University of Warsaw
Faculty of Mathematics, Informatics and Mechanics

Piotr Hofman

Equivalence of infinite-state systems with silent steps

PhD dissertation

Supervisor

prof. dr hab. Sławomir Lasota

Institute of Mathematics
University of Warsaw

June 2013

Author's declaration:
aware of legal responsibility I hereby declare that I have written this dissertation myself and all the contents of the dissertation have been obtained by legal means.

June 7, 2013 ....
date *Piotr Hofman*

Supervisor's declaration:
the dissertation is ready to be reviewed

June 7, 2013 ....
date *prof. dr hab. Sławomir Lasota*

Abstract

This dissertation contributes to analysis methods for infinite-state systems. The dissertation focuses on equivalence testing for two relevant classes of infinite-state systems: commutative context-free processes, and one-counter automata. As for equivalence notions, we investigate the classical bisimulation and simulation equivalences. The important point is that we allow for *silent steps* in the model, abstracting away from internal, unobservable actions. Very few decidability results have been known so far for bisimulation or simulation equivalence for infinite-state systems with silent steps, as presence of silent steps makes the equivalence problem arguably harder to solve.

A standard technique for bisimulation or simulation equivalence testing is to use the hierarchy of *approximants*. For an effective decision procedure the hierarchy must stabilize (converge) at level ω, the first limit ordinal, which is not the case for the models investigated in this thesis. However, according to a long-standing conjecture, the community believed that the convergence actually takes place at level $\omega+\omega$ in the class of commutative context free processes. In Chapter 2 we disprove the conjecture and provide a lower bound of $\omega\cdot\omega$ for the convergence level. We also show that all previously known positive decidability results for commutative context-free processes can be re-proven uniformly using the improved approximants techniques.

The following Chapter 3 is about an unsuccesfull attack on one of the main open problems in the area: decidability of weak bisimulation equivalence for commutative context-free processes. Our technical development of this section is not sufficient to solve the problem, but we believe it is a serious step towards a solution. Furtermore, we are able to show decidability of branching (stuttering) bisimulation equivalence, a slightly more discriminating variant of bisimulation equivalence. It is worth emphesizing that, until today, our result is the only known decidability result for bisimulation equivalence in a class of inifinite-state systems with silent steps that is not known to admit convergence of (some variant of) standard approximants at level ω.

In Chapter 4, the last one, we consider weak simulation equivalence over one-counter automata without zero tests (allowing zero tests implies undecidability). While weak bisimulation equivalence is known to be undecidable in this class, we prove a surprising result that weak simulation equivalence is actually decidable. Thus we provide a first example going against a trend, widely-believed by the community, that simulation equivalence tends to be computationally harder than bisimulation equivalence.

In short words, the dissertation contains three new results, each of them solving a non-trivial open problem about equivalence testing of infinite-state systems with silent steps.

Acknowledgements

First of all I would like to express my great thanks to my supervisor, Sławomir Lasota. He put a lot of effort teaching me what does it mean to be a scientist, and what are principles of working as a researcher. He shared with me enormous amount of knowledge. I could always rely on him with my problems, even those not scientific. I admire him for a lot of patience to read and to correct my writings. I could not imagine a better mentor and I am extremely grateful for those last four years.

I would also like to thank my friends with whom I worked: for their ideas and questions, for solving some of my problems, and constantly motivating me to work. Wojciech Czerwiński and Partick Totzke, thank you guys.

Furthermore, I am very grateful to Richard Mayr, Partick's supervisor who helped us a lot in our work on one-counter automata. I also wish to thank Diego Figueira with whom I had a joy to work during my first year of studies. Moreover I would like to thank Mikołaj Bojańczyk and Damian Niwiński, for their advices, good example and support.

Finally, huge thanks to my family and friends.

Contents

Chapter 1

Introduction

We observe recently an increasing interest in methods of formal analysis of computer systems. A typical approach is twofold: first, a real system is replaced by a finite-state abstract model, and then the model is analyzed against a correctness criterion. The number of states of a model is a crucial parameter that one wants to minimize, as the state-space explosion is the major bottleneck in efficient analysis. One way of avoiding this negative phenomenon is replacing a model by an *equivalent* one. This leads to investigation of equivalences of systems, with the emphasis on effective decision procedures.

Finite-state abstractions are sometimes too weak, as computer systems typically exhibit infinite-state behaviours. Typical reasons are unboundedness of recursion depth or the number of concurrently executed threads; other possible reasons behind infinite state spaces are, among the others, data of unbounded size, or elapse of time. Moreover, as finite-state abstractions tend to grow in size, their formal verification becomes often infeasible. This is why it is sometimes arguably more efficient to keep the infinity of state-space, since it enforces usage of symbolic techniques instead of enumerative ones. As a toy illustrating example, checking emptiness of a pushdown automaton using a symbolic algorithm is often less costly than testing emptiness of its finite-state abstraction obtained by bounding the stack depth, using an exhaustive state-space exploration.

In this dissertation we investigate two classes of infinite-state systems: commutative context-free processes, and one-counter automata. The former model is a special case of labelled Petri nets, while the latter one is a special case of pushdown automata. In both cases, we allow for silent (i.e. unobservable) steps that abstract away from internal actions of a system. Presence of silent steps makes analysis arguably harder. This claim is confirmed by a very few known decidability results for infinite-state systems with silent steps. As a consequence, there is still a number of open problems in this area that call for a solution.

The topic of this dissertation is decidability of equivalence-checking for commutative context-free processes and one-counter automata. We concentrate on two notions of equivalences, namely bisimulation equivalence and simulation equivalence, as most of the other semantics equivalence are undecidable in the models considered in this dissertation.

A standard technique for bisimulation or simulation equivalence checking is to use the hierarchy of approximating relations (called *approximants* in this dissertation), indexed by ordinals. For an effective decision procedure the hierarchy must stabilize (converge) at level ω, which is not the case for the models we consider. However, according to a long-standing conjecture, the community believed that the stabilization actually takes place at level $\omega+\omega$ for commutative context-free processes. We disprove the conjecture, by providing a suitable counterexample, and thus establishing a new lower bound of $\omega\cdot\omega$ for the convergence level. We also show that all decidability results for commutative context-free processes, known prior to this dissertation, may be uniformly shown using the hierarchies of standard approximants that stabilize at ω.

Technically, the most involved part of this dissertation is an unsuccessful attempt at attacking one of the main open problems in the area: decidability of weak bisimulation equivalence for commutative context-free processes. Our technical development is not sufficient to solve the problem, but we feel it is a serious step towards a solution. On the other hand, our framework is sufficient to show decidability of branching (stuttering) bisimulation equivalence, a slightly more discriminating variant of bisimulation equivalence. It is worth emphasizing that, up to date, our result is the only known decidability result for bisimulation equivalence in a class of infinite-state systems with silent steps that is not known to admit convergence of (some variant of) standard approximants at level ω.

Finally, we consider also simulation equivalence and pre-order over one-counter automata *without* zero tests (for one-counter automata with zero tests, simulation equivalence and pre-order are undecidable). While weak bisimulation equivalence is know to be undecidable in this class, we prove a surprising result that weak simulation equivalence (as well as weak simulation pre-order) is actually decidable. Thus we provide the first example going against a trend, widely-believed by the community, that simulation equivalence tends to be computationally harder than bisimulation equivalence.

The remaining sections of Chapter 1 are devoted to concisely present the main results of this dissertation. We start by introducing the necessary terminology and definitions, in Section 1.1. Then in Section 1.2 we outline briefly the content of the following chapters. Afterwards, in Section 1.3 we sketch the wider context of our research. We also briefly survey state of the art of the field, with emphasis on the relationship between previously known results and the results presented in this dissertation. Most of the results contained in this dissertation have been already published in [13, 28, 27].

1.1 Basic Definitions

1.1.1 Processes

All equivalences investigated by us may be considered over any labelled transition system. The definitions below assume a finite alphabet $\mathbb{A}ct$ of *transition labels*, or *actions*. In addition, we always assume that *silent transitions* are labelled with a special label ε, such that $\varepsilon \notin \mathbb{A}ct$. We shortly write $\mathbb{A}ct_\varepsilon$ for $\mathbb{A}ct \cup \{\varepsilon\}$.

Definition 1. *A* labelled transition system *(*LTS *in short) over a finite alphabet* $\mathbb{A}ct$ *is a tuple* (N, E) *consisting of a possibly infinite set of nodes* N *and a set* $E \subseteq N \times \mathbb{A}ct_\varepsilon \times N$ *of edges labelled with symbols from* $\mathbb{A}ct_\varepsilon$*.*

Note that the definition allows for self loops and for multiple edges (with different labels) between a pair of nodes. In the sequel we write $v \xrightarrow{\zeta} v'$ instead of $(v, \zeta, v') \in E$. For nodes of LTS we will use the name *processes*, and edges will be called *transitions*.

The silent transitions $\xrightarrow{\varepsilon}$ will be written concisely as $\longrightarrow$ in the sequel, and the reflexive-transitive closure of $\xrightarrow{\varepsilon}$ will be written as $\Longrightarrow$. Thus $\alpha \Longrightarrow \beta$ if a process β can be reached from α by a sequence of $\xrightarrow{\varepsilon}$ transitions.

One typically investigates LTSs that are finitely represented, i.e., induced by a finite description, like a grammar or an automaton. In this thesis we focus on LTSs induced by commutative context-free grammars, or by one-counter automata. In both cases, an infinite set of transitions will be induced by a finite set of *transition rules*.

Commutative context-free processes. Fix attention to grammars in Greibach normal form. A commutative context-free grammar is like an ordinary one, with two differences: first, there is no distinguished initial symbol; second, the right-hand sides of productions are interpreted as multisets, not words, over the set of nonterminal symbols. In the formal definition below we deliberately use terminology typically used in process algebra community, rather that that used in formal languages community.

Definition 2. *A commutative context free grammar consists of: a finite alphabet* $\mathbb{A}ct$*, a finite set* V *of variables, and a finite set of transition rules, each of the form* $X \xrightarrow{\zeta} \alpha$*, where* $X \in V$*,* $\zeta \in \mathbb{A}ct_\varepsilon$*, and* α *is a finite multiset of variables.*

A *process* is any finite multiset of variables, thus a mapping that assigns a finite nonnegative multiplicity to each variable, and may be understood as the parallel composition of a given number of copies of respective variables. In particular the *empty process*, denoted ε, is the empty multiset. Let $V^\otimes$ denote the set of all processes, $V^\otimes = \mathbb{N}^V$.

In this thesis we are interested in LTSs induced by grammars, instead of the generated languages. Denote by $\alpha || \beta$ the composition of processes α and β, understood as the multiset union. A grammar induces an LTS, with the set of processes $V^{\otimes}$, and with transitions defined as follows: for any transition rule $X \xrightarrow{\zeta} \alpha$ in the grammar and for every process β, the LTS has a transition

$$\beta || X \xrightarrow{\zeta} \alpha || \beta.$$

Example 1. *As an illustration consider a grammar over an alphabet $\mathbb{A}ct = \{a, b\}$ (recall that we omit the label ε of silent transition rules):*

$$\begin{array}{ccccccccccccccc} P & \longrightarrow & \varepsilon & \quad & P & \longrightarrow & P||A & \quad & P & \xrightarrow{b} & \varepsilon & \quad & P & \longrightarrow & Q \\ Q & \longrightarrow & \varepsilon & \quad & Q & \longrightarrow & Q||A & \quad & A & \xrightarrow{a} & \varepsilon & \quad & A & \longrightarrow & \varepsilon \end{array}$$

The induced LTS contains, among other transitions, the following sequence of transitions:

$$Q||P \longrightarrow Q||P||A \longrightarrow Q||P||A||A \longrightarrow P||A||A \xrightarrow{b} A||A.$$

Writing A^k for the k-ary composition of the process A,

$$A^k \; = \underbrace{A|| \ldots ||A}_{k \text{ times}},$$

one obtains, for any k, the transitions $Q||P \Longrightarrow P||A^k \xrightarrow{b} A^k$.

One-counter automata. Now we define one-counter automata, and LTSs they induce. For the purpose of the dissertation it is sufficient to define automata *without zero tests*.

