{\longrightarrow}^*$$

We extend $\Longrightarrow$ to sequences of actions inductively: for the empty word $\varepsilon \notin \mathrm{Act}$, let $\overset{\varepsilon}{\Longrightarrow} = Id_V$, that is, $\alpha \overset{\varepsilon}{\Longrightarrow} \alpha$ for any $\alpha \in V$. For non-empty sequences define $\overset{aw}{\Longrightarrow} = \overset{a}{\Longrightarrow} \circ \overset{w}{\Longrightarrow}$ for $a \in \mathrm{Act}$ and $w \in \mathrm{Act}^*$.

We will call the relations $\overset{a}{\longrightarrow}$ *strong* steps if we explicitly want to distinguish them from the weak steps defined above. Weak steps are more general than strong steps: For every $w \in \mathrm{Act}^*$ it holds that $\overset{w}{\longrightarrow} \subseteq \overset{w}{\Longrightarrow}$. By the *weak closure* of a system $(V,\mathrm{Act},\longrightarrow)$ we mean the system $(V,\mathrm{Act},\Longrightarrow)$ over the same states and actions but where strong steps are replaced by weak ones. We now define weak trace inclusion and simulation. Intuitively, these notions are defined just as their strong counterparts, but the supposedly bigger process on the right is replaced by its weak closure.

Definition 2.19. A word $w \in \mathrm{Act}^*$ is a *weak trace* of process $\alpha \in V$ iff $\alpha \overset{w}{\Longrightarrow} \beta$ for some $\beta \in V$. Weak trace inclusion ($\subseteq\!\!\!\subseteq$) is the relation that holds for processes $\alpha,\beta \in V$ iff every strong trace of α is a weak trace of β.

In the same spirit, we define weak simulation preorder, weak simulation approximants and the corresponding weak simulation (approximant) games, in which Duplicator moves along weak steps.

Definition 2.20. A binary relation $R \subseteq (V \times V)$ is a *weak simulation* if for every $(\alpha,\alpha') \in R$, and $a \in \mathrm{Act}$ holds that

$$\forall(\alpha \xrightarrow{a} \beta) \quad \exists(\alpha' \xRightarrow{a} \beta') \quad (\beta,\beta') \in R. \tag{2.4}$$

Weak simulations are closed under union, so there exists a unique maximal relation $\lesssim$, called *weak simulation preorder*, that satisfies the weak simulation condition (2.4).

Weak simulation can be characterized in terms of *weak simulation games*, between Spoiler and Duplicator. In a round from position $(\alpha,\alpha') \in V^2$, Spoiler chooses a next step $\alpha \xrightarrow{a} \beta$, and then Duplicator chooses an equally labelled *weak* step $\alpha' \xRightarrow{a} \beta'$ in response. If one of the players cannot move then the other wins and Duplicator wins every infinite play. We have $\alpha \lesssim \alpha'$ iff Duplicator has a winning strategy in the weak simulation game that starts from (α,α').

Lastly, *weak simulation approximants* are ordinal indexed relations $\lesssim_\iota$ that are the result of refining $V \times V$ (transfinitely often) using the refinement function $\mathcal{W}$, where for all $R \subseteq V \times V$,

$$(\alpha,\alpha') \in \mathcal{W}(R) \iff \forall(\alpha \xrightarrow{a} \beta) \quad \exists(\alpha' \xRightarrow{a} \beta') \quad (\beta,\beta') \in R. \tag{2.5}$$

The sequence of weak simulation approximants decreases with increasing index and converges to weak simulation. Analogously to Theorem 2.10, it holds that $\bigcap_{\iota \in Ord} \lesssim_\iota = \lesssim$ and thus $\lesssim_\kappa = \lesssim$ for some *convergence ordinal* $\kappa \in Ord$.

The fact that we use an "asymmetric" definition for weak simulation, in which every strong (possibly τ-labelled) step needs to be matched by a corresponding weak step, is pure convenience. One can alternatively define a "symmetric" notion of weak simulation, where the condition of Equation (2.4) is replaced by

$$\forall(\alpha \xRightarrow{a} \beta) \quad \exists(\alpha' \xRightarrow{a} \beta') \quad (\beta,\beta') \in R. \tag{2.6}$$

The resulting maximal weak simulation $\lesssim'$ then directly corresponds to strong simulation in the weak closure of the LTS. It is however not hard to show that this notion coincides with the original, "asymmetric" notion of Definition 2.20.

Indeed, since Equation (2.6) is more restrictive than the original weak simulation condition in Equation (2.4), $\lesssim' \subseteq \lesssim$ holds trivially. For the converse, notice that $\lesssim$ satisfies the new condition Equation (2.6): assume that $\alpha \lesssim \alpha'$ and $\alpha \xRightarrow{a} \alpha'$. By definition of weak steps, this is due to a finite sequence $\alpha = \alpha_0 \xrightarrow{\tau} \alpha_1 \xrightarrow{\tau} \alpha_2 \xrightarrow{\tau} \cdots \xrightarrow{\tau} \alpha_l = \alpha'$ if $a = \tau$, or

$$\alpha = \alpha_0 \xrightarrow{\tau} \alpha_1 \xrightarrow{\tau} \alpha_2 \xrightarrow{\tau} \cdots \xrightarrow{\tau} \alpha_k \xrightarrow{a} \alpha_{k+1} \xrightarrow{\tau} \alpha_{k+2} \xrightarrow{\tau} \cdots \xrightarrow{\tau} \alpha_{l-1} \xrightarrow{\tau} \alpha_l = \alpha'$$

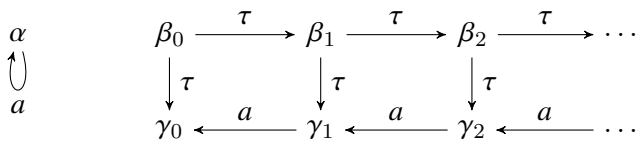
otherwise. In both cases, an induction on l, using the weak simulation condition Equation (2.4), shows the existence of a corresponding sequence $\beta_{i-1} \xRightarrow{} \beta_i$, where $\alpha_i \lesssim \beta_i$ holds for every

$1 \leq i \leq l$. In particular, since the original sequence contained at most one symbol $a \neq \tau$, it holds that $\beta \overset{a}{\Longrightarrow} \beta_l$ and $\alpha' \leqq \beta_l$, which proves the claim.

A similar argument can be used to show that weak trace inclusion coincides with its "symmetric" variant that demands that every *weak* trace of the one process is a weak trace of the other. In the rest of this thesis we use only the "asymmetric" variants of weak trace inclusion and simulation, as defined in Definitions 2.6 and 2.20.

The additional difficulty with checking weak instead of strong simulation is that abstracting from internal actions potentially introduces infinite branching. The weak closure of a system may not be finitely-branching even if the original system is. In such a case, approximants are not guaranteed to converge at level ω and a negative semi-decision procedure as mentioned in Remark 2.12 can therefore not be taken for granted.

Example 2.21. The infinite LTS depicted below has a finite branching degree. So, by Theorem 2.11, strong simulation approximants converge at level ω. In particular, we have $\alpha \not\preceq_1 \beta_0$ because β_0 lacks direct a-successors. Moreover, for every $i \in \mathbb{N}$ it holds that $\alpha \preceq_i \gamma_i$ but $\alpha \not\preceq_{i+1} \gamma_i$, because γ_i can do exactly i many a-steps.



The weak closure of this system is infinitely-branching. For every index $i \in \mathbb{N}$ there is a weak step $\beta_0 \overset{a}{\Longrightarrow} \gamma_i$ induced by $\beta_0 \overset{\tau}{\longrightarrow}^{i+1} \beta_{i+1} \overset{\tau}{\longrightarrow} \gamma_{i+1} \overset{a}{\longrightarrow} \gamma_i$. Similar to the strong case, $\alpha \leqq_i \gamma_i$ and $\alpha \not\leqq_{i+1} \gamma_i$ for every $i \in \mathbb{N}$ because no process γ_i has access to (can reach a configuration that enables) τ-labelled steps. However, since there is a weak step $\beta_0 \overset{a}{\Longrightarrow} \gamma_i$ for every $i \in \mathbb{N}$, we have $\alpha \leqq_i \beta_0$ and therefore $\alpha \leqq_\omega \beta_0$ by definition of weak simulation approximants at the limit ω.

In a weak simulation game from position (α, β_0), Spoiler plays the only available strategy and simply loops on label a. In the first round, this forces Duplicator to move to some process γ_i, which will deadlock after another i rounds. Therefore, Duplicator's first response determines how many further rounds he survives. This means that $\alpha \not\leqq_{\omega+1} \beta_0$ because Spoiler wins the explicit weak simulation approximantion game from position $(\alpha, \beta_0, \omega+1)$. For this particular system, weak simulation approximants indeed converge at level $\omega+1$.

2.3 Decision Problems

We defined strong and weak trace inclusion and simulation relative to a single global LTS, representing the operational semantics of some computational model. If we want to compare

processes of different systems, we can consider the respective preorder defined for their disjoint union.

Definition 2.22. Let $\sqsubseteq$ be some semantic preorder and $\mathcal{M}$ and $\mathcal{M}'$ be computational models inducing LTS over state-sets V and V', respectively. We write $\sqsubseteq_{\mathcal{M},\mathcal{M}'}$ for the relation $\sqsubseteq$ *with respect to* $\mathcal{M},\mathcal{M}'$, which is $\sqsubseteq$ as defined for the disjoint union of their induced LTS and projected into $V \times V'$.

If X and Y abbreviate classes of computational models, we write $X \sqsubseteq Y$ for the decision problem that asks if $x \sqsubseteq y$ holds for given processes x and y definable in X and Y respectively.

For instance, OCA $\preceq$ VASS denotes the problem of checking strong simulation between processes of a one-counter automaton (a 1-CM) and a vector addition system with states:

INPUT: 1) A OCA $\mathcal{M} = (Q,\mathrm{Act},\delta)$ together with a process $(q,c) \in Q \times \mathbb{N}$,
2) A n-dimensional VASS $\mathcal{M}' = (Q',\mathrm{Act},\delta')$ and a process $(q',c') \in Q' \times \mathbb{N}^n$

QUESTION: $(q,n) \preceq_{\mathcal{M},\mathcal{M}'} (q',c')$?

This particular problem turns out to be undecidable [1]. For decidable inclusion problems we can ask what the exact complexity is, i.e., how much time and space relative to the size of the given input one needs to solve it.

We assume familiarity with standard notions of computational complexity. We will write NL, P, NP and PSPACE, for the classes of problems decidable in nondeterministic logarithmic space, deterministic and nondeterministic polynomial time and polynomial space respectively.

Chapter 3

One-Counter Systems: Basic Results and Notation

One-counter automata (OCAs) are counter machines with only one counter. They correspond to the subclass of pushdown automata with a unary stack alphabet. One-counter nets (OCNs) are 1-dimensional VASSs. They form a subclass of OCAs where the counter cannot be fully tested for zero, because transitions enabled at counter value zero are also enabled at nonzero values.

We now introduce a more convenient notation for one-counter systems in order to keep the presentation clear. In Section 3.1 we characterize shortest paths in one-counter systems and derive and NL upper complexity bound for checking reachability. In Section 3.2 we consider the particularities of checking preorders between one-counter processes.

Definition 3.1. A *one-counter automaton* $\mathcal{A} = (Q, \mathrm{Act}, \delta, \delta_0)$ is given by finite sets of control states Q, action labels Act, transitions $\delta \subseteq Q \times \mathrm{Act} \times \{-1,0,1\} \times Q$ and zero-test transitions $\delta_0 \subseteq Q \times \mathrm{Act} \times \{0,1\} \times Q$. It induces an infinite-state labelled transition system over the state set $Q \times \mathbb{N}$, whose elements will be written as pm where $p \in Q$ and $m \in \mathbb{N}$. The transition relation $\longrightarrow = \longrightarrow_+ \cup \longrightarrow_0$ is partitioned into *positive* and *zero-testing* steps. For all states $p,p' \in Q$ and $m,m' \in \mathbb{N}$ these are defined by

1. $pm \xrightarrow{a}_+ p'm' \iff (p,a,(m'-m),p') \in \delta$ and
2. $pm \xrightarrow{a}_0 p'm' \iff (p,a,m',p') \in \delta_0$ and $m = 0$.

Such an automaton is called a *one-counter net* if $\delta_0 = \emptyset$, i.e., if the automaton cannot test if the counter is equal to 0. It is *deterministic* if for every $p \in Q$ and $a \in \mathrm{Act}$, there is at most one $(p,a,d,q) \in \delta$, at most one $(p,a,d,q) \in \delta_0$ and moreover, if $(p,a,d,p') \in \delta_0$ and $(p,a,d',p'') \in \delta$, then $d' = -1$.

When defining a OCN we will omit δ_0 and simply define the net as triple $\mathcal{N} = (Q, \mathrm{Act}, \delta)$. Abusing notation, we moreover write $pm \xrightarrow{t} qn$ if a transition $t = (p,a,d,q) \in \delta \cup \delta_0$ justifies a step $pm \xrightarrow{a} qn$.

Remark 3.2. Unlike the 1-CM from Definition 2.2 (Page 8), the notion of OCA defined here allows zero-testing steps that set the counter to 1. This is purely for convenience and does not increase the expressivity of the model (due to the presence of a finite control) and has no impact on the decidability/complexity of the problems considered here.

Example 3.3. We make use of the usual graphical notation for automata to draw OCA. We draw an edge $q \xrightarrow{a,d} r$ for normal a-labelled transitions with effect d and use double arrow tips for zero-testing transitions. If not explicitly stated, the effect of a transition is $d = 0$. The picture below shows the automata $\mathcal{A} = (\{q,r\}, \{a\}, \{(q,a,0,r),(r,a,-1,q)\}, \emptyset)$, which is a OCN, on the left and $\mathcal{A}' = (\{s\}, \{a,b\}, \{(s,a,-1,s),(s,b,-1,s)\}, \{(s,b,0,s)\})$ on the right. Both systems are deterministic.

3.1 Paths in One-Counter Automata

Definition 3.4. Let $\mathcal{A} = (Q, \mathrm{Act}, \delta, \delta_0)$ be a OCA. For a transition $t = (p,a,d,p') \in \delta \cup \delta_0$ we write $source(t) = p$ and $target(t) = p'$ for the source and target states, $\lambda(t) = a$ for its label and $\Delta(t) = d$ for its effect on the counter.

A *path* (of length k) in $\mathcal{A}$ is a sequence $\pi = p_0 t_1 p_1 t_2 p_2 \dots p_{k-1} t_k p_k$ where all $p_i \in Q$ and $t_i \in \delta \cup \delta_0$ and for every $1 \le i \le k$, $p_{i-1} = source(t_i)$ and $target(t_i) = p_i$. The source and target of π are p_0 and p_k, respectively. Its label is $\lambda(\pi) = \lambda(t_1)\lambda(t_2)\dots\lambda(t_k) \in \mathrm{Act}^*$ and its *effect* is the cumulative effect of its transitions:

$$\Delta(\pi) = \sum_{i=1}^{k} \Delta(t_i) \tag{3.1}$$

A path π as above is *positive* if it makes only positive steps, i.e., $t_i \in \delta \setminus \delta_0$ for all $1 \le i \le k$. It is a *cycle* if $p_0 = p_k$ and a *simple cycle* if it is a cycle and moreover, no proper subpath is itself a cycle.

A path in $\mathcal{A}$ defines a (possibly empty) set of paths in the LTS induced by $\mathcal{A}$. We say π is *enabled* in configuration pm if it prescribes a valid path from state pm in the transition system of $\mathcal{A}$, i.e., if there are natual numbers $m_0, m_1, \dots, m_k$ such that $p_0 m_0 = pm$ and $p_{i-1} m_{i-1} \xrightarrow{t_i} p_i m_i$ for all $1 \le i \le k$. In this case we write $p_0 m_0 \xrightarrow{\pi} p_k m_k$ and say π is a *run* or path *of* $\mathcal{A}$ from $p_0 m_0$ to $p_k m_k$. Note that $m_k = m_0 + \Delta(\pi)$.

If a path in $\mathcal{A}$ is not positive (contains at least one zero-testing transition), then it defines at most one run of $\mathcal{A}$. On the other hand, a positive path π in $\mathcal{A}$ defines infinitely many positive runs of $\mathcal{A}$ and there is a minimal sufficient counter value $\Gamma(\pi)$ that enables it. This *guard* of π can be defined as the minimal $m \in \mathbb{N}$ such that no prefix of π has an effect less than $-m$. Writing ${}^{i}\pi$ for the prefix of path π of length i, the guard of π is given as

$$\Gamma(\pi) = -\min\{\Delta({}^{i}\pi) \mid 0 \leq i \leq k\}. \tag{3.2}$$

Note that there are different paths of length 0 because the initial state forms part of a path. Any zero-length path π has effect and guard $\Delta(\pi) = \Gamma(\pi) = 0$. Surely, both the effect and the guard of any path are bounded by its length. The next lemmas characterize shortest runs of one-counter automata. We fix a OCA $\mathcal{A} = (Q, \mathrm{Act}, \delta, \delta_0)$ with $K = |Q| \in \mathbb{N}$ states.

Lemma 3.5. *Let π be a shortest run from $p_0 m_0$ to $p_k m_k$. Then,*

1. *All configurations $p_i m_i$ along π have counter values $m_i \leq \max\{m_0, m_k\} + (K-1)K$.*
2. *$|\pi| < (\max\{m_0, m_k\} + (K-1)K) \cdot K$.*

Proof. Let $\pi = p_0 m_0 \longrightarrow p_1 m_1 \longrightarrow \ldots \longrightarrow p_k m_k$ be a shortest run from $p_0 m_0$ to $p_k m_k$ and consider the set $U = \{m_i \mid 0 \leq i \leq k,\ m_i \geq \max\{m_0, m_k\}\}$ of counter values visited by π which are greater or equal to $\max\{m_0, m_k\}$. Pick an index x such that $m_x \in U$ is maximal. For each value $m \in U$ we can mark the largest index $0 \leq l(m) \leq x$ and the smallest index $x \geq f(m) \geq k$ such that $m_{l(m)} = m_{f(m)} = m$. These indices guarantee that along the subpath from index $l(m)$ to $f(m)$, the counter value does not drop below m.

Due to the minimality assumption, π cannot repeat any configurations, as otherwise the intermediate path could be removed. This means in particular that $p_{l(m)} \neq p_{f(m)}$ for every $m \in U \setminus \{m_x\}$. Furthermore, no two values $z, y \in U$ with $z < y \leq m_x$ can satisfy $p_{l(z)} = p_{l(y)}$ and $p_{f(y)} = p_{f(z)}$ as otherwise it would be possible to remove the nonempty paths from $p_{l(z)}$ to $p_{l(y)}$ and from $p_{f(y)}$ to $p_{f(z)}$ and obtain a valid path from $p_0 m_0$ to $p_k m_k$ that is shorter than π (see Figure 3.1). This leaves exactly $(K-1)K$ assignments of pairs $(p_{f(m)}, p_{l(m)}) \in Q^2$ to values $m \in U \setminus \{m_x\}$, that are consistent with the above constraints. We conclude that $|U| \leq (K-1)K + 1$ and since $\max\{m_0, m_k\} \in U$, this completes the proof of the first claim.

For the second claim observe that by point 1, all counter values visited along π must be smaller than or equal to $\max\{m_0, m_k\} + (K-1)K$. Since π does not repeat any configurations, it can visit at most $(\max\{m_0, m_k\} + (K-1)K) \cdot K$ distinct configurations. □

Lemma 3.6. *Let $\pi = p_0 m_0 \longrightarrow p_1 m_1 \longrightarrow \ldots \longrightarrow p_k m_k$ be a shortest positive run. Then,*

1. *$\max\{m_0, m_k\} + (K-1)K \geq m_i \geq \min\{m_0, m_k, (K-1)K\}$ for all $0 \leq i \leq k$,*
2. *$|\pi| < ((K-1)K + |m_0 - m_k| + \min\{m_0, m_k, (K-1)K\} + 1) \cdot K$.*

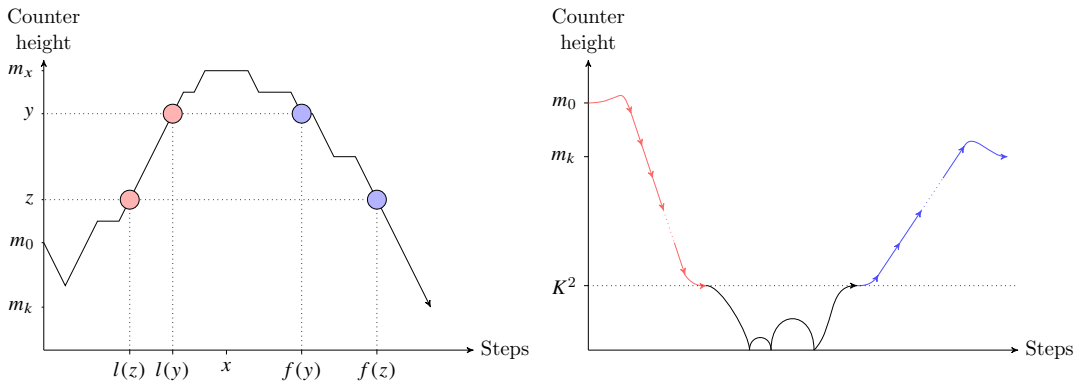


Figure 3.1: The change in counter value along a path of a OCA, starting in m_0 and ending in m_k. Left: If the maximal visited value m_x is above $\max\{m_0,m_k\}+(K-1)K$, one can safely remove subpaths from $f(y)$ to $f(z)$ and from $l(z)$ to $l(y)$ for some values $z < y < m_x$. Right: the shape of a shortest non-positive path. If $m_0,m_k > K^2$, the positive prefix and suffix visiting only values $> K^2$ mostly repeat one most effective cycle.

Proof. The first inequality we simply repeat the argument from Lemma 3.5, point 1, that bounds the set $U = \{m_i \mid 0 \le i \le k,\ m_i \ge \max\{m_0,m_k\}\}$. For the second inequality observe that since π is positive, none of its steps depend on the counter being exactly 0. This means that one can once again repeat the previous argument to show that the set D of counter values $\le \min\{m_0,m_k\}$ visited along π has no more than $(K-1)K+1$ elements. However, since the counter cannot drop below 0, we have $|D| \le \min\{m_0+1, m_k+1, (K-1)K+1\})$ and as $\min\{m_0,m_k\} \in D$, the minimal counter value on the whole path is thus greater or equal to $\min\{m_0,m_k,(K-1)K\}$. This completes the proof of the first claim.

For the second claim notice that all counter values visited along the path are in $U \cup \{m_i \mid m_0 < m_i < m_k\} \cup D$, which has no more than

$$n = ((K-1)K+1) + (|m_0 - m_k| - 1) + (\min\{m_0,m_k,(K-1)K\}+1)$$

many elements. It follows that shortest positive paths visit no more than $n \cdot K$ many distinct configurations, which also bounds their length and thus completes the proof. □

Lemma 3.5 in particular implies that shortest paths connecting two configurations with counter values $m_0,m_k < K^2$ are shorter than $2K^3 - K^2$. Using Lemma 3.6 we can bound a shortest *positive* path from p_0m_0 to p_km_k by $3K^3 - 2K^2 + K$, if the difference $|m_0 - m_k|$ between initial and final counter value does not exceed K^2. The next lemma characterizes the possibly long positive minimal paths from p_0m_0 to p_km_k for the remaining case where the difference $|m_0 - m_k| > K^2$ and moreover, $\min\{m_0,m_k\} \ge K^2$. Roughly, it states that in this case, a minimal path mainly iterates one most effective simple cycle. It is a slight generalization of Lemma 2 in [51].

Lemma 3.7. *Assume there is a positive path $pm \xrightarrow{\pi}_+ qn$ where $|m-n| \geq K^2$ and $\min\{m,n\} \geq K^2$. Then there is a positive path $pm \xrightarrow{\pi'}_+ qn$ of minimal length which is of the form $\pi' = \pi_0 C^l \pi_1$ where C is a simple cycle, $\Delta(C) \cdot (n-m) > 0$ and $|\pi_0\pi_1| \leq K^2$.*

Proof. Note that the condition $\Delta(C) \cdot (n-m) > 0$ only states that the effect of cycle C cannot be 0 and iterating C moves the counter in the "right direction": C decreases the counter if $m > n$ and increases if $m < n$. We prove the claim for the case $m \geq n + K^2$, the other case ($m \leq n - K^2$) is analogous.

Assume a path $pm \xrightarrow{\pi}_+ qn$ of minimal length. Since $m - n \geq K^2$, π must contain a simple cycle C with $\Delta(C) < 0$. Define the *efficiency* of a cycle C to be $(-\Delta(C)/|C|)$. We decompose π into $\pi = \pi_0 C \pi_1$ where C is a simple cycle with maximal efficiency and let $d = -\Delta(C)$. Surely, $0 < d < K$ because C is a simple cycle. At least d steps along C have effect -1 and the total effect of the remaining steps is 0. Thus, no prefix of C has effect $< \frac{1}{2}(|C| - d)$.

Consider a path $\pi'_0 C \pi'_1$ that is the result of removing, from π, a set $S = \{C_1, C_2, \ldots, C_k\}$ of disjoint cycles (not necessarily simple ones), with maximal total length such that d evenly devides its overall effect $\sum_{1 \leq i \leq k} \Delta(C_i)$. We claim that $|\pi'_0\pi'_1| \leq K^2$.

First, observe that the path $\pi'_0\pi'_1$ contains at least $l = |\pi'_0\pi'_1|/K$ simple cycles $\{C'_1, C'_2, \ldots, C'_l\}$. If $l \geq d$, then for some non-empty subset $S' \subseteq \{C'_1, C'_2, \ldots, C'_l\}$, we have that $\sum_{c \in S'} \Delta(c) = 0 \mod d$. But this contradicts the maximality of the initial set S of cycles removed from π. We derive that $|\pi'_0\pi'_1|/K < d$ and thus $|\pi'_0\pi'_1| < d \cdot K \leq K^2$.

Since the set S satisfies $\sum_{C \in S} \Delta(C) = 0 \mod d$, there exists some value $r \in \mathbb{N}$ such that $\Delta(\pi'_0\pi'_1) = \Delta(\pi_0\pi_1) - rd = \Delta(\pi_0\pi_1) + r\Delta(C)$. This means that the effect of path $\pi' = \pi'_0 C^{r+1} \pi'_1$ equals that of the original path π. To see why π' is enabled in pm, observe that the counter cannot drop below

$$\min\{n,m\} - |\Delta(\pi'_0\pi'_1)| - (|C| - d)/2 \quad \geq \quad \min\{n,m\} - d \cdot K - (|K| - d)/2 \tag{3.3}$$

which is non-negative because $d \leq K$ and we assumed that $\min\{m,n\} \geq K^2$. Since π' was obtained from the minimal path π only by replacing cycles by others with greater or equal efficiency, π' must also be a shortest path. □

Theorem 3.8. *The reachability problem for OCA is NL-complete. More specifically, there is a nondeterministic algorithm that, given a OCA $\mathcal{A} = (Q, Act, \longrightarrow)$ and configurations $pm, qm \in Q \times \mathbb{N}$ decides if $pm \longrightarrow^* qn$ in $O(\log(|Q|) + \log^2(m+n))$ space.*

Proof. NL hardness already holds for reachability in finite graphs. For the upper bound, we solve three subproblems and show that their solutions can be combined to solve the overall problem. For each subproblem we show that if a witness exists, then there is a small certificate that can be guessed and verified in logarithmic space.

Subproblem 1. Fix some bound $c \in \mathbb{N}$ and consider the subproblem of checking $pm \longrightarrow^* qn$ for instances with counter values $m,n < c$. If there is a path $pm \xrightarrow{\rho} qn$, then we can assume by point 2 of Lemma 3.5 (page 21), ρ is no longer than $(c-1)\cdot K + K^3 - K^2$. Recall that the length of a path bounds the effect it has on the counter. The binary representation of $\Delta(\rho)$ therefore requires only $\log(|\rho|) < \log((c-1)\cdot K + K^3 - K^2)$ space. In this case, one can simply guess the witness stepwise, explicitly memorize its (binary encoded) effect and in the end verify that indeed $n = m + \Delta(\rho)$ holds.

Subproblem 2. Consider instances where the difference between initial and terminal counter values is $|m-n| < K^2$. Is there a *positive* path $pm \xrightarrow{\rho}_+ qn$? If such a path exists, then by Lemma 3.6 (page 21), there is also one which is no longer than $3K^3 - 2K^2 + K$. As in the first case, one can guess and verify a short witness.

Subproblem 3. Consider instances where $|m-n| \geq K^2$ and moreover, $\min\{m,n\} \geq K^2$. Is there a *positive* path $pm \xrightarrow{\rho}_+ qn$? If the answer is yes, then we know by Lemma 3.7 that a witness exists that can be decomposed into $\rho = \pi_0 C^l \pi_1$, where C is a simple cycle (and thus has effect $-K < \Delta(C) < K$), and the total effect of the prefix and suffix is $\Delta(\pi_0\pi_1) < K^2$. It suffices to stepwise guess π_0, C and π_1, memorize their effects and verify that there exists $l \in \mathbb{N}$ such that $m + \Delta(\pi_0\pi_1) + \Delta(C)\cdot l = n$. This check can be implemented deterministically in $log(m + \Delta(\pi_0\pi_1) + \Delta(C))$ space (see also Lemma 7.13 on page 106 for details).

Finally, let us consider the unrestricted problem. The idea is that if a witness exists, then there is one that can be decomposed (see Figure 3.1 on page 22) into smaller paths that can be individually verified. We can first check if the initial and terminating counter values satisfy the preconditions for subproblem 2 or 3, and conclude "yes" if one of these subroutines finds a (positive) witness. Otherwise, if $pm \xrightarrow{\rho} qn$ holds but so far, no witness is detected, then there must be one witnessing path along which the counter drops below K^2. We can guess a first position rl with counter value $l = K^2$ on this path and verify (using subroutine 3) that $pm \longrightarrow^*_+ rl$. Further, if the terminal counter value n is below K^4, we can use subroutine 1 to directly check if $rl \longrightarrow^* qn$. Otherwise, if $n \geq K^4$, we can again decompose the path $rl \xrightarrow{\pi} qn$ into $rl \longrightarrow^* sk \longrightarrow^*_+ qn$, where sk is the last position with counter value $k = K^2$. We can then individually verify, using subroutines 1 and 3 respectively, that $rl \longrightarrow^* sk$ and $sk \longrightarrow^*_+ qn$. □

3.2 Two-Player Games on One-Counter Systems

We saw that strong and weak simulation can be interpreted as two player, turn-based (weak) simulation games. Other relations can be characterized by similar games with slightly modified rules. Trace inclusion for instance can be seen as a simulation game in which Spoiler must first announce a finite sequence of actions (the witness), and then the game is played such that Spoiler moves as prescribed by this sequence, i.e., if the chosen witness is $a_1 a_2 \ldots a_k$ then she

must make an a_i labelled step in the ith round (and automatically loses in round $k+1$).

All these inclusion games are "single-sided" in the sense that each player is assigned one process (side) and during any unfolding of the game can only move from their respective side. This means that a player cannot gain from making a move that restricts the further behaviour of the process, for instance by moving to a deadlock[1]. Conversely, Duplicator surely wins any sensibly defined inclusion game from two identical processes or if his process is universal, i.e., has self-loops for every possible action. This is a useful observation if one wants to show lower complexity bounds using the *defenders forcing* technique (cf. [34]): Duplicator might "threaten" to move to a universal (or syntactically equal) process in the next round unless Spoiler moves in some specific way. This technique has been used for example in [47] to show a PSPACE lower bound for simulation on OCN.

So far, the above is not at all specific to games played on one-counter systems. If we only consider inclusion relations and their corresponding games on OCA, we can make further assumptions. We now introduce some common notions and notation for inclusion games played on one-counter systems.

3.2.1 Normal Form Assumption

We prove a simple normal-form theorem (Lemma 3.10) for inclusion games on OCAs, that essentially states that Spoiler can only win if she forces Duplicator to empty his counter.

Definition 3.9. A OCA $\mathcal{A} = (Q,\mathrm{Act},\delta,\delta_0)$ is *complete* if for every state $p \in Q$ and every action $a \in \mathrm{Act}$, there exists at least one transition $(p,a,d,p') \in \delta$. $\mathcal{A}$ is *non-blocking* if none of its processes is a deadlock, i.e., if for every state $p \in Q$ either there is some transition $(p,a,d,p') \in \delta$ with $d \in \{0,1\}$ or there are transitions $(p,a,-1,q) \in \delta$ and $(p,b,d,r) \in \delta_0$.

A pair $\mathcal{A},\mathcal{A}'$ of OCAs is defined to be in *normal form* if $\mathcal{A}$ is deterministic and non-blocking, and $\mathcal{A}'$ is complete.

Lemma 3.10. *For any two OCAs $\mathcal{A} = (Q,Act,\delta,\delta_0)$ and $\mathcal{A}' = (Q',Act',\delta',\delta'_0)$, one can construct a pair $\mathcal{B},\mathcal{B}'$ of OCAs in normal form with state sets Q and $S \supseteq Q'$, respectively, such that for all $(q,n,q',n') \in Q\times\mathbb{N}\times Q'\times\mathbb{N}$ and for every $\sqsubseteq \in \{\preceq,\leqq,\subseteq\}$ it holds that*

$$qn \sqsubseteq q'n' \text{ w.r.t. } \mathcal{A},\mathcal{A}' \iff qn \sqsubseteq q'n' \text{ w.r.t. } \mathcal{B},\mathcal{B}'. \tag{3.4}$$

Moreover, the reduction uses logarithmic space and $\mathcal{B}'$ is deterministic if the original automaton $\mathcal{A}'$ is deterministic.

Proof. If $\mathcal{A}$ is not already deterministic, we can make it so by uniquely re-labeling all its transitions t by actions a_t and adding corresponding transitions (p',a_t,d',q') to the other system

[1] This is not the case in games corresponding to equivalences like bisimulation [40, 49], in which Spoiler is allowed to start a round from a side of her choice. There, Spoiler could actually win by moving one process to a deadlock.

$\mathcal{A}'$ for any existing $(p',\lambda(t),d',q')$. So assume $\mathcal{A}$ is already deterministic and pick a new action label $\$ \notin Act$. We turn $\mathcal{A}$ into a non-blocking net $\mathcal{B}$ by adding \$-labelled cycles with effect 0 to all states: $\mathcal{B} = (Q,\mathrm{Act}\cup\{\$\},\overline{\delta},\delta_0)$ with $\overline{\delta} = \delta \cup \{(s,\$,0,s) \mid s \in Q\}$. To compensate for this, we add \$-cycles to all states of $\mathcal{A}'$ in the same way. To complete the second net, add a sink state L (for "losing"), which has counter-decreasing cycles for all actions, and connect all states without outgoing a-transitions to L by a-labelled transitions. $\mathcal{B}' = (Q'\cup\{L\},\mathrm{Act}\cup\{\$\},\overline{\delta}',\delta_0')$ where

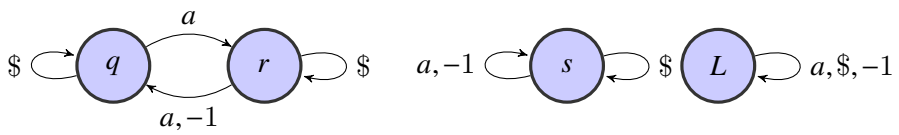
$$\begin{aligned}\overline{\delta}' = {} & \{(p,a,d,q) \in \delta' \mid a \in \mathrm{Act}\}\\ & \cup \{(s,\$,0,s) \mid s \in Q'\}\\ & \cup \{(s,a,0,L) \mid s \in Q',\ a \in \mathrm{Act},\ \neg\exists (s,a,d,s') \in \delta'\}\\ & \cup \{(L,a,-1,L \mid a \in \mathrm{Act}\cup\{\$\}\}.\end{aligned}$$

Assume Spoiler, playing on $\mathcal{A}$, wins the (weak) simulation game against Duplicator playing on $\mathcal{A}'$. In the game on $\mathcal{B}$ and $\mathcal{B}'$, Spoiler can move according to a winning strategy in the original game and thus force a play ending in a position $(pm,p'm')$ that is immediately winning in the game on $\mathcal{A}$ and $\mathcal{A}'$, i.e., $pm \xrightarrow{a} rl$ but $p'm' \not\xrightarrow{a}$ for some action a. In case $p'm' \not\xrightarrow{a}$ because $\delta' \cup \delta_0'$ does not contain a-labelled transitions that originate in p', the game on $\mathcal{B},\mathcal{B}'$ continues in the position rl,Lm', which is clearly winning for Spoiler because she can exhaust her opponent's counter and win using finitely many \$-moves.

Conversely, if Duplicator wins the (weak) simulation game on $\mathcal{A}$ and $\mathcal{A}'$ this means that each play is either infinite or ends in a position $pm,p'm'$ where $pm \not\xrightarrow{a}$ for all actions a. In the game on $\mathcal{B}$ and $\mathcal{B}'$, the latter case means that Spoiler has no choice but to make \$-moves indefinitely, which is losing for her.

The above argument justifies the correctness of the construction for strong and weak simulation. Considering trace inclusion, we see that if a word w of length k witnesses non-inclusion $qn \not\subseteq q'n'$ w.r.t. $\mathcal{A},\mathcal{A}'$ then surely $w\$^{k+n'+1}$ witnesses non-inclusion $qn \not\subseteq q'n'$ w.r.t. $\mathcal{B},\mathcal{B}'$. To see this, observe that in this case any path π in $\mathcal{B}'$ that starts in state q' and with $\lambda(\pi) = w$ must end in state L. This means any such path takes the initial configuration $q'n'$ to some configuration Ln'' where $n'' \leq n' + k$. Conversely, if $qn \not\subseteq q'n'$ w.r.t. $\mathcal{B},\mathcal{B}'$ then, as \$ labelled steps only affect processes with control state L, a shortest witness must be of the form $w = w'\$^k$ where w' does not contain actions \$. This means that w' witnesses $qn \not\subseteq q'n'$ w.r.t. $\mathcal{A},\mathcal{A}'$. □

Example 3.11. If we normalize the systems $\mathcal{A},\mathcal{A}'$ from Example 3.3 on page 20 according to the construction in the proof of Lemma 3.10, we derive the one-counter nets $\mathcal{B}$ and $\mathcal{B}'$ with state sets $\{q,r\}$ and $\{s,L\}$, respectively, as depicted below. Notice that the b-labelled transitions of the original automaton $\mathcal{A}'$ are not caried over, since the action $b \in \mathrm{Act}' \setminus \mathrm{Act}$ does not occur in $\mathcal{A}$.



Remark 3.12. Note that the construction above does *not* work for weak trace inclusion $\subseteq$. This can be seen for the systems $\mathcal{A} = q \xrightarrow{a} q$ vs. $\mathcal{A}' = q' \xrightarrow{\tau,+1} q'$. It holds that $q0 \not\subseteq q'0$ because $q'0 \stackrel{a}{\not\Longrightarrow}$. However, in any *complete* OCA $\mathcal{B}' \supseteq \mathcal{A}'$ over alphabet Act $\supseteq \{a\}$, the process $q'0$ is weakly trace universal: Because of the silent, counter increasing self loop from q' and the completeness-assumption, there must be a state L such that $q'0 \stackrel{a}{\Longrightarrow} Ln$ for any action a and value $n \geq 1$. Now any $w \in \text{Act}^*$ shorter than n is a weak trace of Ln because $\mathcal{B}'$ is complete. We will later show (in Chapter 7) that weak trace inclusion between DOCN and weak trace universality of DOCAs are in fact undecidable.

Lemma 3.10 allows us to focus on instances of the (weak) simulation or trace inclusion problems where the given systems are normalized. In particular, we may assume that the systems are such that both players win "purely", if at all: Spoiler cannot get stuck and only loses infinite plays. Moreover, Duplicator can only be stuck (and lose the game) if his counter equals zero. Therefore, every branch in any winning strategy for Spoiler ends in a position where Duplicator has counter value 0.

3.2.2 Monotonicity

We already mentioned the undecidability of simulation for OCA. However, simulation *is* decidable for the subclass of one-counter nets, as we will see in Chapter 4. Ultimately, what makes the problem decidable for OCN is the monotonicity of steps, derived from vector addition systems.

A OCN $\mathcal{N} = (Q, \text{Act}, \delta)$ has no zero-testing transitions by definition. This means a step $pm \xrightarrow{a} qn$ in a OCN must be due to some transition $(p,a,d,q) \in \delta$ with $d = n - m$. The same transition then justifies a step $p(m+l) \xrightarrow{a} q(n+l)$ for any number $l \in \mathbb{N}$. We thus observe that for all OCN processes pm and $l \in \mathbb{N}$,

$$pm \preceq p(m+l) \tag{3.5}$$

because Duplicator can mimic the behaviour of Spoiler's process to win the simulation game. Seen as a function, this "copycat" strategy is simply the identity. Seen as a tree, it has the property that every node is of the form $[qn, q(n+l)]$, where $q \in Q$ and $n \in \mathbb{N}$. Equation (3.5) implies that on OCNs, all preorders that are coarser than $\preceq$, the maximal strong simulation, are monotonic in the following sense.

Lemma 3.13 (Monotonicity)**.** *Let pm be a OCN process, s an arbitrary process and $\sqsubseteq$ be any transitive relation that subsumes strong simulation $\preceq$. Then, for every $n \geq m$,*

1. *$pm \not\sqsubseteq s$ implies $pn \not\sqsubseteq s$, and*

2. *$s \sqsubseteq pm$ implies $s \sqsubseteq pn$.*

Proof. By Equation (3.5) we have $pm \preceq pn$ and thus $pm \sqsubseteq pn$. The claim directly follows from this observation and the transitivity of $\sqsubseteq$. □

Remark 3.14. This holds in particular for $\sqsubseteq$ being strong or weak simulation, trace inclusion or any relation $\sqsubseteq_\alpha$ that approximates one of them from above.

The following is a direct consequence of Lemma 3.13 that we state here only because we are particularly interested in games played between two OCN processes.

Corollary 3.15. *Let pm and $p'm'$ be two OCN processes and $\sqsubseteq$ be any transitive relation that subsumes stong simulation. Then $pm \sqsubseteq p'm'$ implies $pn \sqsubseteq p'n'$ for all $n \leq m$ and $m' \leq n'$.*

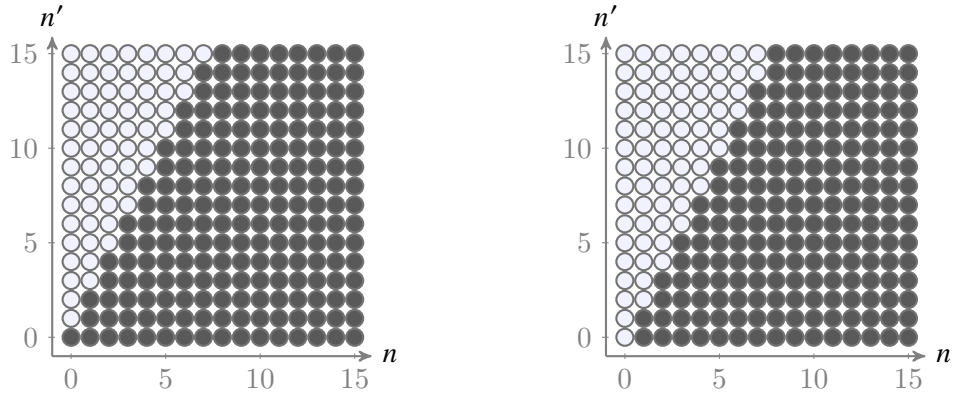
3.2.3 Colourings

We will use a geometric interpretation of relations between processes of one-counter systems in terms of colourings of planes, that is due to Jančar (see for instance [27]).