Definition 3. *A one-counter automaton without zero tests consists of: a finite alphabet $\mathbb{A}ct$, a finite set of control-states Q, and a finite set of transition rules, each of the form $q \xrightarrow{\zeta, d} p$, where $p, q \in Q$, $\zeta \in \mathbb{A}ct_{\varepsilon}$, and $d \in \{-1, 0, 1\}$ encodes a change of the counter value along the transition.*

An automaton induces an LTS, with the set of processes $Q \times \mathbb{N}$; the processes will be written as qm. The transitions are defined as follows: for every transition rule $q \xrightarrow{\zeta, d} p$, and every $m \in \mathbb{N}$ with $m + d \geqslant 0$, the LTS has a transition

$$qm \xrightarrow{\zeta} p(m + d).$$

Example 2. *As an illustration consider an automaton over an alphabet $\mathbb{A}ct = \{a, b\}$, with only one state p and the following transition rules (again, labels of silent transition rules are omitted):*

$$p \quad \xrightarrow{a,1} \quad p \qquad\qquad p \quad \xrightarrow{b,-1} \quad p \qquad\qquad p \quad \xrightarrow{\varepsilon,-1} \quad p \qquad\qquad .$$

The induced LTS contains, among other transitions, the following sequence of transitions:

$$p0 \xrightarrow{a} p1 \xrightarrow{a} p2 \xrightarrow{b} p1 \xrightarrow{a} p2 \longrightarrow p1 \longrightarrow p0.$$

Normedness is, intuitively, capability of reaching the empty process or zero counter value.

Definition 4 (Normed processes)**.** *A commutative context-free grammar is normed if for every process there is a sequence of transitions to the empty one.*[1] *A one-counter automaton is normed if for every process there is a sequence of transitions to a process with zero counter value.*

For instance, the systems in Examples 1 and 2 are normed.

1.1.2 Simulation and bisimulation

To simplify the definitions, in all LTSs considered in this thesis we assume that every process α has additionally a silent self-loop $\alpha \longrightarrow \alpha$ (including the transition $\varepsilon \longrightarrow \varepsilon$).

Now we state definitions necessary to formulate our results, namely definitions of bisimulation equivalence (bisimilarity), as well as simulation pre-order and equivalence. It is widely known that bisimulation and simulation appears in various variants. In general there are many reasons for this proliferation, but in the dissertation we consider only one: different ways of treating silent transitions. The proliferation motivates us to separate the part of definition that varies from one variant to another (namely *expansion*), from the remaining invariant part, as illustrated in the definition of *weak*[2] and *branching* (bi)simulation below. All the definitions in Section 1.1 apply to an arbitrary fixed LTS (N, E).

We use the notation $\alpha \stackrel{\zeta}{\Longrightarrow} \beta$ to mean that $\alpha \Longrightarrow \gamma \xrightarrow{\zeta} \gamma' \Longrightarrow \beta$ for some γ, γ'. In particular, when $\zeta = \varepsilon$, the transition $\stackrel{\varepsilon}{\Longrightarrow}$ denotes a finite sequence of silent transitions, maybe the empty one (using a silent self-loop).

Definition 5. *Let $S \subseteq N \times N$. A pair (α, β) of processes satisfies* **weak simulation expansion** *wrt. S if for every $\zeta \in \mathbb{A}ct_{\varepsilon}$:*

$$\textit{if } \alpha \xrightarrow{\zeta} \alpha' \textit{ then } \beta \stackrel{\zeta}{\Longrightarrow} \beta' \textit{ such that } (\alpha', \beta') \in S.$$

Definition 6. *Let $B \subseteq N \times N$. A pair (α, β) satisfies* **weak bisimulation expansion** *wrt. B if (α, β) satisfies weak simulation expansion wrt. B, and (β, α) satisfies weak simulation expansion wrt. B^{-1}.*

[1] In other words, a grammar is normed if every nonterminal generates at least one word. The restriction, being irrelevant from the language perspective, becomes relevant from the LTS perspective.

[2] As opposed to *strong simulation/bisimulation*, which assumes that there are no silent transitions.

Unwinding the above definition one obtains: a pair (α, β) satisfies weak bisimulation expansion wrt. B if and only if for every $\zeta \in \mathbb{A}ct_\varepsilon$:

- if $\alpha \xrightarrow{\zeta} \alpha'$ then $\beta \xRightarrow{\zeta} \beta'$ such that $(\alpha', \beta') \in B$;
- if $\beta \xrightarrow{\zeta} \beta'$ then $\alpha \xRightarrow{\zeta} \alpha'$ such that $(\alpha', \beta') \in B$.

The expansion of Definition 5 is asymmetric while the one of Definition 6 is symmetric. In consequence, the former one will yield a pre-order on processes, while the latter one will yield an equivalence of processes, as defined below.

Definition 7. *$S \subseteq N \times N$ is* **weak simulation** *if every pair $(\alpha, \beta) \in S$ satisfies weak simulation expansion wrt. S.* **Weak pre-order** *is the union of all weak simulations.*

Definition 8. *$B \subseteq N \times N$ is* **weak bisimulation** *if every pair $(\alpha, \beta) \in B$ satisfies weak bisimulation expansion wrt. B.* **Weak equivalence** *is the union of all weak bisimulations.*

We thus prefer to write shortly *weak pre-order* and *weak equivalence*, instead of *weak simulation pre-order* and *weak bisimulation equivalence*, respectively.

In other words, processes α and β are related by weak pre-order, if there exists a weak simulation S containing (α, β). Similarly, processes α and β are weak equivalent, if there exists a weak bisimulation B containing (α, β). In general, weak equivalence in finer than the symmetric part of weak pre-order. In other words, two simulations in both directions do not imply existence of bisimulation. The symmetric part of weak pre-order is called weak simulation equivalence in the literature.

Proposition 1. *Weak pre-order is a pre-order indeed, and weak equivalence is an equivalence indeed.*

Proposition 2. *Weak pre-order is the greatest weak simulation, while weak equivalence is the greatest weak bisimulation.*

The latter proposition follows immediately from Knaster-Tarski Fixpoint Theorem. Indeed, weak simulation expansion defines a monotonic refinement function

$$S \quad \mapsto \quad \exp(S) \tag{1.1}$$

which takes an arbitrary binary relation S over processes to the relation

$$\exp(S) \ = \ \{(\alpha, \beta) : (\alpha, \beta) \text{ satisfies weak simulation expansion wrt. } S\}.$$

With this notation weak simulations are exactly those relations S that satisfy

$$S \subseteq \exp(S)$$

and weak pre-order is the greatest fixed point of the refinement function (1.1). Similar facts hold for weak bisimulation expansion as well.

In presence of silent transitions, weak simulation and bisimulation are not the unique choices. The best known among the competitors are *branching simulation* and *branching bisimulation*, to be defined now. We will exploit the fact that Definition 7 is universal, and makes sense not only for weak simulation expansion, but also for branching simulation expansion, to be defined now:

Definition 9. *Let $S \subseteq N \times N$. A pair (α, β) of processes satisfies the* **branching simulation expansion** *wrt. S if for every $\zeta \in \mathbb{A}ct_\varepsilon$:*

$$\textit{if } \alpha \xrightarrow{\zeta} \alpha' \textit{ then } \beta \Longrightarrow \bar{\beta} \xrightarrow{\zeta} \beta' \textit{ such that } (\alpha, \bar{\beta}) \in S \textit{ and } (\alpha', \beta') \in S.$$

(Note that $\bar{\beta}$ may be equal to β', due to silent self-loops.)

By substituting the new expansion in place of weak simulation expansion in Definition 7, we obtain *branching pre-order*. Furthermore, Definition 6 may be adapted similarly to the new expansion, yielding *branching bisimulation expansion*:

Definition 10. *Let $S \subseteq N \times N$. A pair (α, β) of processes satisfies the* **branching bisimulation expansion** *wrt. S if (α, β) satisfies branching simulation expansion wrt. S and (β, α) satisfies branching simulation expansion wrt. S^{-1}.*

Then one obtains in turn *branching equivalence*, due to the analogous adaptation of Definition 8. Propositions 1 and 2 still hold if 'weak' is replaced by 'branching'.

Branching equivalence (resp. pre-order) is finer than the weak counterpart, as it essentially requires that along a matching sequence of transitions $\beta \Longrightarrow \bar{\beta} \xrightarrow{a} \beta'$, the bisimulation class is only changed once (via the last transition). Indeed, if two branching equivalent processes β and $\bar{\beta}$ are related by a sequence of silent transitions $\beta \Longrightarrow \bar{\beta}$ then all intermediate processes are also equivalent to β (and $\bar{\beta}$)[3] . Thus in general there are the following inclusions between the relations (arrows stand for 'finer than'):

$$\begin{tikzcd}
 & \text{weak pre-order} & \\
\text{weak equivalence} \arrow[ur, dotted] & & \text{branching pre-order} \arrow[ul, dotted] \\
 & \text{branching equivalence} \arrow[ul, dotted] \arrow[ur, dotted] &
\end{tikzcd}$$

When there is no silent steps (apart from self-loops), weak and branching semantics clearly coincide. We may say thus about the bisimulation equivalence, and the simulation pre-order.

[3]The same fact holds for weak equivalence too. (Lemma 4.)

Example 3. *As an illustration of differences between weak equivalence and branching equivalence, consider the following grammar:*

$$\begin{array}{lllllllll} A & \longrightarrow & \varepsilon & A & \xrightarrow{a} & \varepsilon & & & \\ B & \longrightarrow & \varepsilon & B & \longrightarrow & A & B & \xrightarrow{a} & A \end{array}$$

and two processes B and $A||A$. The two processes are easily seen to be in weak equivalence. On the other hand, they are not branching equivalent (even not related by branching pre-order), as the transition

$$B \longrightarrow \varepsilon$$

can not be matched properly. A natural candidate for a matching sequence, namely the sequence $A||A \longrightarrow A \longrightarrow \varepsilon$, does not satisfy the requirement of branching bisimulation expansion, as its second last process A is not branching equivalent to B. Another possible candidate, namely $A||A \Longrightarrow \varepsilon \longrightarrow \varepsilon$, doesn't work either for similar reasons.

1.1.3 Decision problems

This thesis is about few instantiations of the following generic pattern of decision problems, parametrized by a class C of finitely representable LTSs and a notion $\equiv$ of equivalence or pre-order:

INPUT:	an LTS from the class C and two processes α and β
QUESTION:	Does $\alpha \equiv \beta$ hold ?

We focus on the following instantiations:

- weak and branching equivalence for commutative context-free processes;
- weak and branching pre-order for one-counter automata without zero tests.

Clearly, the choice of these decision problems is not accidental. Decidability status of the four problems has been unknown until recently. In this thesis we describe solutions of three of them; decidability of weak equivalence over commutative context-free processes remains open.[4] Decidability status of three of the other combinations, namely pre-orders on commutative context-free processes, and weak equivalence over one-counter automata, are already known (decidability of branching equivalence is still unknown). Furthermore, all other reasonable semantic equivalences or pre-orders are undecidable for both models, even without silent steps, as well as for other natural

[4]We claim however that we have done a substantial progress towards a solution of this long-standing open problem.

models, like pushdown automata or labeled Petri nets. The detailed discussion of the area, with emphasis on the relationship between our contribution and previously known results, is contained in Section 1.3.

1.1.4 Approximations

For any of expansions defined so far (and for many others) we may define a hierarchy of relations that approximate the equivalence (resp. pre-order) from above.

Definition 11 (approximants)**.** *Define the family of relations $R_\kappa \subseteq N \times N$ indexed by ordinals $\kappa \in Ord$.*

- $R_0 = N \times N$ *contains all pairs of processes* (α, β)*;*
- $(\alpha, \beta) \in R_{\kappa+1}$ *if and only if* (α, β) *satisfies expansion wrt.* R_κ*;*
- *for a limit ordinal* κ*, we define* $R_\kappa = \bigcap_{\lambda<\kappa} R_\lambda$.

Note that we could define $R_{\kappa+1}$ using the refinement function defined in (1.1) as $R_{\kappa+1} = \exp(R_\kappa)$.

The above definition is generic, as it does not specify which particular expansion is used. It may be actually instantiated with any of simulation or bisimulation expansions defined in Section 1.1.2. In each case, the hierarchy forms a decreasing chain of relations that finally stabilizes and hence converges to the appropriate notion of pre-order or equivalence, by Knaster-Tarski Fixpoint Theorem. Formally speaking:

Definition 12. *For a class of LTSs, we say that the hierarchy of approximants $(R_\kappa)_{\kappa \in Ord}$* **stabilizes** *(***converges***) at ordinal λ if for every LTS from the class, $R_\lambda = R_{\lambda+1}$.*

In general, however, the hierarchy may stabilize far beyond the first infinite ordinal ω. This is provably the case for weak and branching, simulation and bisimulation expansions, on the classes of LTSs induced by commutative context-free grammars or one-counter automata.

Example 4. *For illustration consider again the grammar form Example 3, and the hierarchy of approximants $(R_\kappa)_\kappa$ induced by weak bisimulation expansion. Clearly the pair (A, ε) is in R_0 but not in R_1, as ε has no a-transition. Similarly, the pair $(A||A, A)$ is in R_1 but not in R_2, as the only possibility of matching the transition $A||A \xrightarrow{a} A$ is $A \xrightarrow{a} \varepsilon$. Intuitively speaking, processes $A||A$ and A can not be distinguished at level 1 but can be so at level 2. Generally, if $n > m$ then the pair (A^n, A^m) is in R_m but not in R_{m+1}, thus the hierarchy is strict on finite ordinals. Finally, for this particular transition system the hierarchy stabilizes at ω, i.e. the relation R_ω coincides with weak equivalence.*

The aim of introducing approximants is that they are typically decidable for finite ordinals. On the other hand, decidability is usually hard to prove beyond ordinal ω. This motivates introducing *faster* approximants, which converge to the same relation. To state our results for weak equivalence we need to define two hierarchies of faster approximants. For one of them we need to extend the relation $\stackrel{a}{\Longrightarrow}$ to words $w \in \mathbb{A}ct^*$: let $\alpha \stackrel{a_1 \ldots a_n}{\Longrightarrow} \beta$ if $\alpha \stackrel{a_1}{\Longrightarrow} \gamma_1 \stackrel{a_2}{\Longrightarrow} \ldots \gamma_{n-1} \stackrel{a_n}{\Longrightarrow} \beta$ for some $\gamma_1, \ldots, \gamma_{n-1}$.

Formally speaking, the symbol $\stackrel{\varepsilon}{\Longrightarrow}$ is a notational clash as it has been actually defined twice: for the silent action $\varepsilon \in \mathbb{A}ct_\varepsilon$, as well as for the empty word $\varepsilon \in \mathbb{A}ct^*$. Fortunately, the two definitions coincide, and amount to a finite sequence of silent transitions.

Definition 13. *Let $S \subseteq N \times N$. A pair (α, β) of processes satisfies the* **long weak simulation expansion** *wrt. S if for every $\zeta \in \mathbb{A}ct_\varepsilon$:*

$$\textit{if } \alpha \stackrel{\zeta}{\Longrightarrow} \alpha' \textit{ then } \beta \stackrel{\zeta}{\Longrightarrow} \beta' \textit{ such that } (\alpha', \beta') \in S.$$

Definition 14. *Let $S \subseteq N \times N$. A pair (α, β) of processes satisfies the* **word weak simulation expansion** *wrt. S if for every $w \in \mathbb{A}ct^*$:*

$$\textit{if } \alpha \stackrel{w}{\Longrightarrow} \alpha' \textit{ then } \beta \stackrel{w}{\Longrightarrow} \beta' \textit{ such that } (\alpha', \beta') \in S.$$

In the last definition the quantification could also range over $w \in \mathbb{A}ct_\varepsilon^*$, instead of $w \in \mathbb{A}ct^*$. This would define the same notion of expansion, due to our assumption that every process has a silent self-loop.

As before, we naturally obtain symmetric versions of the expansions: long (resp. word) weak bisimulation expansion. Interestingly, the two new expansions give rise to precisely the same notion of bisimulation, namely weak bisimulation, and thus to the same weak equivalence (the same holds for simulation). On the other hand, surprisingly, the hierarchies of approximants differ: in general, word weak bisimulation expansion yields faster convergence than long weak bisimulation expansion, and the latter yields faster convergence than weak bisimulation expansion.

For $\kappa = 1$, the relation R_κ induced by word weak simulation (resp. bisimulation) expansion is trace[5] inclusion (resp. equivalence). Both relations are undecidable in models considered in this thesis.

There are also branching analogues of the weak expansions defined above.

1.2 Results

Our results are described in detail in Chapters 2–4 of this thesis. The first two chapters focus mostly on normed commutative context-free processes: in Chapter 2 we

[5] Trace inclusion is the special case of language inclusion, when all states are accepting.

investigate different hierarchies of approximating relations for weak and branching equivalence, and then in Chapter 3 we prove decidability of branching equivalence. Decidability of weak equivalence remains an open problem, in fact one of the main open problems in equivalence checking for infinite-state systems. Finally, in Chapter 4 we prove decidability of weak and branching pre-orders for one-counter automata without zero tests. The three chapters are based on the results of papers [28, 13, 27], respectively.

Below we outlined the content of Chapters 2–4 in detail.

1.2.1 Approximations

In Chapter 2 we investigate various hierarchies of approximants for weak and branching equivalence over commutative context-free processes. Our study is twofold: we investigate both *limitations* and *usefulness* of standard approximants, for instance those introduced in Section 1.1.4. On one hand, we prove high inapproximability of weak equivalence, by providing examples showing that the hierarchies of approximants do not stabilize up to large ordinals. On the other hand, we prove that all decidability results known prior to this thesis for subclasses of commutative context-free processes may be re-proved using ω-stabilization of one of the hierarchies of decidable approximants. Thus the general conclusions from this results are quite disappointing.

Chapter 2 contains a number of results. Below we formulate two of them, illustrating both the limitations and usefulness of approximants.

Concerning the former aspect, we prove the following theorem, thus disproving the conjecture of Hirsfeld and Jančar that approximants induced by long weak bisimulation expansion stabilize at ordinal $\omega + \omega$.

Theorem. *The hierarchy of approximants induced by long weak bisimulation expansion does not stabilize below level ω^2 in the class of normed commutative context-free processes.*

We also demonstrate that even the fastest hierarchy, namely that induced by word expansion, does not stabilize below $\omega + \omega$.

Concerning the usefulness of approximations, we demonstrate that the non-trivial decidability proof of Stirling [53] for a subclass of commutative context-free processes may be re-done using the approximants.

Theorem. *The hierarchy of approximants induced by long weak bisimulation expansion does stabilize at level ω in the subclass investigated in [53].*

The word expansion is the only one among the expansions defined in Section 1.1.2 that is not effective in the class of commutative context-free processes; indeed, already on the first level it expresses trace equivalence, which is undecidable even for commutative context-free grammars without silent steps. In Chapter 2 we propose a new expansion that is effective, and surprisingly, very close to the word weak expansion.

1.2.2 Decidability of branching equivalence

The following theorem, whose proof occupies almost entire Chapter 3, is probably the most difficult to prove among the result of this thesis:

Theorem. *Branching equivalence is decidable over normed commutative context-free processes.*

The decision procedure consists of two separate semi-decision procedures. The positive one is entirely standard and known well in advance, and builds on two fundamental facts:

- congruences over $\mathbb{N}^k$ are semilinear;
- Presburger's arithmetic is decidable.

Branching equivalence is a congruence. Hence the first fact we know that if there is a branching bisimulation witnessing equivalence, then there is also a semilinear one. Branching bisimulation expansion property is expressible in Presburger arithmetic, hence the positive procedure enumerates all semilinear relations and checks them for being a branching bisimulation. The same procedure works for weak equivalence as well.

The whole contribution of Chapter 3 is thus to provide a negative semi-decision procedure for branching equivalence. Concerning the approximants, we know that the hierarchy induced by branching bisimulation expansion does not stabilize at level ω; moreover, the hierarchy induced by the branching analogue of long weak bisimulation expansion is not known do stabilize at level ω. Thus the proof cannot be based on standard techniques.

Our way of proving decidability is by *bounded response property*, which says that whenever α and β are branching equivalent, they necessarily satisfy a sharpening of branching bisimulation expansion. Namely, there is a constant $c \in \mathbb{N}$ such that the size of $\bar{\beta}$ in Definition 9 is at most c times greater than the size of α, and similarly for the size of β' compared to the size of α'. Furthermore, the constant c may be effectively computed from the input grammar.

Our approach may be understood as proposing a new non-standard expansion, call it c-branching bisimulation expansion; the clue here is clearly to show correctness of this new expansion with respect to the branching equivalence, the main technical core of the proof. Once the correctness is proved, we easily get the negative semi-decision procedure due to the following simple facts:

- the hierarchy of approximants induced by c-branching bisimulation expansion stabilizes at ω;
- finite approximants in this hierarchy are decidable.

The proof presented in Chapter 3 required developing a framework that includes, among the others, a result on transforming a process into a normal form of bounded size. It is worth emphasizing that most of this development works for weak equivalence just as well. In particular, weak equivalence also satisfies bounded response property. However, we were not able to estimate the constant c effectively in this case. Clearly, any such estimation would immediately lead to decidability of weak equivalence.

1.2.3 Decidability of simulation pre-orders

Here is the second main results of this thesis, proved in Chapter 4:

Theorem. *Over processes of one-counter automata without zero tests, weak and branching pre-orders are effectively semilinear and thus decidable.*

As a corollary one obtains decidability of weak (resp. branching) *simulation equivalence*, defined as the symmetric part of weak (resp. branching) simulation pre-order.

On technical level, we propose a new effective notion of simulation expansion. The notion is intricate: in order to compute each next approximant one needs to solve, as a subroutine, the pre-order problem for one-counter automata without zero tests and without silent transitions[6]. This time our technique works, equally well, both for weak and branching pre-order.

It has been widely observed that bisimulation equivalence tends to be computationally less expensive than the analogous simulation pre-order. Theorem 8, surprisingly, presents an exception in this trend: while weak equivalence is undecidable, weak pre-order is decidable for one-counter automata without zero tests. As far as we know, this is the only *natural* class of infinite-state systems that exhibits this exceptional property.

1.3 Related research

Decidability questions for infinite-state systems, such as Petri nets, pushdown automata, or process algebras, have been investigated since 70ies. However, the real interest has been attracted since the discovery by Beaten, Bergstra and Klop [2, 3], in late 80ies, that bisimulation equivalence [45, 52] is decidable, under normedness assumption, for plain (non-commutative) context-free processes. This result initiated a successful line of research, in a substantial part devoted to plain context-free processes, and commutative context-free processes. An excellent reference concerning the early

[6] Decidability of this problem has been shown in [1] and in [39], as we discuss in detail in Section 1.3.2.

results on both classes is Chapter 9 of [46]. Both commutative and plain context-free processes have been classified in the so called Process Rewrite Systems (PRS) hierarchy by Mayr [43], as special cases of Petri nets and pushdown processes, respectively. An exhaustive survey on decidability and complexity of bisimulation equivalence in all PRS classes is [50].

Except for bisimulation equivalence and simulation pre-order, there are other relevant semantic equivalences and pre-orders on processes. A widely accepted taxonomy of those is contained in [16], in absence of silent steps; extension of this taxonomy to systems with silent steps is given in [17].

1.3.1 Plain and commutative context-free processes

Both for commutative and plain context-free grammars, even in the normed case, all semantic equivalences and pre-orders from van Glabbeek's spectrum, except from bisimulation equivalence, are undecidable. This negative result was shown by Hüttel in [30] and [22] for commutative processes, and in [18] and [31] for non-commutative ones. Bisimulation equivalence is thus the only interesting one, from the point of view of decidability.

Plain context-free processes. This class is also often called *Basic Process Algebra*, see for instance [46]. For unnormed processes without silent steps, decidability of bisimulation equivalence has been first shown by Christensen, Hüttel and Stirling in [9]. The best known upper bound on complexity is 2-EXPTIME, shown by Burkart, Caucal and Steffen [6] (see also the recent simpler proof [36]). On the other hand EXPTIME is the best known lower bound for complexity of this problem, shown recently by Kiefer [41].

On *normed* context-free processes without silent steps, bisimulation equivalence can be solved in polynomial time, as proved by Hirshfeld, Jerrum and Moller in [24] (full version in [25]). The decision procedure runs in time $O(n^{13})$. This breakthrough paper was followed by a sequence of improvements [42, 14], culminating in the best currently known algorithm, described in Czerwiński's PhD thesis [10], running in time $O(n^4 polylog(n))$.

On the other hand, very little is know about context-free processes with silent steps. Examples of partial positive result are decidability of weak equivalence [23] and branching equivalence [29, 23] in the subclass of *totally normed* context-free processes. Decidability status of weak and branching equivalence in the whole class, or even for normed processes, remains an intriguing open problem.

Commutative context-free processes. This class is also often called *Basic Parallel Processes* [7]. Consider processes without silent steps now. The first positive results was decidability of bisimulation equivalence by Christensen, Hirshfeld and Moller [8]. The decidability has been later reproved, using the result of Hirshfeld [21], independently shown by Jančar [33], stating that every congruence on a finitely generated commutative semigroup is finitely generated (and semilinear). The bisimulation equivalence is PSPACE-complete, due to [49] and [35].

Under normedness assumption, but still without silent steps, bisimulation equivalence is solvable in polynomial time, as shown by Hirshfeld, Jerrum and Moller in [26]. A fast decision procedure running in time $O(n^3)$ has been later given in [37] by Jančar and Kot. Existence of separate polynomial time procedures for commutative and non-commutative context-free processes revealed apparent similarities between the two classes of systems; despite this similarities, the decision procedures in [24] and [26] were entirely different. This raised a question about a common framework to capture both commutative and non-commutative processes. A successful answer appeared in [11, 12], in terms of *partially-commutative* context-free processes.