Any relation R between the processes of OCA $\mathcal{A}$ with states Q and $\mathcal{A}'$ with states Q' is a subset $R \subseteq Q \times \mathbb{N} \times Q' \times \mathbb{N}$. We interpret R as 2-colouring of $K = |Q \times Q'|$ Euclidean planes, one for each pair of control states $(q,q') \in Q \times Q'$. Every point $(n,n') \in \mathbb{N}^2$ in the plane for $(q,q') \in Q \times Q'$ is coloured white if $qn \; R \; q'n'$ and black otherwise.

Example 3.16. Consider trace inclusion $\subseteq$ with respect to systems $\mathcal{A}$ and $\mathcal{A}'$ from Example 3.11 on page 26. Because $\mathcal{A}'$ is deterministic and contains no τ-labelled transitions, trace inclusion $\subseteq$ actually coincides with strong and weak simulation and weak trace inclusion.

The traces of a process qn of $\mathcal{A}$ are all words in $\{a,\$\}^*$ with no more than $2n+1$ many a-letters. The traces of process sn' of $\mathcal{A}'$ are exactly all words in $\{a,\$\}^*$ with at most n' letters a. We thus get $qn \subseteq sn'$ iff $n' \geq 2n+1$. Similarly, we see that $rn \subseteq sn'$ iff $n' \geq 2n$. The colouring of $\subseteq$ relative to $\mathcal{A},\mathcal{A}'$ is shown below for the planes for (q,s) on the left and (r,s) on the right.



The colourings of the planes for (s,L) and (r,L) are completely black, because $qn \not\subseteq Ln'$ and $rn \not\subseteq Ln'$ for all values $n,n' \in \mathbb{N}$ as witnessed by $\$^{n'+1}$.

3.2.4 Product Graphs

When we consider simulation (and other) games played on LTS induced by one-counter systems, it is convenient to identify individual plays with paths in the *synchronous product* of the two given OCA. In later constructions we will in particular be interested in the effects of cyclic paths in this product.

Definition 3.17. The *product graph* of two OCAs $\mathcal{A} = (Q,\mathrm{Act},\delta,\delta_0)$ and $\mathcal{A}' = (Q',\mathrm{Act},\delta',\delta'_0)$ is the finite, edge-labelled graph with nodes $V = Q \times Q'$ and transitions $E \subseteq (\delta \cup \delta_0) \times (\delta' \cup \delta'_0)$ where E contains any pair of transitions of $\mathcal{A}$ and $\mathcal{A}'$ with the same label:

$$(t,t') \in E \iff \lambda(t) = \lambda(t'). \tag{3.6}$$

A *path* in the product is a sequence $\Pi = v_0 T_1 v_1 T_2 v_2 \dots v_{k-1} T_k v_k$ where for all $i \leq k$, $v_i \in V$ and $T_i = (t_i,t'_i) \in E$ such that $\pi = t_1 t_2 \dots t_k$ and $\pi' = t'_1 t'_2 \dots t'_k$ are paths in $\mathcal{A}$ and $\mathcal{A}'$ respectively. A path is *positive* if it does not contain any zero-testing transitions, i.e., $T_i \in (\delta \times \delta')$ for all $1 \leq i \leq k$.

The path Π is *enabled in* $(pm,p'm')$ if both π is enabled in pm and π' is enabled in $p'm'$. In this case we write $(pm,p'm') \xrightarrow{\Pi} (qn,q'n')$ to mean that both $pm \xrightarrow{\pi} qn$ and $p'm' \xrightarrow{\pi'} q'n'$.

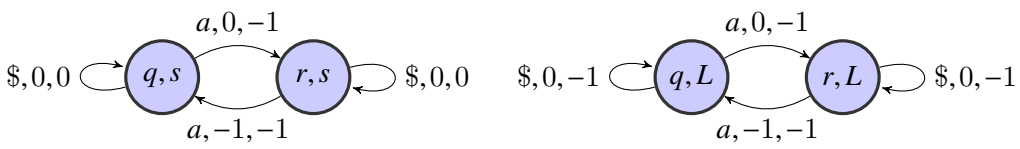
We write $T \in \Pi$ if $T = T_i$ for some index $1 \leq i \leq k$. The $source$, and $target$ of paths in OCA are lifted to paths in products in a natural way: We define $source(\Pi) = (source(\pi),source(\pi'))$, $target(\Pi) = (target(\pi),target(\pi'))$. We write $\Delta(\Pi) = \Delta(\pi)$ and $\Gamma(\Pi) = \Gamma(\pi)$ as well as $\Delta'(\Pi) = \Delta(\pi')$ and $\Gamma'(\Pi) = \Gamma(\pi')$ for the effect and guard of Π on the counter of $\mathcal{A}$ and $\mathcal{A}'$ respectively.

A nonempty path Π is a *cycle* if $source(T_1) = target(T_k)$. It is a *simple cycle* or *loop* if it is a cycle but none of its proper subpaths is a cycle. A *lasso* is a path that contains a cycle while none of its strict prefixes does. That is, a path Π as above is a lasso if there exists $l \leq k$ such that $target(T_k) = source(T_l)$ and for all $1 \leq i \leq j < k$, $target(T_j) \neq source(T_i)$. A lasso Π naturally splits into $\textsc{prefix}(\Pi) = v_0 T_1 v_1 T_2 \dots T_{l-1} v_l$ and $\textsc{cycle}(\Pi) = v_l T_l v_{l+1} T_{l+1} \dots T_k v_k$.

Example 3.18. The product of the systems $\mathcal{B}$ and $\mathcal{B}'$ from Example 3.11 (page 26) has nodes $V = \{(q,s),(r,s),(q,L),(r,L)\}$ and transitions

$$\begin{aligned} E = \{ \quad & ((q,a,0,r),(s,a,-1,s)),\ ((q,a,0,r),(L,a,-1,L)),\ ((r,a,-1,q),(s,a,-1,s)), \\ & ((r,a,-1,q),(L,a,-1,L)),\ ((q,\$,0,q),(s,\$,0,s)),\ ((q,\$,0,q),(L,\$,-1,L)), \\ & ((r,\$,0,r),(s,\$,0,s)),\ ((r,\$,0,r),(L,\$,-1,L)) \quad \} \end{aligned}$$

and is drawn below.



If we let $T_1 = ((q,a,0,r),(s,a,-1,s))$ and $T_2 = ((r,a,-1,q),(s,a,-1,s))$, then the path $\Pi = (q,s)T_1(r,s)T_2(q,s)$ is a lasso where $\textsc{prefix}(\Pi)$ is the empty path from (q,s) and $\textsc{cycle}(\Pi)$ is Π itself. Π defines paths $\pi = q \xrightarrow{a,0} r \xrightarrow{a,-1} q$ and $\pi' = s \xrightarrow{a,-1} s \xrightarrow{a,-1} s$ in $\mathcal{B}$ and $\mathcal{B}'$, respectively. Note that both π and π' are cycles but only π is a simple cycle. The effects of Π are $\Delta(\Pi) = -1$ and $\Delta'(\Pi) = -2$.

Chapter 4

Strong Simulation

Jančar, Moller, and Sawa [30] showed the undecidability of strong simulation between OCA processes by reduction from the halting problem of 2-Counter Minsky Machines. We focus on one-counter nets, for which the decidability of simulation for OCN was established by Abdulla and Čerāns [2]. Jančar, Moller, and Sawa [29, 30] provided a simplified proof of this result using a geometric argument. In [27], they showed that for a fixed pair of nets, the maximal simulation $\preceq$ is in fact semilinear and its description can be effectively computed. Srba [47] provided a PSPACE lower bound for checking simulation between OCN processes as well as for the problems OCA $\preceq$ NFA and NFA $\preceq$ OCA by reduction from the emptiness problem for alternating finite automata over a unary alphabet [21, 31].

Abdulla and Čerāns [2] showed that, above a certain level, the maximal simulation relation between two OCNs has a regular structure. An important parameter for this structure is the *ratio* of the respective counter values of the two given processes. They show the existence of winning strategies for a player from positions where his counter value exceeds some offset plus the counter value of his opponent multiplied by some predetermined factor called the *quality*. These qualities are different for every pair of control states and are derived from the structure of the nets.

In [29], this is made explicit by the *Belt Theorem*, a geometric statement about the colourings of the simulation preorder over a fixed pair of OCN. For a pair of OCN with sets of control states Q and Q', we interpret $\preceq$ as 2-colouring of $|Q \times Q'|$ planes of naturals, one for each pair of control states $(q,q') \in Q \times Q'$. A point $(n,n') \in \mathbb{N}^2$ in the plane for $(q,q') \in Q \times Q'$ is coloured black if $qn \not\preceq q'n'$ and white otherwise. The *Belt Theorem* 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 4.1. Let us make this statement precise.

Definition 4.1. A vector $(\rho,\rho') \in \mathbb{R} \times \mathbb{R}$ is called *positive* if $\rho \geq 0$, $\rho' \geq 0$ and $(\rho,\rho') \neq (0,0)$. Its *direction* is the set $\mathbb{R}^+ \cdot (\rho,\rho') = \{(r \cdot \rho, r \cdot \rho') : r \in \mathbb{R}^+\}$ of points that lie on the half-line defined by (ρ,ρ') from the origin. For a positive vector (ρ,ρ') and a number $c \in \mathbb{N}$ we say that

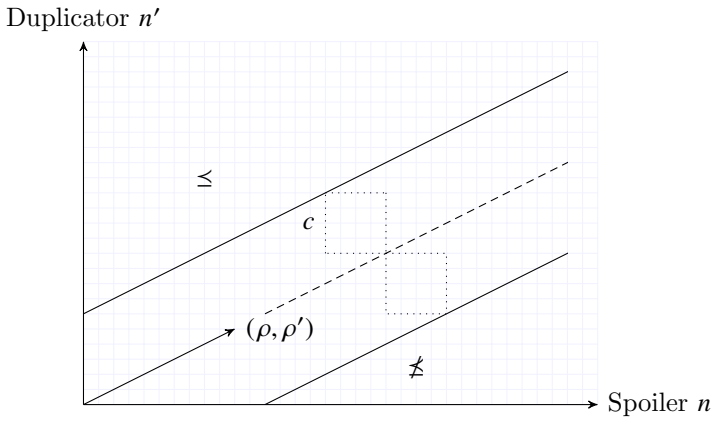


Figure 4.1: A belt with slope ρ/ρ'. The dashed half-line indicates the direction of the vector (ρ,ρ').

the point $(n,n') \in \mathbb{Z}\times\mathbb{Z}$ is *c-above* (ρ,ρ') if there exists some point $(r,r') \in \mathbb{R}^+ \cdot (\rho,\rho')$ in the direction of (ρ,ρ') such that

$$n < r - c \qquad \text{and} \qquad n' > r' + c. \tag{4.1}$$

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

$$n > r + c \qquad \text{and} \qquad n' < r' - c. \tag{4.2}$$

Theorem 4.2 (Belt Theorem)**.** *For every two one-counter nets $\mathcal{N}$ and $\mathcal{N}'$ with sets of 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 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 $qn \not\preceq q'n'$.*

Notice that a point $(n,n') \in \mathbb{N}^2$ is c-below the positive vector $(0,1)$ iff $n > c$ and that no point in $\mathbb{N}^2$ is c-above this vector. In the particular case of a pair of states (p,p') with $pm \not\preceq p'm'$ for all $m,m' \in \mathbb{N}$, the vertical vector $(\rho,\rho') = (0,1)$ satisfies the claim of the Belt Theorem.

Using this theorem one can easily show that strong simulation is semilinear and therefore decidable for OCN processes. The non-trivial part of the colouring of $\preceq$ is contained in a linear belt with fixed width. If one considers sufficiently many cuts (consisting of points on a horizontal line) through a belt then eventually the complete colouring of two such cuts and their one-neighbourhood must be the same. Due to the locality of the simulation condition, we observe that if the colouring of a belt up to the first repetition of a cut does not violate the simulation condition, then the same is true for the derived colouring that replicates the part between the two equal cuts indefinitely. One concludes that whenever $qn \preceq q'n'$ holds,

this is witnessed by a semilinear simulation which can be described by the finite colouring of a sufficiently big initial rectangle and some repetitive colouring within each belt. By enumerating and locally verifying semilinear colourings, one derives a positive semi-decision procedure. Non-simulation is semi-decidable using the standard approximation approach (cf. Remark 2.12 on page 12).

In [27], it is shown how to approximate the exact semilinear colouring of strong simulation for a fixed pair of OCN. The argument is again based on the locality of the simulation condition and the Belt Theorem.

However, the sole existence of certain coefficients c and ratios (ρ,ρ') in the claim of the Belt Theorem is not sufficient to derive an upper complexity bound for the strong simulation problem. Unfortunately, neither the proof of the Belt Theorem in [29] nor the proof of the corresponding lemmas in [2] directly provide bounds for these coefficients.

In the remainder of this section, we provide a new constructive proof of Theorem 4.2 that allows us to derive polynomial bounds on the coefficients of all belts. The proof is based on the analysis of symbolic *slope games* which are finite games played directly on the control graphs of the nets. We show (as Lemmas 4.9 and 4.10) that given a sufficiently high excess of counter values, both players can re-use winning strategies for the slope game also in the simulation game.

In Section 4.1 we discuss slope games and in Section 4.2 prove the strategy transfer lemmas. Section 4.3 contains the proof of the Belt Theorem. In Section 4.4 we formalize the notion of locality of simulation as a geometric property of the colouring and precisely state what we mean by locally verifying a semilinear candidate relation. Finally, in Section 4.5, we show that the largest simulation for two OCN is an effectively constructable semilinear relation that can be explicitly represented in space exponential in the sizes of the input nets. We derive a PSPACE decision procedure for OCN $\preceq$ OCN based on this characterization.

4.1 Slope Games

We fix two one-counter nets $\mathcal{N} = (Q,\mathrm{Act},\delta)$ and $\mathcal{N}' = (Q',\mathrm{Act},\delta')$. We are interested in the strong simulation $\preceq$ relative to these systems. By Lemma 3.10 (page 25), we can assume without loss of generality that $\mathcal{N},\mathcal{N}'$ are in normal form (Definition 3.9) and it is thus Spoiler's objective to exhaust her opponent's counter. Intuitively, her local goal is to maximize the ratio n/n' between the counter values along a play.

Consider the product graph of $\mathcal{N}$ and $\mathcal{N}'$ (Definition 3.17) and let $K = |Q \times Q'|$ be the number of states in this product. If we ignore the actual counter values, any play of the simulation game starting in two processes of $\mathcal{N}$ and $\mathcal{N}'$ respectively, describes a path in this product graph. Moreover, after at most K rounds, a pair of control states is revisited, which means the

corresponding path in the product is a lasso.

The effects of cycles in the product will play a central role in our further construction. The intuition is that if a play of a simulation game describes a lasso then both 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 alternatingly 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 effect of the last cycle was strictly smaller than all previous ones and the next phase starts. The number of different effects of simple cycles thus bounds the maximal length of a game.

Definition 4.3. Let (ρ,ρ') and (α,α') be two vectors in $\mathbb{R}\times\mathbb{R}$ and consider the clockwise oriented angle from (ρ,ρ') to (α,α') with respect to the origin $(0,0)$. We say that (α,α') is *behind* (ρ,ρ') if this oriented angle is strictly between $0°$ and $180°$. See Figure 4.2 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').

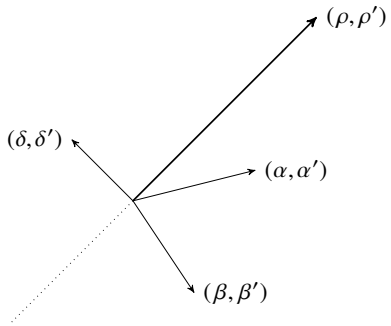


Figure 4.2: Vectors (α,α') and (β,β') are behind (ρ,ρ'), but (δ,δ') is not.

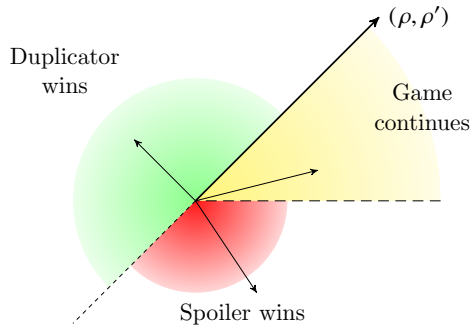


Figure 4.3: Evaluating the winning condition in position $(\pi,(\rho,\rho'))$ after a phase of the slope game.

Notice 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 4.4. *Let (ρ,ρ') be a positive vector and $c,m,n \in \mathbb{N}$.*

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').

Definition 4.5. A *slope game* is a strictly alternating two player game played on a pair $\mathcal{N},\mathcal{N}'$ of one-counter nets in normal form. The game positions are pairs $(\pi,(\rho,\rho'))$, where π is an acyclic path in the product graph of $\mathcal{N}$ and $\mathcal{N}'$, and (ρ,ρ') is a positive vector called *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 $t = (q \xrightarrow{a,d} p)$, then Duplicator responds with a transition rule $t' = (q' \xrightarrow{a,d} p')$. If the extended path $\pi' = \pi(t,t')$ 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 $\Delta(\textsc{cycle}(\pi)) = (\alpha,\alpha')$ 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 $\pi' = \mathit{target}(\pi)$ is the path of length 0 consisting of the pair of ending states of π.

Figure 4.3 on page 34 illustrates the winning condition. Note that if there is no immediate winner it is guaranteed that (α,α') is a positive vector that is behind the slope (ρ,ρ') of the last phase. The number of different positive vectors that derive from the effects of simple cycles thus bounds the maximal length of a game.

The connection between the slope and simulation games is that the outcome of a 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.

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 4.6. *For a fixed pair $\mathcal{N},\mathcal{N}'$ of OCNs in normal form,*

1. *any slope game ends after at most $(\mathrm{K}+1)^2$ 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 $\mathrm{K} = |Q \times Q'|$. It follows that the total number of different possible values for (ρ,ρ'), and therefore the maximal number of phases played, is at most $(\mathrm{K}+1)^2$. Point 2 is a direct consequence as one can find and verify winning strategies by an exhaustive search. □

Definition 4.7. Consider all the non-zero effects (α,α') of all simple 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 $(\alpha,\alpha') \in V$,

$$(\alpha,\alpha') \text{ is behind } (\rho,\rho') \iff (\alpha,\alpha') \text{ is behind } (\sigma,\sigma'). \tag{4.3}$$

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 4.8. *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. □

4.2 Strategy Transfer

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 as there, both Spoiler and Duplicator have a strategy to either win immediately or continue the slope game, i.e., to avoid an immediate loss.

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 a *segment tree*. Regardless of 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 point 1 of Lemma 4.6 (page 36), we know that a slope game ends after at most $d_{\max} = (\mathrm{K}+1)^2$ phases. Consequently, the segment depths of strategies are at most $d_{\max}$ as well.

Let C be the maximal length of a simple cycle in the product graph, i.e., the maximal length of any acyclic path plus 1. We claim that this value is sufficient for the claim of Theorem 4.2. In particular, the following two Lemmas 4.9 and 4.10 state that if a player wins the slope game, an excess of counter value of C is sufficient to be able to safely "replay" a winning strategy in the simulation game.

Lemma 4.9. *If Spoiler has a winning strategy in the slope game from position* $((p,p'),(\rho,\rho'))$ *then Spoiler wins the simulation game from every position* $(pm,p'm')$ *which is* C*-below* (ρ,ρ').

Proof. A position in the slope game contains a positive vector (ρ,ρ'), while a position in the simulation game contains a pair $(m,m') \in \mathbb{N}\times\mathbb{N}$ of counter values, that can also be interpreted as a positive vector. 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 m/m' 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 4.7 on page 36. Then, Spoiler can safely continue to play according to the next lower segment.

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" 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 below the slope of the phase. In particular, this means that Spoiler's counter is always ≥ 0.

Formally, the proof of Lemma 4.9 proceeds by induction on the segment depth d of the assumed winning strategy in the slope game.

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 σ. 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 $(pm,p'm')$ with corresponding position $((p,p'),(\rho,\rho'))$, Spoiler starts playing the simulation game according to σ, until the path in the corresponding position of the slope game say π_1, describes a lasso (this must happen after at most C rounds). Thus π_1 splits into:

$$\pi_1 = \alpha_1\beta_1 \tag{4.4}$$

where the suffix β_1 is a cycle. Let $(a_1,a_1') = (\Delta(\alpha_1),\Delta'(\alpha_1))$ and $(b_1,b_1') = (\Delta(\beta),\Delta'(\beta_1))$ be the effects of α_1 and β_1, respectively. The current values of counters are clearly

$$m + a_1 + b_1 \qquad \text{and} \qquad m' + a_1' + b_1' \tag{4.5}$$

assuming that the play did not end by now with Spoiler's win. As the length of path π_1 is at most C and (m,m') is assumed to be C-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 β_1, namely changes the corresponding position in the slope game from $(\pi_1,(\rho,\rho'))$ to $(\alpha_1,(\rho,\rho'))$ and continues playing according to σ. 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:

$$\pi_2 = \alpha_2\beta_2. \tag{4.6}$$

Denote the respective effects by (a_2,a_2') and (b_2,b_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

$$m + a_2 + b_1 + b_2 \qquad \text{and} \qquad m' + a_2' + b_1' + b_2', \tag{4.7}$$

i.e. the effect (a_1,a_1') of the prefix α_1 of the previous lasso does not contribute any more. As (b_1,b_1') is behind (ρ,ρ') we may apply Lemma 4.4 (page 34) to (b_1,b_1') with $c = 0$ in order to deduce, similarly as before, that all positions by now were below (ρ,ρ'). Now Spoiler rolls back β_2 by establishing $(\alpha_2,(\rho,\rho'))$ as the new corresponding position in the slope game. Continuing in this way, after k rollbacks the counter values are:

$$\begin{aligned} n &= m + a_k + (b_1 + b_2 + \ldots + b_{k-1}) + b_k \qquad \text{and} \\ n' &= m' + a_k' + (b_1' + b_2' + \ldots + b_{k-1}') + b_k', \end{aligned} \tag{4.8}$$

assuming that Spoiler did not win earlier. All the effect-vectors (b_i,b_i') and thus also the sum

$$(b_1 + b_2 + \ldots + b_{k-1}, b_1' + b_2' + \ldots + b_{k-1}') \tag{4.9}$$

are behind (ρ,ρ'), hence similarly as before all positions by now have been below (ρ,ρ'), by Lemma 4.4 applied to the vector (4.9) above. This in particular means that Spoiler's counter remains non-negative. However, as by assumption all observed cycles come from a final segment in her slope game strategy, the vector (4.9) 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 drop below 0 and hence Spoiler eventually wins.

Case $d > 1$. By assumption, Spoiler has a strategy with segment depth d to win the slope game. As before, we prescribe a strategy for her in the simulation game that will re-use her slope game strategy 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 cycle is less steep than (ρ,ρ'). After some rollbacks, the counter values will be of the form:

$$\begin{aligned} n &= m + a + (b_1 + \ldots + b_l) + (c_1 + \ldots + c_k) \quad \text{and} \\ n' &= m' + a' + (b_1' + \ldots + b_l') + (c_1' + \ldots + c_k'), \end{aligned} \tag{4.10}$$

where the absolute values of a and a' are at most C, the vectors (c_i,c_i') are behind (ρ,ρ') and positive, and the vectors (b_i,b_i') are behind (ρ,ρ') and non-positive. We apply Lemma 4.4 and obtain that all the positions so far have been below (ρ,ρ').

In general Spoiler has no power to choose whether the effect of the cycle at the 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 (4.10), suppose (c_k,c_k') 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 (n,n') of counter values in the simulation game is C-below some vector (γ,γ') which is equivalent to the positive effect (c_k,c_k') of the last rolled back cycle.

Proof. By an easy geometric argument. Ignore vectors (b_i,b_i') as they preserve being C-below all positive vectors 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 $(qn,q'n')$ be a position of the simulation game satisfying the claim. We know that Spoiler has a winning strategy in the slope game from $((q,q'),(c_k,c_k'))$, of segment depth at most $d-1$. Because (c_k,c_k') is equivalent to (γ,γ'), we can apply Lemma 4.8 (page 36) and know that the same strategy is winning in the slope game from $((q,q'),(\gamma,\gamma'))$. By the induction assumption we conclude that Spoiler wins the simulation game from $(qn,q'n')$, which completes the proof of Lemma 4.9. □

Lemma 4.10. *Suppose Duplicator has a winning strategy in the slope game from a position* $((p,p'),(\rho,\rho'))$. *Then Duplicator wins the simulation game from every position* $(pm,p'm')$ *which is C-above* (ρ,ρ').

Proof. We will again build on the concept of rollbacks. There are two cases, depending on the segment height d of Duplicator's supposed winning strategy in the slope game.

Case $d = 1$. Duplicator wins the slope game immediately after the first phase. Then he can enforce that the effects $(\Delta(\beta),\Delta'(\beta))$ of the cycles β of all observed lassos are not behind (ρ,ρ'). Let π be an arbitrary play of the simulation game, in which Duplicator uses rollbacks to play according to the segment of this slope game strategy. The effects of π can be decomposed as

$$\begin{aligned}\Delta(\pi) &= \Delta(\alpha) + \Delta(\beta_1\beta_2\dots\beta_k)\\ \Delta'(\pi) &= \Delta'(\alpha) + \Delta'(\beta_1\beta_2\dots\beta_k)\end{aligned} \tag{4.11}$$

where α is some acyclic path and β_i are simple cycles with effect-vector not behind (ρ,ρ'). Because the initial pair of counter values (m,m') is C-above (ρ,ρ') and $|\alpha| \leq C$, we know that $(m,m') + (\Delta(\alpha),\Delta'(\alpha))$ is above (ρ,ρ'). Moreover, as all the effects of all β_i are not behind (ρ,ρ'), their sum $\sum_{i \leq k}(\Delta(\beta_i),\Delta'(\beta_i))$ is also not behind (ρ,ρ'). Using part 2 of Lemma 4.4 we get that $(n,n') = (m,m') + (\Delta(\pi),\Delta'(\pi))$ is still above (ρ,ρ'). This in particular means that Duplicator's counter value n' remains non-negative. Since π was arbitrary, this shows that Duplicator can prevent his counter from ever decreasing below 0 and thus enforce an infinite play and win.

Case $d > 1$. We prescribe a winning strategy for Duplicator in the simulation game that is based on the assumed winning strategy σ in the slope game. Intuitively, Duplicator plays according to σ until a leaf-segment is reached and then continues to play according to this segment using rollbacks. Since $d_{\max} = (K+1)^2$ bounds the maximal number of segments in σ and every path in a segment is no longer than C, we know that an offset of $d_{\max} \cdot C$ is sufficient to ensure that some position in a bottom segment can be reached. From then on, all visited points must be above the slope of this segment, as in the previous case.

We can accelerate this strategy, allowing *forward jumps*: Duplicator starts to play according to the initial segment of σ at height d. At any given position in a segment at height h, Duplicator first checks if the same pair of control states appears in a segment at a lower height $h' < h$. If such a position exists, Duplicator continues to play from there, otherwise he plays as prescribed by the current position. See Figure 4.4 on page 42 for an illustration.

If Duplicator plays as described above, he guarantees that no control states are repeated unless he is already in a leaf segment. Let π be some play of the simulation game in which Duplicator plays as above. This play must have the form $\pi = \gamma\delta$, where γ is some short acyclic prefix that contains all moves played according to non-leaf segments and δ is an unfolding of some leaf segment T.

Because Duplicator uses forward jumps as soon as possible, we know that no pair of states visited in the prefix γ can be contained in T. Moreover, as T itself is a winning strategy in the slope game for some slope (φ,φ'), we know that the effect of the suffix δ, and therefore also that of the whole path π, can be decomposed as in Equation (4.11), where again, α is some acyclic path and the effects of all cycles β_i (and their sum) are not behind (φ,φ'). We can now make the same estimation as in the previous case.

The initial point of counter values (m,m') is C-above (ρ,ρ') and thus also C-above (φ,φ') because $(\varphi,\varphi') \prec (\rho,\rho')$. Since α is acyclic it cannot be longer than C, which means that $(m,m') + (\Delta(\alpha),\Delta'(\alpha))$ is above (φ,φ'). Knowing that $(\Delta(\beta_1\beta_2\ldots\beta_k),\Delta'(\beta_1\beta_2\ldots\beta_k))$ is not behind (φ,φ'), we can one more time apply part 2 of Lemma 4.4 and derive that $(n,n') = (m,m') + (\Delta(\pi),\Delta'(\pi))$ is still above (φ,φ') and therefore that Duplicator's counter value n'

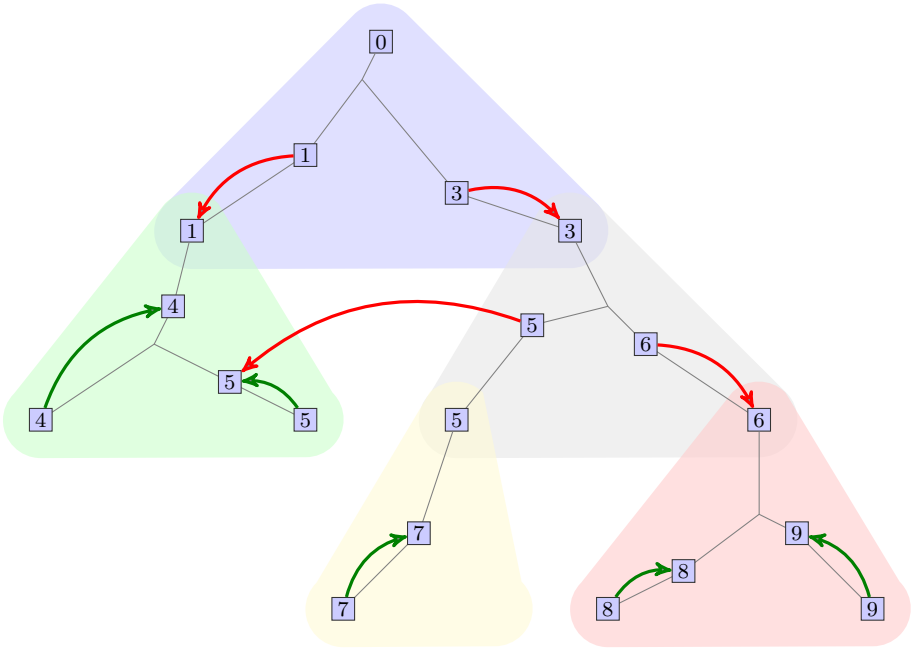


Figure 4.4: A strategy for Duplicator in the slope game is turned into a strategy in the simulation game by inserting forward jumps (red) and rollbacks (green). The green, yellow and red segments have height 1, the grey and blue segments have height 1 and 2 respectively. Nodes with the same labeling indicate positions with the same pair of control states.

remained non-negative.

Just as in the first case, we observe that the play π was arbitrarily chosen (by Spoiler). The prescribed strategy therefore allows Duplicator to maintain a non-negative counter value and hence to ensure an infinite play. □

4.3 Proof of the Belt Theorem

Assume a pair $\mathcal{N},\mathcal{N}'$ of OCNs in normal form and fix the value $C \in \mathbb{N}$ from Lemmas 4.9 and 4.10. It is the maximal length of a simple cycle in the product graph of the nets $\mathcal{N}$ and $\mathcal{N}'$. For two states $q \in Q$ and $q' \in Q'$ we will determine the ratio (ρ,ρ') that, together with C, characterizes the belt of the plane (q,q'). First observe the following monotonicity property of the slope game.

Lemma 4.11. *If Spoiler wins the slope game from a position* $((q,q'),(\rho,\rho'))$ *and* $(\alpha,\alpha') \prec (\rho,\rho')$ *then Spoiler also wins the slope game from* $((q,q'),(\alpha,\alpha'))$.

Proof. Assume that Spoiler wins from the position $((q,q'),(\rho,\rho'))$ while Duplicator wins from $((q,q'),(\alpha,\alpha'))$, for some slope $(\alpha,\alpha') \prec (\rho,\rho')$. This means that a point $(n,n') \in \mathbb{N}\times\mathbb{N}$ exists which is both C-above (α,α') and C-below (ρ,ρ'). Applying both Lemmas 4.9 and 4.10

immediately yields a contradiction. □

Equivalently, if Duplicator wins the slope game from $((q,q'),(\rho,\rho'))$ and (α,α') is steeper than (ρ,ρ') then he also wins from $((q,q'),(\alpha,\alpha'))$. We conclude that for every pair (q,q') of states, there is a *boundary slope* $(\beta,\beta') \in \mathbb{R}\times\mathbb{R}$ such that

1. Spoiler wins the slope game from $((q,q'),(\alpha,\alpha'))$ for every $(\alpha,\alpha') \prec (\beta,\beta')$;
2. Duplicator wins the slope game from $((q,q'),(\alpha,\alpha'))$ for every $(\alpha,\alpha') \succ (\beta,\beta')$.

Note that we claim nothing about the winner from the position $((q,q'),(\beta,\beta'))$ itself. Applying Lemmas 4.9 and 4.10 we see that this boundary slope (β,β') satisfies the claims 1 and 2 of Theorem 4.2. 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 vector (α,α') less steep than (β,β') such that (n,n') is C-below (α,α'). By point 1 above, Spoiler wins the slope game from $((q,q'),(\alpha,\alpha'))$. By Lemma 4.9, Spoiler wins the simulation game from $(qn,q'n')$. Analogously, one can use point 2 above together with Lemma 4.10 to show the second condition of Theorem 4.2. This concludes the proof of the Belt Theorem. □

We recall Definition 4.7 (page 36) of equivalent vectors, that is based on the set V, of ratios of simple cycles and their opposite vectors. Two vectors are equivalent if they are behind the same vectors from V. Lemma 4.8 states that the outcome of a slope game from a fixed pair of states is the same for equivalent initial slopes.

By Lemma 4.8, a boundary slope (β,β') as used in the proof above must correspond to a slope contained in V. Indeed, otherwise (β,β') must be between two vectors (γ,γ') and (δ,δ') of V and there is some (α,α') satisfying $(\gamma,\gamma') \prec (\beta,\beta') \prec (\alpha,\alpha') \prec (\delta,\delta')$, and which is equivalent to (γ,γ'). By Lemma 4.8, the outcome of a slope game for (γ,γ') or (α,α') is the same, contradicting that (β,β') is a boundary.

We conclude that the slope (β,β') of any belt must be the effect of a simple cycle of the product graph. Such paths are no longer than C and because along a path of length C the counter values cannot change by more than C, we get that $\beta,\beta' \leq C$ as well. Below we explicitly state the stronger version of the Belt Theorem that includes the derived bounds for future reference.

Theorem 4.12 (Belt Theorem)**.** *Let $\mathcal{N}$ and $\mathcal{N}'$ be two OCNs in normal form, with sets of states Q and Q' respectively and let $C \leq |Q\times Q'| \in \mathbb{N}$ be the maximal length of an acyclic path in the product graph of $\mathcal{N}$ and $\mathcal{N}'$. Then for every pair $(q,q') \in Q\times Q'$ of states there is a positive vector (ρ,ρ') such that*

1. *if (n,n') is C-above (ρ,ρ') then $qn \preceq q'n'$,*
2. *if (n,n') is C-below (ρ,ρ') then $qn \not\preceq q'n'$,*
3. *$\rho,\rho' \leq C$.*

4.4 Locality

We mentioned before that simulation enjoys a certain locality property due to the simulation condition. Intuitively, the outcomes of all possible successor positions after one round of the simulation game determine the outcome of the game. For OCAs, this can be stated as a precise geometric property. Whether or not one process simulates another is completely determined by their control states and the colouring of its surrounding pairs.

Definition 4.13. Let $R \subseteq (Q \times \mathbb{N} \times Q' \times \mathbb{N})$ be some relation on the configurations of two OCA with statesets Q and Q' respectively. The R-neighbourhood of $(m,m') \in \mathbb{N}^2$ is the function $NH_R^{(m,m')} : (Q \times Q' \times \{-1,0,1\} \times \{-1,0,1\}) \to \{0,1,\perp\}$ with

$$NH_R^{(m,m')}(q,q',l,l') = \begin{cases} 1, & \text{if } (qm+l, q'm'+l') \in R \\ 0, & \text{if } (qm+l, q'm'+l') \in (Q \times \mathbb{N} \times Q' \times \mathbb{N}) \setminus R \\ \perp, & \text{if } (qm+l, q'm'+l') \notin (Q \times \mathbb{N} \times Q' \times \mathbb{N}) \end{cases} \tag{4.12}$$

The R-neighbourhood of (m,m') determines the colouring of R on all points surrounding (m,m'). Observe that there are at most $3^{|Q \times Q'| \cdot 3 \cdot 3}$ different neighbourhoods. The $\perp$-values ensure that if two points (m,m') and (n,n') in $\mathbb{N}^2$ have the same neighbourhood, then they have the same relative position to the axes, i.e., $m = 0 \iff n = 0$ and $m' = 0 \iff n' = 0$.

We can now precisely state what we mean with the *locality* of simulation on OCA.

Lemma 4.14 (Locality)**.** *Consider a pair $(p,p') \in (Q \times Q')$ of states and naturals $m,m',n,n' \in \mathbb{N}$. If the $\preceq$-neighbourhoods of (m,m') and (n,n') agree on every $(q,q',l,l') \neq (p,p',0,0)$, then they also agree on $(p,p',0,0)$, i.e., $pm \preceq p'm' \iff pn \preceq p'n'$.*

Proof. Assume otherwise and let $pm \preceq p'm'$ but $pn \not\preceq p'n'$ and consider an optimal initial winning Spoiler-move $pn \xrightarrow{a} qn+l$ in the simulation game from $(pn,p'n')$. For the move $pm \xrightarrow{a} qm+l$ in the game from $(pm,p'm')$, Duplicator has a response $p'm' \xrightarrow{a} q'm'+l'$ such that $qm+l \preceq q'm'+l'$. Due to the optimality of Spoiler's initial move we know that $(q,q',l,l') = (p,p',0,0)$ and therefore that $NH_{\preceq}^{(m,m')}(q,q',l,l') = 1 = NH_{\preceq}^{(n,n')}(q,q',l,l')$. In particular, this means that $qn+l \preceq q'n'+l'$. Using the analogous response $p'n' \xrightarrow{a} q'n'+l'$ to Spoiler's initial move in the game from $(pn,p'n')$, Duplicator can ensure that the game continues from position $(qn+l,q'n'+l)$, which is winning for him.

We contradicted the existence of an optimal initial Spoiler-move in the simulation game from $(pn,p'n')$. This shows that indeed $pn \preceq p'n'$ holds, which contradicts our assumption. □

Since the simulation condition for a pair of processes depends only on their neighbourhood, we can locally verify that some finite colouring is not self-contradictory. Moreover, if a relation

on the configurations of two OCA is not a simulation, then this is witnessed locally by some inconsistent neighbourhood.

Lemma 4.15. *A relation $R \subseteq (Q \times \mathbb{N} \times Q' \times \mathbb{N})$ is a simulation if for every $(pm,p'm') \in R$ there exists $(n,n') \in \mathbb{N}^2$ with $NH_R^{(m,m')} = NH_{\leq}^{(n,n')}$.*

Proof. The condition immediately implies that R satisfies the simulation condition. □

4.5 Upper Bounds

As before, we fix two OCN $\mathcal{N}$ and $\mathcal{N}'$ in normal form, with sets of control states Q and Q', respectively and let $C \leq |Q \times Q'|$ be the maximal length of an acyclic path in their product, as used in Theorem 4.12.

For convenience, we will write $(pm,p'm') + k \cdot (n,n')$ to mean $(p(m + k \cdot n), p'(m' + k \cdot n'))$ for any $(p,p') \in (Q \times Q')$ and $m,m',n,n',k \in \mathbb{N}$. Similarly, for a relation $R \subseteq (Q \times \mathbb{N} \times Q' \times \mathbb{N})$ we write $R + k \cdot (n,n') = \{(pm,p'm') + k \cdot (n,n') \mid (pm,p'm') \in R\}$.

Definition 4.16. The *slope* of a pair $(p,p') \in Q \times Q'$ of control states, is the positive vector $\textsc{slope}(p,p') = (\rho,\rho')$ satisfying the claim of the Belt Theorem. The *belt* with slope (ρ,ρ') is the set of points $(n,n') \in \mathbb{N}^2$ which are neither C-above nor C-below (ρ,ρ'). The *extended belt* is the relation $\textsc{belt}(p,p') \subseteq (Q \times \mathbb{N} \times Q' \times \mathbb{N})$ that contains $(qn,q'n')$ iff (n,n') is in the belt with slope $\textsc{slope}(p,p')$.

Recall that simulation preorder on the configurations with control states p and p' is trivial outside of $\textsc{belt}(p,p')$: it contains all pairs $(pm,p'm')$ s.t. (m,m') is C-above $\textsc{slope}(p,p')$, and contains no pairs $(pm,p'm')$ where (m,m') is C-below $\textsc{slope}(p,p')$. We show (Lemma 4.18) that the non-trivial part

$$\preceq_{p,p'} = \preceq \cap \textsc{belt}(p,p')$$

is repetitive in the sense defined in Definition 4.17 below. Essentially, one can cut through the belt at two levels $n_1,n_2 \in \mathbb{N}$ such that the colouring of $\textsc{belt}(p,p')$ above level n_2 repeats the (finite) colouring between n_1 and n_2 indefinitely. This implies that $\preceq_{p,p'}$ and hence also $\preceq$ are semilinear, and each $\preceq_{p,p'}$ can be represented by the finite colouring up to level n_2. This is already enough to decide strong simulation, and to compute a representation of the maximal simulation, since one can enumerate candidate relations $R \subseteq (Q \times \mathbb{N} \times Q' \times \mathbb{N})$ that are represented in this way and check that they satisfy the simulation condition.

Due to the polynomial bounds on the width and the slopes of belts provided by Theorem 4.12, we can further bound the cut-levels n_1,n_2 and thus the representation of periodic candidate relations, exponentially in the size of the input nets. The crucial idea for deciding

OCN $\preceq$ OCN in PSPACE is that one can stepwise guess and locally verify the colouring of a (extended) belt by shifting a polynomially bounded window along the belt.

By Theorem 4.12, we know that coefficients ρ and ρ' of any slope $\textsc{slope}(p,p') = (\rho,\rho')$ are bounded by C. Consequently, there are at most C^2 different slopes and belts and apart from vertical and horizontal slopes (those with $\rho = 0$ or $\rho' = 0$ respectively), the maximally and minimally steep (cf. Definition 4.3 on page 34) possible slopes are $(1,C)$ and $(C,1)$ respectively. We can therefore find polynomially bounded $l_0,l_0' \in \mathbb{N}$ such that belts are pairwise disjoint outside the *initial rectangle* L_0 between corners $(0,0)$ and (l_0,l_0'). For technical convenience we assume w.l.o.g. that only horizontal belts (those with $\textsc{slope}(p,p') = (n,0)$ for some n) cross the vertical border of L_0. This can always be achieved by extending L_0, if necessary.

By our definition of belts, shifting a point along the vector $\textsc{slope}(p,p')$ preserves membership in $\textsc{belt}(p,p')$, i.e., for every $(qn,q'n') \in (Q \times \mathbb{N} \times Q' \times \mathbb{N})$,

$$(qn,q'n') \in \textsc{belt}(p,p') \iff (qn,q'n') + k \cdot \textsc{slope}(p,p') \in \textsc{belt}(p,p'). \tag{4.13}$$

This is why we restrict our focus to multiples of vectors $\textsc{slope}(p,p')$.