Now we move to commutative context-free processes with silent steps. Similarly as for non-commutative processes, not much is known about decidability status of weak or branching equivalence.

Concerning decidability results for restricted subclasses, totally normed processes have been investigated in [23]. Decidability of weak equivalence was shown by Stirling [53], under a restriction on *generators* in a grammar. This restricted subclass is interesting since, as opposed to totally normed processes, the standard approximants do not stabilize at ω in this class. A simpler decidability proof have been given in [28] by demonstrating that approximants induced by *long* weak expansion do stabilize in the subclass of [53] at level ω. Yet another subclass was investigated by Jitka Stribrna in her PhD thesis [54], under restriction to singleton alphabets and non-zero norms; approximants induced by long weak expansion converge in this subclass only at level $\omega + \omega$. Again, a simpler decidability proofs have been given in [28] by demonstrating that approximants induced by an effective variant of word weak expansion, a new expansion introduced in [28], do stabilize at ω. The results of [28] are described in Chapter 2 of this thesis.

Decidability of weak equivalence for commutative context-free processes, even under normedness assumption, remains a major open problem, similarly as for non-commutative processes. However, branching equivalence has been recently shown decidable in [13]. The proof is the topic of Chapter 3 of this thesis.

1.3.2 One-counter automata

One-counter automaton is a special case of pushdown automaton (singleton stack alphabet); if an automaton has no zero tests, as assumed in this thesis, it is also a special case of labelled Petri net (a net with one unbounded place).

Most of semantic equivalences and pre-orders from van Glabbeek's spectrum are undecidable for Petri nets, as well as for one-counter automata with zero tests, even without silent steps. For Petri nets, the undecidability result due to Jančar [33] applies to all equivalences in van Glabbeek's spectrum. For one-counter automata *with* zero tests, undecidability of simulation pre-order was given by Jančar, Moller and Sawa in [40], and trace equivalence was known to be undecidable since 70's [56]. As the only positive result, Jančar showed that bisimulation equivalence is decidable [34], using a nice geometrical technique of *coloring*.[7] Thus the two models are quite hard to analyze, which motivates searching for an interesting but simpler subclasses. One-counter automata *without zero tests* seem to be a reasonable choice.

A result relevant for this thesis is that simulation pre-order is decidable for one-counter automata without zero tests, which was shown by Abdulla and Cerans in [1]; a much simpler proof, based again on a coloring technique, has been given soon afterwards by Jančar and Moller [39]. The complexity of the problem is still unknown, except for the PSPACE lower bound shown in [51]. On the other hand, trace equivalence remains undecidable even without zero tests, as recently shown in [27].

In view of the above results, an interesting question arises whether decidability of bisimulation equivalence and simulation pre-order is preserved when silent steps as allowed. For the bisimulation equivalence, the negative answer has been given by Mayr [44]: weak equivalence is undecidable, even for normed automata. Decidability of weak pre-order remained an open problem since more than a decade, until it has been recently solved positively by Hofman, Mayr and Totzke, who showed decidability of weak (resp. branching) pre-order in [27]. This result, together with its adaptation to branching pre-order, is the topic of Chapter 4 of this thesis. It is worth stressing that our result closes the last open question concerning equivalence- or pre-order-checking of one-counter automata with silent transition rules. As explained in the introduction, the decidability of weak pre-order, combined with undecidability of weak equivalence, contradict the common trend that simulation problems tend to be computationally harder than bisimulation problems.

A somewhat separate line of research focused on analysis of *deterministic* one-counter automata, where bisimulation equivalence (resp. simulation pre-order) coincides with trace equivalence (resp. trace inclusion). Several decidability and complexity results have been obtained in deterministic case, see for instance [5, 19, 20, 55]; basically for deterministic automata both weak pre-order and weak equivalence are solvable in *logarithmic space*. Equivalence of deterministic one-counter automata is a special case of famous equivalence of deterministic pushdown automata which, after being an open question for few decades, was proved decidable by Senizeurges in [47, 48].

[7] Recently, the problem has been shown PSPACE-complete in [4], and even in NL for *deterministic* automata [5].

Chapter 2

Approximations of equivalences and pre-orders

The following three sections, namely Sections 2.1 – 2.3, are preparatory ones before formulating and proving the results of this chapter. In Section 2.1 we analyze how different approximating hierarchies relate to each other, and to the corresponding equivalence or pre-order. In addition to the expansions defined before, we will also introduce a new one, namely *multiset expansion*. Furthermore we discuss computability of *finite approximants*, i.e. those indexed by finite ordinals. Then in Section 2.2 we recall well-known characterizations of equivalences and pre-orders in terms of games, in the style of Bisimulation Game of [45, 53].

In Section 2.3 few simples example LTSs are presented as witnesses of non-stabilization of various approximating hierarchies over commutative context-free processes and over one-counter automata. These examples are either known from the literature, or may be easily deduced.

The last two sections present the contribution of this chapter. In Section 2.4 we prove new lower bounds for stabilization level of weak bisimulation approximants over normed commutative context-free processes. Finally, in Section 2.5 we investigate two restricted subclasses of commutative context-free processes studied in the literature, for which decidability of weak equivalence has been established before. For each of them, we prove that some of approximating hierarchies stabilizes at level ω, thus immediately reproving the decidability of weak equivalence.

2.1 Approximating hierarchies

As we argue below, the word expansion admits the fastest convergence among all expansions defined so far. On the other hand, it is the only expansion that is not effective, namely one can not decide finite approximants. We claim one can find a consensus between the convergence speed and effectiveness. We define below a new expansion that is quite similar to word expansion on one hand, and on the other hand it admits a decision procedure for finite approximants. The expansion is called *multiset weak simulation expansion* as it requires the matching move to agree only up to the multiset of alphabet letters, instead of the word. Later in Section 2.5.2 we will use the hierarchy induced by multiset weak bisimulation expansion, the symmetric version, to re-prove one of partial decidability result known in the literature.

For a word $w \in \mathbb{A}ct^*_\varepsilon$, the observable Parikh image $P(w)$ of w is the multiset of letters of $\mathbb{A}ct$ appearing w. Formally, the cardinality of a letter $a \in \mathbb{A}ct$ in the multiset $P(w)$ is the same as the number of occurrences of a in w. Note that we deliberately ignore the occurrences of the silent action ε in w. For example, $P(\varepsilon ababaa\varepsilon\varepsilon) = P(aaa\varepsilon abb)$.

Definition 15. *Let $S \subseteq N \times N$. A pair (α, β) satisfies* **multiset weak simulation expansion** *wrt. S if for every $w \in \mathbb{A}ct^*$:*

$$\text{if } \alpha \stackrel{w}{\Longrightarrow} \alpha' \text{ then } \beta \stackrel{v}{\Longrightarrow} \beta' \text{ such that } (\alpha', \beta') \in S \text{ and } P(w) = P(v).$$

Similarly as before, one obtains *multiset weak bisimulation expansion* as a symmetric version of the above definition.

Before we relate approximating hierarchies induced by different expansions, let's establish a bundle of notation that is used in this and the following chapters. First, the equivalences and pre-orders will be denoted by the following symbols:

$\approx$	weak equivalence
$\simeq$	branching equivalence
$\preccurlyeq$	weak pre-order
$\preceq$	branching pre-order

Second, we will use ordinals κ in lower indices to denote the approximants induced by the expansions corresponding to an equivalence or pre-order at hand:

approximants	induced by
$\approx_\kappa$	weak bisimulation expansion
$\simeq_\kappa$	branching bisimulation expansion
$\preccurlyeq_\kappa$	weak simulation expansion
$\preceq_\kappa$	branching simulation expansion

Finally, as there are typically several different expansions defining the same equivalence or pre-order, and hence also several different approximating hierarchies, we will use additional symbols to specify a particular expansion. The following table defines notation for approximants induced by different weak bisimulation expansions, which will be the most intensively investigated in the sequel.

approximants	induced by
$\approx_\kappa$	weak bisimulation expansion
$\stackrel{\text{L}}{\approx}_\kappa$	long weak bisimulation expansion
$\stackrel{\text{M}}{\approx}_\kappa$	multiset weak bisimulation expansion
$\stackrel{\text{W}}{\approx}_\kappa$	word weak bisimulation expansion

Along the same lines one can define notation for approximants induced by other expansions: $\simeq$, $\stackrel{\text{L}}{\simeq}$, $\preccurlyeq$, $\stackrel{\text{L}}{\preccurlyeq}$, $\stackrel{\text{M}}{\preccurlyeq}$, $\stackrel{\text{W}}{\preccurlyeq}$, $\preceq$, $\stackrel{\text{L}}{\preceq}$ (recall that we did not define all variants of branching simulation/bisimulation expansions, and it is not straightforward to define its word and multiset variants).

Lemmas 1 and 2 below apply to any LTS.

Lemma 1. *For every ordinal κ,*

$$\approx \;\subseteq\; \stackrel{W}{\approx}_\kappa \;\subseteq\; \stackrel{M}{\approx}_\kappa \;\subseteq\; \stackrel{L}{\approx}_\kappa \;\subseteq\; \approx_\kappa \;.$$

Proof. The first inclusion $\approx \subseteq \stackrel{\text{W}}{\approx}_\kappa$ follows easily by transfinite induction on ordinal κ, knowing that weak equivalence $\approx$ is the fixed point of the refinement function (cf. (1.1) in Section 1.1.2) induced by word weak bisimulation expansion, and that the refinement is monotonic.

Out of the remaining inclusions we will only prove one, say $\stackrel{\text{M}}{\approx}_\kappa \subseteq \stackrel{\text{L}}{\approx}_\kappa$, as the remaining ones are shown similarly. The proof is again by transfinite induction. For a limit ordinal κ the inclusion follows directly, as $\stackrel{\text{M}}{\approx}_\kappa \subseteq \stackrel{\text{L}}{\approx}_k$ for all $k < o$. For the successor ordinal, assume $\stackrel{\text{M}}{\approx}_\kappa \subseteq \stackrel{\text{L}}{\approx}_\kappa$, aiming at showing $\stackrel{\text{M}}{\approx}_{\kappa+1} \subseteq \stackrel{\text{L}}{\approx}_{\kappa+1}$. We will use two easy observations about the refinement functions induced by both expansions (as in (1.1)), call them $\stackrel{\text{M}}{\exp}$ and $\stackrel{\text{L}}{\exp}$:

- For every fixed relation R, the results of the two refinement functions are related by inclusion: $\stackrel{\text{M}}{\exp}(R) \subseteq \stackrel{\text{L}}{\exp}(R)$.
- Both the refinement functions are monotonic: $R \subseteq S$ implies $\stackrel{\text{M}}{\exp}(R) \subseteq \stackrel{\text{M}}{\exp}(S)$, and $\stackrel{\text{L}}{\exp}(R) \subseteq \stackrel{\text{L}}{\exp}(S)$.

Using the above observations we derive the inclusion $\stackrel{\text{M}}{\approx}_{\kappa+1} \subseteq \stackrel{\text{L}}{\approx}_{\kappa+1}$ in few steps:

$$\stackrel{\text{M}}{\approx}_{\kappa+1} \;=\; \stackrel{\text{M}}{\exp}(\stackrel{\text{M}}{\approx}_\kappa) \;\subseteq\; \stackrel{\text{L}}{\exp}(\stackrel{\text{M}}{\approx}_\kappa) \;\subseteq\; \stackrel{\text{L}}{\exp}(\stackrel{\text{L}}{\approx}_\kappa) \;=\; \stackrel{\text{L}}{\approx}_{\kappa+1} \;.$$

The first inclusion follows by the first item above, and the second one by the second item together with the induction assumption. □

Recall that we denote the class of all ordinals by Ord.

Lemma 2. $\approx \;=\; \bigcap_{\kappa\in Ord} \stackrel{W}{\approx}_\kappa \;=\; \bigcap_{\kappa\in Ord} \stackrel{M}{\approx}_\kappa \;=\; \bigcap_{\kappa\in Ord} \stackrel{L}{\approx}_\kappa \;=\; \bigcap_{\kappa\in Ord} \approx_\kappa$.

Proof. The chain of inclusions holds by transfinite induction using Lemma 1. The equality between $\approx$ and $\bigcap_{\kappa \in Ord} \approx_\kappa$ follows by Knaster-Tarski Fixpoint Theorem, applied to the refinement function (1.1) induced by weak bisimulation expansion. $\square$

Analogs of Lemmas 1 and 2 hold also for weak simulation pre-order. Namely, for every ordinal κ we have

$$\preccurlyeq \;\subseteq\; \overset{\text{W}}{\preccurlyeq}_\kappa \;\subseteq\; \overset{\text{M}}{\preccurlyeq}_\kappa \;\subseteq\; \overset{\text{L}}{\preccurlyeq}_\kappa \;\subseteq\; \preccurlyeq_\kappa$$

and moreover

$$\preccurlyeq \;=\; \bigcap_{\kappa \in Ord} \overset{\text{W}}{\preccurlyeq}_\kappa \;=\; \bigcap_{\kappa \in Ord} \overset{\text{M}}{\preccurlyeq}_\kappa \;=\; \bigcap_{\kappa \in Ord} \overset{\text{L}}{\preccurlyeq}_\kappa \;=\; \bigcap_{\kappa \in Ord} \preccurlyeq_\kappa$$

Furthermore, analogs of Lemmas 1 and 2 hold also for branching equivalence and pre-order. We focus on plain and long expansions only. For every ordinal κ we have

$$\simeq \;\subseteq\; \overset{\text{L}}{\simeq}_\kappa \;\subseteq\; \simeq_\kappa \quad \text{and} \quad \preceq \;\subseteq\; \overset{\text{L}}{\preceq}_\kappa \;\subseteq\; \preceq_\kappa$$

and moreover

$$\simeq \;=\; \bigcap_{\kappa \in Ord} \overset{\text{L}}{\simeq}_\kappa \;=\; \bigcap_{\kappa \in Ord} \simeq_\kappa \quad \text{and} \quad \preceq \;=\; \bigcap_{\kappa \in Ord} \overset{\text{L}}{\preceq}_\kappa \;=\; \bigcap_{\kappa \in Ord} \preceq_\kappa .$$

All the proofs are exactly the same as the proofs of Lemmas 1 and 2.

2.1.1 Computability of finite approximants

In this section we focus on LTSs induced by commutative context-free grammars. We will show that for finite ordinals $\kappa < \omega$, the approximants induced by plain/long/multiset weak/branching simulation/bisimulation expansions are effectively computable. In particular this means that all these relations admit finite representations. (In fact a similar computability result is also valid for LTSs induced by one-counter automata without zero tests.)

On the other hand, finite approximants induced by all the variants of word expansions are not computable for both models. Indeed, already for $\kappa = 1$ the approximating relation is trace inclusion or trace equivalence, both undecidable for commutative context-free processes as well as for one-counter automata without zero tests (even without silent transitions).

For computability we will use *Presburger arithmetic*, i.e. the first order theory of natural numbers with addition. A set $R \subseteq \mathbb{N}^k$ of k-tuples of natural numbers is said to be *Presburger-definable* if there is a Presburger formula $\phi_R(x_1, x_2, \ldots, x_k)$ with k

free variables such that

$$\phi_R(n_1, n_2, \ldots, n_k) \iff (n_1, n_2, \ldots, n_k) \in R.$$

It is well-known that it is decidable if a given a Presburger formula ϕ without free variables is valid. Thus Presburger-definable sets are decidable.

Assuming an arbitrary linear order on variables of a commutative context-free grammar, processes of a grammar with k variables are tuples from $\mathbb{N}^k$. Thus any binary relation R over such processes is a subset of $\mathbb{N}^{2k}$. (Similarly, any binary relation over processes of a one-counter automaton may be seen as a family of $|Q|^2$ subsets of $\mathbb{N}^2$, where n is the number of states.) We will focus on the approximating hierarchy induced by one of expansions, say multiset weak simulation expansion (other expansions may be treated similarly), and will show that for finite ordinals κ, the approximants $\overset{\scriptscriptstyle M}{\preccurlyeq}_\kappa$ are Presburger-definable. Moreover, we claim that a Presburger formula defining $\overset{\scriptscriptstyle M}{\preccurlyeq}_\kappa$ may be effectively constructed, given a commutative context-free grammar and a finite ordinal κ. We will build on an immediate corollary of the result of [15] (Theorem 3.3), stated as a lemma below. For k the number of variables and l the cardinality of $\mathbb{A}ct$, by the reachability relation we mean the set of all triples

$$(\alpha, m, \beta) \in \mathbb{N}^k \times \mathbb{N}^l \times \mathbb{N}^k = \mathbb{N}^{2k+l}$$

such that there is a path from α to β labelled by a word $w \in \mathbb{A}ct_\varepsilon$ whose observable Parikh image is m, $P(w) = m$.

Lemma 3. *Given a commutative context-free grammar, one may compute a Presburger formula defining the reachability relation.*

Using the above lemma, the proof of computability of approximants $\overset{\scriptscriptstyle M}{\preccurlyeq}_\kappa$ is by induction on a finite ordinal κ. For $\kappa = 0$ the claim is obviously true as the relation $\overset{\scriptscriptstyle M}{\preccurlyeq}_0$ contains all pairs of processes. Assuming that we have already constructed a Presburger formula ψ for $\overset{\scriptscriptstyle M}{\preccurlyeq}_\kappa$, we will now describe a formula defining $\overset{\scriptscriptstyle M}{\preccurlyeq}_{\kappa+1}$. Recalling that $\overset{\scriptscriptstyle M}{\preccurlyeq}_{\kappa+1} = \overset{\scriptscriptstyle M}{\exp}(\overset{\scriptscriptstyle M}{\preccurlyeq}_\kappa)$ is the refinement of $\overset{\scriptscriptstyle M}{\preccurlyeq}_\kappa$, we simply encode the definition of multiset weak simulation expansion in Presburger arithmetic (we denote by r the formula defining the reachability relation):

$$\forall \alpha', m.\ r(\alpha, m, \alpha') \implies \exists \beta'.\ r(\beta, m, \beta') \wedge \psi(\alpha', \beta').$$

The symbols α, β, etc. stand for vectors of variables of length k; similarly m stands for a vector of variables of length l.

2.2 Simulation and bisimulation via games

This section is devoted to characterization of equivalences and pre-orders defined so far, in term of games played between Spoiler and Duplicator. The definitions are in the style of *bisimulation game* of [45, 52]. Analogous characterizations will be provided for approximants as well. The game characterizations will be useful later as a convenient tool in the proofs.

The games we consider in this thesis are played in *rounds*. Different notions of expansion will lead to different variants of game. On one hand, the overall structure of the game will remain intact when changing one expansion into another; on the other hand, the definition of a single round of a game will change accordingly. This is why we start with a generic definition that does not specify the round at all, and then we provide definitions of a single round that correspond to particular expansions. The definition below applies to an arbitrary LTS (N, E).

Definition 16 (abstract game)**.** *The game is played by two players, Spoiler (she) and Duplicator (he), over an arena $N \times N$ consisting of all pairs of processes (called* positions*), and proceeds in rounds. Each round starts with a Spoiler's move followed by a Duplicator's response. Every consecutive round starts in a position in which the preceding round ended.*

When one of players gets stuck, the other player wins. Otherwise the play is infinite and then it is Duplicator who wins.

As the first case, define the round of the game corresponding to weak simulation expansion. The corresponding instantiation of the abstract game we will call *weak simulation game*.

<u>*weak simulation round*</u>

In a position (α, β), Spoiler chooses one transition $\alpha \xrightarrow{\zeta} \alpha'$ of process α. Then Duplicator answers by a sequence of transitions of the form $\beta \stackrel{\zeta}{\Longrightarrow} \beta'$. The round ends in (α', β').

Note that Spoiler is never stuck in weak simulation game, as she has always an available move induced by a silent self-loop. The same happens in other games defined below, but will not necessarily be true in approximant games defined in Section 2.2.1.

All the games used in this dissertation are *determined*: for every starting position (α, β), exactly one of players has a strategy to win the game, regardless of the opponent choices. One may also prove that there always exists a winning strategy independent from history. Thus if Spoiler wins, she can choose every her move solely on the basis of the starting position of the current round and if Duplicator wins, he can do his choice on the basis of the last Spoiler's move.

There is a tight connection between weak simulations and Duplicator's winning strategies in weak simulation game. To justify this connection, we recall a well-known fact that weak simulation game *characterizes* weak pre-order $\preccurlyeq$:

Fact 1. *$\alpha \preccurlyeq \beta$ if and only if Duplicator has a winning strategy from (α, β) in weak simulation game.*

Proof. For the only if implication, assume that α and β are related by weak pre-order, and consider the game starting from (α, β). In order to win, Duplicator's strategy is always to choose a response that leads to a pair of processes that are related again by weak pre-order. This is always doable as weak pre-order satisfies weak simulation expansion.

For the opposite implication, collect the set R of all the positions (α, β) such that Duplicator wins the game starting from (α, β). Using co-induction, it is enough to show that R is weak simulation, i.e. that every pair from R satisfies weak simulation expansion wrt. R. Indeed, whatever pair $(\alpha, \beta) \in R$ is chosen, Duplicator can play one round so that the resulting pair of processes (α', β') in again winning for him, as otherwise Spoiler would win from (α, β). This means that (α, β) satisfies weak simulation expansion wrt. R as required. $\square$

Analogously we define *weak bisimulation round*, in order to obtain *weak bisimulation game*.

weak bisimulation round

In a position (α, β), Spoiler chooses one of α and β, say α, and one transition of the chosen process, say $\alpha \xrightarrow{\zeta} \alpha'$. Then Duplicator answers by a sequence of transitions of the form $\beta \overset{\zeta}{\Longrightarrow} \beta'$ from the other process. The round ends in (α', β').

The round corresponding to branching simulation/bisimulation expansion is defined analogously, but with an additional Spoiler's choice ending every round. This leads to *branching simulation/bisimulation game*.

branching simulation round

In a position (α, β), Spoiler chooses one transition $\alpha \xrightarrow{\zeta} \alpha'$ of process α. Then Duplicator answers by a sequence of transitions of the form $\beta \Longrightarrow \bar{\beta} \xrightarrow{\zeta} \beta'$ from the other process. Finally, Spoiler chooses $(\alpha, \bar{\beta})$ or (α', β') for the ending position of the round.

branching bisimulation round

In a position (α, β), Spoiler chooses one of α and β, say α, and one transition of the chosen process, say $\alpha \xrightarrow{\zeta} \alpha'$. Then Duplicator answers by a sequence of transitions of the form $\beta \Longrightarrow \bar{\beta} \xrightarrow{\zeta} \beta'$ from the other process. Finally, Spoiler chooses $(\alpha, \bar{\beta})$ or (α', β') for the ending position of the round.

All the definitions of rounds strictly follow the definitions of corresponding expansions in a straightforward way, by allowing Spoiler to govern universal quantification and Duplicator to govern the existential one. Using this rule one may also obtain games corresponding to other expansions defined in Section 1.1, like long/word weak simulation/bisimulation expansion, as well as to multiset weak simulation/bisimulation expansion. In each case, the analog of Fact 1 holds. In particular, weak bisimulation game characterizes weak equivalence, and branching bisimulation game characterizes branching equivalence. Moreover, there are typically many different games characterizing a given equivalence or pre-order, similarly as there are different but equivalent expansions. For instance, weak equivalence is characterized also by long weak bisimulation game, as well as by word weak bisimulation game.

Example 5. *For the context-free grammar from Example 3. let's analyze weak bisimulation game form the position* $(B, A||A)$.

$$
\begin{array}{lll}
A \longrightarrow \varepsilon & A \xrightarrow{a} \varepsilon & \\
B \longrightarrow \varepsilon & B \longrightarrow A & B \xrightarrow{a} A
\end{array}
$$

Formally, Spoiler has 4 possible moves. One possibility, not stated explicitly in the context-free grammar above, is to use a silent self-loop and thus stay in the same process. However this move can be always responded by Duplicator by staying in the same process as well. If she moves $B \longrightarrow \varepsilon$ *then Duplicator in response will play* $A||A \longrightarrow A \xrightarrow{\varepsilon}$. *The game essentially ends at this point because the only possible move now is* $\varepsilon \longrightarrow \varepsilon$. *If Spoiler chooses* $B \longrightarrow A$ *or* $B \xrightarrow{a} A$ *then Duplicator responses via* $A||A \longrightarrow A$ *or* $A||A \xrightarrow{a} A$, *respectively. Next round in both cases will be played from the position* $(A,\ A)$ *and Duplicator will use the copy-paste strategy to mimic Spoiler's moves. Thus Duplicator wins.*

Any Duplicator's response which is played according to his winning strategy will be called *matching* in the sequel. We will also say that a Spoiler's move is *matched* by a Duplicator's response.