Definition 4.17. Fix a pair $(p,p') \in Q \times Q'$ and $j,k \in \mathbb{N}$ and let $l_0,l_0' \in \mathbb{N}$ define the initial rectangle L_0 discussed above. We write $\textsc{rect}(p,p',j)$ for the rectangle between corners $(0,0)$ and $(l_0,l_0') + j \cdot \textsc{slope}(p,p')$. A subset $R \subseteq \textsc{belt}(p,p')$ is called (j,k)-*ultimately-periodic* if for all $(n,n') \in \mathbb{N}^2 \setminus \textsc{rect}(p,p',j)$ and every $(q,q') \in (Q \times Q')$,

$$(qn,q'n') \in R \iff (qn,q'n') + k \cdot \textsc{slope}(p,p') \in R. \tag{4.14}$$

One can represent a (j,k)-ultimately-periodic set R by the two numbers $n_1' = l_0' + j \cdot \rho'$ and $n_2' = n_1' + k \cdot \rho'$ and two finite sets

$$\{(qn,q'n') \in R \mid n' \leq n_1'\} \qquad \text{and} \qquad \{(qn,q'n') \in R \mid n_1' \leq n' < n_2'\}. \tag{4.15}$$

This in particular means that R is semilinear, where the subsets above form the basis and period respectively. We continue to show that the non-trivial part $\preceq_{p,p'}$ of the colouring of simulation is such a (j,k)-ultimately periodic set for every pair (p,p') of states.

Lemma 4.18. *For every pair $(p,p') \in Q \times Q'$, the set $\preceq_{p,p'}$ is (j,k)-ultimately periodic for some $j,k \in \mathbb{N}$ exponentially bounded in C.*

Proof. Fix states p,p' and let $(\rho,\rho') = \textsc{slope}(p,p')$. W.l.o.g. suppose that $\textsc{slope}(p,p')$ is positive and $\textsc{belt}(p,p')$ therefore intersects the horizontal border of L_0 (if the belt is horizontal and intersects the vertical border of L_0 the proof is analogous).

By a *cross-section* at level n' we mean the set of all points in $\textsc{belt}(p,p')$ on a horizontal line at that level, i.e., $\{(qn,qn') \in \textsc{belt}(p,p') \mid n \in \mathbb{N}\}$. 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 $\textsc{slope}(p,p') = (\rho,\rho')$

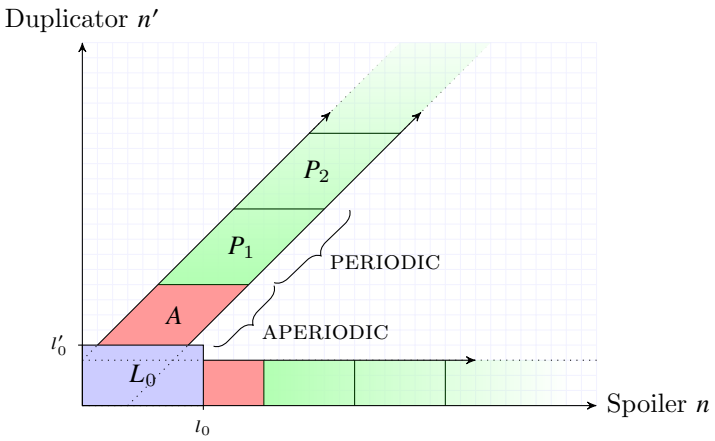


Figure 4.5: The initial rectangle L_0 (blue) and two belts. Outside L_0, the colouring of a belt consists of some exponentially bounded block (red), and another exponentially bounded non-trivial block (green) which repeats ad infinitum along the rest of the belt.

and moreover, the $\preceq$-neighbourhoods of any two corresponding points are the same. Formally, we require that for some $k \in \mathbb{N}$,

1. $s_2 = s_1 + k \cdot (\rho, \rho')$
2. $NH_{\preceq}^{(m,m')} = NH_{\preceq}^{(n,n')}$ for any $(qn, q'n') \in s_1$ and $(m,m') = (n,n') + k \cdot \textsc{slope}(p,p')$.

Notice that there are at most $2^{K \cdot W}$ pairwise different colourings for any cross-section, where $K = |Q \times Q'|$ and W is the maximal width of a belt. By our definition of neighbourhoods, two cross-sections are equal only if their colouring agrees and the same is true for the (pairs of) cross-sections directly above and below. This means that in total, there are no more than $\rho' \cdot 2^{K \cdot W \cdot 3}$ pairwise different cross-sections for a given belt.

We choose two equal cross-sections at levels n'_1 and n'_2 respectively, such that $n'_1 = l'_0 + j \cdot \rho'$ and $n'_2 = n'_1 + k \cdot \rho'$ for some $j,k \in \mathbb{N}$. That is, we demand that $l'_0 < n'_1 < n'_2$ and the respective offsets are divisible by the vertical offset ρ' of $\textsc{slope}(p,p')$. By our observation above it is safe to assume that both j and k are bounded exponentially in C.

Based on n'_1 and n'_2, we decompose $\preceq_{p,p'}$ into finite segments. To this end, first extend n'_1 and n'_2 to an infinite progression $n'_1, n'_2, n'_3, \ldots$ where $n'_{i+1} = n'_i + k \cdot \rho'$ for $i \geq 1$. Now let A be the restriction of $\preceq_{p,p'}$ to the area below n'_1 and for any $i \geq 1$, let P_i be the restriction of $\preceq_{p,p'}$ to the area between n'_i and n_{i+1} (see Figure 4.5):

$$A = \{(qn, qn') \in \preceq_{p,p'} : n' < n'_1\} \qquad P_i = \{(qn, qn') \in \preceq_{p,p'} : n'_i \leq n' < n'_{i+1}\}.$$

We now show that

$$\preceq_{p,p'} = A \cup P_1^*, \quad \text{where } P_1^* = \bigcup_{i \in \mathbb{N}} (P_1 + i \cdot k \cdot (\rho, \rho')). \tag{4.16}$$

That is, apart from the initial fragment A, the colouring of $\preceq_{p,p'}$ is actually an infinite repetition of a finite colouring P_1 along the belt: $P_{i+1} = P_i + k \cdot \text{SLOPE}(p,p')$. This implies the claim of the lemma, since $A \cup P_1^*$ is clearly (j,k)-ultimately periodic. The proof of Equation (4.16) strongly relies on the locality of the simulation condition (Lemma 4.15 on page 45).

For the first inclusion ($A \cup P_1^* \subseteq \preceq_{p,p'}$) we show that the relation

$$R = (\preceq \setminus \preceq_{p,p'}) \cup (A \cup P_1^*) \tag{4.17}$$

obtained from $\preceq$ by replacing $\preceq_{p,p'}$ with $A \cup P_1^*$, is a simulation. Recall that n_1' and n_2' were chosen sufficiently high (above the initial rectangle L_0) such that any two different belts are disjoint. This means that the R-neighbourhood of any point in P_i for some $i > 1$ is the same as the R-neighbourhood and hence also the $\preceq$-neighbourhood of the corresponding point in P_1. By Lemma 4.15, this means that R is a simulation and since $\preceq$ is the largest simulation, the claimed inclusion follows.

It remains to show the other inclusion ($A \cup P_1^* \supseteq \preceq_{p,p'}$). Assume the contrary. We already know that $A \cup P_1^* \subseteq \preceq_{p,p'}$, so we must have $P_1 + i \cdot k \cdot \text{SLOPE}(p,p') \subsetneq P_i$ for some $i > 1$. Since $P_i \subseteq \preceq_{p,p'}$ is part of the colouring of simulation $\preceq$, it is clearly locally consistent. This means if we replace P_1 with the colouring according to P_i, we again derive a consistent colouring. Formally, we let $P = P_i \cdot i \cdot (-k) \cdot \text{SLOPE}(p,p')$ and replace $\preceq_{p,p'}$ with $A \cup P^*$ in the colouring of $\preceq$. Similar to the first case, the resulting relation

$$(\preceq \setminus \preceq_{p,p'}) \cup (A \cup P^*) \tag{4.18}$$

is a simulation due to the locality of the simulation condition. This implies that $P_1 \subsetneq P \subseteq \preceq_{p,p'}$, which means that there exists some point $(qn,q'n') \in \preceq_{p,p'} \setminus P_1$ with $n_1' \leq n' < n_2'$. This contradicts the definition of P_1 as the set of points $(qn,q'n')$ in $\preceq_{p,p'}$ with $n_1' \leq n' < n_2'$. □

Lemma 4.18 implies that the largest strong simulation $\preceq$ is not only semilinear, but the finite union of (j,k)-ultimately periodic sets, for exponentially bounded j,k. It therefore admits an EXPSPACE representation that consists, for every pair of states (p,p'), of:

- a polynomially bounded vector $(\rho,\rho') = \text{SLOPE}(p,p')$
- exponentially bounded natural numbers $n_1', n_2' \in \mathbb{N}$
- two exponentially bounded relations:

$$\text{APERIODIC} = \{(qn,q'n') \in \preceq \mid n' \leq n_1'\}$$
$$\text{PERIODIC} = \{(qn,q'n') \in \preceq \mid n_1' \leq n' < n_2'\}$$

Assume w.l.o.g. that in descriptions of the above form, the coefficients n_1' and n_2' are the same for all pairs (p,p') with the same $\text{SLOPE}(p,p')$. This is a safe assumption as the least common multiples of the respective values are still exponentially bounded.

The above characterization immediately leads to a naïve EXPSPACE-decision procedure for simulation OCN $\preceq$ OCN, between processes of OCNs in normal form: 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. But due to the particular periodic structure of the candidate relation and the locality of simulation (Lemma 4.14 on page 44), it suffices to locally verify the finite initial and periodic parts for every pair of control states.

A PSPACE procedure. The naïve algorithm outlined above may easily be turned into a PSPACE algorithm by a window shifting 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 APERIODIC and later PERIODIC, inside a polynomially bounded rectangle window through the belt and shifts this window along the belt, checking the simulation condition for all contained points along 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. By Lemma 4.18, this repetition needs to occur after an exponentially bounded number of shifts. Therefore, 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.

We summarise our findings as the theorem below.

Theorem 4.19. *Checking strong simulation preorder between two OCNs is in PSPACE. Moreover, the largest simulation relation is semilinear and can be represented in space exponential in the number of states of the input nets.*

In Section 5.4 we will be particularly interested in identifying and computing the exact width of *vertical* belts. We conclude this section by showing that this can be done in polynomial space.

Let us write $suf(q,q')$ for the least value $n \in \mathbb{N}$ such that $qn \not\preceq q'n'$ for every $n' \in \mathbb{N}$ and ω if no such value n exists. In terms of the simulation game, this is the minimal initial counter value that is sufficient for Spoiler to win against any initial value for Duplicator if we fix the initial states to q and q'. Observe that $suf(q,q') = \omega$ iff the belt for the plane (q,q') is *not* vertical.

The following is an easy consequence of Theorem 4.19, because one can check the simulation problem for selected positions.

Lemma 4.20. *Given OCNs $\mathcal{N}$ and $\mathcal{N}'$ in normal form, with state-sets Q and Q'. For any given pair $(q,q') \in Q \times Q'$ of states, the value $suf(q,q')$ can be computed in PSPACE. Moreover, if*

$suf(q,q') \neq \omega$, then it is bounded by C, the maximal length of an acyclic path in the product of $\mathcal{N}$ and $\mathcal{N}'$.

Proof. By Theorem 4.12, we can bound the coefficients of the slopes of all belts polynomially. In particular, we know that if (ρ,ρ') is the slope of some belt then ρ and ρ' are both non-negative and no bigger than $C \leq |Q \times Q'|$. The steepest possible such slope that is not vertical (i.e., with $\rho > 0$) is thus given by the vector $(\alpha,\alpha') = (1,C)$.

To check if $suf(q,q') = \omega$ we can pick a point (n,n') that is both C-above (α,α') and C-below of (α,α') and check if $qn \preceq q'n'$ holds. For instance, $n = C+1$ and $n' = 2(C+1)^2 \cdot C$ is surely such a point. If $suf(q,q') = \omega$, then the belt for (q,q') is vertical and by Theorem 4.12, point 2, we have $qn \not\preceq q'n'$. Otherwise, the belt is not vertical and has slope $(\rho,\rho') \prec (\alpha,\alpha')$. Then by point 1 of Theorem 4.12, we must have $qn \preceq q'n'$.

To compute $suf(q,q') \in \mathbb{N}$ for a vertical belt recall that by point 1 of Theorem 4.12, $qn \not\preceq q'n'$ for all points with $n > C$. Clearly, this means that $suf(q,q')$ is bounded by $C \leq |Q \times Q'|$. By Lemma 4.18, the colouring on this belt must be repetitive from some exponentially bounded level n'_0 onwards. By monotonicity, this means that the colouring of the belt must have stabilized at this level already, so that for all $n' \geq n'_0$, we have $qn \preceq q'n'$ iff $n < suf(q,q')$.

We can now iteratively check the colour of the point (n,n'_0) for decreasing values $n \in \mathbb{N}$, starting with C. By Theorem 4.19, this can surely be done in polynomial space. The value $suf(q,q')$ must be the largest considered $n < C$ where $qn \not\preceq q'n'_0$ still holds. □

Chapter 5

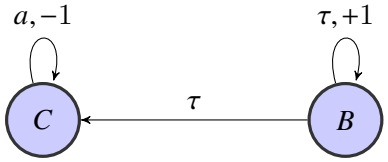
Weak Simulation

Weak and strong simulation preorder are known to be undecidable for one-counter automata. This holds even for deterministic systems, or if one of the systems is fixed [50, 30].

For one-counter nets, we have shown weak simulation to be decidable [20] and PSPACE-complete [19]. In this section we give a unified presentation of the argument for its decidability and the subsequent improvement to PSPACE.

The main obstacle is that, with respect to weak steps, Duplicator's system is infinitely-branching. This implies that non-simulation does not necessarily manifest itself locally, i.e., normal weak simulation approximants do not necessarily converge at level ω.

Example 5.1. Consider the simple process $A \circlearrowright a$, that can only loop on action a, and the OCN depicted below. Notice that the disjoint union of A and the LTS induced by this net is isomorphic to the one of Example 2.21 on page 17.



We see that $A \leq_n Cn$ and $A \not\leq_{n+1} Cn$ hold for every $n \in \mathbb{N}$. Moreover, there is a weak step $B0 \stackrel{a}{\Longrightarrow} Cn$ for every $n \in \mathbb{N}$ and therefore $A \leq_\omega B0$. Still, it holds that $A \not\leq_{\omega+1} B0$ because there is no weak a-step from $B0$ to a process α that satisfies $A \leq_\omega \alpha$. It follows that $\leq_\omega \neq \leq_{\omega+1}$. We will later show (as Theorem 5.18 on page 65) that convergence of weak simulation approximants on OCN can only be guaranteed at level ω^2.

We resolve this problem in two steps. First, the weak simulation problem between OCN is reduced to a *strong* simulation problem between OCN and a slightly generalized model that we call ω-Nets, and that symbolically captures infinite branching. In ω-Nets, there exist dedicated transitions with symbolic effect ω, which allow to arbitrarily increase the counter in a single step. Secondly, this new strong simulation problem is solved using a novel kind of

approximant sequence, that is derived from the representation of Duplicator's system. It is shown that this sequence converges at a finite index and that individual approximant relations are effectively computable semilinear sets. In particular, knowing the representation of the approximant at level k, one can characterize the approximant at the next level $k+1$ in terms of strong simulation over suitably modified OCN, which is an effectively computable semilinear set by Theorem 4.19 (page 49). A description of the weak simulation preorder $\leqq$ can therefore be computed by successively computing the approximant relations and stopping once convergence is detected. This procedure is effective because the semilinear approximants are guaranteed to converge to $\leqq$ at some finite level and equality is decidable for semilinear sets.

The remainder of the section is organized as follows. In Section 5.1, ω-Nets are introduced and the reduction theorem (Theorem 5.3) is proved. In Section 5.2 we introduce and discuss approximants and show that they indeed converge towards simulation at some finite level. The main iterative construction to compute representations of approximants is described in Section 5.3. Afterwards, in Section 5.4, we make a closer analysis of the complexity of this procedure. In Section 5.5 we discuss the subproblems of weak simulation where one process is finite.

5.1 ω-Nets

Definition 5.2. An *ω-net* $\mathcal{N} = (Q, \mathrm{Act}, \delta)$ is given by a finite set of control states Q, a finite set of actions Act and transitions $\delta \subseteq Q \times \mathrm{Act} \times \{-1, 0, 1, \omega\} \times Q$. It induces a transition system over the state set $Q \times \mathbb{N}$ that allows a step $pm \xrightarrow{a} qn$ if either $(p,a,d,q) \in \delta$ and $n = m + d \in \mathbb{N}$ or if $(p,a,\omega,q) \in \delta$ and $n > m$.

A *path* in $\mathcal{N}$ is a sequence $\pi = p_0 t_1 p_1 t_2 \dots t_k p_k$ such that $t_i = (p_{i-1}, a_i, d_i, p_i)$ for every $1 \leq i \leq k$. We write $\lambda(\pi) = a_1 a_2 \dots a_k \in \mathrm{Act}^*$ for the sequence of actions it induces and $|\pi| = k$ for its length. The *effect* $\Delta(\pi)$ of such a path is the minimum of ω and $\sum_{1 \leq i \leq k} d_i$. Its *guard* is $\Gamma(\pi) = -\min\{\Delta({}^i\pi) \mid i \leq k\}$, where ${}^i\pi$ denotes the prefix $p_0 t_1 p_1 t_2 \dots t_i p_i$ of π of length i.

Observe that the effect of a path is ω iff it contains at least one ω-transition and is otherwise bounded by the length of the path. Moreover, the guard of a path π equals the guard of its longest prefix without ω-transitions and therefore satisfies $0 \leq \Gamma(\pi) \leq |\pi|$.

Every one-counter net is an ω-net without ω-transitions. Unlike one-counter nets, ω-nets can yield infinitely branching transition systems, since each ω-transition (p,a,ω,q) introduces steps $pm \xrightarrow{a} qn$ for any two naturals $n > m$. We observe that, just like one-counter nets, ω-nets are monotone in the sense of Lemma 3.13 (page 27):

$$pm \xrightarrow{a} qn \text{ implies } p(m+d) \xrightarrow{a} q(n+d) \text{ for all } d \in \mathbb{N}. \tag{5.1}$$

This means that $pm \preceq p'm'$ implies $pn \preceq p'n'$ for $n \leq m$, $n' \geq m'$. The following theorem justifies our focus on strong simulation games where Duplicator plays on an ω-net process.

Theorem 5.3. *Checking weak simulation between two OCN processes can be reduced to checking strong simulation between a one-counter net process and an ω-net process. Formally, for two OCNs $\mathcal{N}$ and $\mathcal{N}'$ with state sets Q and Q', respectively, one can construct a OCN $\mathcal{M}$ with states $M \supseteq Q$ and an ω-net $\mathcal{M}'$ with states $M' \supseteq Q'$ such that for each pair $q,q' \in Q \times Q'$ of original control states, any $n,n' \in \mathbb{N}$ and ordinal α the following hold.*

1. *$qn \lessapprox q'n'$ w.r.t. $\mathcal{N},\mathcal{N}'$ iff $qn \preceq q'n'$ w.r.t. $\mathcal{M},\mathcal{M}'$.*
2. *If $qn \lessapprox_\alpha q'n'$ w.r.t. $\mathcal{N},\mathcal{N}'$ then $qn \preceq_\alpha q'n'$ w.r.t. $\mathcal{M},\mathcal{M}'$.*

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

The idea of the proof is to look for counter-increasing cyclic paths via τ-labelled transitions in the control graph of $\mathcal{N}'$ and to introduce ω-transitions accordingly. For any path that reads a single visible action and visits a 'generator' state that is part of a silent cycle with positive effect, we add an ω-transition. For all of the finitely many non-cyclic paths that read a single visible action we introduce direct transitions.

The remainder of this section is devoted to proving Theorem 5.3. All further definitions in this section are only relevant locally. Formally, the proof of Theorem 5.3 will be done in two steps. First (Lemma 5.7), we reduce weak simulation for one-counter nets to strong simulation between a one-counter net and yet another auxiliary model called *guarded ω-nets*. These differ from ω-nets in that each transition may change the counter by more than one and is explicitly guarded by an integer, i.e. it can only be applied if the current counter value exceeds the *guard* attached to it. In the second step (Lemma 5.8) we normalize the effects of all transitions to $\{-1,0,1,\omega\}$ and eliminate all integer guards and thereby construct an ordinary ω-net for Duplicator.

Definition 5.4. A *guarded ω-net* $\mathcal{N} = (Q,\mathrm{Act},\delta)$ is given by finite sets Q and Act of states and actions, and a transition relation $\delta \subseteq Q \times \mathrm{Act} \times \mathbb{N} \times \mathbb{Z} \cup \{\omega\} \times Q$. It defines a transition system over the state set $Q \times \mathbb{N}$ where $pm \xrightarrow{a} qn$ iff there is a transition $(p,a,g,d,q) \in \delta$ with

1. $m \geq g$ and
2. $n = m + d \in \mathbb{N}$ or $d = \omega$ and $n > m$.

Specifically, $\mathcal{N}$ is an *ω-net* if $g = 0$ and $d \in \{-1,0,1,\omega\}$ for all transitions $(p,a,g,d,q) \in \delta$. The next construction establishes the connection between weak similarity of one-counter nets and strong similarity between OCN and guarded ω-net processes. In order to avoid confusion we write $\longrightarrow_{\mathcal{N}}$ and $\Longrightarrow_{\mathcal{N}}$ for (weak) steps in the system $\mathcal{N}$.

Lemma 5.5. *For any OCN $\mathcal{N} = (Q,Act,\delta)$ one can effectively construct a guarded ω-net $\mathcal{G} = (Q,Act,\gamma)$ such that for all $a \in Act$,*

1. *whenever* $pm \overset{a}{\Longrightarrow}_{\mathcal{N}} qn$, *there is some* $l \geq n$ *such that* $pm \xrightarrow{a}_{\mathcal{G}} ql$
2. *whenever* $pm \xrightarrow{a}_{\mathcal{G}} qn$, *there is some* $l \geq n$ *such that* $pm \overset{a}{\Longrightarrow}_{\mathcal{N}} ql$.

Proof. The idea of the proof is to introduce direct transitions from one state to another for any path between them that reads exactly one visible action and does not contain silent cycles.

For two states s,t of $\mathcal{N}$, let $D(s,t)$ be the set of *direct* (i.e., acyclic) paths from s to t and let $SD(s,t)$ denote the subset of *silent direct* paths $SD(s,t) = \{\pi \in D(s,t) \mid \lambda(\pi) \in \{\tau\}^*\}$ from s to t. Every path in $D(s,t)$ has acyclic prefixes only and is therefore bounded in length by $|Q|$. Hence $D(s,t)$ and $SD(s,t)$ are finite and effectively computable for all pairs (s,t).

Using this notation, we define the transitions in $\mathcal{G}$ as follows. $\mathcal{G}$ contains a transition $(p,a,\Gamma(\pi),\Delta(\pi),q)$ for each path $\pi = \pi_1(s,a,d,s')\pi_2$ where $\pi_1 \in SD(p,s)$ and $\pi_2 \in SD(s',q)$. This carries over all transitions of $\mathcal{N}$, including the ones with label $a = \tau \in \mathrm{Act}$, because the empty path is in $SD(s,s)$ for all states s. Moreover, introduce ω-transitions in case $\mathcal{N}$ allows paths π_1,π_2 as above to contain direct cycles with positive effect on the counter: If there is a path $\pi = \pi_1'\pi_1''\pi_1'''(s,a,d,s')\pi_2$ with

1. $\pi_1' \in SD(p,t)$, $\pi_1'' \in SD(t,t)$ and $\pi_1''' \in SD(t,s)$
2. $\Delta(\pi_1'') > 0$

for some $t \in Q$, then $\mathcal{G}$ contains a transition $(p,a,\Gamma(\pi_1'\pi_1''),\omega,q)$. Similarly, if for some $t \in Q$, there is a path $\pi = \pi_1(s,a,d,s')\pi_2'\pi_2''\pi_2'''$ that satisfies

1. $\pi_1 \in SD(p,s)$,$\pi_2' \in SD(s',t)$, $\pi_2'' \in SD(t,t)$ and $\pi_2''' \in SD(t,q)$
2. $\Delta(\pi_2'') > 0$

add a transition (p,a,g,ω,q) with guard $g = \Gamma(\pi_1(s,a,d,s')\pi_2'\pi_2'')$. If there is an a-labelled path from p to q that contains a silent and direct cycle with positive effect, G has an a-labelled ω-transition from p to q with the guard derived from that path.

To prove the first part of the claim, assume $pm \overset{a}{\Longrightarrow}_{\mathcal{N}} qn$. By definition of weak steps, there must be a path $\pi = \pi_1(s,a,d,s')\pi_2$ with $\lambda(\pi_1),\lambda(\pi_2) \in \{\tau\}^*$. Suppose both π_1 and π_2 do not contain cycles with positive effect. Then there must be paths $\pi_1' \in SD(p,s),\pi_2' \in SD(s',q)$ with $\Gamma(\pi_i') \leq \Gamma(\pi_i)$ and $\Delta(\pi_i') \geq \Delta(\pi_i)$ for $i \in \{1,2\}$ that can be obtained from π_1 and π_2 by removing all simple cycles with effects less or equal 0. So $\mathcal{G}$ contains a transition (p,a,g',d',q) for some $g' \leq m$ and $d' \geq n - m$ and hence $pm \xrightarrow{a}_{\mathcal{G}} qn'$ for $n' = m + d' \geq n$. Alternatively, either π_1 or π_2 contains a cycle with positive effect. Note that for any such path, another path with lower or equal guard exists that connects the same states and contains only one such counter-increasing simple cycle: If π_1 contains a simple cycle with positive effect, there is a path $\bar{\pi}_1 = \pi_1'\pi_1''\pi_1'''$ from p to s, where π_1',π'' and π_1''' are direct and $\Delta(\pi_1'') > 0$ for the cycle $\pi_1'' \in SD(t,t)$ for some state t. In this case, $\mathcal{G}$ contains an ω-transition (p,a,g,ω,q) with

$g = \Gamma(\pi_1'\pi_1'')$. Similarly, if π_2 contains a counter-increasing cycle, there is a $\bar{\pi}_2 = \pi_2'\pi_2''\pi_2'''$, with $\pi_2' \in SD(s',t), \pi_2'' \in SD(t,t), \pi_2''' \in SD(t,q)$ and $\Delta(\pi_2'') > 0$. This means there is a transition (p,a,g,ω,q) in $\mathcal{G}$ with $g = \Gamma(\pi_1(s,a,d,s')\pi_2'\pi_2'')$. In both cases, $g \leq \Gamma(\pi) \leq m$ and therefore $pm \xrightarrow{a}_{\mathcal{G}} qi$ for all $i \geq m$.

For the second part of the claim, assume $pm \xrightarrow{a}_{\mathcal{G}} qn$. This must be the result of a transition (p,a,g,d,q) in $\mathcal{G}$ for some $g \leq m$. In case $d \neq \omega$, there is a path π from p to q with $\Delta(\pi) = n - m$, $\lambda(\pi) \in \{\tau\}^*\{a\}\{\tau\}^*$ and $\Gamma(\pi) = g$ that witnesses the weak step $pm \overset{a}{\Longrightarrow}_{\mathcal{N}} qn$ in $\mathcal{N}$. Otherwise, if $d = \omega$, there must be a path $\pi = \pi_{11}\pi_{12}\pi_{13}(s,a,d,s')\pi_{21}\pi_{22}\pi_{23}$ from p to q in $\mathcal{N}$ where $\Gamma(\pi) \leq m$, all π_{ij} are silent and direct and one of π_{12} and π_{22} is a cycle with strictly positive effect. This implies that one can "pump" the value of the counter higher than any given value. Specifically, there are naturals k and j such that the path $\pi' = \pi_{11}\pi_{12}^k\pi_{13}(s,a,d,s')\pi_{21}\pi_{22}^j\pi_{23}$ from p to q satisfies $\Gamma(\pi') \leq \Gamma(\pi) \leq m$ and $\Delta(\pi') \geq m - n$. Now π' witnesses the weak step $pm \overset{a}{\Longrightarrow}_{\mathcal{N}} qn'$ in $\mathcal{N}$ for an $l \geq n$. □

Remark 5.6. Observe that no transition of the net $\mathcal{G}$ as constructed above has a guard larger than $|Q| * 3 + 1$, nor any finite effect $> 2|Q| + 1$.

Lemma 5.7. *For a one-counter net $\mathcal{N}'$ one can effectively construct a guarded ω-net $\mathcal{G}'$ over the same set of states, s.t. for any OCN $\mathcal{N}$ and any two configurations $pm, p'm'$ of $\mathcal{N}$ and $\mathcal{N}'$ resp.,*

$$pm \leqq p'm' \text{ w.r.t. } \mathcal{N},\mathcal{N}' \iff pm \preceq p'm' \text{ w.r.t. } \mathcal{N},\mathcal{G}'. \tag{5.2}$$

Proof. Consider the construction from the proof of Lemma 5.5. Let $\leqq_{\mathcal{N},\mathcal{N}'}$ be the largest weak simulation w.r.t. $\mathcal{N},\mathcal{N}'$ and $\preceq_{\mathcal{N},\mathcal{G}'}$ be the largest strong simulation w.r.t. $\mathcal{N},\mathcal{G}'$.

For the "if" direction we show that $\preceq_{\mathcal{N},\mathcal{G}'}$ is a weak simulation w.r.t. $\mathcal{N},\mathcal{N}'$. Assume $pm \preceq_{\mathcal{N},\mathcal{G}'} p'm'$ and $pm \xrightarrow{a}_{\mathcal{N}} qn$. That means there is a step $p'm' \xrightarrow{a}_{\mathcal{G}'} q'n'$ for some $n' \in \mathbb{N}$ so that $qn \preceq_{\mathcal{N},\mathcal{G}'} q'n'$. By Lemma 5.5 part 2, $p'm' \overset{a}{\Longrightarrow}_{\mathcal{N}} q'l$ for some $l \geq n'$. Since simulation is monotonic (point 2 of Lemma 3.13), we know that also $qn \preceq_{\mathcal{N},\mathcal{G}'} q'l$. Similarly, for the "only if" direction, one can use the first claim of Lemma 5.5 to check that $\leqq_{\mathcal{N},\mathcal{N}'}$ is a strong simulation w.r.t. $\mathcal{N},\mathcal{G}'$. □

Lemma 5.8. *For a one-counter net $\mathcal{N}$ and a guarded ω-net $\mathcal{G}'$ with state sets Q and Q' one can effectively construct a one-counter net $\mathcal{M}$ and an ω-net $\mathcal{M}'$ with state sets $M \supseteq Q$ and $M' \supseteq Q'$ respectively, such that for any two configurations $qn, q'n'$ of $\mathcal{N}$ and $\mathcal{G}'$,*

$$qn \preceq q'n' \text{ w.r.t. } \mathcal{N},\mathcal{G}' \iff qn \preceq q'n' \text{ w.r.t. } \mathcal{M},\mathcal{M}'. \tag{5.3}$$

Proof. We first observe (see also Remark 5.6) that for any transition of the guarded ω-net $\mathcal{G}'$, the values of its guard is bounded by some constant. The same holds for all finite effects. Let $\Gamma(\mathcal{G}')$ be the maximal guard and $\Delta(\mathcal{G}')$ be the maximal absolute finite effect of any transition of $\mathcal{G}'$.

The idea of this construction is to simulate one round of the game $\mathcal{N}$ vs. $\mathcal{G}'$ in $k = 2\Gamma(\mathcal{G}') + \Delta(\mathcal{G}') + 1$ rounds of a simulation game $\mathcal{M}$ vs. $\mathcal{M}'$. We will replace original steps of both players by sequences of k steps in the new game, which is long enough to verify if the guard of Duplicator's move is satisfied and adjust the counter using transitions with effects in $\{-1,0,+1,\omega\}$ only.

We use one fresh symbol $b \notin \mathrm{Act}$ and let the new alphabet be $\widehat{\mathrm{Act}} = \mathrm{Act} \cup \{b\}$. We transform the net $\mathcal{N} = (Q, \mathrm{Act}, \delta)$ to the net $\mathcal{M} = (M, \widehat{\mathrm{Act}}, \mu)$ as follows:

$$M = Q \cup \{p_i \mid 1 \leq i < k, p \in Q\} \tag{5.4}$$

$$\mu = \{p \xrightarrow{a,d} q_k \mid p \xrightarrow{a,d} q \in \delta\} \tag{5.5}$$

$$\cup \{p_i \xrightarrow{b,0} p_{i-1} \mid 1 < i < k\} \tag{5.6}$$

$$\cup \{p_1 \xrightarrow{b,0} q\}. \tag{5.7}$$

We see that

$$pm \xrightarrow{a}_{\mathcal{N}} qn \iff pm \xrightarrow{a}_{\mathcal{M}} q_{k-1}n \xrightarrow{b^{k-2}}_{\mathcal{M}} q_1 n \xrightarrow{b}_{\mathcal{M}} qn. \tag{5.8}$$

Now we transform the guarded ω-net $\mathcal{G}' = (Q', \mathrm{Act}, \delta')$ to the ω-net $\mathcal{M}' = (M', \widehat{\mathrm{Act}}, \mu')$. Every original transition will be replaced by a sequence of k steps that test if the current counter value exceeds the guard and adjust the counter accordingly. The new net $\mathcal{M}'$ has all states of $\mathcal{G}'$ plus a chain of k new states for each original transition.

$$M' = Q' \cup \{t_i \mid 0 \leq i < k, t \in \delta'\}. \tag{5.9}$$

For every transition $t = (p,a,g,d,q)$ in $\mathcal{G}'$, we add the following transitions to $\mathcal{M}'$. First, to test the guard:

$$p \xrightarrow{a,0} t_{k-1}, \tag{5.10}$$

$$t_i \xrightarrow{b,-1} t_{i-1}, \text{for } k-g < i < k \tag{5.11}$$

$$t_i \xrightarrow{b,+1} t_{i-1}, \text{for } k-2g < i < k-g. \tag{5.12}$$

Now we add transitions to adjust the counter according to $d \in \mathbb{N} \cup \{\omega\}$. In case $0 \leq d < \omega$ we add

$$t_i \xrightarrow{b,+1} t_{i-1}, \text{for } k-2g-|d| < i < k-2g \tag{5.13}$$

$$t_i \xrightarrow{b,0} t_{i-1}, \text{for } 0 \leq i < k-2g-d. \tag{5.14}$$

In case $d < 0$ we add

$$t_i \xrightarrow{b,-1} t_{i-1}, \text{for } k-2g-|d| < i < k-2g \tag{5.15}$$

$$t_i \xrightarrow{b,0} t_{i-1}, \text{for } 0 \leq i < k-2g+d. \tag{5.16}$$

In case $d = \omega$ we add

$$t_i \xrightarrow{b,\omega} t_{i-1}, \text{ for } i = k-2g \tag{5.17}$$

$$t_i \xrightarrow{b,0} t_{i-1}, \text{ for } 0 \leq i < k-2g. \tag{5.18}$$

Finally, we allow a move to the new state:

$$t_0 \xrightarrow{b,0} q. \tag{5.19}$$

Observe that every transition in the constructed net $\mathcal{M}'$ has effect in $\{-1,0,+1,\omega\}$. $\mathcal{M}'$ is therefore an ordinary ω-net. It is straightforward to see that

$$pm \xrightarrow{a}_{\mathcal{G}'} qn \iff pm \xrightarrow{ab^{k-1}}_{\mathcal{M}'} qn. \tag{5.20}$$

The claim (5.3) now follows from Equations (5.8) and (5.20). □

Theorem 5.3 now follows from Lemmas 5.7 and 5.8. In particular for the second claim of Theorem 5.3, observe that by the construction above, one round of a weak simulation game w.r.t. $\mathcal{N},\mathcal{G}'$ is simulated by k rounds of a simulation game w.r.t. $\mathcal{M},\mathcal{M}'$. Therefore, if Spoiler has a strategy to win the simulation game relative to $\mathcal{M},\mathcal{M}'$ in α rounds then she can derive strategies to win the games relative to $\mathcal{N},\mathcal{G}'$ and to $\mathcal{N},\mathcal{N}'$ in no more than α rounds. Hence, if $qn \not\preceq_\alpha q'n'$ w.r.t. $\mathcal{M},\mathcal{M}'$ then $qn \not\approx_\alpha q'n'$ w.r.t. $\mathcal{N},\mathcal{N}'$.

5.2 Approximants

The basic idea of our procedure for checking simulation between a OCN and an ω-net, and therefore weak simulation between two OCN, is to stepwise compute semilinear overapproximations $\preceq_i \supseteq \preceq$. For such a procedure to be effective, it is crucial that these approximants converge to $\preceq$ at some finite level, i.e., $\preceq_k = \preceq_{k+1} = \preceq$ for some $k < \omega$. Unfortunately, the usual simulation approximants do not have this property, as Example 5.1 (page 51) shows.

We overcome this difficulty by generalizing the notion of $\preceq_\alpha$ simulation approximants in the case of simulation between one-counter and ω-net processes. This yields approximants that indeed converge at a finite level for any pair of nets.

First we define approximants $\preceq_\alpha^\beta$ in two (ordinal) dimensions. From the game perspective the subscript α indicates the number of rounds Duplicator can survive and the superscript β denotes the number of ω-steps Spoiler needs to allow. For example, $qn \preceq_5^2 q'n'$ if Duplicator can guarantee that no play of the simulation game from position $(qn,q'n')$ that contains < 2 ω-steps is losing for him in fewer than 6 rounds. If not stated otherwise we assume that $\mathcal{N} = (Q,\text{Act},\delta)$ is a one-counter net and $\mathcal{N}' = (Q',\text{Act},\delta')$ is an ω-net.

Definition 5.9. For ordinals α and β, the *approximant* $\preceq_\alpha^\beta$ is inductively defined as follows. Let $\preceq_\alpha^0 = \preceq_0^\beta = Q\times\mathbb{N}\times Q'\times\mathbb{N}$, the full relation. For successor ordinals $\alpha+1,\beta+1$ let $pm \preceq_{\alpha+1}^{\beta+1} p'm'$ iff for all $pm \xrightarrow{a} qn$ there is a step $p'm' \xrightarrow{a} q'n'$ such that either

1. $(p',a,\omega,q') \in \delta'$, $m' < n'$ and $qn \preceq_\alpha^\beta q'n'$, or
2. $(p',a,d,q') \in \delta'$, $n' = m' + d \in \mathbb{N}$ and $qn \preceq_\alpha^{\beta+1} q'n'$.

For limit ordinals λ we define $\preceq_\alpha^\lambda = \bigcap_{\beta<\lambda} \preceq_\alpha^\beta$ and $\preceq_\lambda^\beta = \bigcap_{\alpha<\lambda} \preceq_\alpha^\beta$. Finally,

$$\preceq^\beta = \bigcap_{\alpha\in Ord} \preceq_\alpha^\beta \qquad\qquad \preceq_\alpha = \bigcap_{\beta\in Ord} \preceq_\alpha^\beta . \tag{5.21}$$

Notice that the approximant $\preceq_{\alpha+1}^{\beta+1}$ above is defined in terms of both $\preceq_\alpha^{\beta+1}$ and $\preceq_\alpha^\beta$. The first condition in its definition asks that if a response is due to an ω-transition then the resulting pair of processes need to be related by the approximant with reduced superscript β. The second condition is for the case where a response is due to an ordinary transition.

The approximants $\preceq_\alpha$ correspond to the usual notion of simulation approximants and $\preceq^\beta$ is a special notion derived from the syntactic peculiarity of ω-transitions present in the game on one-counter vs. ω-nets.

Example 5.10. Consider the net that consists of a single a-labelled loop in state A and the ω-net with transitions $B \xrightarrow{a,\omega} C \xrightarrow{a,-1} C$ only. This is a variant of the system of Example 5.1 on page 51, but now we are interested in *strong* simulation. We see that for any $m,n \in \mathbb{N}$, $Am \preceq_n Cn_{n+1} \not\preceq Am$. Moreover, $Am \preceq_\omega Bn$ but $Am \not\preceq_{\omega+1} Bn$ and $Am \preceq^1 Bn$ but $Am \not\preceq^2_{\omega+1} Bn$ and therefore $Am \not\preceq^2 Bn$.

We will further use a game characterization of these approximants, similar to the simulation games that characterize strong simulation.

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 ith time. It is easy to see that this game favours Duplicator due to the additional winning condition. Hence, $\forall i \in \mathbb{N}$, $\preceq^i \supseteq \preceq^{i+1}$. With growing index i, this advantage becomes less important and the game increasingly resembles a standard simulation game.

Definition 5.11. An *approximant game* is played in rounds between Spoiler and Duplicator. Game positions are quadruples $(pm,p'm',\alpha,\beta)$ where $pm,p'm'$ are configurations of $\mathcal{N}$ and $\mathcal{N}'$ respectively, and α,β are ordinals called step- and ω-counter. In each round that starts in $(pm,p'm',\alpha,\beta)$:

- Spoiler chooses two ordinals $\hat\alpha < \alpha$ and $\hat\beta < \beta$,
- Spoiler makes a step $pm \xrightarrow{a} qn$,

- Duplicator responds by making a step $p'm' \xrightarrow{a} q'n'$ using some transition t.

If t was an ω-transition the game continues from position $(qn,q'n',\hat{\alpha},\hat{\beta})$, Otherwise the next round starts at $(qn,q'n',\hat{\alpha},\beta)$ (in this case Spoiler's choice of $\hat{\beta}$ becomes irrelevant). If a player cannot move then the other player wins and if α or β becomes 0, Duplicator wins.

Lemma 5.12. *If Duplicator wins the approximant game from $(pm,p'm',\alpha,\beta)$ then he also wins the game from $(pm,p'm',\hat{\alpha},\hat{\beta})$ for any $\hat{\alpha} \leq \alpha$ and $\hat{\beta} \leq \beta$.*

Proof. If Duplicator has a winning strategy in the game from $(pm,p'm',\alpha,\beta)$ then he can use the same strategy in the game from $(pm,p'm',\hat{\alpha},\hat{\beta})$ and maintain the invariant that the pair of ordinals in the game configuration is pointwise smaller than the pair in the original game. Thus Duplicator wins from $(pm,p'm',\hat{\alpha},\hat{\beta})$. □

Lemma 5.13 (Game Characterization)**.** *Duplicator has a strategy to win the approximant game that starts in $(pm,p'm',\alpha,\beta)$ iff $pm \preceq^{\beta}_{\alpha} p'm'$.*

Proof. We say a pair $(\alpha,\beta) \in Ord^2$ of ordinals *dominates* another such pair (α',β') iff $\alpha' \leq \alpha$, $\beta' \leq \beta$ and $(\alpha',\beta') \neq (\alpha,\beta)$. Both directions of the claim are now shown by well-founded induction on pairs of ordinals: If the claim holds for all pairs (α',β') that are dominated by (α,β) then it also holds for (α,β).

For the "if" direction we assume $pm \preceq^{\beta}_{\alpha} p'm'$ and show that Duplicator wins the game from $(pm,p'm',\alpha,\beta)$. In the base case of $\alpha = 0$ or $\beta = 0$ Duplicator directly wins by definition. By induction hypothesis we assume that the claim is true for all pairs dominated by (α,β). Spoiler starts a round by picking ordinals $\hat{\alpha} < \alpha$ and $\hat{\beta} < \beta$ and moves $pm \xrightarrow{a} qn$. We distinguish two cases, depending on whether β is a limit or successor ordinal.

Case 1. β is a successor ordinal. By Lemma 5.12, we can safely assume that $\hat{\beta} = \beta - 1$. By our assumption $pm \preceq^{\beta}_{\alpha} p'm'$ and Definition 5.9, there must be a response $p'm' \xrightarrow{a} q'n'$ that is either due to an ω-transition and then $qn \preceq^{\hat{\beta}}_{\hat{\alpha}} q'n'$ or due to an ordinary transition, in which case we have $qn \preceq^{\beta}_{\hat{\alpha}} q'n'$. In both cases, we know by the induction hypothesis that Duplicator wins from this next position and thus also from the initial position.

Case 2. β is a limit ordinal. By $pm \preceq^{\beta}_{\alpha} p'm'$ and Definition 5.9, we obtain $pm \preceq^{\gamma}_{\alpha} p'm'$ for all $\gamma < \beta$. If α is a successor ordinal then, by Lemma 5.12, we can safely assume that $\hat{\alpha} = \alpha - 1$. Otherwise, if α is a limit ordinal, then, by Definition 5.9, we have $pm \preceq^{\gamma}_{\overline{\alpha}} p'm'$ for all $\overline{\alpha} < \alpha$ and in particular $pm \preceq^{\gamma}_{\hat{\alpha}+1} p'm'$. So in either case we obtain

$$pm \preceq^{\gamma}_{\hat{\alpha}+1} p'm' \text{ for all } \gamma < \beta. \tag{5.22}$$

If there is some ω-transition that allows a response $p'm' \xrightarrow{a}_{\omega} q'n'$ that satisfies $qn \preceq^{\hat{\beta}}_{\hat{\alpha}} q'n'$, then Duplicator picks this response and we can use the induction hypothesis to conclude that he wins the game from the next position. Otherwise, if no such ω-transition exists, Equation (5.22)

implies that for every $\gamma < \beta$ there is a response to some $q'n'$ that uses a non-ω-transition $t(\gamma)$ and that satisfies $qn \preceq^{\gamma}_{\hat{\alpha}} q'n'$. Since β is a limit ordinal, there exist infinitely many $\gamma < \beta$. By the pigeonhole principle, there must be one transition that occurs as $t(\gamma)$ for infinitely many γ, because there are only finitely many transitions in the net. Therefore, a response that uses this particular transition satisfies $qn \preceq^{\beta}_{\hat{\alpha}} q'n'$. If Duplicator uses this response, the game continues from position $(qn,q'n',\hat{\alpha},\beta)$ and he wins by induction hypothesis.

For the "only if" direction we show that $pm \not\preceq^{\beta}_{\alpha} p'm'$ implies that Spoiler has a winning strategy in the approximant game from $(pm,p'm',\alpha,\beta)$. In the base case of $\alpha = 0$ or $\beta = 0$ the implication holds trivially since the premise is false. By induction hypothesis, we assume that the implication is true for all pairs dominated by (α,β). Observe that if α or β are limit ordinals then (by Definition 5.9) there are successors $\hat{\beta} \leq \beta$ and $\hat{\alpha} \leq \alpha$ s.t. $pm \not\preceq^{\hat{\beta}}_{\hat{\alpha}} p'm'$. So without loss of generality we can assume that α and β are successors. By the definition of approximants there must be a move $pm \xrightarrow{a} qn$ such that

- for every response $p'm' \xrightarrow{a}_{\omega} q'n'$ that uses some ω-transition we have $qn \not\preceq^{\beta-1}_{\alpha-1} q'n'$,
- for every response $p'm' \xrightarrow{a} q'n'$ via some normal transition it holds that $qn \not\preceq^{\beta}_{\alpha-1} q'n'$.

So if Spoiler chooses $\hat{\alpha} = \alpha - 1$, $\hat{\beta} = \beta - 1$ and moves $pm \xrightarrow{a} qn$ then any possible response by Duplicator will take the game to a position $(qn,q'n',\gamma,\hat{\alpha})$ for some $\gamma \leq \beta$. By induction hypothesis Spoiler wins the game. □

Lemma 5.14. *For all ordinals α,β the following properties hold.*

1. *$pm \preceq^{\beta}_{\alpha} p'm'$ implies $pn \preceq^{\beta}_{\alpha} p'n'$ for all $n \leq m$ and $n' \geq m'$*
2. *If $\hat{\alpha} \geq \alpha$ and $\hat{\beta} \geq \beta$ then $\preceq^{\hat{\beta}}_{\hat{\alpha}} \subseteq \preceq^{\beta}_{\alpha}$.*
3. *There are ordinals CA,CB such that $\preceq_{CA} = \preceq_{CA+1}$ and $\preceq^{CB} = \preceq^{CB+1}$.*
4. *$\preceq = \bigcap_{\alpha} \preceq_{\alpha} = \bigcap_{\beta} \preceq^{\beta}$*

The first point states that individual approximants are monotonic with respect to the counter values. Points 2-4 imply that both $\preceq_{\alpha}$ and $\preceq^{\beta}$ yield non-increasing sequences of approximants that converge towards simulation. As Example 5.10 on page 58 shows, the approximants $\preceq_{\alpha}$ do not converge at finite levels, and not even at level ω, i.e., $CA > \omega$ in general. We will later show (in Lemma 5.16) that the approximants $\preceq^{\beta}$ converge at a finite level, i.e., CB is *strictly* below ω for any pair of nets, and further bound CB in Section 5.4 to obtain an exact complexity upper bound.

Proof. 1 . By Lemma 5.13, it suffices to observe that Duplicator can reuse a winning strategy in the approximant game from $(pm,p'm',\alpha,\beta)$ to win the game from $(pn,p'n',\alpha,\beta)$ for naturals $n \leq m$ and $n' \geq m'$.

2 . If $pm \preceq^{\hat{\beta}}_{\hat{\alpha}} p'm'$ then, by Lemma 5.13, Duplicator wins the approximant game from position $(pm,p'm',\hat{\beta},\hat{\alpha})$. By Lemma 5.12 he can also win the approximant game from $(pm,p'm',\beta,\alpha)$. Thus $pm \preceq^{\beta}_{\alpha} p'm'$ by Lemma 5.13.

3 . By point 2) we see that with increasing ordinal index α the approximant relations $\preceq_\alpha$ form a decreasing sequence of relations, thus they stabilize for some ordinal CA. The existence of a convergence ordinal for $\preceq^{CB}$ follows analogously.

4 . First we observe that $\bigcap_\alpha \preceq_\alpha = \bigcap_\alpha \bigcap_\beta \preceq^{\beta}_{\alpha} = \bigcap_\beta \bigcap_\alpha \preceq^{\beta}_{\alpha} = \bigcap_\beta \preceq^{\beta}$. It remains to show that $\preceq = \bigcap_\alpha \preceq_\alpha$. In order to show $\preceq \supseteq \bigcap_\alpha \preceq_\alpha$, we use CA from point 3) and rewrite the right side of the inclusion to $\bigcap_\alpha \preceq_\alpha = \preceq_{CA} = \preceq_{CA+1}$. From Definition 5.9 we get that $\preceq_\alpha = \preceq^{\gamma}_{\alpha}$ for $\gamma \geq \alpha$ and therefore $\preceq^{CA+1}_{CA+1} = \preceq_{CA+1} = \preceq_{CA} = \preceq^{CA}_{CA}$. This means $\preceq^{CA}_{CA} = \bigcap_\alpha \preceq_\alpha$ must be a simulation relation and hence a subset of $\preceq$.

To show $\preceq \subseteq \bigcap_\alpha \preceq_\alpha$, we prove by induction that $\preceq \subseteq \preceq_\alpha$ holds for all ordinals α. The base case $\alpha = 0$ is trivial. For the induction step we prove the equivalent property $\not\preceq_\alpha \subseteq \not\preceq$. There are two cases.

In the first case, $\alpha = \gamma + 1$ is a successor ordinal. If $pm \not\preceq_{\gamma+1} p'm'$ then $pm \not\preceq^{\gamma+1}_{\gamma+1} p'm'$ and therefore, by Lemma 5.13, Spoiler wins the approximant game from $(pm,p'm',\gamma+1,\gamma+1)$. Let $pm \xrightarrow{a} qn$ be an optimal initial move by Spoiler. Now either there is no valid response and thus Spoiler immediately wins in the simulation game; or for every Duplicator response $p'm' \xrightarrow{a} q'n'$ that uses an ω-step, we have $qn \not\preceq^{\gamma}_{\gamma} q'n'$ and for every response that does not use an ω move, we have $qn \not\preceq^{\gamma+1}_{\gamma} q'n'$. Either way, we get $qn \not\preceq_\gamma q'n'$ and by induction hypothesis, $qn \not\preceq q'n'$. By Lemma 5.13, we obtain that Spoiler wins the simulation game from $(qn,q'n')$ and thus from $(pm,p'm')$. Therefore $pm \not\preceq p'm'$, as required.

In the second case, α is a limit ordinal. Then $pm \not\preceq_\alpha p'm'$ implies $pm \not\preceq_\gamma p'm'$ for some $\gamma < \alpha$ and therefore $pm \not\preceq p'm'$ by induction hypothesis. □

The following lemma shows a certain uniformity property of the simulation game. Beyond some fixed bound, an increased counter value of Spoiler can be neutralized by an increased counter value of Duplicator, thus enabling Duplicator to survive at least as many rounds in the game as before. This lemma is necessary for the proof of Lemma 5.16, which guarantees the existence of a finite bound for the convergence level CB.

Lemma 5.15. *For any one-counter net $\mathcal{N} = (Q,Act,\delta)$ and ω-net $\mathcal{N}' = (Q'Act,\delta')$ there is a fixed bound $c \in \mathbb{N}$ s.t. for all states $(p,p') \in Q \times Q'$, naturals $n > m > c$ and ordinals α:*

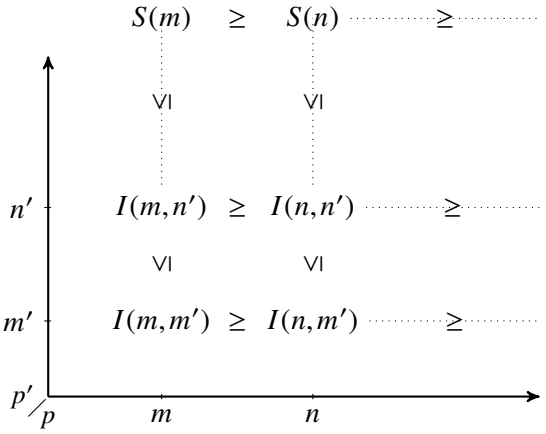
$$\forall m'.\ (pm \preceq_\alpha p'm' \implies \exists n'.\ pn \preceq_\alpha p'n') \tag{5.23}$$

Proof. It suffices to show the existence of a local bound c that satisfies (5.23) for any given pair of states, since we can simply take the global c to be the maximal such bound over all finitely many pairs. Let CA be the convergence ordinal provided by Lemma 5.14, point 3 and consider

a fixed pair $(p,p') \in (Q \times Q')$ of states. For $m,m' \in \mathbb{N}$, we define the following (sequences of) ordinals.

$$\begin{aligned} I(m,m') &= \text{the largest ordinal } \alpha \text{ with } pm \preceq_\alpha p'm' \text{ or } \mathcal{CA} \\ &\quad \text{if no such } \alpha \text{ exists,} \\ I(m) &= \text{the increasing sequence of ordinals } I(m,m')_{m' \geq 0}, \\ S(m) &= \sup\{I(m)\}. \end{aligned}$$

Observe that $I(m,m')$ can be presented as an infinite matrix where $I(m)$ is a column and $S(m)$ is the limit of the sequence of elements of column $I(m)$ looking upwards. Informally, $S(m) = lim_{i \to \infty} I(m,i)$.



By Lemma 5.14 (point 1), we derive that for any $m \leq n \in \mathbb{N}$ and $m' \leq n' \in \mathbb{N}$

$$I(m,n') \geq I(m,m') \geq I(n,m') \tag{5.24}$$

and because of the second inequality, also that $S(m) \geq S(n)$. So the ordinal sequence $S(m)_{m \geq 0}$ of suprema must be non-increasing and by the well-ordering of the ordinals there is a smallest index $k \in \mathbb{N}$ at which this sequence stabilizes:

$$\forall l > k.\ S(l) = S(k). \tag{5.25}$$

We split the remainder of this proof into three cases depending on whether $I(k)$ and $I(l)$ for some $l > k$ have maximal elements. In each case we show the existence of a bound c that satisfies requirement (5.23).

Case 1. For all $l \geq k$ and $m' \in \mathbb{N}$ it holds that $I(l,m') < S(l)$, i.e., no $I(l)$ has a maximal element. In this case $c := k$ satisfies the requirement (5.23). To see this, take $n > m > c = k$ and $pm \preceq_\alpha p'm'$. Then, by our assumption, $\alpha < S(m)$ and $S(m) = S(n) = S(k)$. Therefore $\alpha < S(n)$, which means that there must exist an $n' \in \mathbb{N}$ s.t. $pn \preceq_\alpha p'n'$, as required.

Case 2. For all $l \geq k$ there is a $n'_l \in \mathbb{N}$ such that $I(l,n'_l) = S(l)$, i.e., all $I(l)$ have maximal element $S(l) = S(k)$. Again $c := k$ satisfies the requirement (5.23). Given $n > m > c = k$ and

$pm \preceq_\alpha p'm'$ we let $n' := n'_n$ and obtain $I(n,n') = S(n) = S(k) \geq \alpha$ and thus $pn \preceq_\alpha qn'$, as required.

Case 3. If none of the two cases above holds then there must exist some $l > k$ s.t. the sequences $I(k),\ldots,I(l-1)$ each have a maximal element and for $i > l$ the sequence $I(i)$ has no maximal element. To see this, consider sequences $I(m)$ and $I(n)$ with $n > m \geq k$. If $I(n)$ has a maximal element then so must $I(m)$, by Equation (5.24) and $S(m) = S(n) = S(k)$. Given this, we repeat the argument for the first case, with $c := l$ and again satisfy the requirement (5.23). □

Lemma 5.16. *Consider strong simulation $\preceq$ between a OCN $\mathcal{N} = (Q,Act,\delta)$ and an ω-net $\mathcal{N}' = (Q',Act,\delta')$. There exists a constant $CB \in \mathbb{N}$ such that $\preceq = \preceq^{CB}$.*

Proof. We assume the contrary and derive a contradiction. By Lemma 5.14, part 4, the inclusion $\preceq \subseteq \preceq^\beta$ always holds for every ordinal β. Thus, if there is no $CB \in \mathbb{N}$ with $\preceq = \preceq^{CB}$, then for every finite $\beta \in \mathbb{N}$ there are processes p_0m_0 and $p'_0m'_0$ such that $p_0m_0 \preceq^\beta p'_0m'_0$ but $p_0m_0 \not\preceq p'_0m'_0$. In particular, this holds for the special case of $\beta = |Q \times Q'|(c+1)$, where c is the constant given by Lemma 5.15, which we consider in the rest of this proof.

Since $p'_0m'_0$ does not simulate p_0m_0, we can assume a winning strategy for Spoiler in the simulation game which is optimal in the sense that it guarantees that the simulation level α_i – the largest ordinal with $p_im_i \preceq_{\alpha_i} p'_im'_i$ – strictly decreases along rounds of any play. By monotonicity (Lemma 5.14, part 1), we can thus infer that whenever a pair of control states repeats along a play, then Duplicator's counter must have decreased or Spoiler's counter must have increased: Along any partial play $(p_0m_0,p'_0m'_0)(t_0,t'_0)(p_1m_1,p'_1m'_1)(t_t,t'_1)\ldots(p_km_k,p'_km'_k)$ with $(p_i,p'_i) = (p_k,p'_k)$ for some $i < k$, we have $m_i < m_k$ or $m'_i > m'_k$. By a similar argument we can assume that Duplicator also plays optimally, in the sense that he uses ω-transitions to increase his counter to higher values than in previous situations with the same pair of control states. By combining this with the previously stated property that the sequence of α_i strictly decreases we obtain the following:

$$\text{If } (p_i,p'_i) = (p_k,p'_k) \text{ and } t'_{i-1},t'_{k-1} \in \delta'_\omega \text{ then } m_i < m_k. \tag{5.26}$$

Here δ'_ω denotes the set of transitions with symbolic effect ω in Duplicator's net $\mathcal{N}'$.

Although Duplicator loses the simulation game between p_0m_0 and $p'_0m'_0$, our assumption $p_0m_0 \preceq^\beta p'_0m'_0$ with $\beta = |Q \times Q'|(c+1)$ means that Duplicator can ensure that no play with fewer than β ω-transitions is losing for him, regardless of Spoiler's strategy. So we can safely assume that there is a play in Spoiler's supposed optimal winning strategy along which Duplicator makes use of ω-transitions β times. Let $\pi = (p_0m_0,p'_0m'_0)(t_0,t'_0)(p_1m_1,p'_1m'_1)(t_t,t'_1)$ $\ldots(p_km_k,p'_km'_k)$ be such a play.

Our choice of $\beta = |Q \times Q'|(c+1)$ guarantees that some pair (p,p') of control states repeats at least $c+1$ times directly after Duplicator making an ω-step. Thus there are indices $i(1) <$

$i(2) < \cdots < i(c+1) < k$ s.t. for all $1 \leq j \leq c+1$ we have $(p_{i(j)}, p'_{i(j)}) = (p, p')$ and $t'_{i(j)} \in \delta_\omega$. By observation (5.26) and $m_0 \geq 0$ we obtain that $m_{i(x)} \geq x$ for all x with $0 \leq x \leq c+1$. In particular, $c \leq m_{i(c)} < m_{i(c+1)}$, that is, both of Spoiler's counter values after the last two such repetitions must lie above c. This allows us to apply Lemma 5.15 to derive a contradiction.

Let α be the simulation level before this repetition: α is the largest ordinal that satisfies $pm_{i(c)} \preceq_\alpha p'm'_{i(c)}$. Since $m_{i(c+1)} > m_{i(c)} > c$, Lemma 5.15 ensures the existence of a natural n' s.t. $pm_{i(c+1)} \preceq_\alpha p'n'$. Because Duplicator used an ω-transition in his last response leading to the repetition of states there must be a partial play π' in which both players make the same moves as in π except that Duplicator chooses $m'_{i(c+1)}$ to be n'. Now in this play we observe that the simulation level did in fact not strictly decrease as this last repetition of control states shows: We have $pm_{i(c)} \preceq_\alpha p'm'_{i(c)} \not\preceq_{\alpha+1} pm_{i(c)}$ and $pm_{i(c+1)} \preceq_\alpha p'm'_{i(c+1)}$, which contradicts the assumed optimality of Spoiler's strategy. □

To conclude this section on approximants, we show that ordinary weak simulation approximants $\preceq_\alpha$ indeed converge at level ω^2 for any pair of OCNs.

Lemma 5.17. *For relations between a OCN and an ω-net, we have $\preceq_{\omega i} \subseteq \preceq^i$ for every $i \in \mathbb{N}$.*

Proof. By induction on i. The base case of $i = 0$ is trivial, since $\preceq^0$ is the full relation. We prove the inductive step by assuming the contrary and deriving a contradiction. Let $pm \preceq_{\omega i} p'm'$ and $pm \not\preceq^i p'm'$ for some $i > 0$. Then there exists some ordinal α s.t. $pm \not\preceq^i_\alpha p'm'$. Without restriction let α be the least ordinal satisfying this condition. If $\alpha \leq \omega i$ then we trivially have a contradiction. Now we consider the case $\alpha > \omega i$. By $pm \not\preceq^i_\alpha p'm'$ and Lemma 5.13, Spoiler has a winning strategy in the approximant game from position $(pm, p'm', \alpha, i)$. Without restriction we assume that Spoiler plays optimally, i.e., wins as quickly as possible. Thus this game must reach some position $(qn, q'n', \beta+1, i)$ where $\beta \geq \omega i$ is a limit ordinal, such that Spoiler can win from $(qn, q'n', \beta+1, i)$ but not from $(qn, q'n', \beta, i)$. I.e., $qn \not\preceq^i_{\beta+1} q'n'$, but $qn \preceq^i_\beta q'n'$. Consider Spoiler's move $qn \xrightarrow{a} rl$ according to her optimal winning strategy in the game from position $(qn, q'n', \beta+1, i)$. Since $qn \preceq^i_\beta q'n'$ and β is a limit ordinal, for every ordinal $\gamma_k < \beta$, Duplicator must have some countermove $q'n' \xrightarrow{a} r'_k l'_k$ s.t. $rl \preceq^j_{\gamma_k} r'_k l'_k$, where $j = i - 1$ if the move was due to an ω-transition and $j = i$ otherwise. In particular, $\sup_k\{\gamma_k\} = \beta$. However, since Spoiler's move $qn \xrightarrow{a} rl$ was according to an optimal winning strategy from position $(qn, q'n', \beta+1, i)$, we have that $rl \not\preceq^j_\beta r'_k l'_k$. Therefore, there must be infinitely many different responses $q'n' \xrightarrow{a} r'_k l'_k$. Infinitely many of these countermoves must be due to an ω-transition, because apart from these the system is finitely branching. Thus for every ordinal $\gamma < \beta$ there is some Duplicator countermove $q'n' \xrightarrow{a} r'_k l'_k$ which is due to an ω-transition s.t. $rl \preceq^{i-1}_{\gamma_k} r'_k l'_k$ where $\gamma_k \geq \gamma$ (note the $i-1$ index due to the ω-transition). In particular, we can choose $\gamma = \omega(i-1)$, because $i > 0$ and $\beta \geq \omega i$. Then we have $rl \preceq^{i-1}_{\omega(i-1)} r'_k l'_k$, but $rl \not\preceq^{i-1}_\beta r'_k l'_k$. However, from $rl \preceq^{i-1}_{\omega(i-1)} r'_k l'_k$ and the induction hypothesis, we obtain $rl \preceq^{i-1} r'_k l'_k$ and in

particular $rl \preceq_\beta^{i-1} r'_k l'_k$. Contradiction. □

Theorem 5.18. *Weak simulation approximants on OCN converge at level ω^2, but not earlier in general.*

Proof. First we show that $\leqq_{\omega^2}$ is contained in $\leqq$ for OCN. Let pm and $p'm'$ be processes of OCN $\mathcal{N}$ and $\mathcal{N}'$, respectively, and let $\mathcal{M}$ and $\mathcal{M}'$ be the derived OCN and ω-net from Theorem 5.3 (page 53). Assume $pm \leqq_{\omega^2} p'm'$ w.r.t. $\mathcal{N},\mathcal{N}'$. By point 2) of Theorem 5.3 we conclude that $pm \preceq_{\omega^2} p'm'$ w.r.t. $\mathcal{M},\mathcal{M}'$. In particular we have $pm \preceq_{\omega \cdot CB} p'm'$ w.r.t. $\mathcal{M},\mathcal{M}'$, for the level $CB \in \mathbb{N}$ from Lemma 5.16. From Lemma 5.17 we obtain $pm \preceq^{CB} p'm'$ w.r.t. $\mathcal{M},\mathcal{M}'$. Lemma 5.16 then yields $pm \preceq p'm'$ w.r.t. $\mathcal{M},\mathcal{M}'$. Finally, by Theorem 5.3, we obtain that $pm \leqq p'm'$ w.r.t. $\mathcal{N},\mathcal{N}'$.

To see that ω^2 is needed in general, consider the following class of examples, that are the result of extending the net from Example 5.1 on page 51. Let $\mathcal{N}$ be the simple OCN that consists only of the self-loop $A \xrightarrow{a,0} A$. For every $k \in \mathbb{N}$ and all $1 \leq i \leq k$, the OCN $\mathcal{N}'_k$ has transitions $(C_i,a,-1,C_i)$, $(B_i,\tau,0,C_i)$ $(B_i,\tau,1,B_i)$, and $(C_{i+1},\tau,0,B_i)$. We see that $A \leqq_{\omega \cdot k} B_k 0$, but $A \not\leqq B_k 0$ w.r.t. $\mathcal{N},\mathcal{N}'_k$. So, for every $k \in \mathbb{N}$ there are OCNs for which $\leqq_{\omega \cdot k} \neq \leqq$. □

5.3 Effective Semilinearity

In order to show the decidability of simulation between one-counter nets and ω-Nets we prove a stronger claim, namely that the largest simulation relation is a semilinear set and one can effectively compute its description. To prove this claim for a fixed pair of nets, we consider approximants $\preceq^k$ and show (by repeated reduction to strong simulation over OCN and using Theorem 4.19) that in fact $\preceq^k$ is effectively semilinear for every level $k \in \mathbb{N}$. To be precise, we show the following lemma.

Lemma 5.19. *For any one-counter net $\mathcal{M}$ and ω-net $\mathcal{M}'$ with state sets Q and Q' respectively, there is an effectively computable sequence $(\mathcal{S}_k,\mathcal{S}'_k)_{k\in\mathbb{N}}$ of pairs of OCN with state sets $S_k \supseteq Q$ and $S_k \supseteq Q'$ respectively, such that for all $k,m,m' \in \mathbb{N}$ and states $p \in Q, p' \in Q'$,*

$$pm \preceq^k p'm' \text{ w.r.t. } \mathcal{M},\mathcal{M}' \iff pm \preceq p'm' \text{ w.r.t. } \mathcal{S}_k,\mathcal{S}'_k. \tag{5.27}$$

A direct consequence of this is the effective semilinearity, and thus decidability, of weak simulation $\leqq$ over any fixed pair of one-counter nets.

Theorem 5.20. *Let $\mathcal{N},\mathcal{N}'$ be two one-counter nets. The largest weak simulation relation $\leqq$ with respect to $\mathcal{N},\mathcal{N}'$ is a semilinear set and its representation is effectively computable.*

Proof. By Theorem 5.3, it suffices to show the claim for the largest strong simulation $\preceq$ between a OCN $\mathcal{M}$ and an ω-Net $\mathcal{M}'$. By Lemma 5.19, one can iteratively compute the sequence

$(\mathcal{S}_k,\mathcal{S}'_k)_{k\in\mathbb{N}}$ of nets characterizing $\preceq^k$ for growing k. Because $\mathcal{S}_k$ and $\mathcal{S}'_k$ are one-counter nets, we can apply Theorem 4.19 and derive that strong simulation w.r.t. $\mathcal{S}_k,\mathcal{S}'_k$, and hence the approximant $\preceq^k$ w.r.t. $\mathcal{M},\mathcal{M}'$ are effectively semilinear sets. Recall that for $k\in\mathbb{N}$, $\preceq^{k+1}\subseteq\preceq^k$. Because equality of semilinear sets is decidable, we can check after each iteration if $\preceq^{k+1}\supseteq\preceq^k$ holds, in which case we stop with the description of $\preceq^k=\preceq$. Termination of this procedure is guaranteed by Lemma 5.16. □

Before we prove Lemma 5.19, we introduce two important ingredients for the construction of the nets $\mathcal{S}_k,\mathcal{S}'_k$. The first is a class of simple gadgets called *test chains* that will form part of these nets and allow us to check, by means of a continued simulation game, if the counter value of Spoiler is $\geq i$ for some hard-wired constant $i\in\mathbb{N}$. A *test chain* for $i\in\mathbb{N}$, is a pair $\mathcal{T}_i,\mathcal{T}_i'$ of OCNs with initial states t_i and t'_i over actions $\mathrm{Act}=\{e,f\}$. We let t_i be the starting point of a counter-decreasing chain of e-steps of length i where the last state of the chain can make an f-step, whereas t'_i is a simple e-loop (see Figure 5.1). Then we observe that for all $m,n\in\mathbb{N}$,

$$t_i m \not\preceq t'_i n \iff m\geq i. \tag{5.28}$$

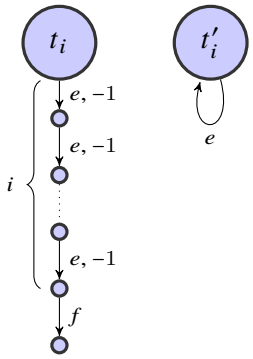


Figure 5.1: A test chain for value $i\in\mathbb{N}$.

The second ingredient for our construction is the notion of *minimal sufficient values*. Consider the approximant $\preceq^k$ for some parameter k, and let $(q,q')\in(Q\times Q')$ be a pair of states. By monotonicity (Lemma 5.14, point 1), there is a minimal value $suf(q,q',k)\in\mathbb{N}\cup\{\omega\}$ satisfying

$$\forall n'\in\mathbb{N}.\ q(suf(q,q',k))\not\preceq^k q'n'. \tag{5.29}$$

Let $suf(q,q',k)$ be ω if no finite value satisfies this condition. The following properties are immediate from the definitions.

Lemma 5.21. *For all $q\in Q,q'\in Q'$ and $k\in\mathbb{N}$,*

1. $suf(q,q',0)=\omega$, *and*
2. $suf(q,q',k)\geq suf(q,q',k+1)$.

We are now ready to present the construction of successive pairs of nets $\mathcal{S}_k, \mathcal{S}'_k$, that satisfy the claim of Lemma 5.19. The idea behind the construction of nets for parameter $k+1$ is as follows. Assuming we have already constructed a semilinear representation of $\preceq^k$ in the form of two OCN $\mathcal{S}_k$ and $\mathcal{S}'_k$, we can compute the values $suf(q,q',k)$ for every pair (q,q').

The nets $\mathcal{S}_{k+1}$ and $\mathcal{S}'_{k+1}$ are constructed so that a simulation game played on the arena $\mathcal{S}_{k+1}, \mathcal{S}'_{k+1}$ mimics the approximant game game played on $\mathcal{M}, \mathcal{M}'$ with ω-parameter $(k+1)$ until Duplicator uses an ω-labelled transition, leading to some game position qn vs. $q'n'$. Afterwards, the approximant game would continue with the next lower parameter k. In the simulation game on $\mathcal{S}_{k+1}$ and $\mathcal{S}'_{k+1}$, Duplicator cannot make the ω-step but can instead enforce the play to continue in some subgame (a test chain) that he wins iff Spoiler's counter is smaller than the hard-wired value $suf(q,q',k)$. This "forcing" of the play can be implemented for OCN simulation using a standard technique called *defender's forcing* (see e.g. [34]), that essentially allows Duplicator to reach a universal process (and thus win) in the next round unless his opponent moves in some specific way.

The nets $\mathcal{S}_{k+1}$ and $\mathcal{S}'_{k+1}$ thus 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 gadget. The only difference between two such pairs of nets for different parameters k is the lengths of the test chains.

Definition 5.22 (The construction of $\mathcal{S}_k$ and $\mathcal{S}'_k$)**.** Fix a OCN $\mathcal{M} = (Q, \mathrm{Act}, \delta)$, an ω-net $\mathcal{M}' = (Q', \mathrm{Act}, \delta)$ and a constant $k \geq 1$. We construct the one-counter nets $\mathcal{S}_k$ and $\mathcal{S}'_k$ that characterize the approximant $\preceq^k$.

For any pair $(p,p') \in Q \times Q'$ of states, let $\mathcal{T}_{p,p'}$ and $\mathcal{T}'_{p,p'}$ be the nets that describe the test chain for $suf(p,p',k-1)$. Let $\mathcal{T}_{p,p'} = (T_{p,p'}, \{e,f\}, \delta_{p,p'})$ and $\mathcal{T}'_{p,p'} = (T'_{p,p'}, \{e,f\}, \delta'_{p,p'})$ and let $t_{p,p'}$ and $t'_{p,p'}$ be the initial states of $\mathcal{T}_{p,p'}$ and $\mathcal{T}'_{p,p'}$ respectively. W.l.o.g. we can assume that $e, f \notin \mathrm{Act}$ are new letters. We define the one-counter nets $\mathcal{S}_k$ and $\mathcal{S}'_k$ over the new alphabet $\overline{\mathrm{Act}}$ as follows. $\overline{\mathrm{Act}}$ contains all letters of the original alphabet, two (new) actions e, f used in test gadgets and a new action (p,p') for every pair of original states.

$$\overline{\mathrm{Act}} = \mathrm{Act} \cup \{f,e\} \cup (Q \times Q'). \tag{5.30}$$

The net $\mathcal{S}_k = (S_k, \overline{\mathrm{Act}}, \delta_k)$ has all original states of $\mathcal{M}$, plus those of all test chains:

$$S_k = Q \cup \bigcup_{p \in Q, p' \in Q'} T_{p,p'} \tag{5.31}$$

Its transitions $\delta_k \supseteq \delta \cup \bigcup_{q \in Q, q' \in Q'} \delta_{q,q'}$ are those of $\mathcal{M}$, all test chains, and the following for all $q \in Q, q' \in Q'$:

$$q \xrightarrow{(q,q'),0} t_{q,q'} \tag{5.32}$$

The net $\mathcal{S}'_k = (S'_k, \overline{\text{Act}}, \delta'_k)$ has states

$$S'_k = Q' \cup \Big(\bigcup_{q \in Q, q' \in Q'} T'_{q,q'} \Big) \cup \{W\}. \tag{5.33}$$

So it contains all original states of $\mathcal{M}'$, those of all test chains and a new "win" state W. Its set of transitions is $\delta'_k \supseteq \{q \xrightarrow{a,x} q' \in \delta' \mid x \neq \omega\} \cup \bigcup_{q \in Q, q' \in Q'} \delta_{q,q'}$. It contains those transitions in $\mathcal{M}'$ which are not labelled by ω, the transitions of the test chains plus the following, that allow Duplicator to force the game into a test chain:

$$p' \xrightarrow{a,0} t'_{p,q'} \quad \text{for all } p \in Q \text{ and } q', p' \in Q' \text{ if } p' \xrightarrow{a,\omega} q' \in \delta', \tag{5.34}$$

$$p' \xrightarrow{(q,q'),0} W \quad \text{for all } q \in Q \text{ and } q', p' \in Q', \tag{5.35}$$

$$t'_{q,q'} \xrightarrow{(q,q'),0} t'_{q,q'} \quad \text{for all } q \in Q \text{ and } q' \in Q', \tag{5.36}$$

$$t'_{q,q'} \xrightarrow{(q,p'),0} W \quad \text{for all } q \in Q \text{ and } q', p' \in Q' \text{ if } p' \neq q', \tag{5.37}$$

$$t'_{q,q'} \xrightarrow{a,0} W \quad \text{for all } q \in Q \text{ and } q' \in Q' \text{ and } a \in \text{Act}, \tag{5.38}$$

$$W \xrightarrow{a,0} W \quad \text{for all } a \in \text{Act}'. \tag{5.39}$$

Figure 5.2 illustrates the forcing mechanism due to these new transitions.

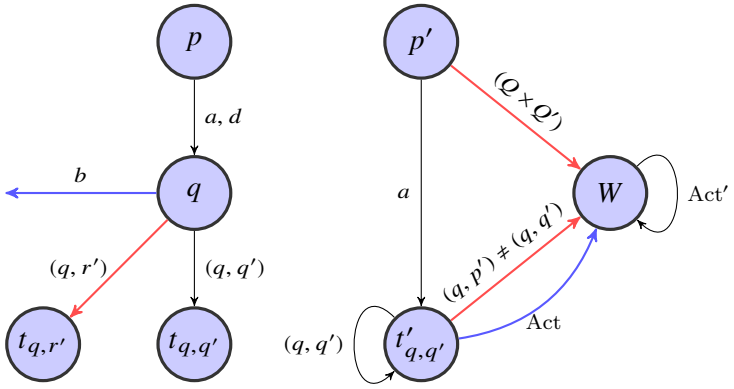


Figure 5.2: The forcing mechanism that replaces a Duplicator transition $p' \xrightarrow{a,\omega} q'$. Counter effects are omitted, individual transitions are grouped and punishing moves are coloured. For instance, the red arrow from p' to W depicts all transitions due to Equation (5.35) that prevent Spoiler from using any actions of the form $(p,p') \in$ Act unless Duplicator already moved to some state $t'_{q,q'}$. Note that Spoiler must prevent Duplicator from reaching the universal state W and that once the players are at states $t_{q,q'}$ and $t'_{q,q'}$, she has no other option but to play the test chain that starts here.

Observe that the definition of the nets $\mathcal{S}_k, \mathcal{S}'_k$ above is relative to the values $suf(p,p',k-1)$ for all original control states p,p'. It is therefore crucial to know these values for this contruction to be effective. The following two lemmas state the correctness of the construction.

Lemma 5.23. *For all control states $(p,p') \in Q \times Q'$ and naturals $k,m,n \in \mathbb{N}$:*

$$pm \not\preceq p'm' \text{ w.r.t. } \mathcal{S}_k,\mathcal{S}'_k \impliedby pm \not\preceq^k p'm' \text{ w.r.t. } \mathcal{M},\mathcal{M}' \tag{5.40}$$

Proof. Note that by definition of approximants, $pm \not\preceq^k p'm'$ implies $pm \not\preceq^k_\alpha p'm'$ for some ordinal α. By the game interpretation (Lemma 5.13) it is thus sufficient to show that for all ordinals α, if Spoiler has a winning strategy in the approximant game from position $(pm,p'm',\alpha,k)$ then she also has a winning strategy in the simulation game between $\mathcal{S}_k,\mathcal{S}'_k$ from position $(pm,p'm')$.

We proceed by ordinal induction on α. The base case trivially holds since Spoiler loses from a position $(pm,p'm',0,k)$ by definition of the approximant game (Definition 5.11).

For the induction step let Spoiler play the same move $pm \xrightarrow{a} qn$ for some $a \in \mathrm{Act}$ in both games according to her assumed winning strategy in the approximant game. Now Duplicator makes his response move in the new game between $\mathcal{S}_k,\mathcal{S}'_k$, which yields two cases. In the first case, Duplicator does not use a transition from Equation (5.34). Then his move induces a corresponding move in the approximant game which leads to a new configuration $(qn,q'n',\gamma,k)$ where $qn \not\preceq^k_\gamma q'n'$ for some ordinal $\gamma < \alpha$. By the induction hypothesis, Spoiler now has a winning strategy to continue the simulation game from position $(qn,q'n')$.

In the second case, Duplicator's response is via a transition from Equation (5.34), which leads to a new configuration $(qn,t'_{r,q'}n')$ for some $r \in Q$. Thus in the approximant game there will exist Duplicator moves to positions $(qn,q'n',\gamma,k-1)$ where $\gamma < \alpha$ and $n' \in \mathbb{N}$ can be arbitrarily high. We can safely assume that Duplicator chooses $r = q$, since otherwise Spoiler can afterwards win in one round by a (q,q') labelled step from qn. Now in the next round Spoiler can play $qn \xrightarrow{(q,q')} t_{q,q'}n$ by Equation (5.32) and Duplicator's only option is to stay in his current state by Equation (5.36). The simulation game thus continues from $(t_{q,q'}n,t'_{q,q'}n')$, which is the beginning of the testing gadget for states q,q'. To show that Spoiler wins the rest of the simulation game, we show that indeed, n must be at least be $suf(k-1,q,q')$. By our initial assumption, Spoiler wins the approximant game from the position $(pm,p'm',\alpha,k)$. Thus there is some ordinal $\gamma < \alpha$ such that Spoiler also wins the approximant game from position $(qn,q'n',\gamma,k-1)$ for every $n' \in \mathbb{N}$. Thus, by Lemma 5.13 and Definition 5.9, we have $qn \not\preceq^{k-1}_\gamma q'n'$ and by Lemma 5.14 (item 2) $qn \not\preceq^{k-1} q'n'$ for all $n' \in \mathbb{N}$. By the definition of sufficient values, we obtain $n \geq suf(q,q',k-1)$. By the construction of the gadgets and Equation (5.28) we get $t_{q,q'}n \not\preceq t'_{q,q'}n'$, which concludes our proof. □

Lemma 5.24. *For all control states $(p,p') \in Q \times Q'$ and naturals $k,m,n \in \mathbb{N}$:*

$$pm \not\preceq p'm' \text{ w.r.t. } \mathcal{S}_k,\mathcal{S}'_k \implies pm \not\preceq^k p'm' \text{ w.r.t. } \mathcal{M},\mathcal{M}' \tag{5.41}$$

Proof. Assume $pm \not\preceq p'm'$ w.r.t. $\mathcal{S}_k$ and $\mathcal{S}'_k$. Since both $\mathcal{S}_k,\mathcal{S}'_k$ are just one-counter nets, non-simulation manifests itself at some finite approximant $\alpha \in \mathbb{N}$, i.e., $pm \not\preceq_\alpha p'm'$. By definition

of $\preceq^k$ it suffices to show that some ordinal γ exists such that $pm \not\preceq^k_\gamma p'm'$ w.r.t. $\mathcal{M},\mathcal{M}'$. By the game characterization of approximants (Lemma 5.13) this amounts to showing a winning strategy for Spoiler in the approximant game from position $(pm,p'm',\gamma,k)$.

We proceed by induction on α. The claim is trivial for the base case $\alpha = 0$. For the induction step we consider a move $pm \xrightarrow{a} qn$ for some $a \in \mathrm{Act}$ by Spoiler in both games according to Spoiler's assumed winning strategy in the simulation game between $\mathcal{S}_k,\mathcal{S}'_k$. It cannot be a Spoiler move $p \xrightarrow{(p,q'),0} t_{p,q'}$ by Equation (5.32), because this would allow Duplicator to reply by moving to the universal state W by Equation (5.35). Now we consider all (possibly infinitely many) replies by Duplicator in the approximant game between $\mathcal{M},\mathcal{M}'$ from a position $(pm,p'm',\gamma,k)$ for some yet to be determined ordinal γ. These replies fall into two classes.

In the first class, Duplicator's move $p'm' \xrightarrow{a} q'n'$ is *not* due to an ω-transition and thus also a possible move in the simulation game between $\mathcal{S}_k,\mathcal{S}'_k$. From our assumption that Spoiler wins the simulation game from position $(pm,p'm')$ in at most $\alpha \in \mathbb{N}$ steps, it follows that Spoiler wins from $(qn,q'n')$ in at most $\alpha - 1$ steps. By induction hypothesis, there is a ordinal β s.t. Spoiler has a winning strategy in the approximant game for $\preceq^k_\beta$ between $\mathcal{M},\mathcal{M}'$ from position $(qn,q'n')$. There are only finitely many such replies. Let γ^0 be the maximal such β.

In the second class, Duplicator's move $p'm' \xrightarrow{a} q'n'$ uses an ω-transition, which does not exist in $\mathcal{S}'_k$. Instead, Duplicator can move $p'm' \xrightarrow{a,0} t'_{r,q'}m'$ by a transition due to Equation (5.34). From our assumption that Spoiler wins the simulation game from position $(pm,p'm')$ in at most $\alpha \in \mathbb{N}$ steps, it follows that Spoiler wins from $(qn,t'_{r,q'}m')$ in at most $\alpha - 1$ steps. If $r \neq q$ then this is trivially true by a move due to Equation (5.32). Otherwise, if $r = q$, then this can only be achieved by a Spoiler move $qn \xrightarrow{(q,q'),0} t_{q,q'}n$ in the next round, because for any other Spoiler move Duplicator has a winning countermove by Equations (5.37) or (5.38). In this case Duplicator can only reply with a move $t'_{q,q'}m' \xrightarrow{(q,q'),0} t'_{q,q'}m'$ due to Equation (5.36), and we must have that Spoiler can win in at most $\alpha - 2$ steps from position $(t_{q,q'}n,t'_{q,q'}m')$, which is the beginning of the testing gadget for states (q,q'). By construction of $\mathcal{S}_k,\mathcal{S}'_k$, in particular by definition of the gadgets and Equation (5.28), this implies that $n \geq \mathit{suf}(q,q',k-1)$. By the definition of sufficient values we obtain $\forall n' \in \mathbb{N}.\, qn \not\preceq^{k-1} q'n'$. Therefore, for every $n' \in \mathbb{N}$ there exists some ordinal β s.t. $qn \not\preceq^{k-1}_\beta q'n'$. Let γ be the least ordinal greater or equal all those β. Each of the finitely many distinct ω-transitions yields such an γ. Let γ^1 be the maximum of them.