2.2.1 Approximants via games

We have already described game-theoretic characterizations of equivalences and pre-orders we work with. Now we adopt the similar characterizations to approximants. The crucial difference is that a position will contain additionally an ordinal, thus being a triple $(\kappa, \alpha, \beta) \in Ord \times N \times N$. The ordinal will only decrease during a play, therefore the plays will be always finite. The choice of the next value of the ordinal will be up to Spoiler.

Below we formalize this new version of game. Similarly as before we prefer to state one abstract definition, without specifying the definition of round. By instantiating with the concrete rounds, exactly as before, one gets different variants of the game that will characterize different approximants.

Definition 17 (abstract approximant game)**.** *The game is played by two players, Spoiler (she) and Duplicator (he), over an arena consisting of all triples $(\kappa, \alpha, \beta) \in Ord \times N \times N$, and proceeds in rounds. In a position (κ, α, β) Spoiler starts by choosing an ordinal $\kappa' < \kappa$. Then a single round is played from (α, β), exactly as before; suppose that the round ends in (α', β'). Then the game continues from the position $(\kappa', \alpha', \beta')$.*

When one of players gets stuck, the other player wins.

Observe that if $\kappa = 0$ then Duplicator wins, as Spoiler is stuck. Thus it is Spoiler's interest to keep the ordinal as large as possible. Therefore, whenever κ is not a limit ordinal, we silently assume that Spoiler chooses for κ' the predecessor of κ, i.e. $\kappa = \kappa' + 1$.

Analogously as before, the abstract approximant game instantiated with weak simulation round will be called *weak simulation approximant game*, and similarly for all other expansions.

Now we are ready to recall an analog of Fact 1: characterization of approximants $\preccurlyeq_\kappa$ by weak simulation approximant game.

Fact 2. *$\alpha \preccurlyeq_\kappa \beta$ if and only if Duplicator has a winning strategy from position (κ, α, β) in weak simulation approximant game.*

The analogs of Fact 2 hold for all other expansions defined by now; for instance, $\alpha \overset{\text{L}}{\simeq}_\kappa$ if and only if Duplicator has a winning strategy from position (κ, α, β) in *long branching bisimulation approximant game*.

Example 6. *Consider the following grammar, and the pair (Z, A) of processes as the initial position.*

$$\begin{array}{ll} Z \longrightarrow Z||A \qquad\qquad & Z \longrightarrow \varepsilon \\ A \longrightarrow \varepsilon & A \xrightarrow{a} \varepsilon \end{array}$$

We will show how Spoiler wins weak simulation approximant game from the initial position $(4, Z, A)$. As her first move, Spoiler chooses $Z \longrightarrow Z||A$. In response, Duplicator can either stay in the same process, or vanish A; the former option seems better for Duplicator. Suppose the second round starts from the position $(3, Z||A, A)$. Then Spoiler does $Z||A \xrightarrow{a} Z$, which has to be responded by the Duplicator's move $A \xrightarrow{a} \varepsilon$ and the third round starts in $(2, Z, \varepsilon)$. Now Spoiler does $Z \longrightarrow Z||A$ and thus the fourth round starts in $(1, Z||A, \varepsilon)$. Finally Spoiler does $Z||A \xrightarrow{a} Z$, a move which can not be matched by Duplicator.

On the other hand Duplicator wins the game from the position $(3, Z, A)$, because Spoiler is not able to play the last $\xrightarrow{a}$.

The game-theoretic setting introduced in this section will be very convenient in the proofs given in the following sections.

2.3 Non-stabilization at level ω

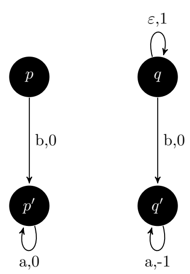
Originally, the definition of weak equivalence due to Milner was based on the ω-approximant. However, it soon become apparent that the hierarchy of approximants does not need to stabilize at level ω. Actually, the stabilization is surely guaranteed only over *finitely branching* LTSs; in term of games, finite branching translates to the requirement that at every position, both Spoiler and Duplicator have only finitely many possible moves to choose. From the game perspective models considered in this thesis, namely LTSs induced by commutative context-free grammars or by one-counter automata (recall that we only work with automata without zero tests), are not finitely branching in general.

Section 2.3 is devoted to studying simple examples witnessing non-stabilization at level ω.

2.3.1 Simulation pre-order over one-counter automata

We will start with the hierarchy of approximants induced by weak simulation expansion over one-counter automata, and show that the hierarchy does not stabilise at level ω. As a byproduct we will explain, among the others, how is it possible that Spoiler wins the weak simulation game starting from a position (α, β) consisting of two processes related by the ω-approximant, $\alpha \preccurlyeq_\omega \beta$. Recall that it means, due to Fact 2, that it is Duplicator who wins the weak simulation approximant game from (ω, α, β).

Example 7. *Consider a one-counter automaton over the alphabet $\{a, b\}$, with four states $\{p, q, p', q'\}$.*



$$q \xrightarrow{\varepsilon,1} q$$

$$p \xrightarrow{b,0} p' \qquad q \xrightarrow{b,0} q'$$

$$p' \xrightarrow{a,0} p' \qquad q' \xrightarrow{a,-1} q'$$

When considering pre-orders, we will often separate states of Spoiler from those of Duplicator, and instead of one automaton consider actually two separate automata. In *this example, the Spoiler's automaton would have states $\{p, p'\}$, and the Duplicator's one $\{q, q'\}$.*

What is the least ordinal κ such that

$$p0 \not\preccurlyeq_\kappa q0?$$

First, observe that $p'0 \preccurlyeq_k q'k$ for every $k \in \mathbb{N}$. Indeed, in the weak simulation approximant game at position $(k, p'0, q'k)$, the most reasonable option for Spoiler is to play $p'0 \xrightarrow{a} p'0$. Duplicator has only one possible response, namely $q'k \xrightarrow{a} q'(k-1)$, and the round ends in position $(k-1, p'0, q'(k-1))$. Thus we have shown:

$$\forall_{k \in \mathbb{N}} \exists_{n \in \mathbb{N}} \; p'0 \preccurlyeq_k q'n. \tag{2.1}$$

One may similarly argue that

$$p'0 \not\preccurlyeq_{k+1} q'k. \tag{2.2}$$

Now if we consider weak simulation approximant game starting from position $(\omega, p0, q0)$, then for every choice of an ordinal $k < \omega$ by Spoiler, followed by the unique possible Spoiler's move $p0 \xrightarrow{b} p'0$, Duplicator may respond with $q0 \overset{b}{\Longrightarrow} q'n$, for an arbitrarily large n. Due to (2.1) this proves

$$p0 \preccurlyeq_\omega q0.$$

On the other hand we claim that $p0 \not\preccurlyeq_{\omega+1} q0$. This time Spoiler chooses ω as the next ordinal, and moves $p0 \xrightarrow{b} p'0$, which must be responded by Duplicator with $q0 \overset{b}{\Longrightarrow} q'n$ for some $n \in \mathbb{N}$. If Spoiler chooses in the next round an ordinal bigger than n, then after n rounds she will win, by (2.2).

The described Spoiler's strategy, seen as a tree, has infinite branching, even if every branch is finite. Thus we can not bound the depth of the Spoiler's strategy, in particular König's Lemma does not apply.

As a conclusion $\preccurlyeq_\omega \neq \preccurlyeq$ over LTSs generated by one-counter automata. However, since the Spoiler's LTS has no silent transitions, our argument works just as well for *long* weak simulation approximant game, which allows us to conclude $\overset{\text{L}}{\preccurlyeq}_\omega \neq \preccurlyeq$ as well.

2.3.2 Bisimulation equivalence over commutative context-free processes

Our next goal is to construct an analogous example for weak equivalence over normed commutative context-free processes. We will apply the same general scheme: to construct a pair of processes α and β that are related by the ω-approximant, but not by the

$(\omega + 1)$-approximant. We need to endow Duplicator with an infinite set of possible responses, among which one is chosen by Duplicator on the basis of a prior Spoiler's move. We will face however two difficulties. The first one is lack of control states in commutative context-free processes. Another one is that we will work with bisimulation expansion, instead of simulation one, and thus Spoiler can change the side in every round.

Example 8. *Consider the following commutative context-free grammar. The left-hand side is a graphical depiction of the rules listed on the right-hand side. A loop*

$$Y \looparrowleft A$$

is used to concisely depict a rule $Y \longrightarrow Y||A$.

$$\begin{tikzcd}
X \arrow[r] \arrow[d, "b"'] & Y \looparrowleft A \arrow[d, "b"] & A \arrow[d, bend right] \arrow[d, bend left, "a"] \\
a \circlearrowright Z \arrow[r] & \varepsilon & \varepsilon
\end{tikzcd}
\qquad
\begin{array}{ll}
X \longrightarrow Y & Y \longrightarrow Y||A \\
X \xrightarrow{b} Z & Y \xrightarrow{b} \varepsilon \\
Z \xrightarrow{a} Z & Z \longrightarrow \varepsilon \\
A \longrightarrow \varepsilon & A \xrightarrow{a} \varepsilon
\end{array}$$

The two processes X and Y are not weak equivalent, $X \not\approx Y$. Indeed, Spoiler wins the bisimulation game by playing $X \xrightarrow{b} Z$; proper Duplicator's responses lead to A^n, for some $n \in \mathbb{N}$. Now Spoiler continues by playing the move $Z \xrightarrow{a} Z$, as many as n times, and then wins in the next round. Actually $X \not\approx_{\omega+1} Y$.

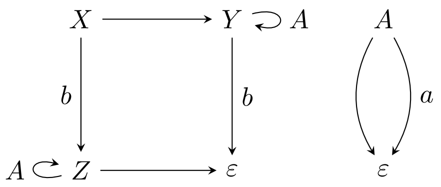
On the other hand $X \approx_\omega Y$ as it is Duplicator to win the weak bisimulation approximant game from position (ω, X, Y). To see this observe that the first Spoiler's move (after choosing an ordinal smaller than ω) has to be $X \xrightarrow{b} Z$ on the left-hand side, as otherwise Spoiler would play on the right-hand side and Duplicator would win by ending his response in exactly the same process as Spoiler (and continuing using copy-paste strategy in consecutive rounds). Then Duplicator responds via a sequence of transitions $Y \stackrel{b}{\Longrightarrow} A^j$, for j equal to the ordinal chosen by Spoiler, which guarantees Duplicator to win the game as

$$Z \approx_j A^j \quad \text{for every } j \in \mathbb{N}.$$

We deduce $\approx_\omega \neq \approx$ over LTSs generated by commutative context-free grammars.

In order to obtain $\stackrel{\mathrm{L}}{\approx}_\omega \neq \approx$ we need to introduce a slight modification. Observe that in the LTS from the previous example, $Z||A^k \stackrel{\mathrm{L}}{\not\approx}_2 A^j$ for every $j, k \in \mathbb{N}$ (while $Z||A^k \approx_j A^j$ was a crucial property before). Indeed, this is due to a Spoiler's move $A^j \Longrightarrow \varepsilon$ on the right-hand side. To prevent Spoiler from playing on the right-hand side, let's replace the loop $Z \xrightarrow{a} Z$ by the rule $Z \longrightarrow Z||A$.

Example 9.



$$
\begin{array}{ll}
X \longrightarrow Y & Y \longrightarrow Y||A \\
X \xrightarrow{b} Z & Y \xrightarrow{b} \varepsilon \\
Z \longrightarrow Z||A & Z \longrightarrow \varepsilon \\
A \longrightarrow \varepsilon & A \xrightarrow{a} \varepsilon
\end{array}
$$

To understand why $Z||A^k \overset{\mathrm{L}}{\approx}_j A^j$ for every k, let's consider one round of long weak bisimulation approximant game. Spoiler chooses the ordinal $j-1$, and then one of two sides. However, if she chooses the right-hand side, and makes any move, then Duplicator is able to reach exactly the same process from the left-hand side, and thus win. Hence Spoiler will play from $Z||A^k$. But during one round she can make only one transition labelled with a, which means that she can force Duplicator to decrease the exponent j only to $j-1$. Therefore Spoiler needs at least $j+1$ rounds to win. Thus $\overset{\mathrm{L}}{\approx}_\omega \neq \approx$.

The next example deals with branching bisimulation expansion in order to show $\simeq_\omega \neq \simeq$ over normed commutative context-free processes.