Finally, we set $\gamma := \max(\gamma^0,\gamma^1) + 1$. Then every reply to Spoiler's initial move $pm \xrightarrow{a} qn$ in the approximant game from $(pm,p'm',\gamma,k)$ leads to a position that is winning for Spoiler. It follows that Spoiler has a winning strategy in the approximant game from $(pm,p'm',\gamma,k)$. □

The proof of Lemma 5.19 is now a formality.

Proof of Lemma 5.19. Let $\mathcal{M} = (Q,\mathrm{Act},\delta)$ and $\mathcal{M}' = (Q',\mathrm{Act},\delta')$. We iteratively construct nets

$(\mathcal{S}_k,\mathcal{S}'_k)$ that characterize $\preceq^k$ for growing $k \in \mathbb{N}$.

For the base case $k = 0$, we observe that $\preceq^0 = Q \times \mathbb{N} \times Q' \times \mathbb{N}$ is the full relation. The claim therefore trivially holds for the pair $\mathcal{S}_0,\mathcal{S}'_0$ of nets that contain no transitions at all. Also, by Lemma 5.21, point 1, the minimal sufficient value $suf(q,q',0)$ equals ω for every pair of states $(q,q') \in Q \times Q'$.

For the induction step, consider $k > 0$. By assumption, we have already constructed the pair $(\mathcal{S}_{k-1},\mathcal{S}'_{k-1})$ of nets correctly characterizing $\preceq^{k-1}$. By Theorem 4.19 (page 49) we know that the simulation preorder w.r.t. $\mathcal{S}_{k-1},\mathcal{S}'_{k-1}$ is effectively semilinear. Since semilinear sets are effectively closed under projections and complements, we can compute the semilinear representation of the approximant $\preceq^{k-1}$ and its complement and therefore also the values $suf(q,q',k-1)$ for all $(q,q') \in Q \times Q'$. Knowing these values, we can construct the next pair $(\mathcal{S}_k,\mathcal{S}'_k)$ of nets according to Definition 5.22. The correctness of this new pair follows from Lemmas 5.23 and 5.24. □

Note that in the proof above, we construct a description of the previous approximants only to compute the values $suf(p,p',k-1)$. We will now show that these values are in fact polynomially bounded and can also be computed in polynomial space.

5.4 Complexity Analysis

We show that the bounds on the coefficients of the Belt Theorem, as derived in Section 4.3, imply that the construction shown in the previous section for checking *weak* simulation actually uses only polynomial space.

To obtain an upper bound for the complexity of this procedure, we will show that the sizes of all nets $(\mathcal{S}_k,\mathcal{S}'_k)$, as constructed in Definition 5.22, are polynomial in the sizes of $\mathcal{M}$ and $\mathcal{M}'$. We start with some observations about the shape of the nets $\mathcal{S}_k$ and $\mathcal{S}'_k$.

Lemma 5.25.

1. *The net $\mathcal{S}'_k$ remains constant from index $k = 1$ on.*
2. *Every net $\mathcal{S}_k$ for $k > 0$ contains precisely $|Q \times Q'|$ many disjoint testing chains, one for each pair of states in $\mathcal{M}$ and $\mathcal{M}'$.*
3. *If $suf(q,q',k-1) \neq \omega$, then the length of the test chain for states q,q' in net $\mathcal{S}_k$ is exactly $suf(q,q',k-1)$. Otherwise, it is a simple e-labelled loop.*

Using these properties above and Lemma 5.21, point 2, we derive that indeed at some $k \in \mathbb{N}$, the sequence $(\mathcal{S}_i,\mathcal{S}'_i)_{i\in\mathbb{N}}$ of nets stabilises to $(\mathcal{S}_k,\mathcal{S}'_1)$. This holds because for any pair (q,q') there can only be one index i such that the respective sufficient value jumps from

$suf(q,q',i) = \omega$ to $suf(q,q',i+1) \in \mathbb{N}$. Because these nets characterize approximants $\preceq^k$ and $\preceq^{k+1}$ w.r.t. $\mathcal{M},\mathcal{M}'$ (by Lemmas 5.23 and 5.24) we obtain that $\preceq^k = \preceq^{k+1} = \preceq$.

Lemma 5.26. *Consider the sequence $(\mathcal{S}_k,\mathcal{S}'_k)_{k\in\mathbb{N}}$ as constructed in Definition 5.22 for the OCN $\mathcal{M}$ and ω-Net $\mathcal{M}'$. For any index $k \in \mathbb{N}$, the nets $\mathcal{S}_k,\mathcal{S}'_k$ are of polynomial size, and can be constructed in polynomial space with respect to the sizes of the original nets $\mathcal{M}$ and $\mathcal{M}'$.*

Proof. For $k = 0$, these nets are defined to be just copies of $\mathcal{M}$ and $\mathcal{M}'$ with no transitions. The claim is therefore trivial for $k = 0$. For all higher indices $k+1$, we consider nets $\mathcal{S}_{k+1}$ and $\mathcal{S}'_{k+1}$ individually.

By Lemma 5.25, point 1, $\mathcal{S}'_{k+1}$ is the same as $\mathcal{S}'_1$, which can easily be seen to be of polynomial size in the sizes of $\mathcal{M}$ and $\mathcal{M}'$ (cf. Definition 5.22). The net $\mathcal{S}_{k+1}$ is completely determined by the original pair of nets and the length of the test chains, which in turn are derived only from the minimal sufficient values $suf(q,q',k)$ for level k. By construction, the size of the net $\mathcal{S}_{k+1}$ is polynomial (actually linear) in the sizes of $\mathcal{M},\mathcal{M}'$ and the maximal length of a test chain in the net $\mathcal{S}_k$. By Lemma 5.25, point 3, it is therefore enough show that one can compute the values $suf(q,q',k)$ for all states $q \in Q$ and $q' \in Q'$ in polynomial space and bound them polynomially w.r.t. $\mathcal{M},\mathcal{M}'$ in case they are finite.

Recall that $suf(q,q',k)$ is defined in terms of the approximant $\preceq^k$, which is characterized as the strong simulation $\preceq$ relative to the nets $\mathcal{S}_k,\mathcal{S}'_k$ by Lemma 5.19.

Let C_k be the maximal length an acyclic path in the product of nets $\mathcal{S}_k$ and $\mathcal{S}'_k$. By Theorem 4.12, C_k is sufficient for the claim of the Belt Theorem applied to the nets $\mathcal{S}_k$ and $\mathcal{S}'_k$. In particular, by Lemma 4.20, it bounds the width of all vertical belts and therefore all finite values $suf(q,q',k)$:

$$suf(q,q',k) \in \mathbb{N} \implies suf(q,q',k) \leq C_k. \tag{5.42}$$

The form of the nets (Lemma 5.25, points 2,3) means that the longest acyclic path in the product of $\mathcal{S}_k$ and $\mathcal{S}'_k$, must actually start within the part described by the original nets, and eventually go through one of the test chains. We can therefore bound C_k by

$$C_k \leq C_1 + C_{k-1}. \tag{5.43}$$

We fix a pair (q,q') of states and consider the length of the test chain for this pair in the net $\mathcal{S}_i$ for growing indices i. By Lemma 5.21 and Lemma 5.25, point 3, we see that there can only be one index i such that the length of the chain increases, namely if $suf(q,q',i) = \omega > suf(q,q',i+1) \in \mathbb{N}$. Because there are always exactly $K = |Q \times Q'|$ many test chains, this means that there can be at most K indices i such that $C_{i+1} \geq C_i$. Together with Equation (5.43) we can therefore globally bound every C_k by

$$C_k \leq K \cdot C_1. \tag{5.44}$$

We conclude that the sizes of all $\mathcal{S}_k, \mathcal{S}'_k$ are polynomial in the sizes of $\mathcal{M}$ and $\mathcal{M}'$. By Lemma 4.20, we can thus compute the exact values of $suf(q,q',k)$ and construct $\mathcal{S}_{k+1}, \mathcal{S}'_{k+1}$ using polynomial space w.r.t. $\mathcal{M}$ and $\mathcal{M}'$ as required. □

Theorem 5.27. *For any pair $\mathcal{N}, \mathcal{N}'$ of one-counter nets one can construct two polynomially bigger OCNs $\mathcal{S}$ and $\mathcal{S}'$ that contain the original states of $\mathcal{N}$ and $\mathcal{N}'$ respectively, such that weak simulation $\leqq$ w.r.t. $\mathcal{N}, \mathcal{N}'$ is the projection of strong simulation w.r.t. $\mathcal{S}, \mathcal{S}'$.*

Proof. The claim follows from Theorem 5.3 and Lemmas 5.19 and 5.26. □

The main result of this section is now a direct consequence of Theorems 4.19 and 5.27. Recall that a PSPACE lower bound already holds for strong simulation.

Theorem 5.28. *Checking weak simulation preorder between two OCNs is PSPACE-complete. Moreover, the largest weak simulation relation is semilinear and can be explicitly represented in space exponential in the sizes of the input nets.*

5.5 Comparison with Finite Systems

We now turn to strong and weak simulation between OCA/OCN and finite systems.

Srba [47] showed a PSPACE lower bound for checking strong simulation between OCA and NFA, in both directions. This result was obtained by reduction from the emptiness problem for alternating finite automata over a unary alphabet, which was already known to be PSPACE-hard [21, 31]. Matching upper bounds (for strong and weak simulation) follow from the fact that μ-calculus model checking is in PSPACE for one-counter systems [46], because finite systems as well as the strong/weak simulation conditions can be expressed as μ-calculus formulae.

Strong simulation between one-counter *nets* and finite systems was known to be P-complete, in both directions [33]. The complexity of the corresponding *weak* simulation problems however were only known to be between P and PSPACE. In this section we show that these problems are in fact polynomial time complete. Note that for the direction $NFA \leqq$ OCN, this is not immediately obvious because one still has to deal with the problem of infinite branching. The usual weak simulation approximants do not converge at level ω, but only at ω^2 (see Theorem 5.18). The easy direction is the following.

Theorem 5.29. *Checking if a finite-state process weakly simulates a OCN process is in P.*

Proof. It suffices to first replace the step relation in the finite system with its weak closure so that $q \stackrel{a}{\Longrightarrow} q' \iff q \stackrel{a}{\longrightarrow} q'$ and then check if the resulting finite system strongly simulates the net. The finiteness of the state space allows us to compute the weak closure in polynomial time. A polynomial time algorithm for checking strong simulation between OCN and finite-state processes can be found in [33]. □

For the other direction, checking if a OCN process weakly simulates a finite-state process, we show that it suffices to consider a finite version of the net where the counter is capped at a polynomially bounded level. The crucial observation is that monotonicity (Lemma 3.13) implies that Duplicator must be able to ensure that his counter never decreases along any partial play that repeats control states.

Definition 5.30. Let $\mathcal{N} = (Q, \mathrm{Act}, \delta)$ be a OCN and $l \in \mathbb{N}$. The *l-capped version* of $\mathcal{N}$ is the finite system $\mathcal{N}_l = (Q_l, \longrightarrow)$ with states $Q_l = \{(q,n) | q \in Q, n \leq l\}$ that has a transition $(p,m) \xrightarrow{a} (q, \min\{n,l\})$ iff $pm \xrightarrow{a}_{\mathcal{N}} qn$.

It is easy to see that $\mathcal{N}_l$ can be constructed from $\mathcal{N}$ in time proportional to $l \times |\mathcal{N}|$. We observe the following properties.

Proposition 5.31. *For $n,l \in \mathbb{N}$ and $q \in Q$,*

1. $(q, \min\{n,l\}) \leqq qn$,
2. $qn \leqq_l (q, \min\{n,l\})$,
3. $(q, \min\{n,l\}) \leqq (q, \min\{n+1,l\})$.

We continue to show that Duplicator can be assumed to play optimally in a sense that depends on cycles in the underlying control graphs. Consider a simulation game between a finite process and a OCN process (or its l-capped version). Recall that OCN subsume finite systems, so we can identify plays of the weak simulation game with paths in the product graph of the given systems. In our scenario, it is actually only Duplicator that could be restricted by his counter. So we are looking at a one-dimensional game and it makes sense what it means for a play to be *decreasing*.

Definition 5.32. A path $\pi = p_0 t_0 \ldots t_{k-1} p_k$ in a OCN is *decreasing* if $\Delta(\pi) < 0$. Similarly, a path in the product graph $(\mathcal{F} \times \mathcal{N})$ of finite system $\mathcal{F}$ and OCN $\mathcal{N}$ is decreasing if the induced path in $\mathcal{N}$ is.

Lemma 5.33. *Suppose $p \leqq qn$. Then Duplicator has a winning strategy in the weak simulation game that moreover guarantees the following properties in every play.*

1. *No round decreases the counter by more than $|Q| \cdot 2 + 1$.*
2. *No simple cycle is decreasing.*

Proof. A weak step $s_0 m_0 \overset{a}{\Longrightarrow} t_l n_l$ made by Duplicator as a response in the weak simulation game is due to some sequence

$$s_0 m_0 \xrightarrow{\tau} s_1 m_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} s_l m_l \xrightarrow{a} t_0 n_0 \xrightarrow{\tau} t_1 n_1 \xrightarrow{\tau} \ldots \xrightarrow{\tau} t_k n_k. \tag{5.45}$$

By monotonicity (Lemma 3.13), it is suboptimal for Duplicator to decrease the counter when silently moving from state s_i to $s_j = s_i$ (or from t_i to $t_j = t_i$) for $i < j$. Also, we can safely assume that a weak step as above will be non-decreasing if there are indices $i < j$ with $s_i = s_j$ and $m_i < m_j$ (or $t_i = t_j$ and $n_i < n_j$). Therefore, if the weak step decreases the counter, both silent paths will be acyclic and hence no longer than $|Q|$. Such a step cannot decrease the counter by more than $|Q| \cdot 2 + 1$.

For the second point observe that if Duplicator cannot avoid the next simple cycle to be decreasing, then Spoiler must have some strategy to enforce cycles to be decreasing. Such a strategy must be winning for Spoiler as it eventually exhausts Duplicator's counter. □

The next lemma uses the previously stated optimality assumption to show that we only need to consider a polynomially capped net to determine if a OCN process weakly simulates a finite-state process.

Lemma 5.34. *For any pair $\mathcal{F} = (S, \longrightarrow)$, $\mathcal{N} = (Q, Act, \delta)$ of a finite-state system and a OCN respectively, there is a polynomial bound c such that for all $n \in \mathbb{N}$:*

$$p \leqq qn \iff p \leqq (q, \min\{n, c\}) \tag{5.46}$$

where $(q, \min\{n,c\})$ is a state of the c-capped version $\mathcal{N}_c$ of $\mathcal{N}$.

Proof. The "if" direction follows directly from Proposition 5.31 (point 1). For the other direction we show that $c = (2|Q|+1)(|S \times Q|+1)$ suffices to contradict $p \leqq qn$ and $p \not\leqq (q, \min\{n,c\})$.

If $p \leqq qn$ then $p \leqq_c qn$ and by Proposition 5.31 (2) it also holds that $p \leqq_c (q, \min\{n,c\})$. Moreover, Duplicator has an *optimal* strategy in the sense of Lemma 5.33. We see that using the same strategy in the game p vs. $(q, \min\{n,c\})$ guarantees that

1. No round decreases the second component of Duplicator's state by more than $2|Q|+1$.
2. For any simple cycle between game positions $p_i, (q_i, n_i)$ and $p_j, (q_j, n_j)$ it holds that $n_j \geq n_i$ or $n_j \geq c - (2|Q|+1)(|S \times Q|)$.

To see the second point observe that the only way a simple cycle can be decreasing is because some of its increases are dropped due to the counter being at its limit c. Then point 1 implies $n_j \geq c - (2|Q|+1)(|S \times Q|)$ because the length of simple cycles is bounded by $|S \times Q|$. By our assumption $p \not\leqq (q, \min\{n,c\})$, we can consider a play

$$\pi = (p_0, (q_0, n_0))(t_0, t_0')(p_1, (q_1, n_1))(t_1, t_1') \ldots (p_l, (q_l, n_l)) \tag{5.47}$$

where $p_0 = p$ and $(q_0, n_0) = (q, \min\{n,c\})$, along which Duplicator plays optimally as described above and which is winning for Spoiler in the smallest possible number of rounds.

Since $c > |S \times Q|$, we know that π must contain cycles as otherwise $l \leq |S \times Q|$ and thus $p \leqq_c (q, \min\{n,c\})$ contradicts that π is won by Spoiler. So assume the last simple cycle in

π is between positions i and j. We know that $n_j < n_i$, as otherwise omitting this last cycle results in a shorter play, which by monotonicity (Lemma 3.13, applied to weak simulation approximants $\leqq_\alpha$) must also be winning for Spoiler. Thus, by Observation 2 above, $n_j \geq c - (2|Q|+1)(|S \times Q|)$ and in particular we get that $n_j \geq (2|Q|+1)$ due to our choice of c.

Finally, recall that the last position $(p_l,(q_l,n_l))$ in the play π must be directly winning for Spoiler. That is, for some action a it holds that $p_l \xrightarrow{a}$ and $(q_l,n_l) \overset{a}{\not\Longrightarrow}$. But since $n_l > (2|Q|+1)$, we know that also $q_l n_l \overset{a}{\not\Longrightarrow}$ in the original OCN process. This contradicts our assumption that Duplicator's original strategy in the unrestricted game was winning. □

Theorem 5.35. *Checking if a OCN process weakly simulates a finite-state process can be done in polynomial time.*

Proof. To check if $p \leqq qn$ holds we can by Lemma 5.34 equivalently check $p \leqq (q,\min\{n,c\})$ where $(q,\min\{n,c\})$ is a state of a polynomially bounded finite system $\mathcal{N}_c$. Checking weak simulation between two finite processes is in P. □

Chapter 6

Trace and Language Inclusion

The *traces* of a process are all (finite) sequences of actions it can emit. Trace inclusion is the problem whether all traces of one process are also traces of another given process. We look at the decidability and complexity of checking trace inclusion between finite systems, OCN and OCA. Where appropriate, we consider weak trace inclusion and trace universality problems.

Recall (Definition 2.6 on page 10) that the set of traces of a process α is given as $\mathcal{T}(\alpha) = \{w \in \mathrm{Act}^* \mid \exists \beta . \alpha \xrightarrow{w} \beta\}$ and trace inclusion $\alpha \subseteq \beta$ holds for two processes α and β iff $\mathcal{T}(\alpha) \subseteq \mathcal{T}(\beta)$. We want to decide if trace inclusion holds for a given pair of one-counter processes, which are configurations of (restricted) one-counter automata.

Related problems have been addressed since the 1970's: Valiant [50] studies *language* inclusion and equivalence of deterministic pushdown automata (DPDAs) and subclasses. The language of a process is a subset of its traces, usually defined by excluding those traces that do not lead to some accepting configuration. Valiant showed that language inclusion for deterministic one-counter automata (DOCA) and language universality for nondeterministic OCAs are already undecidable. Valiant and Paterson [51] showed the decidability of language equivalence for DOCAs. This problem has recently been shown to be NL-complete by Böhm, Göller, and Jančar [6], assuming fixed initial counter values. Jančar, Esparza, and Moller [28, 25] consider trace inclusion between Petri nets/VASS and finite systems and prove decidability in both directions. Jančar [24] showed that trace equivalence, and therefore also inclusion, is undecidable for 2-dimensional VASSs. The status of trace inclusion and equivalence for OCNs was stated as an open question in [12].

In the following section, we discuss how trace inclusion relates to language inclusion and recover the mentioned results by Valiant [50], that trace inclusion between DOCA, as well as trace universality for nondeterministic OCA are undecidable.

In Section 6.2 we show that trace inclusion and equivalence are undecidable for OCNs. This result holds even for processes with initial counter values fixed to 0 and (by Lemma 3.10) extends to the case where only the process on the right is nondeterministic.

In Section 6.3, we study trace inclusion between OCA and finite systems. We show that strong and weak trace inclusion between OCA and NFA are PSPACE-complete. The other direction is undecidable (Corollary 6.6), so we focus on the problem NFA $\subseteq$ OCN in more detail. This problem can be easily seen to be (logspace) inter-reducible with trace universality for OCN. It was known to be decidable, even for the more general class of VASSs [28], but the exact complexity was unknown. We show that trace universality of OCN is in fact Ackermannian, i.e., that no primitive recursive upper bound is possible. We also provide Ackermannian lower bounds on the length of minimal witnesses for non-universality.

6.1 Traces vs. Languages

In formal language theory, automata are seen as as acceptors of action sequences, where acceptance is usually defined by a reachability criterion. Given a distinguished initial configuration α and a set Acc of accepting configurations, an automaton over a (finite) alphabet Act of actions accepts a word $w \in \mathrm{Act}^*$ if it allows a w-labelled run from α to some $\beta \in Acc$. The *language* of the automaton consists of all accepted words.

For one-counter automata with state-set Q, different acceptence criteria can been considered. The most common ones are by final state ($Acc = F \times \mathbb{N}$ for some fixed $F \subseteq Q$), by empty counter ($Acc = Q \times \{0\}$) and by empty counter and final state simultaneously ($Acc = F \times \{0\}$). Naturally, the set $\mathcal{T}(pm)$ of traces of pm can be seen as its language, where every configuration is accepting, i.e., $Acc = Q \times \mathbb{N}$.

For unrestricted OCA, the respective inclusion problems are all reducible (in logspace) to language inclusion where acceptance is by final state, because of the zero-testing capability of the model. For OCN accepting by final state, inclusion is monotone with respect to the counter in the sense of Lemma 3.13 (page 27). This is not true for acceptance by empty counter. For instance, if the system consists of a single, decreasing loop $p \xrightarrow{a,-1} p$, then the languages of $p1$ and $p2$ are $\{a\}$ and $\{aa\}$ respectively, and thus incomparable.

We will go on to show that for OCA and OCN, checking language inclusion where acceptance is *by final state* can be reduced to trace inclusion and vice versa.

Definition 6.1. Let $\mathcal{A} = (Q, \mathrm{Act}, \delta, \delta_0)$ be a OCA and $F \subseteq Q$ a set of *final* states. The *language* of a configuration $pm \in Q \times \mathbb{N}$ is

$$\mathcal{L}_{\mathcal{A},F}(pm) = \{w \in \mathrm{Act}^* \mid pm \xrightarrow{w} fn \text{ for some } fn \in F \times \mathbb{N}\}. \tag{6.1}$$

We will drop the indices $\mathcal{A}, F$ if they are clear from the context. For processes pm and $p'm'$ we write $pm \subseteq_f p'm'$ to mean that $\mathcal{L}(pm) \subseteq \mathcal{L}(p'm')$. If pm and $p'm'$ are processes of different OCA, the languages are assumed to be defined relative to their disjoint union.

By *language inclusion* we mean both the relation $\subseteq_f$ itself and also the decision problem OCA $\subseteq_f$ OCA that asks if $pm \subseteq_f p'm'$ holds for two given OCA processes pm and $p'm'$. Language *universality* is the problem that asks if $\mathrm{Act}^* = \mathcal{L}(pm)$ holds for a given process pm.

Trace inclusion is trivially reducible to language inclusion (and trace universality to language universality), because both notions coincide in case all states are final. The next lemma shows that, for acceptence by final state, the same holds for the other direction. This motivates our later focus on the seemingly less expressive notion of trace inclusion.

Lemma 6.2. *For OCAs, language inclusion and universality can be reduced (in logspace) to trace inclusion and universality, respectively.*

Proof. We show how to reduce language inclusion to trace inclusion. The same construction also works for universality. Assume we are given two processes of the OCA $\mathcal{A}$ with alphabet Act. We construct $\mathcal{B}$, which is a copy of $\mathcal{A}$, where every final state has a f-labelled self-loop for some new action f. Moreover, we add to $\mathcal{B}$ a new universal state U with self-loops $U \xrightarrow{a,0} U$ for every original action $a \in \mathrm{Act}$. This means that in $\mathcal{B}$, $Un \xrightarrow{a}$ holds for all original actions and $n \in \mathbb{N}$. Finally, we connect original states to U in a way that whenever $qn \not\xrightarrow{a}$ in $\mathcal{A}$ for some configuration qn of $\mathcal{A}$, we have $qn \xrightarrow{a} Un$ in $\mathcal{B}$. This can be achieved by adding transition $q \xrightarrow{a,0} U$ if state q has no a-labelled outgoing transition and a zero-testing transition $q \xrightarrow{a,=0} U$ if all a-labelled outgoing transitions of q have negative effect. Observe that this construction preserves determinism, but can only be used if $\mathcal{B}$ allows zero-tests.

To see why this construction is correct assume $w \in \mathrm{Act}^*$ witnesses $pm \not\subseteq_f p'm'$ in $\mathcal{A}$. Then $pm \xrightarrow{w} qn$ for some final state q and thus $pm \xrightarrow{wf}$ in $\mathcal{B}$. At the same time, no run $p'm' \xrightarrow{w} q'n'$ exists to some final state q', either because $p'm' \not\xrightarrow{w}$ or because no such state q' is final. In both cases wf witnesses $pm \not\subseteq qn$ in $\mathcal{B}$. Conversely, every witness for non-trace inclusion in $\mathcal{B}$ must end with the new action f, because this is the only action that can be disabled. So a shortest witness for $pm \not\subseteq qn$ in $\mathcal{B}$ must be of the form wf for some $w \in \mathrm{Act}^*$. This means that w witnesses $pm \not\subseteq_f qn$ in $\mathcal{A}$ because in order to play the last action, the process on the left must reach a final state and in order to not be able to play it, the process on the right must not be able to reach a final state using an equally labelled path. □

Remark 6.3. A similar construction can be used to reduce OCN $\subseteq_f$ OCN to trace inclusion between OCN, where zero-tests are not available. In this case we add transitions $q \xrightarrow{a,0} U$ for all states q and actions a. The correctness of this alternative construction follows using the same argument. This alternative construction does not introduce new zero-testing transitions and thus preserves the system being a net. However, it does *not* preserve determinism.

Later in this chapter we will study trace inclusion where the process on the right is deterministic and moreover, at least one side is restricted to be a one-counter net, and thus does not

contain zero-tests. The reason for this restriction is that both trace inclusion for deterministic OCAs and trace universality of nondeterministic OCAs are already undecidable. This can be seen by the following argument that is originally due to Valiant [50]. We first observe that the emptiness problem of the intersection of two DOCA languages is undecidable.

Theorem 6.4. *It is undecidable to check if $\mathcal{L}(p0) \cap \mathcal{L}(q0) = \emptyset$ holds for two given processes $p0$ and $q0$ of a deterministic one-counter automaton.*

Proof. By reduction from the halting problem for 2-CM (Theorem 2.4). Both OCA processes directly simulate the finite control and keep track of only one counter each.

Suppose we are given a 2-CM with states in Q and transitions in $T \subseteq (Q \times \mathrm{Act} \times Op \times Q)$, where $Op = \{\mathrm{inc}_i, \mathrm{dec}_i, \mathrm{ifz}_i \mid 1 \leq i \leq 2\}$ are the possible operations on the counters. We can contruct a OCA $\mathcal{A} = (Q', T, \delta, \delta_0)$ with states set $Q' = \{p_1, p_2 \mid p \in Q\}$ and action labels in T, such that whenever some transition $t \in T$ justifies a step $(p, c_1, c_2) \xrightarrow{a} (q, c_1 + d_1, c_2 + d_2)$ in the 2-CM, then there are steps $(p_1, c_1) \xrightarrow{t} (q_1, c_1 + d_1)$ and $(p_2, c_2) \xrightarrow{t} (q_2, c_2 + d_2)$ in $\mathcal{A}$. For instance, for $t = (p, a, \mathrm{inc}_1, q) \in T$ we introduce transitions $(p_1, t, 1, q_1) \in \delta$ and $(p_2, t, 0, q_2) \in \delta$ in the OCA. Notice that the so constructed OCA is deterministic.

Assume that $q_{halt} \in Q$ is the halting state of the 2-CM. If we let q_{halt} be the only accepting state in the OCA, then the language of $q_1 0$ describes exactly those paths from state q in the 2-CM that correctly update the first counter when initialized to 0. Similarly, $\mathcal{L}(q_2 0)$ describes those (proto-) runs that are valid w.r.t. the second counter. This means the intersection of their languages is non-empty precisely if there is a valid and terminating run of the 2-CM, starting in configuration $(q, 0, 0)$. □

The class of languages definable by *deterministic* OCAs accepting by final state is effectively closed under complementation because one can simply swap final and non-final states. Together with Theorem 6.4 and Lemma 6.2 we derive the following.

Corollary 6.5. *Language and trace inclusion are undecidable for DOCAs.*

Proof. Language inclusion DOCA $\subseteq_f$ DOCA can be shown to be undecidable by reduction from the problem of checking emptiness of the intersection of DOCA languages, Theorem 6.4. Complementing the language of a process can be achieved by swapping final and non-final states: $\mathrm{Act}^* \setminus \mathcal{L}_{\mathcal{A},F}(pm) = \mathcal{L}_{\mathcal{A},Q\setminus F}(pm)$. This means that $\mathcal{L}_F(p0) \cap \mathcal{L}_F(q0) = \emptyset$ iff both $\mathcal{L}_F(p0) \subseteq \mathcal{L}_{Q\setminus F}(q0)$ and $\mathcal{L}_F(q0) \subseteq \mathcal{L}_{Q\setminus F}(p0)$. The undecidability of trace inclusion follows from the above and Lemma 6.2. □

Corollary 6.6. *Language and trace universality are undecidable for nondeterministic OCAs.*

Proof. $\mathcal{L}_F(p0) \cap \mathcal{L}_F(q0) = \emptyset$ iff $\mathcal{L}_{Q\setminus F}(p0) \cup \mathcal{L}_{Q\setminus F}(q0) = \mathrm{Act}^*$ iff the process $r0$, which can nondeterministically choose between $p0$ and $q0$ has a universal language. This shows that

language universality is undecidable for OCA. The undecidability of trace inclusion follows by Lemma 6.2. □

6.2 Undecidability of Inclusion for One-Counter Nets

In this section, we show that checking trace inclusion (and equivalence) is undecidable for OCN. The proof is by reduction from the containment problem for weighted finite automata over min-plus semirings, which is known to be undecidable [3].

Definition 6.7. A *weighted finite automaton (WFA)* over a semiring $(\mathbb{S},\oplus,\mathbf{0},\otimes,\mathbf{1})$ is an NFA where each transition carries a weight, which is an element of $\mathbb{S}$. Formally, it is a structure $\mathcal{A} = (Q,\Sigma,\delta,q_0)$ where Q is a finite set of states, Σ is a finite alphabet, $\delta \subseteq (Q \times \Sigma \times \mathbb{S} \times Q)$ is a transition relation and $q_0 \in Q$ is a designated initial state.

$\mathcal{A}$ is *complete* if for every $p \in Q$ and $a \in \Sigma$, there is at least one transition $(p,a,d,q) \in \delta$. A *run* of $\mathcal{A}$ on a word $w = w_0 w_1 \dots w_n \in \Sigma^*$ is a sequence $\rho = (q_i,w_i,d_i,q_{i+1})_{0 \le i \le n} \in \delta^*$ of transitions. The *value* of such a run is $\mathcal{A}(\rho) = \bigotimes_{i=0}^{n} d_i$, the semiring product of its weights. The value of a word $w \in \Sigma^*$ is $\mathcal{A}(w) = \bigoplus\{\mathcal{A}(\rho) \mid \rho \text{ is a run of } \mathcal{A} \text{ on } w\}$, the semiring sum of the values of all possible runs on w.

The value $\mathcal{A}(w)$ is well-defined because there can only be finitely many runs of $\mathcal{A}$ on any given word w due to the finite branching property of WFAs. If there is no run of $\mathcal{A}$ on w, then $\mathcal{A}(w) = \mathbf{0}$ is the empty semiring sum.

One can interpret WFAs as a way of defining functions $\mathcal{A} : \Sigma^* \to \mathbb{S}$. In particular, complete WFAs define total functions on Σ^*. Unlike for NFAs, where the containment problem asks if the *language* of one automaton is contained in that of another, in the weighted setting containment is relative to some order on the semiring domain that compares the values of words in the two automata.

We consider the *min-plus* semiring $(\mathbb{Z} \cup \{\infty\},\min,\infty,+,0)$, where the semiring sum and product are minimum and addition on $\mathbb{Z} \cup \{\infty\}$ respectively. For WFAs $\mathcal{A}$ and $\mathcal{B}$ over this semiring and with alphabet Σ, we write $\mathcal{A} \le \mathcal{B}$ if for every word $w \in \Sigma^*$ it holds that $\mathcal{A}(w) \ge \mathcal{B}(w)$. Intuitively, if $\mathcal{A} \le \mathcal{B}$, it is harder for a word to be in $\mathcal{A}$ than in $\mathcal{B}$, so its value in $\mathcal{A}$ is worse than in $\mathcal{B}$. Since we are in a min-plus semiring, worse means larger. The *containment problem* asks if $\mathcal{A} \le \mathcal{B}$ holds for two given WFAs $\mathcal{A}$ and $\mathcal{B}$ over $(\mathbb{Z} \cup \{\infty\},\min,\infty,+,0)$ with the same alphabet. We recall the following result from Almagor, Boker, and Kupferman [3].

Theorem 6.8 ([3], Theorem 4)**.** *The containment problem is undecidable for WFAs over* $(\mathbb{Z} \cup \{\infty\},\min,\infty,+,0)$ *with weights in* $\{-1,0,1\}$.

We turn to WFAs over the complementary *max-plus* semirings $(\mathbb{Z} \cup \{-\infty\},\max,-\infty,+,0)$ and $(\mathbb{N} \cup \{-\infty\},\max,-\infty,+,0)$. Here, the containment problem asks if $\mathcal{A}(w) \le \mathcal{B}(w)$ holds for

all $w \in \Sigma^*$. We again write $\mathcal{A} \leq \mathcal{B}$ if this is the case.

Lemma 6.9. *The containment problem for WFAs over* $(\mathbb{N} \cup \{-\infty\}, \max, -\infty, +, 0)$ *with weights in* $\{0,1\}$ *is undecidable.*

Proof. We can reduce the undecidable containment problem for WFAs over min-plus as considered in Theorem 6.8 to containment for WFAs over max-plus with weights in $\{-1,0,1\}$ simply by inverting all weights in the input automata.

Next, we reduce this problem to the containment for WFAs over $(\mathbb{N} \cup \{-\infty\}, \max, -\infty, +, 0)$ with weights in $\{0,1,2\}$. Let $\mathcal{A}$ and $\mathcal{A}'$ be two WFAs over $(\mathbb{Z} \cup \{-\infty\}, \max, -\infty, +, 0)$ with weights in $\{-1,0,1\}$. Construct WFAs $\mathcal{B}$ and $\mathcal{B}'$ by replacing every weight w by $w' = w + 1$ in transitions of $\mathcal{A}$ and $\mathcal{A}'$ respectively. The resulting automata have weights in $\{0,1,2\}$ and for every word $w \in \Sigma^*$ we have $\mathcal{B}(w) = \mathcal{A}(w) + |w| \in \mathbb{N}$ and $\mathcal{B}'(w) = \mathcal{A}'(w) + |w| \in \mathbb{N}$. It follows that

$$\mathcal{B}(w) \leq \mathcal{B}'(w) \iff \mathcal{A}(w) \leq \mathcal{A}'(w), \tag{6.2}$$

so containment for WFAs over $(\mathbb{N} \cup \{-\infty\}, \max, -\infty, +, 0)$ with weights in $\{0,1,2\}$ is undecidable.

To complete the proof, we show how to normalize the weights to $\{0,1\}$. Let x be a fresh symbol. The WFAs C and C' are the result of replacing every transition $t = (p \xrightarrow{a,d} q)$ in $\mathcal{B}$ and $\mathcal{B}'$ respectively, by two transitions $p \xrightarrow{a,d_0} p_t \xrightarrow{x,d_1} q$ where $d = d_0 + d_1$ is the original weight and p_t is a new state. We see that every word w in the domain of C is of the form $w = a_0 x a_1 x \dots x a_n$ or $w = a_0 x a_1 x \dots x a_n x$. Moreover, for any such word, $C(a_0 x a_1 x \dots x a_n) = \mathcal{B}(a_0 a_1 a_2 \dots a_n)$ and the same holds for C' and $\mathcal{B}'$. We thus get $\mathcal{B} \leq \mathcal{B}' \iff C \leq C'$, which completes the proof. □

Theorem 6.10. *Trace inclusion and equivalence are undecidable for OCNs.*

Proof. Inclusion can trivially be reduced to equivalence for nondeterministic systems like OCN. We show the undecidability of inclusion by reduction from the containment problem of WFAs over $(\mathbb{N} \cup \{-\infty\}, \max, -\infty, +, 0)$ with weights in $\{0,1\}$. The idea is to encode the WFA as OCN, using the counter as accumulator. To ensure a faithful encoding of WFA containment, the OCN can at any point jump to a gadget that compares the counter values by simultaneously counting down.

Given WFAs $\mathcal{A} = (Q, \Sigma, \delta, q)$ and $\mathcal{A}' = (Q', \Sigma, \delta', q')$ we construct nets $\mathcal{B}$ and $\mathcal{B}'$ with states $Q \cup \{D\}$ and $Q' \cup \{D\}$ resp., over alphabet $\mathrm{Act} = \Sigma \cup \{d\}$ where d is a fresh symbol. We add transitions $D \xrightarrow{d,-1} D$ to both nets as well as $q \xrightarrow{d,-1} D$ for any original state (in Q or Q' respectively). We argue that $\mathcal{A} \leq \mathcal{A}'$ if and only if $\mathcal{T}(q0) \subseteq \mathcal{T}(q'0)$.

Assume a witness w with $\mathcal{A}(w) = v > L_{\mathcal{A}'}(w)$. Then there is a run of $\mathcal{A}$ on w with a value higher than that of any run of $\mathcal{A}'$ on w. So the word wd^v must be a valid trace from $q0$, but not

from $q'0$. Conversely, if $\mathcal{A}(w) \leq \mathcal{A}'(w)$ for all $w \in \Sigma^*$, then for any run of $\mathcal{B}$ there is a run of $\mathcal{B}'$ over the same sequence of actions which accumulates a higher or equal counter value. Thus no such word can be extended to a counterexample for trace inclusion by appending finitely many d's. □

6.3 Comparison with Finite Systems

We turn to checking strong and weak trace inclusion between finite-state systems and one-counter systems. First, we focus on the question OCA ⊆ NFA, if the (weak) traces of a given OCA process are subsumed by those of a finite process, and show that this problem is PSPACE-complete. We rely on the following lemma about the reachability of a fixed control state in a one-counter automaton.

Lemma 6.11 (Control state Reachability)**.** *Let $p,q \in Q$ be states of a OCA, $K = |Q|$ and $m \in \mathbb{N}$. If there exists some $n \in \mathbb{N}$ such that $pm \longrightarrow^* qn$, then $pm \longrightarrow^k qm_k$ for some $m_k \in \mathbb{N}$ and $k \leq mK + 2K^3 - K^2$.*

Proof. We distinguish two cases depending on whether or not there is a positive path from pm to control state q.

Case 1:. Assume that a positive path $pm \longrightarrow_+^* qn$ exists and consider a shortest positive path $(p_0m_0) \longrightarrow_+ (p_1m_1) \longrightarrow_+ \cdots \longrightarrow_+ (p_km_k)$ from $pm = p_0m_0$ to $qn = p_km_k$. We know that there is a path from p_0 to p_k in the control graph of the automaton that uses transitions in δ only. So there must be such a path in the control graph that is shorter than K. Thus, if $m_0 \geq K$, then there is a n_k such that $p_0m_0 \longrightarrow_+^k p_kn_k$ for some $k \leq K$. Otherwise, if $m_0 < K$, we recall that $p_im_i \longrightarrow^k p_jm_j$ implies $p_i(m_i+1) \longrightarrow^k p_j(m_j+1)$. After at most $K-1$ steps, our minimal path will repeat some control state $p_i = p_j$ at positions $i < j < K$. By minimality we can assume that then, $m_i < m_j$. Therefore, after at most K such repetitions the counter will reach a value $\geq K$, wich means that the remaining path must be no longer than K. This allows us to bound the length of the shortest positive path from p_0m_0 to control state p_k by $(K-1)K + K = K^2$.

Case 2:. No positive path $pm \longrightarrow_+^* qn$ exists for any $n \in \mathbb{N}$. Consider a shortest path

$$p_0m_0 \xrightarrow{\pi_1}_+^* p_1m_1 \xrightarrow{\pi_2}^* p_2m_2 \xrightarrow{\pi_3}_+^* p_km_k \tag{6.3}$$

from $p_0m_0 = pm$ to p_km_k, where $p_k = q$ and π_1 and π_3 are the longest positive prefix and suffix, respectively. The prefix π_1 is a positive path from counter value m_0 to $m_1 = 0$. By Lemma 3.5 on page 21, it is no longer than $m_0K + K^3 - K^2$. The intermediate path π_2 starts and ends with a zero-testing transition and is therefore no longer than $K^3 - K^2$ by Lemma 3.5. The length of the suffix π_3 can be bounded by K^2 as in the first case. To conclude, the length of the shortest path from pm to some configuration with control state q is bounded by $mK + 2K^3 - K^2$. □

Theorem 6.12. *Checking strong trace inclusion $pm \subseteq q$ or weak trace inclusion $pm \subsetneq q$ for a OCA process pm and a finite process q is PSPACE-complete.*

Proof. A PSPACE lower bound holds already for strong trace universality for finite-state systems [38]. The weak trace inclusion problem can trivially be reduced to the strong one in cubic time by taking the transitive closure of the finite system w.r.t. silent (τ-labelled) transitions. It remains to show a PSPACE upper bound for OCA $\subseteq$ NFA. Let pm be a configuration of the OCA $\mathcal{A} = (Q, Act, \delta, \delta_0)$ and q a state of the NFA $\mathcal{B} = (S, Act, \delta)$ and let $\bar{\mathcal{B}}$ denote the deterministic powerset automaton for $\mathcal{B}$.

To check if $\mathcal{T}(pm) \not\subseteq \mathcal{T}(q)$ holds we can equivalently test $\mathcal{T}(pm) \cap \mathcal{T}(q)^c \neq \emptyset$. That is, if in the product automaton $\mathcal{A} \times \bar{\mathcal{B}}$ some control state $(p', \emptyset)$ is reachable from initial configuration $(p, \{q\})m$. This can be checked by nondeterministically guessing a path stepwise. Note that this product automaton is a OCA with $K = |Q| * 2^{|S|}$ states. By Lemma 6.11, we know that the shortest path that witnesses such a control state reachability is bounded by $b = mK + 2K^3 - K^2$. This bounds the number of steps we need to consider until we can safely terminate and conclude that in fact trace inclusion holds. The bound b is polynomial in m and $|Q|$ and exponential in $|S|$. However, we need only polynomial space to store a configuration of $\mathcal{A} \times \bar{\mathcal{B}}$ (with control state numbers and counter values encoded in binary) and the binary coded values of the search-depth and its bound b. Thus we can check the condition in PSPACE. □

The converse problem is known to be undecidable. In fact, Valiant [50] showed that already trace universality is undecidable for nondeterministic OCAs (see Corollary 6.6 in Section 6.1). We now study the restriction to OCNs. The decidability of trace inclusion between a NFA and a OCN has been shown by Jančar, Esparza, and Moller [25], even for the more general class of Petri nets. The proof uses a well-quasi-order based saturation method that basically determinizes the net on the fly. This yields a non-primitive recursive procedure with space requirements bounded in terms of the longest antichains in an application of Higman's lemma. We will show that for one-counter nets, this naïve approach is optimal: The problem NFA $\subseteq$ OCN is Ackermanian, i.e., lies at exactly level ω in the fast-growing hierarchy [13] and the mentioned algorithm achieves this upper bound.

We first observe that one can without much effort reduce the problem NFA $\subseteq$ OCN to the trace universality problem for OCNs. We now prove this for OCNs, but in fact the same construction works for any class of systems that is closed under synchronous products with finite systems.

Lemma 6.13. *Trace inclusion NFA $\subseteq$ OCN is logspace-reducible to trace universality of OCNs.*

Proof. Let $\mathcal{A} = (Q, \mathrm{Act}, \delta)$ be an NFA and $\mathcal{A}' = (Q', \mathrm{Act}, \delta)$ be a OCN. We consider the OCN $\mathcal{B}$ with states $\{U\} \cup (Q \times Q')$ and actions $\mathrm{Act}' = \delta$ that has a transition $(p, p') \xrightarrow{(p,a,q),d} (q, q')$ for

every $(p,a,q) \in \delta$ and $(p',a,d,q') \in \delta'$. The OCN $\mathcal{B}$ moreover contains transitions $U \xrightarrow{t,0} U$ for all actions $t \in \delta$ and $(p,p') \xrightarrow{(q,a,q'),0} U$ for all $p = q \in Q$ and $p',q' \in Q'$.

Let $wa \in \delta^+$ be a witness for non-universality of some configuration $(p,p')m$ of $\mathcal{B}$. This means any run of $\mathcal{B}$ that is labelled with w must end in a configuration $(q,q')n$ where $q \xrightarrow{a}$ and $q'n \not\xrightarrow{a}$, and so, $w \in \mathcal{T}(p) \setminus \mathcal{T}(p'm)$. Conversely, if w witnesses $p \not\subseteq pm$ and $\pi = t_0 t_1 \dots t_k \in \delta$ is a w-labelled run of $\mathcal{A}$, then π witnesses non-universality of the process $(p,p')m$ of $\mathcal{B}$. □

We now present a simple nondeterministic algorithm that is based on coverability trees and solves the trace universality problem for OCNs.

Let $\mathbb{N}_\perp$ denote the set of non-negative integers plus a special least element $\perp$ and define $\max : 2^{\mathbb{N}}_\perp \to \mathbb{N}_\perp$ as the function that returns the maximal element of any nonempty finite subset and $\perp$ otherwise. We lift the definition of traces to sets $S \subseteq Q \times \mathbb{N}$ of processes in the natural way: The *traces* of S are $\mathcal{T}(S) = \bigcup_{qn \in S} \mathcal{T}(qn)$. By the monotonicity of trace inclusion (Lemma 3.13), the traces of a finite set of processes are determined only by the traces of its maximal elements.

Definition 6.14. For a finite set $S \subseteq Q \times \mathbb{N}$ define the *macrostate* as the vector $M_S \in \mathbb{N}^k_\perp$ of dimension $k = |Q|$ where for each $0 \le i < k$, $M_S(i) = \max\{n \mid q_i n \in S\}$. In particular, the macrostate for a singleton set $S = \{q_i n\}$ is the vector with value n at the i-th coordinate and $\perp$ on all others. The norm $|M|$ of M is the value in its biggest dimension. We define a step relation $\overset{a}{\Longrightarrow}$ for all $a \in \mathrm{Act}$ on the set of macrostates as follows:

$$(n_0,n_1,\dots,n_k) \overset{a}{\Longrightarrow} (m_0,m_1,\dots,m_k) \tag{6.4}$$

iff for all indices $0 \le i < k$, $m_i = \max\{n \mid \exists n_j \neq \perp.\ q_j n_j \xrightarrow{a} q_i n\}$. The *traces* of macrostate M are $\mathcal{T}(M) = \bigcup_{0 \le i \le k} \mathcal{T}(q_i m_i)$. For two macrostates M,N we say M is *covered* by N and write $M \sqsubseteq N$, if it is pointwise smaller, i.e., $M(i) \le N(i)$ for all indices $0 \le i < k$. For convenience, we will write $\{q_0 = n_0, q_1 = n_1, \dots, q_l = n_l\}$ to denote the macrostate with value n_i in dimension $q_i \in Q$ and $\perp$ for all unlisted dimensions.

Steps on macrostates correspond to the classical powerset construction and each macrostate represents the finite set of possible configurations the OCN can be in, where all non-maximal configurations are pruned out. The next lemma directly follows from these definitions and monotonicity (Lemma 3.13).

Lemma 6.15.

1. *The covering-order $\sqsubseteq$ is a well quasi order on $\mathbb{N}^k_\perp$, the set of all macrostates. Moreover, $M \sqsubseteq N$ implies $\mathcal{T}(M) \subseteq \mathcal{T}(N)$.*
2. *If $M \overset{a}{\Longrightarrow} N$ then $|N| \le |M| + 1$.*

3. *For any finite set* $S \subseteq Q \times \mathbb{N}$ *it holds that* $\mathcal{T}(S) = \mathcal{T}(M_S)$.

Dealing with macrostates allows us to treat universality as a reachability problem: By Lemma 6.15, point 3, we see that configuration qn is *not* trace universal, $\mathrm{Act}^* \neq \mathcal{T}(qn)$, if and only if $M_{\{qn\}} \Longrightarrow^* (\bot,\bot,\ldots,\bot)$. We take the perspective of a Pathfinder, who wants to reach $(\bot)^k$.

Point 1 of Lemma 6.15 allows us to use a naïve procedure to decide universality by exhaustively unfolding the coverability-tree of all reachable macrostates in search for a path to $(\bot)^k$, and thus a witness for non-universality. Whenever we see a macrostate that covers one of its ancestors, we can safely stop exploring this branch, because a minimal witness, if one exists, can omit the intermediate path.

To analyze the complexity of this procedure, we recall the definition of the fast-growing hierarchy and a result from [13], that allows us to compute a bound on the maximal number of incomparable macrostates visited.

Definition 6.16. Consider the family of functions $F_n : \mathbb{N} \to \mathbb{N}$ where for $x,k \in \mathbb{N}$,

$$F_0(x) = x+1 \quad \text{and} \qquad F_{k+1}(x) = F_k^{x+1}(x).$$

Here, F^k denotes the k-fold application of F. Moreover, define $F_\omega(x) = F_x(x)$ for the first limit ordinal ω. For $k \leq \omega$, $\mathfrak{F}_k$ denotes the least class of functions that contains all constants and is closed under substitution, sum, projections, limited recursion and applications of functions F_n for $n \leq k$.

Already $\mathfrak{F}_2$ contains all elementary functions and the union $\bigcup_{k\in\mathbb{N}} \mathfrak{F}_k$ of all finite levels contains exactly the primitive-recursive functions. A function is called *Ackermannian* if it is in $\mathfrak{F}_\omega \setminus \bigcup_{k\in\mathbb{N}} \mathfrak{F}_k$.

A sequence $x_0,x_1,\ldots,x_l$ of macrostates is called *good* if there are indices $0 \leq i < j \leq l$ such that $x_i \sqsubseteq x_j$ and *bad* otherwise. Such a sequence is called *t-controlled* by $f : \mathbb{N} \to \mathbb{N}$ if $|x_i| < f(i+t)$ for every index $0 \leq i \leq l$.

Theorem 6.17 ([13])**.** *Let* $f : \mathbb{N} \to \mathbb{N}$ *be a monotone function in* $\mathfrak{F}_\gamma$ *such that* $f(x) \geq \max\{1,x\}$ *for some* $\gamma \geq 1$. *There is a function* $L_{k,f}(t)$ *in* $\mathfrak{F}_{k+\gamma-1}$ *that computes a bound on the maximal length of bad sequences in* $\mathbb{N}_\bot^k$ *which are t-controlled by* f.

Corollary 6.18. *Trace universality of OCNs with no more than k states is in* $\mathfrak{F}_k$. *Trace universality for unresticted OCNs is in* $\mathfrak{F}_\omega$.

Proof. Observe that the time and space requirements to implement the naïve procedure outlined above are bounded in terms of the longest non-increasing (w.r.t. covering) sequence in this unfolding. These are bad sequences where the norm of the initial macrostate is n_0, the counter value of the process to check for universality. By point 2 of Lemma 6.15, all such sequences

are n_0-controlled by the successor function $succ(x) = x+1$, which is in $\mathfrak{F}_1$. By Theorem 6.17, one can compute a bound $L_{k,succ}(n_0)$ on the maximal length of minimal witnesses and thus implement the procedure in $\mathfrak{F}_k$, if the dimension of the macrostates – the number of states in the input net – is fixed to k.

For the general case, the number k of states of the OCN is part of the input. The bound on the maximal length of witnesses one needs to consider therefore not only depends on the initial counter value n_0 but also on k. This bound is $B(k,n_0) = L_{k,succ}(n_0)$. Sicne $L_{k,succ} \in \mathfrak{F}_k$ for every $k \in \mathbb{N}$, the function $B : \mathbb{N}\times\mathbb{N} \to \mathbb{N}$ must be in $\mathfrak{F}_\omega$. Computing the necessary bound and implementing the procedure can thus be done in $\mathfrak{F}_\omega$. □

Remark 6.19. In a similar fashion, we can determinize OCN with respect to their *weak* traces and derive an $\mathfrak{F}_\omega$ upper bound for the weak trace universality of OCNs, as well as the problem OCN $\subseteq$ OCN. In this setting, a OCN process either has only finitely many possible weak a-successors for a particular action a, in which case each such step is short (cf. Remark 5.6 on page 55), or for some control states q, there are weak a-steps to qn for unbounded counter values $n \in \mathbb{N}$.

The crucial observation is that if a weak trace can lead to some process qn for arbitrarily high n, then every extension of this trace that is a witness for non-universality must contain some action that is disabled in all qn. Therefore, one can use a symbolic value $\top > n \in \mathbb{N}$ when defining steps between macrostates: the dimension corresponding to control state q is set to $\top$ if the current prefix of the constructed witness can lead to unbounded qn, and this value is only removed (and set to $\bot$) when the candidate witness prescribes an action that is disabled in all qn. The natural extension of the well-order $\sqsubseteq$ on these generalized macrostates is again monotone with respect their weak traces (cf. Lemma 6.15), and implies a decision procedure.

At first glance, $\mathfrak{F}_\omega$ seems like a rather crude upper bound. However, we will show that this bound is optimal and strong and weak trace universality are indeed Ackermannian, i.e., not primitive recursive. Before we prove this lower bound (in Theorem 6.25), let us introduce some convenient notation.

Let $\mathcal{N} = (Q,\mathrm{Act},\delta)$ be a OCN that contains a *universal* state U: one that has self-loops $U \xrightarrow{a,0} U \in \delta$ for every action $a \in \mathrm{Act}$. Naturally, a Pathfinder who wants to prove non-universality must avoid macrostates with $M(U) \neq \bot$, because no continuation of a path leading to such a macrostate can be a witness. We can use this observation to construct macrostates that inhibit Pathfinder to make certain actions.

Definition 6.20. Let $S \subseteq \mathrm{Act}$ be a set of actions in a OCN that contains a universal state U. A state $q \in Q$ is called an *S-obstacle* if $q \xrightarrow{a,0} U \in \delta$ for all actions $a \in S$. We say q *ignores* S, if $q \xrightarrow{a,0} q \in \delta$ for all $a \in S$.

Note that if a macrostate contains an S-obstacle, then Pathfinder must avoid all actions of S. In order to remove an obstacle, Pathfinder must play an action that is not the label of any of its incoming transitions.

Example 6.21. We show how to construct a one-counter net $\mathcal{N}_k$, parametrized with $k \in \mathbb{N}$, together with a process that is not trace universal, but any witness for non-universality must be longer than $F_k(1)$.

We define the net $\mathcal{N}_k = (Q, \mathrm{Act}, \delta)$ with states $Q = \{I, U, A\} \cup \{F_i \mid 0 \le i \le k\}$ and actions $\mathrm{Act} = \{0, 1, \ldots, k\} \cup \{e\}$. The set δ of transitions contains

$$U \xrightarrow{a,0} U \qquad \text{for all } a \in \mathrm{Act}, \tag{6.5}$$

$$F_i \xrightarrow{i,-1} F_i \qquad \text{for all } 0 \le i \le k, \tag{6.6}$$

$$F_i \xrightarrow{j,0} U \qquad \text{for all } 0 \le i < j \le k, \tag{6.7}$$

$$F_i \xrightarrow{e,0} U \qquad \text{for all } 0 \le i < k, \tag{6.8}$$

$$A \xrightarrow{(i+1),0} F_i \qquad \text{for all } 0 \le i < k, \tag{6.9}$$

as well as $A \xrightarrow{0,+1} A$, $I \xrightarrow{e,0} F_k$ and $I \xrightarrow{e,0} A$. See Figure 6.1 below for the net $\mathcal{N}_3$.

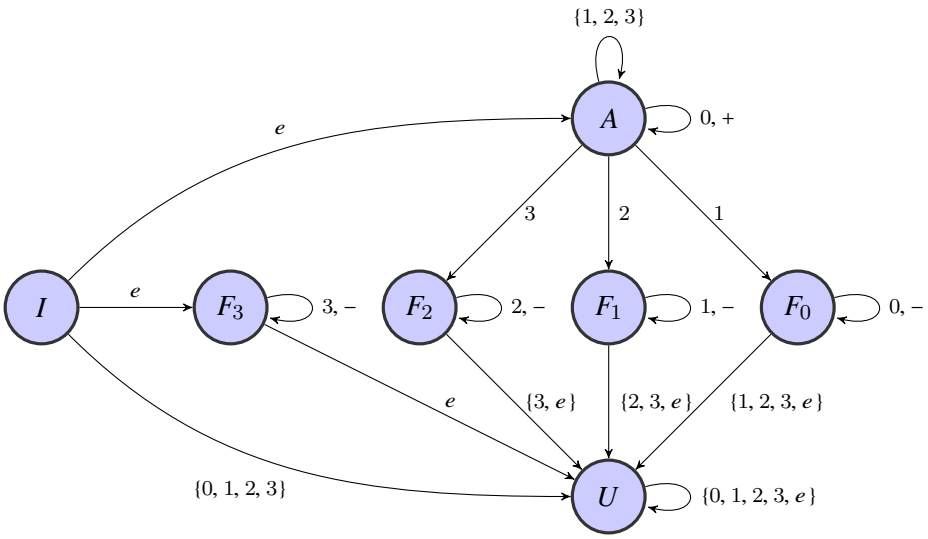


Figure 6.1: The net $\mathcal{N}_3$. Edges labelled with sets of actions indicate multiple transitions, one for each action in the set. The process $I(1)$ is not trace universal but any witness for non-universality is longer than $F_3(1)$.

The interesting property of the net $\mathcal{N}_k$ is that for all $m, n \in \mathbb{N}$, the macrostate $\{A = m, F_k = n\}$ is not universal, but has no witnesses shorter than $F_k^n(m)$.

Intuitively, in order to show non-universality, a Pathfinder must play the "end"-action e eventually, in order to remove the accumulator state A, i.e., exclude the possibility that the net is in configuration $A(n)$ for some n. For this to be safe, all occurrences of the F_i must first be

removed, because they all are e-obstacles. Removing $\{F_i = n\}$ amounts to playing the action i at least $n+1$ times, which in turn requires first removing all F_j for $j < i$. But every action i respawns another m instances of F_{i-1}, where m is the current value of the accumulator A, which gets incremented every time the action 0 is played. This means a shortest witness for non-universality of $\{A = m, F_i = n\}$ must faithfully recur according to the definition of $F_i^n(m)$, and thus count up the accumulator to at least $F_k^n(m)$.

To find a single non-universal configuration with the mentioned lower bound on the length of witnesses, consider the process $I(1)$. The initial state I has transitions $I \xrightarrow{e,0} F_k, I \xrightarrow{e,0} A$ and $I \xrightarrow{i,0} U$ for all actions $i \neq e \in \mathrm{Act}$. This makes it an $\mathrm{Act} \subseteq \{q\}$-obstacle and hence, any witness for non-universality of $I(1)$ must start with action e to avoid reaching a universal macrostate. This action leads to $\{A = 1, F_i = 1\}$, which has no witnesses shorter than $F_i(1)$.

We now show that trace universality for OCNs is not primitive recursive, by reduction from the control state reachability problem in incrementing counter machines, which is Ackermanian-hard [11, 13]. An *incrementing* counter machine is a counter machine (cf. Definition 2.2 on page 8) with imprecise counter-behaviour: It has the possibility to spontaneously increment any of its counters.

Definition 6.22. A *k-dimensional incrementing counter machine (k-ICM)* is a triple $\mathcal{M} = (Q, \mathrm{Act}, \delta)$ where Q is a finite set of control states, Act is a finite alphabet of *actions* and δ is a transition relation of the form $\delta \subseteq Q \times \mathrm{Act} \times OP \times Q$, where $OP = \{\mathrm{inc_i}, \mathrm{dec_i}, \mathrm{ifz_i} \mid 0 < i \leq k\}$ are the possible operations on the counters.

A *configuration* $(q, c_1, c_2, \ldots, c_k) \in Q \times \mathbb{N}^k$ of $\mathcal{M}$ consists of a state and a valuation of the counters. $\mathcal{M}$ can make an a-labelled step

$$(q, c_1, c_2, \ldots, c_k) \xrightarrow{a} (q', c_1', c_2', \ldots, c_k') \tag{6.10}$$

from one configuration to another, if there is a transition $(q, a, \mathrm{op}, q') \in \delta$ and the following are satisfied.

1. If $\mathrm{op} = \mathrm{inc_i}$ then $c_i' \geq c_i + 1$ and $c_j' \geq c_j$ for all $j \neq i$.
2. If $\mathrm{op} = \mathrm{dec_i}$ then $c_i' \geq c_i - 1 \geq 0$ and $c_j' \geq c_j$ for all $j \neq i$.
3. If $\mathrm{op} = \mathrm{ifz_i}$ then $c_i' \geq c_i = 0$ and $c_j' \geq c_j$ for all $j \neq i$.

Control state reachability is the decision problem that asks if there is a run of the ICM from a given initial configuration to some configuration in a given final state. This problem has non-primitive recursive complexity [11, 13].

Remark 6.23. Syntactically, there is no difference between incrementing counter machines and ordinary counter machines. Semantically, the difference is that a ICM can always set the value

of a counter higher than prescribed by the current transition. ICM therefore induce infinitely branching transition systems.

Remark 6.24. Incrementing counter machines are monotone with respect to counter valuations: If a step $(q,c_1,c_2,\ldots,c_k) \xrightarrow{a} (q',c_1',c_2',\ldots,c_k')$ is possible, then so is $(q,d_1,d_2,\ldots,d_k) \xrightarrow{a} (q',c_1',c_2',\ldots,c_k')$ for all pointwise smaller counter valuations $(d_1,d_2,\ldots,d_k) \sqsubseteq (c_1,c_2,\ldots,c_k)$. This is because a transition that decreases counter i is enabled even if the value of the ith counter is 0. Since ICMs can contain zero-testing transitions that are enabled *only* if the corresponding counter value is 0, a smaller valuation usually *strictly* simulates any larger one.

Theorem 6.25. *Trace universality for nondeterminisic OCNs is non-primitive recursive.*

Proof. By reduction from the control state reachability for incrementing counter machine, which is non-primitive recursive [11]. We construct a OCN-process that it is *not* universal iff a given configuration of a k-ICM can reach a given final state. The idea is to enforce a faithful simulation of the ICM by Pathfinder, who wants to show non-universality of the OCN process.

We construct a net $\mathcal{N}$ which has a unique action for every transition of the ICM, as well as actions τ_i that indicate incrementing errors for every counter c_i, and actions ♯ and \$ to mark the beginning and end of a run respectively. This ensures a strict correspondence between words and runs of the ICM. The states of $\mathcal{N}$ are

1. an initial state *Init* and a universal state U,
2. a state q_i for every state q_i of the ICM.
3. a state C_i for every $0 < i \geq k$, representing the ith counter of the ICM.
4. a state Z, that will be used to access the constant 0 and that ignores every action but the end marker \$.

A configuration $(q,c_1,c_2,\ldots,c_k)$ of the ICM is represented by a macrostate $\{q = 0,Z = 0,C_1 = c_1,C_2 = c_2,\ldots,C_k = c_k\}$. The Pathfinder will announce transitions of the ICM (as well as actions demanding increment errors) in order to move between macrostates according to a faithful simulation of the ICM.

Initialization. To set up $M_0 = \{q_0 = 0,Z = 0,C_0 = 0,C_1 = 0,\ldots,C_k = 0\}$, representing the initial ICM configuration, we add ♯-labelled transitions with effect 0 from *Init* to q_0,Z and C_i for all $0 \leq i \leq k$. Moreover, we make *Init* an obstacle for every action but ♯. This way, Pathfinder must initially play the action ♯ (and set up M_0) in order to avoid a universal macrostate.

Finite control. For any transition $t = q \xrightarrow{a,\mathrm{op}} q'$ of the ICM, we add a transition $q \xrightarrow{t,0} q'$ to $\mathcal{N}$ that, in a macrostate-step, will replace the value 0 in dimension q by $\perp$ and introduce value 0

in dimension q'. Moreover, we make every state q an obstacle for all actions announcing ICM-transitions not originating in q. This prevents Pathfinder from announcing transitions from q unless the current macrostate has $M(q) = 0$ and $M(q_i) = \bot$ for all $q_i \neq q$.

Simulation of the counters. Every transition operates on one of the counters $0 \leq i \leq k$. Below we list the corresponding transitions in the OCN $\mathcal{N}$ for this counter. Every state of $\mathcal{N}$ not explicitly mentioned ignores the action in question. In the macrostate, the value of these states are therefore unchanged.

inc_i For every ICM-transition t that increases the ith counter, $\mathcal{N}$ contains a t-labelled transition from state C_i to C_i with effect $+1$. Additionally, to deal with spontanious increment errors, there is a τ_i-labelled increasing self-loop in state C_i. All other states ignore the action τ_i.

dec_i For ICM-transitions t that decreases the ith counter, $\mathcal{N}$ contains a t-labelled transition from state C_i to C_i with effect -1. This means that the next macrostate M could lose the value for this counter and have $M(C_i) = \bot$ if previously, the value was 0. Note that the decrementing step from value 0 to value 0 is valid in the ICM. In order to avoid losing the state C_i in the macrostate, the OCN contains a transition $Z \xrightarrow{t,0} C_i$ from the constant-zero state Z to state C_i. This way, all correctly set up macrostates will never have $M(C_i) = \bot$.

ifz_i For every ICM-transition t that test the ith counter for 0, we add a t-labelled transition $C_i \xrightarrow{t,-1} U$ from state C_i to the universal state. This prevents Pathfinder from using these actions if the current macrostate has $M(C_i) > 0$ because it would make the next macrostate universal. If however $M(C_i) = 0$, such a step is safe because the punishing transition is not enabled in the OCN-process $C_i 0$.

Lastly, we only need to add transitions to $\mathcal{N}$ so that the final state q_f is the only original ICM-state which is not an obstacle for \$. This prevents Pathfinder from playing the end-action \$ unless the simulation has reached the final state. □

Chapter 7

Semantic Preorders Between Deterministic Systems

For deterministic systems, simulation and trace inclusion coincide. In fact, trace inclusion ($\subseteq_{\mathcal{A},\mathcal{A}'}$) and simulation ($\preceq_{\mathcal{A},\mathcal{A}'}$) relative to OCAs $\mathcal{A}$ and $\mathcal{A}'$ already coincide if only $\mathcal{A}'$ is deterministic (cf. Section 2.2.2 on page 11). We now focus on the problem OCA $\subseteq$ DOCA, trace inclusion between one-counter systems where the process on the right is deterministic. This problem is undecidable (Corollary 6.5), so we will have to further restrict the input to regain decidability. Our main focus lies on deterministic one-counter nets (DOCNs) but later, in Section 7.2, we will consider inclusion between deterministic OCAs and OCNs, so let's first consider otherwise unrestricted systems in order to introduce some notation.

Due to the lack of nondeterministic choice in the system on the right, OCA $\subseteq$ DOCA can be seen as a reachability problem in the product of the two systems. We ask if from a given initial pair of processes, some pair $(qn,q'n')$ is reachable such that both $qn \xrightarrow{a}$ and $q'n' \not\xrightarrow{a}$ for some action label a. This termination criterion can be verified locally in constant time.

By Lemma 3.10, we can assume that in fact, both input systems are deterministic and moreover, the DOCA on the right is complete. That is, we want to check if $pm \subseteq p'm'$ holds for proceses pm and $p'm'$ of a DOCA $\mathcal{A} = (Q,\mathrm{Act},\delta,\delta_0)$ and a complete DOCA $\mathcal{A}' = (Q',\mathrm{Act},\delta',\delta'_0)$, respectively. Any trace $w \in \mathrm{Act}^*$ of pm uniquely determines a path in the product of the two automata and vice versa. We therefore identify witnesses for non-inclusion with the corresponding paths they induce in the product.

Definition 7.1. Assume processes pm and $p'm'$ of a DOCA $\mathcal{A}$ and a complete DOCA $\mathcal{A}'$ respectively. A *witness for* $pm \not\subseteq p'm'$ is a path π in the product of $\mathcal{A}$ and $\mathcal{A}'$ such that $(pm,p'm') \xrightarrow{\pi} (qn,q'n')$ and for some action $a \in \mathrm{Act}$, $qn \xrightarrow{a}$ but $q'n' \not\xrightarrow{a}$.

Since $\mathcal{A}'$ is complete, any witness for $pm \not\subseteq p'm'$ exhausts the counter in the process of $\mathcal{A}'$, i.e., it takes $(pm,p'm')$ to some pair of configurations $(qn,q'0)$. This is because a process

of a complete OCA can only *not* make an a-step in case the counter is empty.

We now further restrict the input and consider the problem DOCN $\subseteq$ DOCN, trace inclusion between processes of a DOCN $\mathcal{A}$ and a complete DOCN $\mathcal{A}'$. In Section 7.2 we will later see how to generalize our technique to the case where only one side forbids zero-tests.

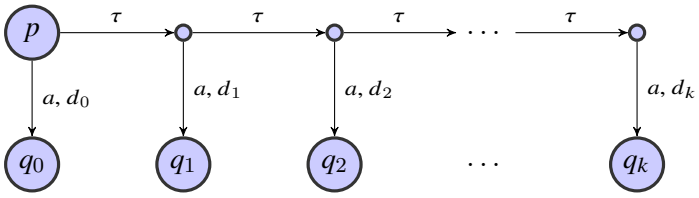
7.1 Inclusion for Deterministic One-Counter Nets

For deterministic one-counter nets, a PSPACE upper bound for checking trace inclusion follows from our previous result for simulation (Theorem 4.19). Similar problems have already been considered by Higuchi, Wakatsuki, and Tomita, who compared the classes of languages defined by DOCNs with various acceptance modes [18] and in a series of papers [17, 15, 16] studied the respective inclusion problems. In each case, they showed bounds on the length of shortest witnesses for non-inclusion that are polynomial in the size of the given automata and the initial counter values. They derived procedures that exhaustively search for a witness and work in time and space polynomial in the size of the automata, assuming fixed initial counter values. We now consider the trace inclusion problem DOCN $\subseteq$ DOCN, where the initial counter values are part of the input and given in binary.

7.1.1 Silent Steps and Nondeterminism

We assume here what in formal language theory are sometimes called *realtime* automata: those that do not have ε-labelled transitions, which silently advance a computation without prolonging the trace it prescribes. This is safe because of the customary definition of deterministic (pushdown) automata, that ensures that not only the induced LTSs, but their weak closures (abstracting ε-steps) have to be deterministic. The usual syntactic restriction for DPDAs that no state with outgoing ε-transitions may have outgoing transitions labelled by $a \neq \varepsilon$, together with the monotonicity of steps in OCNs implies that all states on ε-cycles must be deadlocks. This means one can eliminate ε-transitions by first removing ε-cycles, replacing every remaining short path that contains an ε step by a uniquely labelled new transition, and then normalizing the effects of single transitions to $\{-1,0,1\}$. Such a reduction works in $O(\log n)$ space (see the appendix of [27] for details). Allowing ε-transitions as in DPDAs therefore does not change the complexity of trace inclusion.

However, using τ-labelled transitions, that are not subject to the syntactic restriction for ε-steps in DPDAs, one can easily simulate nondeterministic choice. That is, one can construct a DOCN which induces a deterministic LTS, but its weak closure is isomorphic to the LTS induced by a given nondeterministic OCN. For example, the choice introduced by transitions $\{(p,a,d_0,q_0),(p,a,d_1,q_1),\ldots,(p,a,d_k,q_k)\}$ can be simulated by the following chain.



Therefore, by Theorem 6.10, *weak* trace inclusion DOCN $\subsetneqq$ DOCN and equivalence as well as weak trace universality for DOCA (Corollary 6.6) are undecidable.

7.1.2 Reachability in Vector Addition Systems

Before we go on and show how to solve DOCN $\subseteq$ DOCN let us point out that this problem is actually closely related to the reachability problem for vector addition systems.

Lemma 7.2. *The problem VASS $\subseteq$ DOCN is logspace reducible to the reachability problem of VASS. More precisely, given a k-dimensional VASS $\mathcal{A} = (Q,Act,\delta)$ and a DOCN $\mathcal{A}' = (Q',Act,\delta')$, one can construct an only polynomially larger, $(k+1)$-dimensional VASS $\mathcal{B}$ such that for all $p \in Q$, $p' \in Q'$ and $m_1,m_2,\ldots,m_k,m \in \mathbb{N}$, there are states init,halt of $\mathcal{B}$ such that*

$$p(m_1,m_2,\ldots,m_k) \not\subseteq p'm' \iff init(m_1,m_2,\ldots,m_k,m) \longrightarrow^* halt(0,0,\ldots,0). \tag{7.1}$$

Proof. Assume w.l.o.g., that the OCN $\mathcal{A}'$ is complete (the construction for Lemma 3.10 can be used). Define $\mathcal{B}$ as the product of $\mathcal{A}$ and $\mathcal{A}'$: it has states-set $S \subseteq Q \times Q'$ and a transition $(p,p') \xrightarrow{(t,t'),D,d} (q,q')$ for all $t = (p,a,D,q) \in \delta$ and $t' = (p',a,d,q') \in \delta'$. Since $\mathcal{A}'$ is complete, every witness for non-inclusion leads to some pair $q(n_1,n_2,\ldots,n_k),q'0$ of processes of $\mathcal{A}$ and $\mathcal{A}'$ respectively, such that $q(n_1,n_2,\ldots,n_k) \xrightarrow{a}$ but $q'0 \not\xrightarrow{a}$. There are only finitely many transitions $t = (q,a,D,r) \in \delta$ in $\mathcal{A}$ that enable this winning step. For each $q \in Q$, $q' \in Q'$ and $t \in \delta$ as above, we add a new transition

$$(q,q') \xrightarrow{Stop(t),D,0} E \tag{7.2}$$

to the VASS $\mathcal{B}$, where $Stop(t)$ is a fresh symbol and E is the (unique) halting state of $\mathcal{B}$. Finally, add transitions $E \xrightarrow{\$,C_i} E$ for every $0 < i \leq k+1$, where $C_i \in \{0,-1\}^{k+1}$ is the vector where only the ith component is -1. The constructed VASS $\mathcal{B}$ now satisfies Equation (7.1) for every $p \in Q$ $p' \in Q'$ and $m_1,m_2,\ldots,m_k,m' \in \mathbb{N}$, where $init = (p,p')$ and $halt = E$. □

Lemma 7.3. *The reachability problem of VASSs is logspace reducible to VASS $\subseteq$ DOCN. More precisely, for a given k-dimensional VASS $\mathcal{B} = (B,Act,\delta)$, together with initial and terminal configurations $p(m_1,m_2,\ldots,m_k)$ and $q(n_1,n_2,\ldots,n_k)$, one can construct a k-dimensional VASS $\mathcal{A}$ and a DOCN $\mathcal{A}'$, both with state-sets $B \cup \{E\}$, such that*

$$p(m_1,m_2,\ldots,m_k) \not\subseteq pm \iff p(m_1,m_2,\ldots,m_k,m) \longrightarrow^* q(n_1,n_2,\ldots,n_k). \tag{7.3}$$

Proof. We let $\mathcal{A}$ and $\mathcal{A}'$ be copies of the original system $\mathcal{B}$, where every transition is labelled by some unique action symbol. $\mathcal{A}$ tracks the effect of an original transition precisely, and the effect of a transition in $\mathcal{A}'$ is the sum of the effects of the corresponding transtion in $\mathcal{B}$. This way, any path $t_1 t_2 \dots t_l$ from the initial process in $\mathcal{B}$ determines a trace of both processes $p(m_1,m_2,\dots,m_k)$ in $\mathcal{A}$ and $p(m_1+m_2+\dots+m_k)$ in $\mathcal{A}'$ and vice versa.

Finally, add new states E,R to $\mathcal{A}$, and transitions $q \xrightarrow{\$,-(n_1,n_2,\dots,n_k)} E \xrightarrow{\$,0,0,\dots,0} R$, where \$ is a fresh symbol. In $\mathcal{A}'$, we add states E,R and transitions $q \xrightarrow{\$,-(n_1+n_2+\dots+n_k)} E \xrightarrow{\$,-1} R$.

Notice that any path $p(m_1,m_2,\dots,m_k,m) \longrightarrow^* q(n_1,n_2,\dots,n_k)$ in $\mathcal{B}$ prescribes a trace w of $\mathcal{A}$ and $\mathcal{A}'$, that leads to processes $q(n_1,n_2,\dots,n_k)$ and $q(n_1+n_2+\dots+n_k)$, respectively. The word $w\$\$$ then witnesses non-inclusion. Conversely, if $p(m_1,m_2,\dots,m_k) \not\subseteq p(\sum_{1\le i\le k} m_i)$, then there must be a witness ending in \$\$, which means that the process $q(n_1,n_2,\dots,n_k)$ was reached because the witnessing trace leads to a deadlock in $\mathcal{A}$. □

The previous two lemmas show that trace inclusion VASS ⊆ DOCN is essentially the same problem as VASS reachability. For the one-dimensional case, this connection seems not very exciting: after all, Lemma 7.3 only allows to reduce the already NL-complete reachability problem for OCN (Theorem 3.8) to OCN ⊆ DOCN. In the other direction, Lemma 7.2 allows to reduce OCN ⊆ DOCN to reachability in 2-dimensional VASS, which has a known double exponential time upper bound in the number of transitions [23]. However, a PSPACE upper bound already follows from the fact that strong simulation checking is decidable in polynomial space (Theorem 4.19), because simulation and trace inclusion coincide for deterministic systems. An interesting observation is that both reductions also work for VASS with additional zero-testable counters. Hence, trace inclusion between a VASS with one zero-testable counter and a DOCN corresponds to the reachability problem for VASSs with one zero-testable counter, which is decidable [44, 7].

7.1.3 Characterizing Witnesses

We now characterize the form of possible witnesses for non-inclusion between DOCN processes. This characterization will be sufficient in the sense that if any witness exists at all, then there is also one in our prescribed form. In the next subsection, this will guide our (nondeterministic) procedure that guesses and verifies witnesses.

Since we are looking at trace inclusion between one-counter *nets*, witnesses for non-inclusion are monotone with regards to counter values. The next lemma states this formally. It follows easily from the monotonicity of the steps in OCN-processes. As before, we always assume w.l.o.g. that the two given automata are in normal form (cf. Definition 3.9 and Lemma 3.10).

Lemma 7.4 (Witness Monotonitiy)**.** *Let $\mathcal{A} = (Q,Act,\delta)$ and $\mathcal{A}' = (Q',Act,\delta')$ be two DOCNs in normal form and pm, and $p'm'$ processes of $\mathcal{A}$ and $\mathcal{A}'$, respectively. Then for all $l \in \mathbb{N}$,*

every witness for $pm \not\subseteq p'm'$ is a witness for $p(m+l) \not\subseteq p'm'$

Intuitively, finding a witness is a kind of coverability question due to the monotonicity of steps in OCNs: Any witness for $pm \not\subseteq p'm'$ must be enabled, in particular in the process pm on the left, and moreover exhaust the counter in the process of the complete net on the right. It takes the initial pair of processes to some pair $(qn, q'0)$ where the exact value n is not important. Since any sufficiently long path in the product will revisit control states, we can compare paths with respect to their effect on the counters and see that some are "better" than others in the sense that they have a smaller effect on the counter on the left. For instance, a cycle that only increments the counter on the right and decrements the one on the left is surely a suboptimal choice when looking for a shortest witness, because omitting such a cycle leads to a shorter witness by monotonicity (Lemma 7.4).

The characterization Theorem 7.6 below states that if a positive witness exists, then there is also one that, apart from short paths, combines only the most productive cycles and in a sensible way. It is very similar to Lemma 3.7 that characterizes positive paths in OCA.

Definition 7.5. Let π be a path in the product of $\mathcal{A}$ and $\mathcal{A}'$. The *slope* of π is the ratio $S(\pi) = \Delta(\pi)/\Delta'(\pi)$ of its effects on the counter of $\mathcal{A}$ and $\mathcal{A}'$ respectively, where for $n > 0$ and $k \in \mathbb{Z}$ we let $n/0 = \infty > k$, $0/0 = 0$ and $-n/0 = -\infty < k$. We partition the set of paths into four disjoint types: $(<,<),(>,\geq),(\leq,\geq)$, and $(\geq,<)$. The *type* of π is $Type(\pi) = (\blacktriangleleft,\blacktriangleright)$ iff $\Delta(\pi) \blacktriangleleft 0$ and $\Delta'(\pi) \blacktriangleright 0$.

Recall (Definition 3.4) that a *loop* is a non-empty cyclic path in the product such that none of its proper subpaths is a cycle. No loop is longer than $K = |Q \times Q'|$ because it visits exactly one node twice.

Theorem 7.6. *Let $\mathcal{A}$ and $\mathcal{A}'$ be DOCNs in normal form, with state-sets Q and Q', respectively and let $K = |Q \times Q'|$. There is a bound $c \in \mathbb{N}$ that depends polynomially on K, such that the following holds for all $p \in Q$, $p' \in Q'$ and $m,m' \in \mathbb{N}$.*

If there is a witness for $pm \not\subseteq p'm'$ then there is one that is either no longer than c or has one of the following forms:

1. *$\pi_0 L_0^{l_0} \pi_1$, where L_0 is a loop of type $(\geq,<)$ and π_0,π_1 are no longer than c,*
2. *$\pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \pi_2$, where L_0 and L_1 are loops of type $(>,\geq)$ and $(<,<)$ with $S(L_0) > S(L_1)$ and π_0,π_1,π_2 are no longer than c,*
3. *$\pi_0 L_0^{l_0} \pi_1$, where L_0 is a loop of type $(<,<)$ and π_0,π_1 are no longer than c,*

where in all cases, the number of iterations $l_0,l_1 \in \mathbb{N}$ are polynomial in K and the initial counter value m' of the given processes.

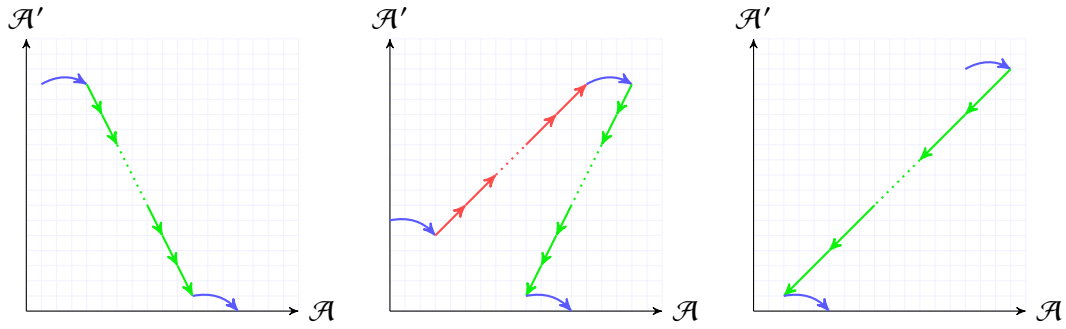


Figure 7.1: Illustration of the changes in counter values during paths of forms 1, 2 and 3 as given in Theorem 7.6. The bent blue arrows indicate the effect of short paths π_i, the red and green arrows indicate the counter-effect of loops that are iterated.

Note that the bound c in the claim of Theorem 7.6 depends only on the number of states of the given nets and not on the size of the action alphabet.

Example 7.7. Consider two systems that each consist only of a single, a-labelled self-loop: $p \xrightarrow{a,0} p$ and $p' \xrightarrow{a,-1} p'$ repectively. The product of these nets has only one edge, namely $L = (p,p') \xrightarrow{a,0,-1} (p,p')$, which is a loop with effects $\Delta(L) = 0$ and $\Delta'(L) = -1$, type $(\geq,<)$ and slope ∞. The only witness for $pm \not\subseteq p'm'$ is $\pi = L^{m'}$, which is a path of form 1.

The remainder of this section is devoted to the proof of Theorem 7.6. We show that it is safe to consider only paths in a reduced form, that allows to derive bounds on the length of certain subpaths. For this, we introduce path rewriting rules that exchange occurrences of some loops by others. We then show (in Lemma 7.9) that these rules preserve witnesses and (in Lemma 7.10) cannot be applied indefinitely. For *reduced* witnesses, those to which no rules are applicable, we derive (Lemma 7.12) bounds on the multiplicities of loops that are less productive than others, which will finally enable us to prove Theorem 7.6.

We start with an easy observation: No loop L is longer than K, so there are only $F_0 := (2K)^2$ possible different values for the pair $(\Delta(L),\Delta'(L))$ of effects. Moreover, if a witness for $(pm,p'm')$ exists, then there is also one that does not contain different loops with the same effects: If $\pi_0 L_0 \pi_1 L_1 \pi_2$ is a witness and $L_0 \neq L_1$ are two loops with $(\Delta(L_0),\Delta'(L_0)) = (\Delta(L_1),\Delta'(L_1))$, then either $\pi_0 L_0^2 \pi_1 \pi_2$ (if $\Delta(L_0) \geq 0$) or $\pi_0 \pi_1 L_1^2 \pi_2$ (if $\Delta(L_0) < 0$) must also be a witness by Lemma 7.4. We can therefore consider only paths of the form

$$\pi = \pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \dots \pi_r L_r^{l_r} \pi_{r+1} \tag{7.4}$$

where $r < F_0$, all π_i are acyclic and all loops have pairwise different effects. We call a positive path *sane* if it is of this form.