Let's start by analysing why the previous example does not show $\simeq_\omega \neq \simeq$. The problem is that the crucial relation

$$Z \simeq_j A^j$$

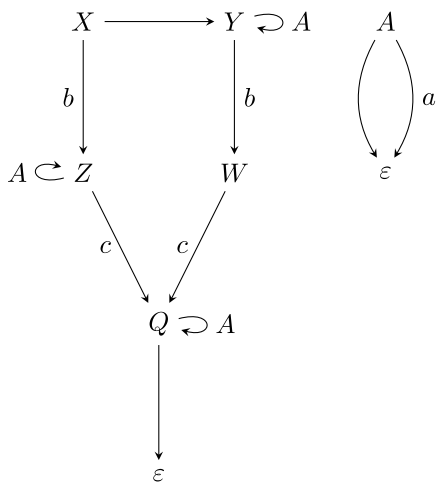
does not hold for every j. Indeed, Spoiler does a move $Z \longrightarrow \varepsilon$. In response, Duplicator has to vanish all A's, $A^j \Longrightarrow \varepsilon$. However, at that point Spoiler can choose one of the following two pairs of processes as the next position in the game:

$$(Z, A) \qquad (\varepsilon, \varepsilon).$$

The first one is due to, intuitively speaking, a "rollback" of the last transition performed by Duplicator. Spoiler clearly chooses the first pair; it is easy to check that Spoiler wins from $(4, Z, A)$. Thus she also wins from $(5, Z, A^j)$ and thus $Z \not\simeq_5 A^j$ for every j.

The problem is that Spoiler is able to win using a smart trick allowing her to force Duplicator to vanish all A's but one. A quick solution of that problem could be to remove the transition rule $Z \longrightarrow \varepsilon$, which however would result in an unnormed system. In order to preserve normedness, we can modify the system as follows:

Example 10.



$$
\begin{array}{ll}
X \longrightarrow Y & Y \longrightarrow Y||A \\
X \xrightarrow{b} Z & Y \xrightarrow{b} W \\
Z \longrightarrow Q & Z \longrightarrow Z||A \\
W \xrightarrow{c} Q & \\
Q \longrightarrow Q||A & Q \longrightarrow \varepsilon \\
A \longrightarrow \varepsilon & A \xrightarrow{a} \varepsilon
\end{array}
$$

The crucial property is $Q \simeq Q||A^j$. Thus, if Spoiler does the move $Z \xrightarrow{c} Q$ then Duplicator does not have to remove A's, and hence a rollback does not help Spoiler. Thus $\simeq_\omega \neq \simeq$.

The next natural question is:

$$\text{Does } \overset{\mathrm{L}}{\simeq}_\omega = \simeq \text{ over normed commutative context-free processes?} \tag{2.3}$$

Unfortunately, we do not know the answer to this question. We conjecture however that the answer is positive. Observe that once one establishes the positive answer, the proof of decidability of branching equivalence, being the main results of Chapter 3, may be drastically simplified.

2.3.3 Other combinations

We have shown non-stabilization at level ω of:

- approximants $\overset{\mathrm{L}}{\preccurlyeq}_\kappa$ over normed[1] one-counter automata, and
- approximants $\overset{\mathrm{L}}{\approx}_\kappa$ over normed commutative context-free processes.

It turns however that the examples may be easily modified to work for the two remaining combinations: approximants $\overset{\mathrm{L}}{\approx}_\kappa$ over normed one-counter automata, and approximants $\overset{\mathrm{L}}{\preccurlyeq}_\kappa$ over normed commutative context-free processes. Furthermore, the example for non-stabilization of approximants $\simeq_\kappa$ over commutative context-free processes may be easily adapted to approximants $\preceq_\kappa$, and even to $\overset{\mathrm{L}}{\preceq}_\kappa$. Finally, the latter example may be easily translated to a one-counter automaton, as well as the example witnessing non-stabilization of approximants $\simeq_\kappa$.

[1]For one-counter automata, normedness means that from every process there is a sequence of transitions that ends with the counter value 0.

These similarities are not accidental. All the grammars appearing in the examples are of a special form, with all reachable processes of the form $B||A^n$, for some $n \in \mathbb{N}$, and for some variable B. Thus all the examples may be easily translated to one-counter automata, preserving the induced LTS up to isomorphism.

Unnormed processes. By now all the examples used *normed* processes. Without normedness assumption, the answer to question (2.3) is negative, as witnessed by an example similar to Example 8: the main difference is that A and Z can not vanish silently.

Example 11.

$$\begin{tikzcd}
X \arrow[r] \arrow[d, "b"'] & Y \arrow[loop right, "A"] \arrow[d, "b"] & A \arrow[d, "a"] \\
Z \arrow[loop left, "a"] & \varepsilon & \varepsilon
\end{tikzcd}
\qquad
\begin{array}{ll}
X \longrightarrow Y & Y \longrightarrow Y||A \\
X \xrightarrow{b} Z & Y \xrightarrow{b} \varepsilon \\
Z \xrightarrow{a} Z & A \xrightarrow{a} \varepsilon
\end{array}$$

2.4 Lower bounds for stabilization levels

In this section we focus mostly on normed commutative context-free processes. We show that long approximants $\stackrel{\text{L}}{\approx}_\kappa$ are not guaranteed to stabilize at level $\omega + \omega$ and that word approximants $\stackrel{\text{W}}{\approx}_\kappa$ do not necessarily stabilize at level ω. From our examples we derive lower bounds of ω^2 and $\omega + \omega$ for the stabilization levels of approximants $\stackrel{\text{L}}{\approx}_\kappa$ and $\stackrel{\text{W}}{\approx}_\kappa$, respectively.

A simple but very useful tool is the following:

Lemma 4. *If* $\alpha \Longrightarrow \beta \Longrightarrow \alpha'$ *and* $\alpha \approx \alpha'$ *(resp.* $\alpha \simeq \alpha'$*) then* $\alpha \approx \beta$ *(resp.* $\alpha \simeq \beta$*).*

Proof. Immediate using games. If Spoiler plays from α, Duplicator uses its response from α', precomposed with $\beta \Longrightarrow \alpha'$. On the other hand, if Spoiler plays from β, Duplicator moves $\alpha \Longrightarrow \beta$ and then copies the Spoiler's transition. $\Box$

Remark 1. *Lemma 4 holds also for approximants induced by long/multiset/word weak/ branching bisimulation expansions.*

As announced in Stríbrná's PhD thesis [54], the following long-standing conjecture is attributed to Jančar and Hirshfeld:

Conjecture 1. $\stackrel{L}{\approx}_{\omega+\omega} = \approx$ *over commutative context-free processes.*

Our first theorem in this section falsifies this conjecture:

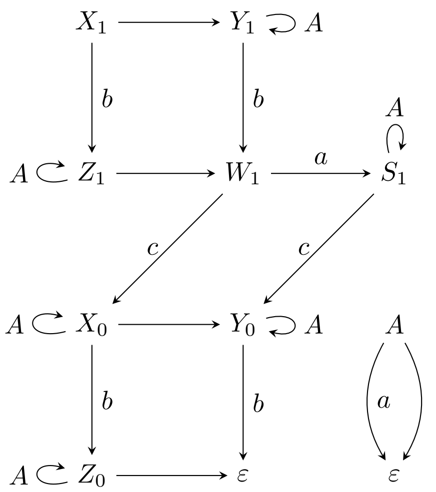
Theorem 1. *Long weak bisimulation approximants* $\stackrel{L}{\approx}_\kappa$ *do not stabilize below level* ω^2 *over normed commutative context-free processes:* $\approx \neq \stackrel{L}{\approx}_{\omega\cdot k}$ *for all* $k \in \mathbb{N}$.

Proof.

We will show how to construct, for a given k, a commutative grammar and two nonequivalent processes that are in relation $\stackrel{\text{L}}{\approx}_{\omega\cdot k}$. For $k = 1$ the claim is trivial, e.g. by Example 9. We first show how to construct a system with $\approx \neq \stackrel{\text{L}}{\approx}_{\omega+\omega}$. For this we recycle Example 9 to which we add the rule $X \longrightarrow X||A$. This modification does not influence the approximant game and we have that $X||A^k \stackrel{\text{L}}{\approx}_\omega Y||A^j \not\approx X||A^k$ for any $k, j \in \mathbb{N}$.

To construct a counter-example to convergence at level $\omega + \omega$ we combine two copies of this system as indicated in Figure 2.1 below.

Figure 2.1: Combining two copies of the context-free grammar from Example 11 yields $X_1 \stackrel{\mathrm{L}}{\approx}_{\omega+\omega} Y_1 \not\approx X_1$.



$$
\begin{array}{ll}
X_1 \longrightarrow Y_1 & X_1 \xrightarrow{b} Z_1 \\
Y_1 \xrightarrow{b} W_1 & Y_1 \longrightarrow Y_1||A \\
Z_1 \longrightarrow W_1 & Z_1 \longrightarrow Z_1||A \\
W_1 \xrightarrow{a} S_1 & S_1 \longrightarrow S_1||A \\
W_1 \xrightarrow{c} X_0 & S_1 \xrightarrow{c} Y_0
\end{array}
$$

$$
\begin{array}{ll}
X_0 \longrightarrow Y_0 & X_0 \xrightarrow{b} Z_0 \\
Y_0 \xrightarrow{b} W_0 & X_0 \longrightarrow X_0||A \\
Y_0 \longrightarrow Y_0||A & Z_0 \longrightarrow Z_0||A \\
Z_0 \longrightarrow \varepsilon &
\end{array}
$$

$$
A \longrightarrow \varepsilon \quad A \xrightarrow{a} \varepsilon
$$

The bottom part of the construction is the gadget as discussed previously. Observe that variables X_0, Y_0, Z_0 are not able to produce variables from the top part of the diagram 2.4, those variables with an index 1. Thus we preserve that $X_0||A^n \stackrel{\mathrm{L}}{\approx}_\omega Y_0||A^m$ for any $m, n \in \mathbb{N}$. Our aim is to show that indeed $X_1 \stackrel{\mathrm{L}}{\approx}_{\omega+\omega} Y_1 \not\approx X_1$. For this it suffices to show that the only possibility for Spoiler to win is to force the game from (κ, X_1, Y_1) to end up in $(\kappa', X_0||A^n, Y_0||A^m)$ where $\kappa' > \omega$, and to achieve that she needs that $\kappa > \omega + \omega$.

The long weak bisimulation approximant game starts from the position

$$(\omega + \omega, X_1, Y_1)$$

and it goes through the upper square pattern X_1, Y_1, Z_1, W_1. By our previous discussion of this gadget, we know that Spoiler, having chosen an ordinal $\omega + n$, has to start by $X_1 \stackrel{b}{\Longrightarrow} Z_1||A^m$; Duplicator will respond to $W_1||A^{n-2}$. Spoiler must continue to play from the left hand side in order to prevent a perfect match to identical positions. Moreover for the same reason she cannot move to a $W_1||A^i$ for $i \leqslant n-2$. Furthermore if she makes a move $Z_1||A^m \stackrel{c}{\Longrightarrow} X_0||A^i$, while the other position still contains a W_1, Duplicator is able to match to the same position. So the only option left for Spoiler is to force Duplicator to remove all variables A one by one by performing transition labelled with a. Eventually, from a position $(\omega+2, Z_1, W_1)$ (or $(\omega+2, W_1||A^{>0}, W_1)$, Spoiler chooses one last transition labelled with a and thus forces Duplicator to rewrite W_1 to S_1. Afterwards, Spoiler can force the game to the position $(\omega, X_0||A^n, Y_0||A^m)$ by choosing a transition labelled with c from either side. As $X_0 \approx_\omega Y_0$, we conclude that $X_1 \approx_{\omega+\omega} Y_1$. This completes the proof for $k = 2$.

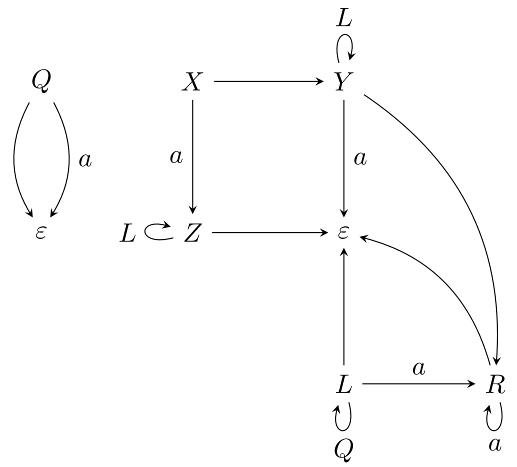
The construction above can be extended to provide a counter-example for convergence at level $\omega \cdot k$ for any natural k by stacking k copies of the square gadget on top of each other. This can also be modified to a system which contains only variables of the norm zero. $\square$

Next we focus on word approximants and falsify a conjecture of Stríbrná [54] about their convergence at level ω.

Theorem 2. *Word weak bisimulation approximants do not stabilize below level $\omega + \omega$ over normed commutative context-free processes.*

Proof. We start by proving that $\approx \neq \overset{\mathrm{w}}{\approx}_\omega$. Consider the commutative context-free grammar depicted in Figure 2.2.