Definition 7.8. Consider the rules given below.

$$\textit{UUL}\quad \frac{\begin{array}{l}\pi = \pi_0 L_0^{l_0}\pi_1 L_1^{l_1}\pi_2\\ \textit{Type}(L_0) = (>,\geq)\\ \textit{Type}(L_1) = (>,\geq)\\ \Delta'(L_0)\cdot x = \Delta'(L_1)\cdot y\\ S(L_0)\geq S(L_1)\\ l_1 - y > 0\end{array}}{\rho = \pi_0 L_0^{l_0+x}\pi_1 L_1^{l_1-y}\pi_2}$$

$$\textit{UUR}\quad \frac{\begin{array}{l}\pi = \pi_0 L_0^{l_0}\pi_1 L_1^{l_1}\pi_2\\ \textit{Type}(L_0) = (>,\geq)\\ \textit{Type}(L_1) = (>,\geq)\\ \Delta'(L_0)\cdot x = \Delta'(L_1)\cdot y\\ S(L_0) < S(L_1)\\ l_0 - x > |\pi_1 L_1|\end{array}}{\rho = \pi_0 L_0^{l_0-x}\pi_1 L_1^{l_1+y}\pi_2}$$

$$\textit{UD}\quad \frac{\begin{array}{l}\pi = \pi_0 L_0^{l_0}\pi_1 L_1^{l_1}\pi_2\\ \textit{Type}(L_0) = (>,\geq)\\ \textit{Type}(L_1) = (<,<)\\ \Delta'(L_0)\cdot x = -\Delta'(L_1)\cdot y\\ S(L_0)\leq S(L_1)\\ l_0 - x \geq |\pi_1|\\ l_1 - y > 0 \wedge l_0 - x > 0\end{array}}{\rho = \pi_0 L_0^{l_0-x}\pi_1 L_1^{l_1-y}\pi_2}$$

$$\textit{DDL}\quad \frac{\begin{array}{l}\pi = \pi_0 L_0^{l_0}\pi_1 L_1^{l_1}\pi_2\\ \textit{Type}(L_0) = (<,<)\\ \textit{Type}(L_1) = (<,<)\\ \Delta'(L_0)\cdot x = \Delta'(L_1)\cdot y\\ S(L_0) < S(L_1)\\ l_1 > |L_0|\cdot x + 2|\pi_1|\\ l_1 - y > 0\end{array}}{\rho = \pi_0 L_0^{l_0+x}\pi_1 L_1^{l_1-y}\pi_2}$$

$$\textit{DDR}\quad \frac{\begin{array}{l}\pi = \pi_0 L_0^{l_0}\pi_1 L_1^{l_1}\pi_2\\ \textit{Type}(L_0) = (<,<)\\ \textit{Type}(L_1) = (<,<)\\ \Delta'(L_0)\cdot x = \Delta'(L_1)\cdot y\\ S(L_0)\geq S(L_1)\\ l_0 - x > 0\end{array}}{\rho = \pi_0 L_0^{l_0-x}\pi_1 L_1^{l_1+y}\pi_2}$$

Each rule consists of *conditions* (lines above the bar) and a conclusion ρ, which is a path, below the bar. Their names indicate which type of loops are handled: E.g., *UUL* exchanges loops of type $(>,\geq)$ (up) for others of the same type to its left.

We say that a rule is *applicable* to a sane path π if there are $0 < x,y,l_0,l_1 \in \mathbb{N}$ and two different loops L_0 and L_1 such that all conditions are satisfied. In this case the rule can rewrite π to ρ, its conclusion and we say ρ is the result of applying the rule to π.

Lemma 7.9. *Let π be a sane witness for $pm \not\subseteq p'm'$ and ρ be the result of applying one of the rules to π. Then ρ is a sane witness for $pm \not\subseteq p'm'$.*

Proof. Each rule only modifies the number of times some loops are iterated, and never completely removes a loop. Therefore, sane paths are always rewritten to other sane paths.

Let's say we rewrite $\pi = \pi_0 L_0^{l_0}\pi_1 L_1^{l_1}\pi_2$ to ρ. The key observation is that the conditions of the rule imply that we can always decompose the paths π and ρ into $\pi = \alpha\gamma$ and $\rho = \beta\gamma$, such that $\Delta'(\alpha) = \Delta'(\beta)$ and $\Delta(\alpha) \leq \Delta(\beta)$. By monotonicity (Lemma 7.4) and the assumption that π is a witness, it is therefore sufficient to show that the result ρ is still enabled in the initial position $(pm,p'm')$. We proceed by case distinction for the used rule.

UUL. Since π is a witness, its prefix $\alpha = \pi_0 L_0^{l_0}\pi_1 L_1^{l_1}$ must be enabled in $(pm,p'm')$ and because $\textit{Type}(L_0) = (>,\geq)$, so is the prefix $\beta = \pi_0 L_0^{l_0+x}\pi_1 L_1^{l_1-y}$ of the result ρ. Assume that $(pm,p'm') \xrightarrow{\alpha} (qn,q'n')$. Since $\Delta'(\alpha) = \Delta'(\beta)$ we have $(pm,p'm') \xrightarrow{\beta} (q\hat{n},q'n')$. The condition $S(L_0) \geq S(L_1)$ implies that $\hat{n} \geq n \geq \Gamma(\pi_2)$ and therefore that ρ is enabled in $(pm,p'm')$.

UUR. The prefix $\pi_0 L_0^{l_0-x}$ of π must be enabled and since the last condition of the rule demands that $l_0 - x > |\pi_1 L_1|$, so is the path $\pi_0 L_0^{l_0-x}\pi_1 L_1$. The fact that $Type(L_1) = (>,\geq)$, means that also $\pi_0 L_0^{l_0-x}\pi_1 L_1^{l_1+y}$ and therefore the result ρ is enabled in $(pm,p'm')$.

UD. $Type(L_1) = (<,<)$ implies $S(L_1) < \infty$. Since $S(L_0) < S(L_1)$, we know that $S(L_0) < \infty$ and hence $\Delta'(L_0) > 0$. The path $\pi_0 L_0^{l_0-x}$ is a prefix of π and is therefore enabled in $(pm,p'm')$. As $l_0 - x \geq |\pi_1|$ by assumption, we get that

$$m + \Delta(\pi_0 L_0^{l_0-x}) \geq l_0 - x \geq |\pi_1| \geq \Gamma(\pi_1) \tag{7.5}$$

and similarly, by $\Delta'(L_0) > 0$,

$$m' + \Delta'(\pi_0 L_0^{l_0-x}) \geq l_0 - x \geq |\pi_1| \geq \Gamma'(\pi_1). \tag{7.6}$$

This means that the prefix $\beta = \pi_0 L_0^{l_0-x}\pi_1$ of ρ is enabled in $(pm,p'm')$. Let us now consider the prefix $\alpha = \pi_0 L_0^{l_0-x} L_0^x \pi_1 L_1^y$ of π. Because $\Delta'(L_0)\cdot x = -\Delta'(L_1)\cdot y$ we get $\Delta'(\alpha) = \Delta'(\beta)$. By $S(L_0) < S(L_1)$ we obtain that $\Delta(\alpha) \leq \Delta(\beta)$. Because $\pi = \alpha L_1^{l_1-y}\pi_2$ is a witness for $pm \not\leq p'm'$, we can apply Lemma 7.4 to conclude $\rho = \beta L_1^{l_1-y}\pi_2$ must be a witness for $pm \not\leq p'm'$.

DDL. We know that $m + \Delta(\pi_0 L_0^{l_0}) + \Delta(\pi_1) \geq \Gamma(L_1^{l_1})$, because π is enabled in $(pm,p'm')$. As L_1 is a type $(<,<)$ loop we also know that $\Delta(L_1) < 0$. Therefore, $\Gamma(L_1^{l_1}) \geq l_1$ and

$$m + \Delta(\pi_0 L_0^{l_0}) \geq l_1 - \Delta(\pi_1). \tag{7.7}$$

Assume towards a contradiction that $m + \Delta(\pi_0 L_0^{l_0}) < \Gamma(L_0^x \pi_1)$. This means that

$$m + \Delta(\pi_0 L_0^{l_0}) < \Gamma(L_0^x) + |\pi_1| \leq |L_0|\cdot x + |\pi_1|. \tag{7.8}$$

This, together with Equation (7.7) yields $l_1 - \Delta(\pi_1) < |L_0|\cdot x + |\pi_1|$ and thus $l_1 < |L_0|\cdot x + 2|\pi_1|$ which contradicts the condition that $l_1 > |L_0|\cdot x + 2|\pi_1|$. Hence, $m + \Delta(\pi_0 L_0^{l_0}) \geq \Gamma(L_0^x \pi_1)$. By the same argument we get that $m' + \Delta'(\pi_0 L_0^{l_0}) \geq \Gamma'(L_0^x \pi_1)$. So the prefix $\beta = \pi_0 L_0^{l_0+x}\pi_1$ of ρ is enabled in $(pm,p'm')$. Consider the prefix $\alpha = \pi_0 L_0^{l_0}\pi_1 L_1^y$ of π. By the assumption that $\Delta'(L_0^x) = \Delta'(L_1^y)$ we get that $\Delta'(\alpha) = \Delta'(\beta)$. Because of $S(L_0) < S(L_1)$ we get $\Delta(L_0^x) \geq \Delta(L_1^y)$ and therefore that $\Delta(\alpha) \leq \Delta(\beta)$. By Lemma 7.4 we conclude that the path $\rho = \beta L_1^{l_1-y}\pi_2$ is a witness for $pm \not\leq p'm'$.

DDR. Let $\alpha = \pi_0 L_0^{l_0}\pi_1$ and let $(pm,p'm') \xrightarrow{\alpha} (qn,q'n')$. Due to the type of L_0 and because π is a witness, we know that the prefix $\beta = \pi_0 L_0^{l_0-x}\pi_1 L_1^y$ of ρ is enabled in $(pm,p'm')$. Since $\Delta'(L_0)\cdot x = \Delta'(L_1)\cdot y$, we get that $(pm,p'm') \xrightarrow{\beta} (q\hat{n},q'n')$ for some $\hat{n} \in \mathbb{N}$. The condition $S(L_0) \geq S(L_1)$ of the rule implies that $\Delta(L_0^x) \leq \Delta(L_1^y) < 0$, and therefore that $\hat{n} \geq n$. We conclude that the path $L_1^{l_1}\pi_2$ is enabled in $(qr,q'r')$ and therefore that $\rho = \pi_0 L_0^{l_0-x}\pi_1 L_1^{l_1+y}\pi_2$ is enabled in $(pm,p'm')$ as required. □

Lemma 7.10. *Any sequence of successive applications of rules to a given path π must eventually terminate.*

Proof. Consider a π to which we apply the rewriting rules. W.l.o.g. assume π is sane, as otherwise no rule is applicable by definition. Let V and E be the sets of nodes and transitions of the product of $\mathcal{A}$ and $\mathcal{A}'$. The *decomposition* of π is the sequence

$$Dec(\pi) = (\pi_0, L_0, l_0)(\pi_1, L_1, l_1) \dots (\pi_k, L_k, l_k)\pi_{k+1} \tag{7.9}$$

in $(E^* \times E^* \times \mathbb{N})^* E^*$ such that $\pi = \pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \dots \pi_k L_k^{l_k} \pi_{k+1}$, where $k \leq F_0$ and for all $0 \leq i \leq k$,

1. L_i is a loop,
2. π_i is acyclic,
3. For any two transitions $t \in \pi_i$ and $t' \in L_i$ with $target(t) = target(t')$ it holds that $target(L_i) = target(t)$.

The last condition demands that any loop L_i shares exactly one node with the acyclic path π_i it succeeds and thus ensures that the decomposition of a path is unique. As no application of a rule completely removes all occurrences of loops nor introduces new ones nor touches the intermediate paths, we observe that rule applications only change the exponents l_i in the decomposition of the path.

Based on the order of loops in the decomposition of π, and their potential for rule application, we now define a notion of *weights* for paths, and show that these weights have to strictly decrease along a well-order whenever a rule is applied.

Let $(L_0, L_1, \dots, L_k)$ be the sequence of loops that occur in the decomposition of π. Furthermore, we fix some linear order $\prec$ on $\{L_0, L_1, \dots, L_k\}$ that satisfies the following conditions for any two different loops L_i, L_j with $i < j$.

1. If $Type(L_i) = Type(L_j) = (>, \geq)$ and $S(L_i) \geq S(L_j)$ then $L_i \prec L_j$.
2. If $Type(L_i) = Type(L_j) = (>, \geq)$ and $S(L_i) < S(L_j)$ then $L_i \succ L_j$.
3. If $Type(L_i) = Type(L_j) = (<, <)$ and $S(L_i) < S(L_j)$ then $L_i \prec L_j$.
4. If $Type(L_i) = Type(L_j) = (<, <)$ and $S(L_i) \geq S(L_j)$ then $L_i \succ L_j$.

Surely, such a linearisation exists because the conditions above only restrict $\prec$ between loops of the same type and slopes are linearly ordered. Consider the permutation $\sigma : \underline{k} \to \underline{k}$ given by $\sigma(i) < \sigma(j) \iff L_i \prec L_j$. The *weight* of π is

$$W(\pi) = (l_{\sigma(k)}, l_{\sigma(k-1)}, \dots, l_{\sigma(0)}) \in \mathbb{N}^{k+1}. \tag{7.10}$$

The weight of π is the ordered tuple of exponents l_i of loops that occur in π. Since the rules do not change the order of loop occurrences, the path before and after applying a rule have comparable weights. The very definition of weights ensures that rule applications must strictly

reduce the weight of a path. We finish the proof of Lemma 7.10 by showing the following claim.

If ρ is the result of applying one of the rewriting rules to π then $W(\pi') \sqsubset_{lex} W(\pi)$ where $\sqsubset_{lex}$ is the lexicographic extension of the pointwise ordering of tuples of naturals.

This implies the claim of the lemma, because $\sqsubset_{lex}$ is a well-ordering for $\mathbb{N}^{k+1}$. To prove this claim, assume the decompositions of π and ρ are

$$\begin{aligned} Dec(\pi) &= (\pi_0,L_0,l_0)(\pi_1,L_1,l_1)\dots(\pi_k,L_k,l_k)\pi_{k+1} \text{ and} \\ Dec(\rho) &= (\pi_0,L_0,r_0)(\pi_1,L_1,r_1)\dots(\pi_k,L_k,r_k)\pi_{k+1}. \end{aligned} \tag{7.11}$$

We show for every type of rule that if the occurrences of loop L_i increase then those of some loop L_j with $L_i \prec L_j$ strictly decrease.

If the rule was *UUL*, then $r_i = l_i + x$ and $r_j = l_j - y$ for some $i < j$, $0 < x,y$ and type $(>,\geq)$ loops L_i,L_j with $S(L_i) \geq S(L_j)$. By condition 1) in the definition of $\prec$ we get $L_i \prec L_j$.

For rule *UUR*, we know $r_i = l_i - x$ and $r_j = l_j + y$ for some $0 < x,y$ and type $(>,\geq)$ loops L_i,L_j with $S(L_i) < S(L_j)$. By condition 2) in the definition of $\prec$, we get $L_i \succ L_j$.

For rule *DDL*, we know $r_i = l_i + x$ and $r_j = l_j - y$ for type $(<,<)$ loops L_i,L_j with $S(L_i) < S(L_j)$. By condition 3) in the definition of $\prec$, we know $L_i \prec L_j$.

For rule *DDR*, we know $r_i = l_i - x$ and $r_j = l_j + y$ for some $0 < x,y$ and type $(<,<)$ loops L_i,L_j with $S(L_i) > S(L_j)$. So condition 4) in the definition of $\prec$, implies $L_i \succ L_j$.

Finally, if the rule used to derive ρ was *UD* we immediately see that $r_i < l_i$ and $r_j < l_j$, which implies the claim. □

Lemmas 7.9 and 7.10 allow us to focus on witnesses that are *reduced*, i.e., which are sane and to which none of the rewriting rules is applicable. We will show (in Lemma 7.12) that on those reduced paths, the multiplicities of loops with suboptimal effect can be bounded.

Example 7.11. Consider two DOCNs such that their product is the graph depicted on the left in Figure 7.2, in which we identify transitions with their action labels for simplicity and let $v_0 = (p,p') \in V$. The paths $t_0t_1t_2$, t_3t_4 and t_6 are loops with effects (3,1), (2,1) and (−1,−1) respectively.

The path $\pi = (t_0t_1t_2)(t_3t_4)^9t_5(t_6)^{20}$ is a witness for $p0 \not\subseteq p'10$ of length 42. However, this path is not reduced because we can apply rule *UUL* to change occurrences of (t_3t_4) by the more effective loop $(t_0t_1t_2)$, for instance setting $x = y = 8$. The resulting path is $\pi' = (t_0t_1t_2)^9(t_3t_4)^1t_5(t_6)^{20}$, which is a reduced witness for $p0 \not\subseteq p'10$ of length 50. Shorter reduced witnesses exists, for example $(t_0t_1t_2)^6t_5t_6^{16}$, but because of their different loop structure, these cannot be obtained from π by applying rewriting rules from Definition 7.8.

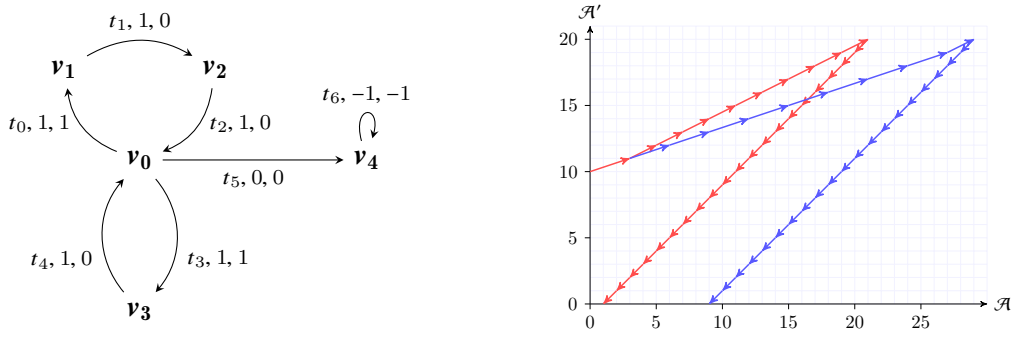


Figure 7.2: Product of two DOCN (left). The right shows how counter values evolve along paths π (in red) and π' (in blue) from initial point $(0,10)$.

The previous example shows that our path rewriting rules do not necessarily preserve minimality of witnesses. However, we can bound the multiplicities of loops with suboptimal effect on reduced paths.

Lemma 7.12. *Let $\pi = \pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \pi_2$ be a reduced path where L_0, L_1 are loops occurring with multiplicities $l_0 > 0$ and $l_1 > 0$.*

1. *If $Type(L_0) = Type(L_1) = (>, \geq)$ and $S(L_0) \geq S(L_1)$ then $l_1 < K$*
2. *If $Type(L_0) = Type(L_1) = (>, \geq)$ and $S(L_0) < S(L_1)$ then $l_0 \leq |\pi_1| + 2K$*
3. *If $Type(L_0) = Type(L_1) = (<, <)$ and $S(L_0) < S(L_1)$ then $l_1 < K^2 + 2|\pi_1|$*
4. *If $Type(L_0) = Type(L_1) = (<, <)$ and $S(L_0) \geq S(L_1)$ then $l_0 < K$*
5. *If $Type(L_0) = (>, \geq)$, $Type(L_1) = (<, <)$ and $S(L_0) \leq S(L_1)$ then $l_0 \leq |\pi_1| + K$ or $l_1 \leq K$.*

Proof. The fourth condition of any rule is satified e.g. by $x = \Delta'(L_1)$ and $y = \Delta'(L_0)$. So if $0 < x, y \in \mathbb{N}$ is the smallest satisfying pair we know $x, y \leq K$. The bounds are now easily derived by contradiction:

1. If $l_1 \geq K$ then $l_1 - y \geq l_1 - K > 0$ and rule *UUL* is applicable.
2. If $l_0 > |\pi_1| + 2K$ then $l_0 - x > |\pi_1| + 2K - x \geq |\pi_1| + |L_1| \geq |\pi_1 L_1|$ and rule *UUR* is applicable.
3. If $l_1 \geq K^2 + 2|\pi_1|$ then $l_1 \geq |L_0| \cdot x + 2|\pi_1|$ and $l_1 - y \geq l_1 - K > 0$, so rule *DDL* is applicable.
4. If $l_0 > K$ then $l_0 - x > 0$, so rule *DDR* is applicable.
5. If $l_1 > K$ and $l_0 > |\pi_1| + K$, then $l_1 - y > 0$, $l_0 - x > 0$ and $l_0 - x > |\pi_1|$, so rule *UD* is applicable.

In each case we conclude that one of the rules is applicable, which contradicts the assumption that π is reduced. □

Proof of Theorem 7.6. We show that we can sufficiently increase the bound c such that whenever there is a witness for $pm \not\subseteq p'm'$ but there is no such witness shorter than c or of form 1) or 2), then there must be a witness of form 3).

Assume a reduced witness π for $pm \not\subseteq p'm'$ that is minimal in length: no shorter witness is reduced. Recall that this also means that π is sane: it is of the form described in Equation (7.4). By monotonicity (Lemma 7.4) and because π is of minimal length among the reduced witnesses, we see that it cannot contain loops of type $(\leq,\geq)$. Since π is not of form 1), it therefore contains only loops of types $(>,\geq)$ and $(<,<)$. Relaxing the bound on the length of paths between loops to $F_1 := F_0(2K + K^2)$, we can write π as

$$\pi = \pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \dots \pi_k L_k^{l_k} \pi_{k+1} \tag{7.12}$$

where $k \leq F_0$, all $|\pi_i| < F_1$ and the number of iterations of loop L_i is $l_i > K$.

Consider a block $\pi_{pos} = L_i^{l_i} \pi_{i+1} L_{i+1}^{l_{i+1}} \pi_{i+2} \dots \pi_j L_j^{l_j}$ that is part of the decomposition above, such that all loops are of type $(>,\geq)$. If for indices $i \leq x < y \leq j$ we have $S(L_x) \geq S(L_y)$, then by Lemma 7.12.1 we get $l_y < K$. Therefore, π_{pos} is of the form

$$\pi_{pos} = L_i^{l_i} \pi_{i+1} L_{i+1}^{l_{i+1}} \pi_{i+2} \dots \pi_j L_j^{l_j} \pi_{j+1} \tag{7.13}$$

where the lengths of π_i are bounded by $F_2 := F_0 \cdot (K^2 + F_1)$ and the slopes of loops are strictly increasing: $S(L_x) < S(L_y)$ for any two indices $i \leq x < y \leq j$. By Lemma 7.12.2 this means that $l_x \leq |\pi_{x+1}| + 2K \leq F_2 + 2K =: F_3$. We conclude that the prefix $\pi' = L_i^{l_i} \pi_{i+1} L_{i+1}^{l_{i+1}} \pi_{i+2} \dots \pi_{j-1} L_{j-1}^{l_{j-1}}$ is no longer than $(j-i) \cdot (K \cdot F_3 + F_2)$ and therefore

$$\pi_{pos} = \pi' L_j^{l_j} \pi_{j+1} \tag{7.14}$$

where $|\pi'|$ is bounded by $F_4 := F_0(K \cdot F_3 + F_2)$ and $|\pi_{j+1}|$ by F_2.

We continue to show by a similar argument that we can bound the number of iterations of all but the most productive loop in a block consisting of only type $(<,<)$ loops. Consider a block $\pi_{neg} = L_i^{l_i} \pi_{i+1} L_{i+1}^{l_{i+1}} \pi_{i+2} \dots \pi_j L_j^{l_j}$ that is part of the decomposition in Equation (7.12), where all loops are of type $(<,<)$. If $S(L_x) \geq S(L_y)$ for some indices $i \leq x < y \leq j$, then by Lemma 7.12.4 we know $l_x < K$. This means that π_{neg} is of the form

$$\pi_{neg} = \pi_i L_i^{l_i} \pi_{i+1} L_{i+1}^{l_{i+1}} \pi_{i+2} \dots \pi_j L_j^{l_j} \pi_{j+1} \tag{7.15}$$

where all π_i have lengths bounded by F_2 and $S(L_x) < S(L_y)$ for any two indices $i \leq x < y \leq j$. By Lemma 7.12.3 we get $l_y \leq K^2 + 2|\pi_x| \leq K^2 + 2F_2 =: F_3'$ and conclude that the suffix $\pi'' = \pi_{i+1} L_{i+1}^{l_{i+1}} \pi_{i+2} \dots \pi_j L_j^{l_j} \pi_{j+1}$ is no longer than $(j-i) \cdot (K \cdot F_3' + F_2)$. Therefore, π_{neg} is of the form

$$\pi_{neg} = \pi_i L_i^{l_i} \pi'' \tag{7.16}$$

where π_i is bounded by F_2 and π'' by $F'_4 := F_0(K \cdot F'_3 + F_2)$.

Equations (7.14) and (7.16) characterize the form of maximal subpaths of the witness π in Equation (7.12), along which the type of loops does not change. They allow us to write π as

$$\pi = \pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \pi_2 \dots \pi_k L_k^{l_k} \pi_{k+1} \tag{7.17}$$

where for all indices $0 \le i < k$:

1. π_i is no longer than $F_5 := F_3 + F'_3 + F_4 + F'_4$.
2. $l_i > K$.
3. Consecutive loops L_i and L_{i+1} have different types.
4. If loops L_i, L_j for $0 \le i < j \le k$ have the same type then $S(L_i) < S(L_j)$.

In the remainder of this proof, we further increase the polynomial bound for the gaps π_i between the loops; this allows to conclude that π contains at least one type $(<,<)$ loop and finally, that π is of form 3).

Observe that if all loops L_i in Equation (7.17) are of type $(>,\ge)$ then the witness is already of form $\pi = \pi_0 L^l \pi_1$ as in Equation (7.14), where π_0, π_1 are short and L is the most effective loop. In this case, consider the run

$$(pm, p'm') \xrightarrow{\pi_0 L^l} (qn, q'n') \tag{7.18}$$

induced by the prefix $\pi_0 L$. Since $\mathcal{A}'$ is complete we know $\Delta'(\pi) = -m'$. Together with $\Delta'(\pi_1) \le |\pi_1| \le F_5$ we get $n' \le F_5$. We know that $l \le |\pi_1| \le F_5$, because $\Gamma(\pi_1) \le |\pi_1|$ and otherwise, fewer iterations l would result in a shorter witness and we assumed π to be minimal in length. Hence, we could bound π by $F_6 := F_5 + K \cdot F_5 + F_5$. So if we let $c \ge F_6$, our witness π must contain type $(<,<)$ loops as it is assumed not to be no shorter than c.

Finally, fix an index $0 \le x \le k$ such that in Equation (7.17), L_x is a loop of type $(<,<)$ with most efficient decrease (minimal slope). That is, π is of the form

$$\pi = \pi_0 L_x^{l_x} \pi_1. \tag{7.19}$$

We now bound π_0 and π_1 and thereby prove that π is of form 3). We start with the suffix π_1.

If L_x is the only loop of type $(<,<)$, we are done, because then $|\pi_1| \le F_5$. Suppose we have two indices $0 \le y < y+2 \le k$, where both L_y and L_{y+2} are of type $(<,<)$. This means that L_{y+1} is of type $(>,\ge)$ with $S(L_{y+1}) < S(L_{y+2})$. By Lemma 7.12.5 and the fact that $l_{y+2} > K$ we know that $l_{y+1} < |\pi_{y+1}| + K \le F_6$. So $\pi_{y+1} L_{y+1}^{l_{y+1}} \pi_{y+2}$ is no longer than $2 \cdot F_5 + K \cdot F_6 =: F_7$. Applying Lemma 7.12.3 to L_y and L_{y+2} we get $l_{y+2} \le K^2 + 2 \cdot F_7 =: F_8$ and thus $\pi_{y+1} L_{y+1}^{l_{y+1}} \pi_{y+2} L_{y+2}^{l_{y+2}}$ is no longer than $F_9 := F_5 + (K \cdot F_6) + F_5 + (K \cdot F_8)$. Now the above argument can be repeated for any successive pair of type $(<,<)$ loops in π_1 of which there are at most F_0. So, $|\pi_1| < F_0 \cdot F_9$.

To bound the prefix π_0 in Equation (7.19), we recall (point 3 above) that consecutive loops in Equation (7.17) have different types and therefore $x \leq 1$. In case $x = 0$, we immediately get $|\pi_0| \leq F_5$. If $x = 1$, then L_0 is a type $(>,\geq)$ loop with $S(L_0) < S(L_x)$ and so by Lemma 7.12.5 and point 2), we get $l_0 \leq |\pi_1| + K < F_6$. This means $|\pi_0| \leq 2F_5 + K \cdot F_6 = F_7$.

We conclude that $c := F_9 \cdot F_0$ is sufficient to ensure that any witness π, longer than c which is not of form 1) or 2) must have form 3).

To see why l_0 and l_1 can always be bounded polynomially in K and m', we look at the types of the loops involved. For paths of form 1 and 3, L_0 decreases the counter on the right at least once in every iteration. Since the value $m' + \Delta'(\pi_0)$ before the first iteration is at most $m' + c$, we have $l_0 \leq m' + c$.

Paths of the second form can be decomposed into a prefix $\pi_0 L_0^{l_0}$ and a suffix $\pi_1 L_1^{l_1} \pi_2$, which is a path of form 3. Let $y_0 \in \mathbb{N}$ be minimal such that the effect of the path $\gamma_0 = \pi_0 L_0^0 \pi_1 L_1^{y_0} \pi_2$, in which L_0 is not iterated at all is sufficient to reduce the initial value m' below 0. That is, we have $m' + \Delta'(\pi_0 L_0^0 \pi_1 L_1^{y_0} \pi_2) \leq 0$. Note that as for forms 1 and 3, we can bound y_0 by $m' + 2c$ and therefore, $|\gamma_0|$ is no larger than $3c + K \cdot (m' + 2c)$. This path might not be a witness because it is not enabled on the left side. However, because of the condition on the slopes, there are $x, y \leq K$ such that the effect of the loops satisfy

$$\Delta'(L_0) \cdot x = -\Delta'(L_1) \cdot y \qquad \text{and} \qquad \Delta(L_0) \cdot x > -\Delta(L_1) \cdot y. \tag{7.20}$$

This means, increasing the iterations of the loops L_0 and L_1 by x and y respectively, does not change the effect of the path on the right, but strictly increases the effect on the left. We increase the iterations $(l_0, l_1) = (0, y_0)$ in γ_0 as suggested above for $\Gamma(\gamma_0) < |\gamma_0| < 3c + K \cdot (m' + 2c)$ times. The resulting path $\gamma_1 = \pi_0 L_0^{x_1} \pi_1 L_1^{y_1} \pi_2$ is then surely a witness, and iterates the loops not more than $x_1 = 3c + K \cdot (m' + 2c)$ and $y_1 = m' + 5c + K \cdot (m' + 2c)$ times. □

7.1.4 An NL-Upper bound

We will now use the characterization of witnesses from the previous section to show an NL upper bound for the problem OCN $\subseteq$ DOCN, trace inclusion between processes of one-counter nets and deterministic one-counter nets, where initial counter values are part of the input. This matches the trivial lower bound that already holds for DFA $\subseteq$ DFA.

Under certain conditions, one can show that the length of a shortest witness for non-inclusion is polynomial in the size of the input nets. Theorem 7.6 (on page 96) implies that this is the case for instances $(\mathcal{A}, \mathcal{A}', pm, p'm')$, if the initial value m' (not its binary encoding) is itself polynomially bounded in the size of the input nets. Then, one can simply stepwise guess and verify a witness, storing the binary encoded effects along the way in logarithmic space.

In general however, the length of shortest witnesses depends (polynomially) not only on the size of the input nets, but also on the initial counter values. Recall the simple self-looping

processes with effects 0 and -1 from Example 7.7 on page 97. A witness for non-inclusion $pm \not\subseteq p'm'$ must iterate these loops exactly m' times. For practical purposes it is reasonable to assume that the initial counter values are given in binary encoding. The length of minimal witnesses can therefore only be *exponentially* bounded in the size of the input.

This means that if the initial counter values are part of the input, the simple procedure mentioned above is not sufficient. However, Example 7.7 and indeed Theorem 7.6 suggest that, although we might not be able to store and manipulate the counter effects directly, we can guess the general form of a witness and then (efficiently) check if a witness of that form exists.

In order to address the issue of large initial counter values, we first present a lemma about the complexity of verifying inequalities of (weighted) binary numbers. It can be shown using standard $O(\log n)$-space algorithms for multiplication and addition on the fly.

Lemma 7.13. *Inequalities of the form $m \cdot A + B \geq n \cdot C + D$ where all coefficients are non-negative integers given in binary can be verified in $O(\log(A+B+C+D))$ deterministic space, regardless of the size of m and n.*

Proof. Assume w.l.o.g. that the bit-representations of m and n are of the same length, as are those of A, B, C and D, and we have the least significant bit on the right.

To check $m \geq n$, we can stepwise read their binary representation from right to left, flipping an "output" bit *Out* on the way: Initially, $Out := 1$; in every step set $Out := 0$ if the current bit in m is strictly smaller than that in n; set $Out := 1$ if the current bit in m is strictly bigger than that in n and otherwise proceed without touching *Out*. The inequality holds iff $Out = 1$ after completely reading the input.

To check the weighted variant, we use the same algorithm but multiply $m \cdot A$, and $n \cdot C$ on the fly, using standard long binary multiplication. We use a scratchpad to store the intermediate sums, starting with values B and D. In a step that reads the ith bit $m[i]$ of m, we want to add $A \cdot 2^i$ to the intermediate sum if $m[i] = 1$. We can do that by shifting the binary representation of A left i times and then adding the result to the current scratchpad. We see that none of the bits up to $i-1$ in the scratchpad are affected by this operation. We can therefore discard (and use for the comparison in our simple algorithm above) the rightmost bit of the scratchpad in every step. The relevant part of the intermediate sum on each side thus uses no more than $x+1$ bits, where x is the length of the bit representation of A, B, C and D. □

Theorem 7.14. *The problem OCN $\subseteq$ DOCN is in NL.*

Proof. Let pm and $p'm'$ be processes of (given) nets $\mathcal{A} = (Q, \mathrm{Act}, \delta)$ and $\mathcal{A}' = (Q', \mathrm{Act}, \delta')$ respectively and let $K = |Q \times Q'| \in \mathbb{N}$ be the number of states in their product. By Lemma 3.10 (page 25), we can assume w.l.o.g. that $\mathcal{A}$ is in fact deterministic and $\mathcal{A}'$ is complete and deterministic, and thus Theorem 7.6 applies.

Assume that $pm \not\subseteq p'm'$. We argue that one can nondeterministically guess a template (consisting of short paths) and verify in logspace that there is indeed some witness that fits this template. Theorem 7.6 allows us to either guess a short ($\leq c$) path π or one of forms 1,2 or 3, together with matching short paths π_i, L_i. The effect and guard of these paths are bounded by their lengths and hence by c, which is polynomial in K. This means $O(\log K)$ space suffices to compute the binary representation of these values and verify that the conditions the form imposes on the types and slopes of the loops are met. It remains to check if exponents $l_i \in \mathbb{N}$ exist, that complete the description of a witness π.

Case 1. There is a witness $\pi = \pi_0 L_0^{l_0} \pi_1$ of form 1. Then surely, there is also a positive witness for $pm \not\subseteq p'm'$ of this form, that just iterates the type $(\geq, <)$ loop L_0, i.e., the π_1 is a prefix of L_0. The type of L_0 implies that $\Gamma(\pi_0 L_0^l) \leq \Gamma(\pi_0 L_0)$ for all $1 < l \in \mathbb{N}$. Therefore, it suffices to check if the prefix $\pi_0 L_0$ is enabled in $(pm, p'm')$, which can be done using logarithmic space as $|\pi_0 L_0| \leq c + K$.

Case 2. There is a witness $\pi = \pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \pi_2$ of form 2. As in the first case, it suffices to check if the prefix $\pi_0 L_0$ is enabled in $(pm, p'm')$, because then the process on the left can repeat the loop L_0 arbitrarily often. The existence of $l_0, l_1 \in \mathbb{N}$ completing the description of a witness is then guaranteed because the slope of the first loop is bigger than that of the second.

Case 3. There is a witness $\pi = \pi_0 L_0^{l_0} \pi_1$ of form 3, Recall that, because $\mathcal{A}'$ is complete, a path π is a witness iff there is some edge T in the product such that $\Delta'(T) = -1$, and both $m \geq \Gamma(\pi T)$ and $n + \Delta'(\pi) = 0$. Equivalently, we can write this as

$$m + \Delta(\pi_0 L_0^{l_0}) = m + \Delta(\pi_0) + \Delta(L_0) \cdot l_0 \geq \Gamma(\pi_1 T) \text{ and} \tag{7.21}$$

$$n + 1 = -\Delta'(\pi T) = -\Delta'(\pi_0) - \Delta'(L_0) \cdot l_0 - \Delta'(\pi_1 T). \tag{7.22}$$

Eliminating l_0, we see that this is true iff

$$m + \Delta(\pi_0) + \Delta(L_0) \cdot \frac{\Delta'(\pi_0) + \Delta'(\pi_1 T) + n}{-\Delta'(L_0)} \geq \Gamma(\pi_1 T). \tag{7.23}$$

Depending on the sign of the actual counter-effects of the guessed short paths, we can simplifying this further and bring it into the form $m \cdot A + B \geq n \cdot C + D$ where $A, B, C, D \in \mathbb{N}$ are polynomial in K. By Lemma 7.13 the condition can be verified in $O(\log K)$.

We conclude that, if any witness for non-inclusion $pm \not\subseteq p'm'$ exists, then by Theorem 7.6, also one in one of the four possible forms. For each of these forms, one can guess a polynomially-size description, consisting of short paths, and then verify the existence of a witness that fits this description (by Lemma 7.13) in $O(\log K)$ space. □

7.2 Automata vs. Nets: Allowing Zero-Tests on One Side

We now generalize the result of the previous section and show that checking trace inclusion OCA $\subseteq$ DOCN is also NL-complete. Again, by Lemma 3.10, it suffices to consider DOCAs

$\mathcal{A} = (Q,\mathrm{Act},\delta,\delta_0)$ and DOCN $\mathcal{A}' = (Q',\mathrm{Act},\delta')$ in normal form. Let $K = |Q \times Q'|$ be the number of states of their product (cf. Definition 3.17 on page 29) let pm and $p'm'$ be processes of $\mathcal{A}$ and $\mathcal{A}'$ respectively.

Any witness for non-inclusion $pm \not\subseteq p'm'$ can be split into a positive prefix, along which no zero-testing transitions are used, a zero-touching intermediate path and the remaining path, which is again positive. Theorem 7.6 characterizes the positive suffix. We then show that the number of zero-testing transitions, and therefore the length of the intermediate path, can be bounded polynomially. Finally, we show (in Lemma 7.18) that the positive prefix can be assumed to have a very simple form: it essentially only iterates one most effective simple cycle. This allows us to check, just as in the proof of Theorem 7.14, the existence of witnesses by exhaustively checking a small number of templates, which can be done in $O(\log K)$ space.

We first focus on zero-touching paths in the product of $\mathcal{A}$ and $\mathcal{A}'$, and start by introducing a notion of *rank* for positions $(pm,p'm')$, that is the necessary number of zero-testing transitions on paths witnessing $pm \not\subseteq p'm'$.

Definition 7.15. The *rank* of a path in the product of $\mathcal{A}$ and $\mathcal{A}'$ is the number of zero-testing transitions used. The rank $\#(pm,p'm')$ of two processes pm and $p'm'$ of $\mathcal{A}$ and $\mathcal{A}'$ is the minimal rank of any witness for $pm \not\subseteq p'm'$ and ω if none exists.

Of course, the interesting case is if the rank of a given pair is strictly bigger than 0: no positive witness for non-inclusion exists. Otherwise, we can turn $\mathcal{A}$ into a OCN by removing all zero-testing transitions, and preserve the existence or non-existence of positive witnesses, which can be verified already by Theorem 7.14.

If however, all witnesses for $(pm,p'm')$ contain zero-testing transitions, we can make additional assumptions on positive prefixes of witnesses: If $0 < \#(pm,p'm')$, then no positive path that is enabled in $(pm,p'm')$ must contain a cycle C with $\Delta(C) \geq 0 > \Delta'(C)$, nor a subpath $L_0\pi L_1$, where $Type(L_0) = (>,\geq)$, $Type(L_1) = (<,<)$ and $S(L_0) > S(L_1)$. This is because both conditions would imply the existence of a positive witness contrary to the assumption on the rank. The next lemmas characterize the positive prefix and the intermediate part, under the assumption that the rank of the positions they visit remains positive.

Lemma 7.16. *Assume* $(pm,p'm') \longrightarrow_+^* (qn,q'n')$ *but* $(pm,p'm') \not\longrightarrow_+^* (qn,q'x)$ *for all* $x < n'$. *Moreover, assume that the rank* $\#(pm,p'm')$ *is strictly positive and let* π *be a shortest positive path from* $(pm,p'm')$ *to* $(qn,q'n')$. *Then* π *does not visit any counter value* m_i *of* $\mathcal{A}$ *more than* $(K-1)$ *times.*

Proof. Assume towards a contradiction that π visits a pair $(s,s') \in Q \times Q'$ of control states and a counter value i of $\mathcal{A}$ more than once: $\pi = \alpha\beta\gamma$ with

$$(pm,p'm') \xrightarrow{\alpha}_+ (si,s'i') \xrightarrow{\beta}_+ (si,s'j') \xrightarrow{\gamma}_+ (qn,q'n'). \tag{7.24}$$

We know that $\Delta'(\beta) \geq 0$, because otherwise some positive path $\alpha\beta^l$ contradicts the assumption that no positive witness for $pm \not\leq p'm'$ exists. If $\Delta'(\beta) = 0$, then the path $\alpha\gamma$ from $(pm,p'm')$ to $(qn,q'n')$ contradicts the minimality of π. If $\Delta'(\beta) > 0$, then the path $\alpha\gamma$ is from $(pm,p'm')$ to $(qn,q'x)$ for some $n' - \Delta'(\beta)$, contrary to the minimality of n'. This means that whenever a counter value i of $\mathcal{A}$ is revisited along π, the corresponding control states must differ, and therefore no value is visited more than $(K-1)$ times. □

Lemma 7.17. *Assume that* $0 < \#(p1,p'm') < \omega$ *and let* $n' \in \mathbb{N}$ *be minimal such that a positive path* $(p1,p'm') \longrightarrow_+^* (q0,q'n')$ *exists. Then* $(p1,p'm') \longrightarrow_+^k (q0,q'n')$ *for some* $k < K^3$.

Proof. Consider a shortest path

$$\pi = (p_0m_0,p'_0m'_0) \longrightarrow_+ (p_1m_1,p'_1,m'_1) \longrightarrow_+ \ldots \longrightarrow_+ (p_km_k,p'_k,m'_k) \tag{7.25}$$

where $(p_0m_0,p'_0m'_0) = (p1,p'm')$ and $(p_km_k,p'_km'_k) = (q0,q'n')$. By Lemma 7.16, we know that no counter value m_i of $\mathcal{A}$ is visited more than $(K-1)$ times. This observation allows us to recycle the proof of Lemma 3.5 on page 21, to show that maximal counter value m_i visited along π cannot be larger than $(K-1)K$.

Let $0 \leq x \leq k$ be the index of the position $(p_xm_x,p'_xm'_x)$ along our path π where m_x is maximal. For each value $0 \leq y < m_x$, we can mark the maximal index $0 \leq l(y) < x$ and the minimal index $x < f(y) \leq k$ such that $m_{l(y)} = m_{f(y)} = y$. If the maximal counter value m_x is at least $(K-1)K$, then there must be two values $0 \leq z < y \leq m_x$ such that $(p_{l(z)},p'_{l(z)}) = (p_{l(y)},p'_{l(y)})$ and $(p_{f(z)},p'_{f(z)}) = (p_{f(y)},p'_{f(y)})$. This means we can decompose π according to y and z into

$$\begin{aligned}(p_0m_0,p'_0m'_0) &\xrightarrow{\alpha_0}_+ (p_{l(z)}m_{l(z)},p'_{l(z)}m'_{l(z)})\\ &\xrightarrow{\beta_0}_+ (p_{l(y)}m_{l(y)},p'_{l(y)}m'_{l(y)})\\ &\xrightarrow{\gamma}_+ (p_{f(y)}m_{f(y)},p'_{f(y)}m'_{f(y)})\\ &\xrightarrow{\beta_1}_+ (p_{f(z)}m_{f(z)},p'_{f(z)}m'_{f(z)})\\ &\xrightarrow{\alpha_1}_+ (p_km_k,p'_k,m'_k)\end{aligned} \tag{7.26}$$

such that $m_i \geq y$ for all $l(y) \leq i \leq f(y)$ and $m_i \geq y$ for all $l(z) \leq i \leq f(z)$. This, and the assumption that there is no positive witness for $p1 \not\leq p'm'$ mean that $\alpha_0\beta_0^i\gamma\beta_1^i\alpha_1$ is enabled from $(p1,p'm')$ for all $i \in \mathbb{N}$. We distinguish three cases, depending on how the difference $d = m'_{l(z)} - m'_{f(z)} = \Delta'(\gamma)$ compares to the difference $c = m'_{l(y)} - m'_{f(y)} = \Delta'(\beta_0\gamma\beta_1)$, and in each case derive a contradiction.

Case $d = c$. Then $\alpha_0\gamma\alpha_1$ is a path from $(p1,p'm')$ to $(q0,q'n')$ that is shorter than π, which contradicts that π is a shortest such path.

Case $d > c$. Then $\alpha_0\beta_0\beta_0\gamma\beta_1\beta_1\alpha_1$ is a positive path from from $(p1,p'm')$ to $(q0,q'n'')$ for some $n'' < n'$, contrary to the minimality of n'.

Case $d < c$. Then $\alpha_0\gamma\alpha_1$ is a positive path from from $(p1,p'm')$ to $(q0,q'n'')$ for some $n'' < n'$, which again contradicts minimality of n'.

We conclude that the maximal counter value m_x seen along the path π must be smaller than $(K-1)K$. By our previous observation that $(K-1)$ bounds the number of times any value $0 \leq m \leq m_x$ is visited along π, we get that there are no more than $(K-1)K(K-1) < K^3$ distinct positions visited by π. □

Lemma 7.18. *Consider* $p,q \in Q$, $p'q' \in Q'$ *and* $m,m'n,n' \in \mathbb{N}$ *such that*

1. $\#(pm,p'm') > 0$,
2. n' *is minimal such that* $(pm,p'm') \longrightarrow_+^* (q0,q'n')$,
3. $m \geq l + K$, *where* $l = K^4 + 2K^2 + 2K$.

Then there is a positive path from $(pm,p'm')$ *to* $(q0,q'n')$ *of the form* $\pi = \pi_0 L_0^{l_0} \pi_1 \pi_2$, *where* $|\pi_0\pi_1| < K$, $|\pi_2| < K \cdot l$ *and* L_0 *is a loop with* $\Delta(L_0) < 0$.

Proof. Consider a shortest path π from $(pm,p'm')$ to $(q0,q'n')$. We decompose the path into $\pi = (pm,p'm') \xrightarrow{\alpha}_+ (rl,r'l') \xrightarrow{\beta}_+ (q0,q'n')$ such that α is a maximal prefix that ends with l. By Lemma 7.16, we know that the suffix β visits fewer than $(K-1) \cdot l$ different positions, so $|\beta| < (K-1) \cdot l$. Because n' was assumed to be minimal, also l' must be minimal and $(pm,p'm') \not\longrightarrow_+^* (rl,r'x)$ for $x < l$.

We now show that there is a possibly different positive path of the form $\alpha' = \alpha_0 L^0 \alpha_1$ from $(pm,p'm')$ to $(rl,r'l')$ such that L_0 is a simple cycle with $\Delta(L_0) < 0$ and $|\alpha_0\alpha_1| < l$, which completes the proof. The prefix α drops the counter of $\mathcal{A}$ by $m - l > K$. Consequently, it contains at least one simple cycle L_0 with effect $\Delta(L_0) < 0$. Let $\alpha = \alpha_0 L_0 \alpha_1$ where L_0 is such a decreasing loop that moreover is maximally steep: the ratio $\Delta'(L_0)/|\Delta(L_0)|$ is minimal.

Case 1. $\Delta'(L_0) \geq 0$. Let S be a maximal set of non-overlapping cyclic subpaths of $\alpha_0\alpha_1$ such that $-\Delta(L_0)$ divides $\sum_{C \in S} \Delta(C)$ and let $\alpha_0'\alpha_1'$ be the remainder of removing all cycles of S from $\alpha_0\alpha_1$. Further, let $l_0 \in \mathbb{N}$ be the unique solution of $\sum_{C \in S} \Delta(C) = -\Delta(L_0) \cdot l_0$. A crucial observation is that the choice of L_0 implies

$$\sum_{C \in S} \Delta'(C) \;\geq\; -\Delta'(L_0) \cdot l_0. \tag{7.27}$$

We now repeat the argument from Lemma 3.7, only in two dimensions. If the remainder $\alpha_0'\alpha_1'$ contains more than K non-overlapping simple cycles, then the overall effect of some of them on the counter of $\mathcal{A}$ is divisible by $-\Delta(L_0)$, contrary to the maximality of S. Therefore, the remainder $\alpha_0'\alpha_1'$ is no longer than K^2. By Equation (7.27), we know that the path $\alpha' = \alpha_0' L_0^{l_0} \alpha_1'$ has effects

$$\Delta(\alpha') = \Delta(\alpha) \quad \text{and} \quad \Delta'(\alpha') \leq \Delta'(\alpha). \tag{7.28}$$

If we start moving according to α' from position $(pm,p'm')$, the counter of $\mathcal{A}$ cannot drop below $m+\Delta(\alpha)-|\alpha'_0\alpha'_1|-K > l-K^2-K$, which is positive. This, together with the assumption on the rank of the initial position, means the path α' is enabled in $(pm,p'm')$. By minimality of l', we conclude that $(pm,p'm') \xrightarrow{\alpha'}_+ (rl,r'l')$.

Case 2. $\Delta'(L_0) < 0$. In this case it is not guaranteed that any set S of cycles occurring along $\alpha_0\alpha_1$ with effect divisible by $-\Delta(L_0)$ satisfies the condition Equation (7.27). This is because there might be a positive loop U on the path α_1 with $\Delta(U) > 0$ and $\Delta'(U)/\Delta(U) < \Delta'(L_0)/\Delta(L_0)$. If no such loop exists we can proceed as in case 1.

Otherwise, the assumption on the rank of the initial position implies that U must not occur before the decreasing loop L_0. So we can decompose α as $\alpha_0 L_0 \alpha_1 U \alpha_2$. Just as in the first case, we can replace the prefix $\alpha_0 L_0 \alpha_1$ with $\alpha'_0 L_0^{l_0} \alpha'_1$ where $|\alpha'_0\alpha'_1| < K^2$. W.l.o.g. assume that the suffix $\alpha'_1 U \alpha_2$ is longer than K^2, as otherwise the claim directly holds. Also, we can assume w.l.o.g. that α_2 does not contain any loop with the exact same effects as L_0. Now one of two cases must hold: Either the suffix $U\alpha_2$ does not contain any decreasing loops L_1 with $\Delta(L_1) < 0$, or it does. If if does not, then the suffix α_2 cannot be longer than $K^4 + 2K^2 + 2K \cdot K$, because every simple loop must increase the counter on $\mathcal{A}$, and the final value is $K^4 + 2K^2 + 2K$. Otherwise, if there are decreasing loops, then there must be one with minimal ratio $\Delta'(L_1)/|\Delta(L_1)|$. We repeat the argument for case 1 for the suffix α_2 and the loop L_1. Because there are at most $K \cdot 2K$ different pairs of counter-effects for decreasing loops L_i, we eventually end up with a path

$$\alpha' = \pi'_0 L_0^{l_0} \pi'_1 L_1^{l_1} \pi'_2 \dots \pi'_{k-1} L_{k-1}^{l_{k-1}} \pi'_k \tag{7.29}$$

from $(pm,p'm')$ to $(rl,r'l')$ where $k \leq 2K^2$ and for all $0 \leq i \leq k$, $|\pi'_i \pi'_{i+1}| < K^2$. We derive that $\sum_{2 \leq i \leq k} |\pi'_i| < K^2 \cdot (2K^2)/2 = K^4$.

Finally, pick a maximal subset S of the cycles $\{L_1^{e_1}, L_2^{e_2}, \dots L_k^{e_k}\}$ and $d \in \mathbb{N}$ such that $\sum_{C \in S} \Delta(C) = -\Delta(L_0) \cdot d$. Observe that this is possible if $\sum_{1 \leq i \leq k} l_i > -\Delta(L_0)$. Because all L_i for $i \geq 1$ are less steep than L_0, we again observe that $\sum_{C \in S} \Delta'(C) \geq -\Delta'(L_0) \cdot d$. We replace the cycles in S by d iterations of the initial decreasing loop L_0. That is, we construct the path

$$\alpha'' = \pi'_0 L_0^{l_0+d} \pi'_1 L_1^{l_1-e_1} \pi'_2 \dots \pi'_{k-1} L_{k-1}^{l_{k-1}-e_{k-1}} \pi'_k. \tag{7.30}$$

The suffix $\pi'_1 L_1^{l_1-e_1} \pi'_2 \dots \pi'_{k-1} L_{k-1}^{l_{k-1}-e_{k-1}} \pi'_k$ is no longer than $K^2 + K + K^4 < l$. This means that the guard $\Gamma(\alpha'')$, the minimal counter value that enables α'' for $\mathcal{A}$ is no larger than $|\pi'_0 L_0| + K^2 + K + K^4 < K^2 + K + K^2 + K + K^4 = l$. Therefore, α'' must be enabled in $(pm,p'm')$. By minimality of l' we derive that $(pm,p'm') \xrightarrow{\alpha''}_+ (rl,r'l')$, which concludes our proof. □

Theorem 7.19. *The problem OCA ⊆ DOCN is in NL.*

Proof. We use Lemmas 7.17 and 7.18 as well as Theorem 7.6 and consider all possible forms of witnesses for non-inclusion. For each form, we show that one can verify the existence of a witness of this form nondeterministically, using only logarithmic space. Assume $pm \not\subseteq p'm'$.

Case 1: $\#(pm,p'm') = 0$. This means there must be a positive witness, which means we can turn $\mathcal{A}$ into a OCN by removing all zero-testing transitions without changing that $pm \not\subseteq p'm'$. The existence of such witnesses can be checked as in the proof of Theorem 7.14.

Case 2: $\#(pm,p'm') > 0$. Then, no positive path witnesses $pm \not\subseteq p'm'$. We can decompose any witness into a positive prefix α, an intermediate path β that uses zero-testing transitions, and a positive suffix γ. Let π be some witness and cut it into

$$\begin{aligned}(pm,p'm') &\xrightarrow{\alpha}_+ (p_0 0,p'_0 m'_0)\\ &\xrightarrow{\beta_1} (p_1 0,p'_1 m'_1) \xrightarrow{\beta_2} (p_2 0,p'_2 m'_2) \xrightarrow{\beta_3} \dots \xrightarrow{\beta_l} (p_l 0,p'_l m'_l)\\ &\xrightarrow{\gamma}_+ (p_k m_k,p'_k 0)\end{aligned} \tag{7.31}$$

where $(p_0 0,p'_0 m'_0)$ is the first position at where the counter of $\mathcal{A}$ reaches value 0, and w.l.o.g., m'_0 is minimal. That is, $(pm,p'm') \not\xrightarrow{}{}^*_+ (p_0 0,p'_0 x)$ for any $x < m'_0$. Moreover, each β_i on the zero-touching path starts with a zero-testing transition and is otherwise positive. By Lemma 7.17, each such β_i is no longer than K^3.

Case 2a: The rank $\#(pm,p'm')$ is at least K^2. This means there is a shortest witness of the above form with at least K^2 intermediate phases β_i. Then, at least one pair (q,q') of control states repeats as starting states for two different $\beta_i \neq \beta_j$, $i < j$. By minimality of the witness, the path $\rho = \beta_i \beta_{i+1} \dots \beta_{j-1}$ must strictly decrease the counter of $\mathcal{A}'$. We observe that $|\rho| < K^3 \cdot K^2$ and that some prefix of $\alpha\rho^r$ witnesses $pm \not\subseteq p'm'$ for some $r \in \mathbb{N}$. It suffices to check the existence of paths α and ρ satisfying $-\Delta(\alpha) = m$, $\Delta(\rho) = 0$ and $\Delta'(\rho) < 0$.

Case 2b: The rank $\#(pm,p'm')$ is strictly between 0 and K^2. Again we consider a shortest witness of form Equation (7.31). We know that the intermediate path $\beta = \beta_0 \beta_1 \dots \beta_l$ from $(p_0 0,p'_0,m'_0)$ to $(p_l 0,p'_l,m'_l)$ is no longer than $K^3 \cdot K^2$.

By Theorem 7.6, we can assume that the positive suffix γ, which is a witness for $p_l 0 \not\subseteq p'_l m'_l$ has one of three possible forms. We show that for each form one can extract polynomially bounded paths that suffice as certificates for the existence of a witness of this form. This means that just as in the proof of Theorem 7.14, we can in nondeterministic logarithmic space guess and verify these paths. Let c be the polynomial bound provided by Theorem 7.6.

Form 1: $\gamma = \pi_0 L_0^{l_0} \pi_1$, where L_0 is a loop with $\Delta'(L_0) < 0$ and $\Delta(L_0) \geq 0$ and $|\pi_0\pi_1| \leq c$. It suffices to guess and verify the prefix $\pi_0 L_0$, because then L_0 can be iterated sufficiently often to eventually reduce the counter of $\mathcal{A}'$ below 0. This certificate is no longer than $c + K^2$. To check if a witness for $pm \not\subseteq p'm'$ of this form exists, we need to check if there is a path $\alpha\beta\pi_0 L_0$ with $-\Delta(\alpha) = m$, $\Delta(\beta\pi_0 L_0) \geq 0$, $\Delta'(L_0) < 0$ and $\Delta'(L_0) \geq 0$. This can be done in logspace by Lemma 7.13.

Form 2: $\gamma = \pi_0 L_0^{l_0} \pi_1 L_1^{l_1} \pi_2$, where π_0,π_1,π_2 are no longer than c, L_0 is a loop with effects $\Delta'(L_0) > 0$ and $\Delta(L_0) \leq 0$, L_1 is a loop with effects $\Delta'(L_1) < 0$ and $\Delta(L_1) < 0$, and $\Delta(L_0)/\Delta'(L_0) > \Delta(L_1)/\Delta'(L_1)$. To show the existence of such a path, it suffices to show that L_0 can be iterated at least once and that afterwards, the initial states of L_1 can be reached. The latter can be checked in $\log(K)$ space, as it is just reachability in the finite (product) control graph. The conditions on the effects can also be checked in logspace, because all guessed paths are polynomially bounded in K and their effects can thus be stored in $\log(K)$. The existence of a witness for $pm \not\leq p'm'$ of this form can thus be shown by providing paths $\alpha\beta\pi_0 L_0$, as well as L_1 satisfying the above conditions and $-\Delta(\alpha) = m$. Note that $|\beta\pi_0 L_0|$ is no longer than $(2K^2 + K^5) + c$.

Form 3: $\gamma = \pi_0 L_0^{l_0} \pi_1$, where L_0 is a loop with effects $\Delta'(L_0) < 0$ and $\Delta(L_0) < 0$ and $|\pi_0\pi_1| \leq c$. Because initially, the path γ starts with counter value 0 in $\mathcal{A}$, this means that the decreasing loop L_0 cannot be iterated more than c times. The whole path γ is therefore no longer than $c + K^2 \cdot c$. To check if a witness of this form exists, we can check (in logspace) if there is a path $\alpha\beta\gamma$ with $-\Delta(\alpha) = m$, $\Delta(\beta\gamma) \geq 0$, $\Delta'(\beta\gamma) < 0$.

We have considered all possible forms of witnesses for $pm \not\leq p'm'$, and in each case demonstrated that the existence of a witness of this form can be verified using logarithmic space. □

We turn to the problem OCN $\subseteq$ DOCA, where zero-tests are only allowed in the system on the right. There is a subtle difference to the problem DOCA $\subseteq$ OCN considered earlier in this section, regarding the applicability of Theorem 7.6. We present here only a partial solution.

Let $\mathcal{A} = (Q,\mathrm{Act},\delta)$ be a DOCN, $\mathcal{A}' = (Q',\mathrm{Act},\delta',\delta'_0)$ be a DOCA and $K = |Q \times Q'|$. Any witness for $p_0m_0 \not\leq p'_0m'_0$ can be split into a positive prefix, along which no zero-testing transitions are used, and the remaining path, that starts with a zero-testing transition of $\mathcal{A}'$. It is true that intermediate positions $(q_1n_1,q'_1 0)$ and $(q_2n_2,q'_2 0)$ along the witness are comparable if $(q_1,q'_1) = (q_2,q'_2)$, and due to the monotonicity in the net $\mathcal{A}$ (Lemma 3.13 on page 27), such a repetition of control states implies a strict increase in the counter values: $n_1 < n_2$. However, the problem is bounding the length of the path between two such positions along a shortest witness. Here, we cannot directly derive a polynomial bound by applying Theorem 7.6, because the theorem does not deal with possible witnesses that end in (or further use) zero-tests. In order to adjust the proof of Theorem 7.6 so that it can be used here, one has to deal with the problem that a witness could depend on a zero-test from some position $(qn,q'0)$ where the value n has to be sufficiently high to enable the remaining witness. This means that

1. Cycles L with effects $\Delta(L) < 0 < \Delta'(L)$, that were considered "obviously bad" before, cannot so easily be dismissed.

2. It might be necessary to alternate between two different loops.

Example 7.20. Assume states W and W' of a DOCN $\mathcal{A}$ and a DOCA $\mathcal{A}'$,respectively, such that $Wn \subseteq W'0$ iff $n \geq 20$. Consider now two other states p_0 and q_0 of $\mathcal{A}$ and $\mathcal{A}'$, where the relevant parts of $\mathcal{A}$ and $\mathcal{A}'$ are given on the left and the middle of Figure 7.3 below. The systems are constructed so that whenever $p_0 n \not\subseteq q_0 m$, some witness exists that leads to a position $Wn \subseteq W'0$, where $n \geq 20$.

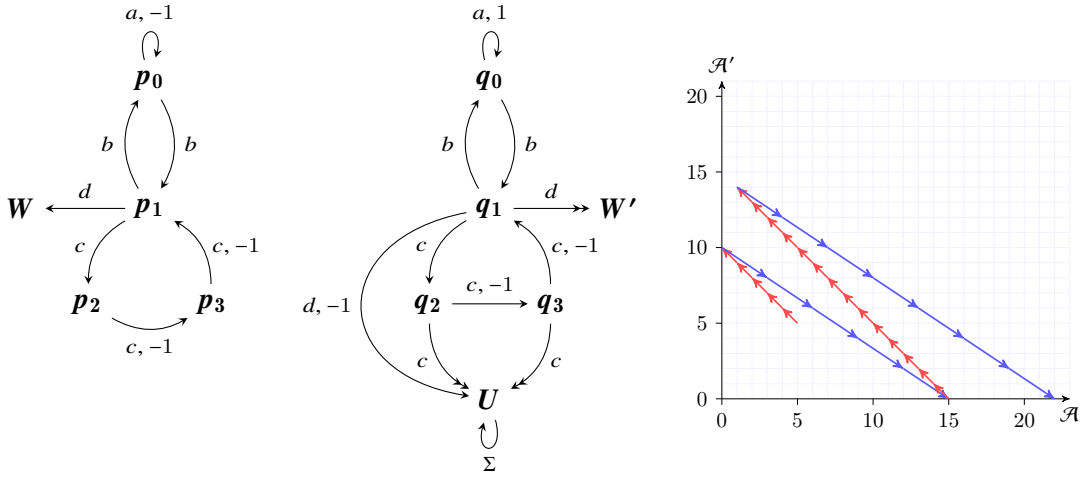


Figure 7.3: Part of DOCN $\mathcal{A}$ (on the left), part of DOCA $\mathcal{A}'$ (middle) and the change of counter value along the path π from position $(p_0 5, q_0 5)$. Notice the double-arrows indicating zero-testing transitions in $\mathcal{A}'$.

To illustrate the first problem above, observe that $p_0 2 \not\subseteq q_0 13$, but every witness uses the a-labelled loop $L = ((p_0,a,-1,p_0),(q_0,a,1,q_0))$ with effects $\Delta(L) = -1$ and $\Delta'(L) = 1$ (red in the picture). For the second problem, notice that $p_0 5 \not\subseteq q_0 5$, as witnessed for instance by a trace with the prefix $\pi = a^4 b (ccc)^5 b a^{14} b (ccc)^7 b a^8 b^4 d$, that leads to $(W22, W'0)$. The change in counter value induced by this witness is drawn on the right in Figure 7.3, where the a-labelled loop is coloured red and the c-labelled loop is blue. However, every witness for $p_0 5 \not\subseteq q_0 5$ must initially alternate between the a-labelled loop and the c-labelled loop.

Despite the additional complications, it seems reasonable that with further effort, one can show that certificates for non-inclusion based on short paths like in Theorem 7.6 are sufficient. We conjecture that the problem DOCN $\subseteq$ DOCA is also NL-complete, even with arbitrary initial counter values as part of the input.

We conclude this section with an NL upper bound for a subproblem that makes additional assumptions on how the initial counter values compare. Specifically, we demand that the counter values $m, m' \in \mathbb{N}$ for the given processes pm and $p'm'$ satisfy $m \geq K^3$ and $m' = 0$, where K is the size of the product of the input systems.

Lemma 7.21. *Let $p,q \in Q$, $p'q' \in Q'$. If there is a path from $(pm,p'0)$ to $(qn,q'0)$ for some $m,n \in \mathbb{N}$, then $(pk,p'0) \longrightarrow^k (qx,q'0)$ for $k \leq K^3$ and $x \geq 1$.*

Proof. Consider the automaton $\mathcal{B}$, that is the product of the control-graph of $\mathcal{A}$ with $\mathcal{A}'$, i.e. the product of $\mathcal{A}$ and $\mathcal{A}'$ that ignores the counter of $\mathcal{A}$. $\mathcal{B}$ is itself a OCA with K states. By assumption, for some $m,n \in \mathbb{N}$ there is a path $(pm,p'0) \longrightarrow^* (qn,q'0)$ in the product $\mathcal{A} \times \mathcal{A}'$. This means in $\mathcal{B}$, there is a valid path from $(p,p')0$ to $(q,q')0$. By Lemma 3.5 (page 21), there must also be such a path of length $k < K^3$, which in turn prescribes a path in the product of $\mathcal{A}$ and $\mathcal{A}'$. Because the absolute effect of a path is bounded by its length, this path must already be enabled in $(pk,p'0)$, which implies the claim. □

Theorem 7.22. *Suppose we are given a OCN $\mathcal{A} = (Q,Act,\delta)$, a DOCA $\mathcal{A}' = (Q',Act,\delta',\delta'_0)$ and processes pm and $p'm'$ of $\mathcal{A}$ and $\mathcal{A}'$, respectively, where $m \geq |Q \times Q'|^3$ amd $m' = 0$. There is a procedure that uses $O(\log(|Q \times Q'|))$ space to decide if $pm \subseteq p'm'$.*

Proof. Assume w.l.o.g. that $\mathcal{A}'$ is complete, so every witness for $pm \not\subseteq p'm'$ ends in some position $(qn,q'0)$, where some transition $t \in \delta$ allows a step in $\mathcal{A}$ and no equally labelled transition of $\mathcal{A}'$ is enabled. By Lemma 7.21, there is a path π that takes $(pm,p'0)$ to $(qx,q'0)$, for some $x \geq 1$, and which is shorter than K^3. Our assumption that $m \geq K^3$ means that this path must be enabled, as it cannot decrease the counter of $\mathcal{A}$ below 0 due to its length. Since $x \geq 1$, we must have $qx \xrightarrow{t}$, which means that π is a witness for $pm \not\subseteq p'm'$.

To check if $pm \subseteq p'm'$ holds it therefore suffices to guess one of the finitely many possible witnesses with length bounded by K^3. This can be done stepwise where only the (binary encoded) effects on the counters and the current pair of control states are memorized. □

Chapter 8

Conclusion and Outlook

We studied the decidability and complexity of inclusion problems for processes definable by one-counter automata (OCA), focussing on well-known semantic partial orders: simulation preorder and trace inclusion, both in their ordinary strong and also their weak variants, that abstract from internal behaviour. Unlike some semantic equivalences that are decidable for much more expressive classes of systems, these notions are already undecidable for OCAs, a fairly limited model of computation. This motivated a more fine-grained investigation of further restricted models, in particular of inclusion problems between OCAs, one-counter nets (OCNs) and finite systems (NFAs) in all remaining combinations. Tables 8.1 to 8.3 summarize the state of the art for these problems and points to the relevant literature or theorems in this dissertation for the respective results.

We were especially interested in one-counter nets, that have access to an unbounded integer counter but lack the explicit zero-testing capability of OCAs. In terms of their expressibility, OCNs lie strictly between NFAs and OCAs. An important property of OCNs is that steps are monotone with respect to counter values. This implies that any two OCN processes with the same control state are comparable w.r.t. simulation: for any state p of a OCN, and for all $m,n \in \mathbb{N}$ where $m \leq n$, it holds that $pm \preceq pn$. Almost all our positive results crucially rely on this monotonicity property.

Simulation preorder. All problems studied in this thesis concern variations or extensions of one central game: simulation for OCNs. It was known [2, 29] that checking simulation is decidable for OCNs but already undecidable for the slightly more general model OCA [30]. Moreover, previous work by Jančar, Moller, and Sawa [30] showed that the maximal simulation relation relative to a given pair of nets is a semilinear set and one can effectively compute a semilinear representation. All previous arguments were based on the locality of the simulation condition, as well as a characterization now called the *Belt Theorem*, that captures a certain regularity of the maximal simulation relation. However, the proofs for this theorem were not

constructive and consequently, no bounds on the parameters in the derived semilinear description of simulation, and also no upper bound on the complexity of simulation checking were known.

Our first main result (Theorem 4.19) contributed to our understanding of strong simulation between OCN processes. We were able to pinpoint the exact complexity of this problem to PSPACE and showed that one can effectively compute a semilinear description of the maximal simulation relation of size exponential only in the total number of control states of the input nets. This was made possible by a new and constructive proof of the Belt Theorem (Theorem 4.2). Our proof yields polynomial bounds for the crucial parameters in the description of the maximal simulation. Based on this, we derived a nondeterministic decision procedure for the simulation checking problem, that works in polynomial space and thus matches the known PSPACE lower bound from Srba [47].

$\preceq$	NFA	OCN	OCA
NFA	**P-complete** [45]	**P-complete** UB: [33]	**PSPACE-complete** LB: [47] UB: [46]
OCN	**P-complete** UB: [33]	**PSPACE-complete** LB: [47] UB: [20], Theorem 4.19	**?**
OCA	**PSPACE-complete** LB: [47] UB: [46]	**?** decidable [1]	**undecidable** [30]

Table 8.1: Results for strong simulation checking between OCAs, OCNs, and NFAs. For example, the cell on the lower left states that the problem OCA $\preceq$ NFA is PSPACE-complete and that lower and upper bounds are due to [47] and [46] respectively.

The PSPACE upper bound for checking strong and weak simulation between NFAs and OCAs is due to the fact that the μ-calculus model checking problem for OCAs is in PSPACE [46]. This is because finite systems and also the strong and weak simulation conditions can be expressed as μ-calculus formulae [26].

There are still some gaps in the knowledge about simulation between OCAs and OCNs, where one of the two given processes is allowed to have zero-tests.

We were recently able to establish the decidability of the problem OCA $\preceq$ OCN [1]. It turns out that simulation problems PDA $\preceq$ VASS between pushdown automata and vector addition systems correspond to multidimensional *pushdown energy games*: these are two-player games played on a (shared) pushdown graph, where the goal of the energy player (Duplicator) is to forever maintain non-negative values in all energy dimensions. One can show [1], that simulation between PDAs and VASSs is undecidable even in the 1-dimensional case (OCN), and simulation between OCAs and VASSs becomes undecidable from dimension 2 on. However, 1-dimensional energy games on OCAs, which correspond to the simulation problem

OCA $\preceq$ OCN are decidable: based on the monotonicity of steps in the net, we observe a certain regularity of the maximal simulation relation and derive that it is semilinear. The decidability then follows from this, and the finite branching property of OCNs, using two complementary semi-decision procedures that search for certificates. The exact complexity of this problem remains open. Our argument does not immediately provide bounds that would allow an effective construction of the whole relation. Also, it is not clear yet how to extend this technique to weak simulation.

Weak simulation. Our second main result was the PSPACE-completeness of weak simulation for OCNs (Theorem 5.28). The decidability of weak simulation for OCNs was published in [20] and the improvement to PSPACE-completeness appeared subsequently in [19]. Sections 5.1 to 5.4 contain a simplified and unified presentation of these results.

This positive result is particularly surprising if one recalls that, for most types of systems, checking bisimulation is computationally easier than simulation: In [35], Kučera and Mayr showed how to construct reductions from simulation to bisimulation checking, that are effective for a wide range of systems including OCNs. For OCNs, strong bisimulation is decidable and NL-complete, even for OCAs [6], but *weak* bisimulation is undecidable [37].

The main complication with weak simulation arises from the fact that with regard to weak steps, the LTSs induced by a OCNs can be infinitely branching. One consequence is that Kučera and Mayr's reductions are not effective. Moreover, the standard weak simulation approximants (cf. Definition 2.20) do not converge at level ω, which means there is no trivial semi-decision procedure for non-inclusion. We later showed (in Theorem 5.18), that weak simulation approximants are only guaranteed to converge at level ω^2.

Our approach for weak simulation on OCNs is based on previous results for strong simulation, particularly on the effectiveness (and bounds) of the semilinear description of the maximal simulation relation. We showed that weak simulation between OCNs can in fact be approximated finitely, using a customized notion of approximants. These approximant relations converge at a polynomially bounded level and moreover, can be represented in terms of strong simulation between OCNs. The fact that the maximal simulation can be effectively computed makes it possible to construct representations of our approximants, and hence of the largest weak simulation for the original input nets.

At the end of Chapter 5, we considered the remaining gaps regarding the complexity of checking strong and weak simulation between OCA/OCN and finite systems. In particular, we showed (Theorems 5.29 and 5.35) polynomial-time completeness of weak simulation between OCNs and NFAs, in both directions.

The decidability of the problems OCA $\leqq$ OCN and OCN $\leqq$ OCA, weak simulation between OCAs and OCNs remains open. These problems are of course PSPACE-hard, but unlike the

$\leqq$	NFA	OCN	OCA
NFA	**P-complete** LB: [45]	**P-complete** UB: [20], Theorem 5.35	**PSPACE-complete** LB: [47] UB: [46]
OCN	**P-complete** UB: [20], Theorem 5.29	**PSPACE-complete** LB: [47] UB: [20], Theorem 5.28	**?**
OCA	**PSPACE-complete** LB: [47] UB: [46]	**?**	**undecidable** [30]

Table 8.2: Weak simulation problems between OCAs, OCNs and NFAs.

corresponding strong simulation problems are not known to have semi-decidable complements.

Trace inclusion. In Chapter 6 we discussed strong and weak trace inclusion. We compared trace inclusion with the classical notion of language inclusion from automata theory, and explicitly recovered some negative results about OCA languages from [50]: both trace inclusion for DOCA and trace universality for OCAs are undecidable (Corollaries 6.5 and 6.6). We then showed (Theorem 6.10) the undecidability of trace inclusion between OCNs, by reduction from the undecidable containment problem for weighted finite automata [3]. Table 8.3 lists the known results for checking trace inclusion between systems of OCAs, OCNs and NFAs.

$\subseteq$ / $\subseteqq$	NFA	OCN	OCA
NFA	**PSPACE-complete** [38]	**Ackermann-complete** Theorem 6.25	**undecidable** Corollary 6.6
OCN	**PSPACE-complete** UB: [20], Theorem 6.12	**undecidable** [20], Theorem 6.10	**undecidable**
OCA	**PSPACE-complete** UB: [20], Theorem 6.12	**undecidable**	**undecidable**

Table 8.3: Results for strong and weak trace inclusion checking between OCAs, OCNs and NFAs.

The picture for weak trace inclusion is the same: Lower bounds and undecidability results trivially propagate from strong to weak trace inclusion. The $\mathfrak{F}_\omega$ upper bound for trace universality of OCNs (and the problem NFA $\subseteq$ OCN) also holds for trace universality and NFA $\subseteqq$ OCN (see Lemma 6.13 and Remark 6.19). Our PSPACE upper bound (Theorem 6.12) for the problem OCA $\subseteq$ NFA also holds for the corresponding weak trace inclusion problem, because one can (in logspace) compute the weak closure for NFAs.

The Deterministic Case. For deterministic systems, simulation and trace inclusion coincide. One can show (cf. Lemma 3.10), that this is already the case if only the supposedly larger

process on the right is deterministic. These problems essentially become reachability problems in the synchronous products of the two input systems. A trivial NL-lower bound for all these inclusions therefore follows from NFA $\subseteq$ DFA, which is just reachability in a finite (product) graph, and therefore NL-complete. Table 8.4 gives an overview of the results on checking strong and weak simulation/trace inclusion for deterministic systems.

Our main result in this area is an NL upper bound for OCN $\subseteq$ DOCN, inclusion between a OCN and a deterministic OCN (Theorem 7.14).

Similar questions were addressed by Higuchi, Tomita, and Wakatsuki [17, 15, 16]: they considered different notions of language inclusion between DOCNs, assuming fixed initial counter values. In each case, they computed polynomial bounds on the length of shortest witnesses for non-inclusion and proposed polynomial-time procedures that exhaustively search for a witness of bounded length [17, 15, 16]. Other than these results, our NL-upper bound for OCN $\subseteq$ DOCN holds even with (binary encoded) initial counter values as part of the input. In this case, shortest witnesses can be exponential in the size of the input. Our proof uses a non-trivial characterization of sufficient witnesses for non-inclusion (Theorem 7.6). We were able to extend our result to the problem OCA $\subseteq$ DOCN (Theorem 7.19) and under certain additional assumptions also for OCN $\subseteq$ DOCA (Theorem 7.22), where one of the sides allows zero-testing transitions. The decidability of the unrestricted problem OCN $\subseteq$ DOCA is still open.

$\preceq = \subseteq$	DFA	DOCN	DOCA
NFA	**NL-complete**	**NL-complete**	**NL-complete**
OCN	**NL-complete**	**NL-complete** Theorem 7.14	**NL-complete*** Theorem 7.22
OCA	**NL-complete**	**NL-complete** Theorem 7.19	**undecidable** [50], Corollary 6.5

Table 8.4: simulation/trace inclusion between OCA, OCN and NFA and their respective deterministic variants. Note that weak trace inclusion NFA $\subsetneqq$ DOCA (and vice versa) correspond to the reachability problem for OCA, which is NL-complete by Theorem 3.8.

An NL upper bound for OCA $\subsetneqq$ DFA again follows from the fact that one can compute the weak closure for DFAs in logspace and from the NL procedure for OCA reachability (Theorem 3.8). One can use silent (τ-labelled) transitions in DOCNs to simulate nondeterministic choice with respect to weak steps (see Section 7.1.1). This means that the lower bounds for trace inclusion between nondeterministic systems (Table 8.3) also hold for weak trace inclusion between deterministic systems.

Bibliography

[1] P. A. Abdulla, M. F. Atig, P. Hofman, R. Mayr, K. N. Kumar, and P. Totzke. "Infinite-State Energy Games". In: *LICS*. 2014.

[2] P. A. Abdulla and K. Čerāns. "Simulation Is Decidable for One-Counter Nets". In: *CONCUR*. Vol. 1466. LNCS. 1998, pp. 253–268.

[3] S. Almagor, U. Boker, and O. Kupferman. "What's Decidable About Weighted Automata?" In: *ATVA*. Vol. 6996. LNCS. 2011, pp. 482–491.

[4] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. 1st. New York, NY, USA: Cambridge University Press, 2009.

[5] J. F. A. K. van Benthem. "Modal Correspondence Theory". PhD thesis. Mathematisch Instituut & Instituut voor Grondslagenonderzoek, 1977.

[6] S. Böhm, S. Göller, and P. Jančar. "Equivalence of Deterministic One-Counter Automata is NL-complete". In: *STOC*. 2013, pp. 131–140.

[7] R. Bonnet. "The Reachability Problem for Vector Addition System with One Zero-Test". In: *MFCS*. Vol. 6907. LNCS. 2011, pp. 145–157.

[8] J. Bradfield and C. Stirling. "Modal mu-calculi". In: *Handbook of Modal Logic*. Vol. 3. Studies in Logic and Practical Reasoning. Elsevier, 2007. Chap. 12, pp. 721–756.

[9] O. Burkart, D. Caucal, F. Moller, and B. Steffen. "Verification on Infinite Structures". In: *Handbook of Process Algebra*. Elsevier, 2001. Chap. 9, pp. 545–623.

[10] E. M. Clarke, E. A. Emerson, and A. P. Sistla. "Automatic Verification of Finite-state Concurrent Systems Using Temporal Logic Specifications". In: *ACM Trans. Program. Lang. Syst.* 8.2 (Apr. 1986), pp. 244–263.

[11] S. Demri and R. Lazić. "LTL with the freeze quantifier and register automata". In: *ACM Trans. Comput. Logic* 10.3 (Apr. 2009), 16:1–16:30.

[12] J. Esparza and M. Nielsen. "Decidability Issues for Petri Nets - a Survey". In: *Elektronische Informationsverarbeitung und Kybernetik* 30.3 (1994), pp. 143–160.

[13] D. Figueira, S. Figueira, S. Schmitz, and P. Schnoebelen. "Ackermannian and Primitive-Recursive Bounds with Dickson's Lemma". In: *LICS*. 2011, pp. 269–278.

[14] R. v. Glabbeek. “The Linear Time – Branching Time Spectrum I; The Semantics of Concrete, Sequential Processes”. In: *Handbook of Process Algebra*. Elsevier, 2001. Chap. 1, pp. 3–99.

[15] K. Higuchi, E. Tomita, and M. Wakatsuki. “A Polynomial-Time Algorithm for Checking the Inclusion for Strict Deterministic Restricted One-Counter Automata”. In: *IEICE Transactions* 78-D.4 (1995), pp. 305–313.

[16] K. Higuchi, M. Wakatsuki, and E. Tomita. “A Polynomial-Time Algorithm for Checking the Inclusion for Real-Time Deterministic Restricted One-Counter Automata Which Accept by Accept Mode”. In: *IEICE Transactions* E81-D.1 (Jan. 1998), pp. 1–11.

[17] K. Higuchi, M. Wakatsuki, and E. Tomita. “A Polynomial-Time Algorithm for Checking the Inclusion for Real-Time Deterministic Restricted One-Counter Automata Which Accept by Final State”. In: *IEICE Transactions* 78-D.8 (1995), pp. 939–950.

[18] K. Higuchi, M. Wakatsuki, and E. Tomita. “Some Properties of Deterministic Restricted One-Counter Automata”. In: *IEICE Transactions* E79-D.8 (July 1996), pp. 914–924.

[19] P. Hofman, S. Lasota, R. Mayr, and P. Totzke. “Simulation Over One-Counter Nets is PSPACE-Complete”. In: *FSTTCS*. 2013, pp. 515–526.

[20] P. Hofman, R. Mayr, and P. Totzke. “Decidability of Weak Simulation on One-Counter Nets”. In: *LICS*. 2013, pp. 203–212.

[21] M. Holzer. “On Emptiness and Counting for Alternating Finite Automata”. In: *DLT*. 1995, pp. 88–97.

[22] J. E. Hopcroft and J. D. Ullman. *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley, 1979.

[23] R. R. Howell, L. E. Rosier, D. T. Huynh, and H.-C. Yen. “Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states”. In: *TCS* 46 (1986), pp. 107–140.

[24] P. Jančar. “Undecidability of Bisimilarity for Petri Nets and Some Related Problems”. In: *TCS* 148.2 (Sept. 1995), pp. 281–301.

[25] P. Jančar, J. Esparza, and F. Moller. “Petri Nets and Regular Processes”. In: *JCSS* 59.3 (Dec. 1999), pp. 476–503.

[26] P. Jančar, A. Kučera, and R. Mayr. “Deciding Bisimulation-Like Equivalences with Finite-State Processes”. In: *ICALP*. 1998, pp. 200–211.

[27] P. Jančar, A. Kučera, and F. Moller. “Simulation and Bisimulation over One-Counter Processes”. In: *STACS*. Vol. 1770. LNCS. 2000, pp. 334–345.

[28] P. Jančar and F. Moller. “Checking Regular Properties of Petri Nets”. In: *CONCUR*. 1995, pp. 348–362.

[29] P. Jančar and F. Moller. “Simulation of One-Counter Nets via Colouring”. Tech. rep. Uppsala Computing Science, Feb. 1999.

[30] P. Jančar, F. Moller, and Z. Sawa. “Simulation Problems for One-Counter Machines”. In: *SOFSEM*. Vol. 1725. LNCS. 1999, pp. 404–413.

[31] P. Jančar and Z. Sawa. “A note on emptiness for alternating finite automata with a one-letter alphabet”. In: *Information Processing Letters* 104.5 (2007), pp. 164–167.

[32] D. Kozen. “Results on the propositional μ-calculus”. In: *TCS* 27.3 (1983), pp. 333–354.

[33] A. Kučera. “On Simulation-Checking with Sequential Systems”. In: *ASIAN*. Vol. 1961. LNCS. 2000, pp. 133–148.

[34] A. Kučera and P. Jančar. “Equivalence-checking on Infinite-state Systems: Techniques and Results”. In: *TPLP* 6.3 (2006), pp. 227–264.

[35] A. Kučera and R. Mayr. “Why is Simulation Harder Than Bisimulation?” In: *CONCUR*. Vol. 2421. 2002, pp. 594–609.

[36] A. Levy. *Basic Set Theory*. Dover Books on Mathematics. Dover Publications, 2012.

[37] R. Mayr. “Undecidability of Weak Bisimulation Equivalence for 1-Counter Processes”. In: *ICALP*. Vol. 2719. LNCS. 2003, pp. 570–583.

[38] A. R. Meyer and L. J. Stockmeyer. “The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space”. In: *Symp. Foundations of Computer Science*. IEEE, 1972, pp. 125–129.

[39] R. Milner. “An Algebraic Definition of Simulation Between Programs”. In: *IJCAI*. 1971, pp. 481–489.

[40] R. Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989, pp. I–XI, 1–260.

[41] M. L. Minsky. *Computation: finite and infinite machines*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1967.

[42] C. M. Papadimitriou. *Computational complexity*. Reading, Massachusetts: Addison-Wesley, 1994.

[43] D. Park. “Concurrency and automata on infinite sequences”. In: *TCS*. Vol. 104. LNCS. 1981, pp. 167–183.

[44] K. Reinhardt. “Reachability in Petri Nets with Inhibitor Arcs”. In: *ENTCS* 223 (2008), pp. 239–264.

[45] Z. Sawa and P. Jančar. “P-hardness of Equivalence Testing on Finite-State Processes”. In: *Proceedings of SOFSEM*. Vol. 2234. LNCS. 2001.

[46] O. Serre. “Parity Games Played on Transition Graphs of One-Counter Processes”. In: *FoSSaCS*. 2006, pp. 337–351.

[47] J. Srba. “Beyond Language Equivalence on Visibly Pushdown Automata”. In: *Logical Methods in Computer Science* 5.1 (2009), pp. 1–22.

[48] J. Srba. *Roadmap of Infinite results*. Vol. 2: Formal Models and Semantics. World Scientific Publishing Co., 2004, pp. 337–350.

[49] C. Stirling. “The Joys of Bisimulation”. In: *MFCS* 1450 (1998), pp. 142–151.

[50] L. G. Valiant. “Decision Procedures for Families of Deterministic Pushdown Automata”. PhD thesis. University of Warwick, July 1973.

[51] L. G. Valiant and M. S. Paterson. “Deterministic One-Counter Automata”. In: *JCSS* 10.3 (1975), pp. 340–350.

[52] M. Y. Vardi and P. Wolper. “An automata-theoretic approach to automatic program verification”. In: *LICS*. 1986.