Figure 2.2: Counter-example for finite approximability of $\overset{\mathrm{w}}{\approx}_i$



$$
\begin{array}{ll}
X \xrightarrow{a} Z, & X \longrightarrow Y \\
Y \xrightarrow{a} \varepsilon, & Y \longrightarrow Y||L \\
Y \longrightarrow R & \\
Z \longrightarrow Z||L\,, & Z \longrightarrow \varepsilon \\
L \xrightarrow{a} R, & L \longrightarrow \varepsilon \\
L \longrightarrow L||Q & \\
R \longrightarrow \varepsilon\,, & R \xrightarrow{a} R \\
Q \xrightarrow{a} \varepsilon, & Q \longrightarrow \varepsilon
\end{array}
$$

By Lemma 4, we know that $Z||L^n||Q^m \approx Z$ and $L^{n+1}||Q^m \approx L^{n+1}$ for any two naturals m, n. Moreover it is easy to see that $R^r \approx R$ for any $r \in \mathbb{N}^+$ which means that we don't need to consider positions with more than one variable R.

We claim that $X \overset{\mathrm{w}}{\approx}_\omega Y \not\approx X$ and base our proof on the following claims that are proven individually after the main argument. For $i, j, n \in \mathbb{N}$, $n > 0$ we have

$$Z \overset{\mathrm{w}}{\not\approx}_3 R||L^i \overset{\mathrm{w}}{\not\approx}_3 L^j, \tag{2.4}$$

$$Z \overset{\mathrm{w}}{\approx}_{2n+1} L^n, \tag{2.5}$$

$$Z \overset{\mathrm{w}}{\not\approx}_{2n+2} L^n. \tag{2.6}$$

In the word weak bisimulation approximant game from a position $(\kappa\ ,\ X\ ,\ Y)$, Spoiler must start with a move $X \overset{a}{\Longrightarrow} Z||L^l||Q^q \approx Z$, as otherwise his opponent is able to match to the same position and thereby win. Possible responses for Duplicator from Y are:

- To some $R^r||L^n||Q^m \approx R||L^n$, which allows Spoiler to win in 3 further rounds by Claim 2.4.
- To some $Y||L^n||Q^m \approx Y||L^n$ which allows Spoiler to silently replace the Y by R in the next round and afterwards again win in 3 rounds by Claim (2.4). Note that no silent response from $Z||L^l||Q^q$ to some position that contains R is possible.
- To some Q^m which allows Spoiler to win in one round by playing $Z \overset{a^{m+1}}{\Longrightarrow} Z$.
- (proper response) To some $L^n||Q^m \approx L^n, n \in \mathbb{N}^+$ which allows Spoiler to win but in not fewer than $2n+2$ rounds by Claims (2.5) and (2.6).

The choice of n is made by Duplicator and therefore $X \overset{\mathrm{w}}{\approx}_\omega Y \not\approx X$. Note that this counter-example uses only a single observable transition label and all variables have zero norm.

To construct a counter-example to convergence of word approximants at level $\omega + k$ for finite k, the previous construction can be complemented by a "finite ladder", where X and Y are renamed to X_0 and Y_0: For $0 < i \leqslant k$ add variables $X_i, Y_i, U_i, U_i', W_i, W_i'$ and rules as indicated below.

$$\begin{tikzcd}
\cdots \arrow[r, dotted, no head] & X_i \arrow[r, "a"] \arrow[d] & U_i \arrow[r, "a"] & U_i' \arrow[r, "a"] & X_{i-1} \arrow[r, dotted, no head] \arrow[d] & \cdots \\
\cdots \arrow[r, dotted, no head] & Y_i \arrow[r, "a"] & W_i \arrow[r, "a"] \arrow[ur, "a"] & W_i' \arrow[r, "a"] & Y_{i-1} \arrow[r, dotted, no head] & \cdots
\end{tikzcd}$$

□

It remains to prove claims (2.4) - (2.6). We first prove some auxiliary claims on which we base our arguments for claims (2.4) - (2.6). For all $m, n \in \mathbb{N}^+$,

$$R||L^n \overset{\mathrm{w}}{\not\approx}_1 Q^m \tag{2.7}$$

$$L^n \overset{\mathrm{w}}{\not\approx}_2 R \overset{\mathrm{w}}{\not\approx}_2 Z \tag{2.8}$$

For (2.7), observe that Duplicator cannot respond to a move $R \overset{a^{m+1}}{\Longrightarrow} R$.
For (2.8), Spoiler moves from L^n (or Z) silently to Q and Duplicator can respond to R or to ε. In the first case he loses in one more round by Claim (2.7), in the latter he cannot respond to move $Q \overset{a}{\Longrightarrow} \varepsilon$ from ε.

Claim (2.4): $Z \overset{w}{\not\approx}_3 RL^n \overset{w}{\not\approx}_3 L^m$.

Proof. For both parts Spoiler moves from $R||L^n$ silently to R. Duplicator must respond either to Q^k which is losing for him in one round by Claim (2.7), or to $L^j||Q^k \approx L^j$ or $Z||L^j||Q^k \approx Z$, which is losing for him in two rounds by Claim (2.8). □

Claim (2.5): $Z \overset{w}{\approx}_{2n+1} L^n$ for $n \in \mathbb{N}^+$.

Proof. By induction on $n \geqslant 1$. In order to make an induction step we will need stronger induction hypothesis, namely: for any $m > n$ hold $L^m \overset{w}{\approx}_{2n+1} L^n$ and $Z \overset{w}{\approx}_{2n+1} L^n$. Thus base cases are $L \overset{w}{\approx}_3 L^m$ and $Z \overset{w}{\approx}_3 L$.

Base case $L \overset{w}{\approx}_3 L^m$.
Wlog. assume that $m > 0$ minimizes k in $L \overset{w}{\not\approx}_k L^m$ i.e. we choose m in such a way that it is in fervour of Spoiler as much as possible. Moreover we can assume that Spoiler only makes optimal moves i.e. wins as quickly as possible. This means in particular that she needs to change the equivalence class in the first move. Otherwise Duplicator does not change his equivalence class as well and we get a contradiction with a choice of m.
Thus from position $(3\ ,\ L\ ,\ L^m)$, she can move either:

- to $R||Q^q$, or to Q^q
- to $L^{m'}||Q^q \approx L^{m'}$, for $0 < m' < m$ or
- to $L^{m'}||R||Q^q \approx L^{m'}||R$ for $0 < m' < m$.

In first case Duplicator moves to $R||Q^q$, or to Q^q and wins, thus Spoiler will not do that. In second and third cases Duplicator stays in L. In the second case, because we assume optimal moves, we must have $L \overset{w}{\not\approx}_i L^l$ for some $i < k$, which contradicts the optimality of m. Thus Spoiler has to choose third option and
the game continues from a position $(2\ ,\ L^{m'}||R\ ,\ L)$. Spoiler must again move from $L^{m'}||R$ and change the class.

- If she makes a move labelled with a to R or ends in Q^i then Duplicator can match to the same position.
- Moreover Spoiler's move to some $R||L^{m''}$ or $L^{m''}, m'' < m' < m$ is surely non-optimal.
- The only remaining option is to move silently to R. Duplicator will respond by $L \Longrightarrow L$. Now recall that $L \overset{w}{\approx}_1 R$.

Base case $Z \overset{w}{\approx}_3 L$:
As Z can silently go to L^n, Spoiler needs to make her first move from Z. She has three options to change the class from here:

(1) to some $L^l||Q^q \approx L^l$, or

(2) to $R||L^l||Q^q \approx R||L^l$, or

(3) to something equivalent to $Z||R$.

For any $l, q \in \mathbb{N}$. In all cases Duplicator responds to L. In the first two cases, now we can use previous claims $L \overset{w}{\approx}_3 L^m$ and $L \overset{w}{\approx}_2 R||L^m$ to conclude that this allows Duplicator to survive another 2 rounds. If the second round starts in a position $(2\ ,\ Z||R\ ,\ L)$ (or equivalent), Spoiler again can not move from L and has three options to change the class: to something equivalent to Z which is non-optimal as it repeats the initial position. Alternatively she can go to $R||L^l||Q^1 \approx R||L^l$ or to $R||Q^q$. In both cases we complete by the observation that obviously $R||L^l \overset{w}{\approx}_1 L \overset{w}{\approx}_1 R$.

For the induction step:
assume $L^m \overset{w}{\approx}_{2n+1} L^n$ and $Z \overset{w}{\approx}_{2n+1} L^n$ where $m > n$. We will show that

$$L^m \overset{w}{\approx}_{2(n+1)+1} L^{(n+1)}.$$

Just as in the base case, the only good move for Spoiler is $L^m \overset{a}{\Longrightarrow} L^{m'}||R$ for some $n < m' < m$. Duplicator in his response goes to $L^n||R$. Next one more time Spoiler has the only one reasonable kind of move, to a position equivalent to $L^{m''}$, where $m'' > n$. However now Duplicator responds to L^n and we use the induction assumption to the pair $L^{m''} \overset{w}{\approx}_{2n+1} L^n$.

Observe that because $\overset{w}{\approx}_{2n+1}$ is a congruence this implies also $L^m||R \overset{w}{\approx}_{2n+1} L^n||R$ for $m \geqslant n$.

To show that $Z \overset{w}{\approx}_{2(n+1)+1} L^{(n+1)}$ we assume wlog. that Spoiler initially moves $Z \overset{a}{\Longrightarrow} Z||R$, Duplicator responds by $L^{n+1} \overset{a}{\Longrightarrow} L^n||R$. Now to prevent a perfect match in the next round, Spoiler moves from $Z||R$ to either Z or to $L^m||R$ or L^m. In the first case, Duplicator will remove the R and end up in L^n and we can use the induction assumption, in the last two cases Duplicator stays in $L^n||R$ or goes to L^n. Either way, we can use the previous claims that $L^m||R \overset{w}{\approx}_{2n+1} L^n||R$ and $L^m \overset{w}{\approx}_{2n+1} L^n$ for $m \geqslant n$. □

Claim (2.6): $Z \overset{w}{\not\approx}_{2n+2} L^n$ for $n > 0$.

Proof. By induction on $n \geqslant 1$. In order to make induction step we need stronger induction hypothesis , namely: for any $m > n$, $L^m \overset{w}{\not\approx}_{2n+2} L^n$ and $Z \overset{w}{\not\approx}_{2n+2} L^n$.

Base case: $Z \overset{w}{\not\approx}_4 L \overset{w}{\not\approx}_4 L^m$.

Spoiler plays $L^m \overset{a}{\Longrightarrow} L||R$ (or $Z \overset{a}{\Longrightarrow} L||R$). Possible responses from L are

(1) to $L||Q^q$ or Q^q;

(2) to $R||Q^q$.

In the first case Spoiler wins in 3 more rounds by Claim 2.4.
In the second case Spoiler performs a move $L||R \Longrightarrow Q^{q+1}$ and Duplicator responds to either $R||Q^i$ or Q^i with $i \leqslant q$. In both cases Spoiler wins in one round by Claim (2.7) or playing a sequence of transitions labelled with a^{q+1} from Q^{q+1}.

For the induction step we assume $L^m \overset{\mathrm{W}}{\not\approx}_{2n+2} L^n \overset{\mathrm{W}}{\not\approx}_{2n+2} Z$ and show that both $L^m \overset{\mathrm{W}}{\not\approx}_{2(n+1)+2} L^{(n+1)}$ and $Z \overset{\mathrm{W}}{\not\approx}_{2(n+1)+2} L^{(n+1)}$ hold. Spoiler moves from L^m (or Z) using a transition labelled with a and she ends in a position $L^{n+1}||R$. Duplicator can respond:

(1) to $L^{n+1}||Q^q$ or some Q^q, from which Spoiler wins in 3 rounds by Claim 2.4, or

(2) to $L^{n'}||R||Q^q \approx L^{n'}||R, n' < n+1$. In this case in the next round Spoiler performs a move $L^{n+1}||R \Longrightarrow L^{n+1}$ and Duplicator responds either to $L^{n''}||R||Q^i$ or $L^{n''}||Q^i$ with $n'' \leqslant n' < n+1$. In the first case, Spoiler wins in one round by Claim (2.7). In the last case the game continues from $(2n''+2\ ,\ L^{n+1}\ ,\ L^{n''\leqslant n})$ and we can use the induction assumption for n'' as big as possible namely $n'' = n$.

□

We conjecture that $\omega+\omega$ lowerbound may be increased to ω^2. However an example should be much more complicated than the one for approximants induced by long weak bisimulation expansion.

2.5 Decidability via approximation

We now use the approximation approach to show that two subclasses of commutative context-free processes previously known in the literature have decidable weak equivalence. We start by providing in Lemma 5 a general procedure which returns a witness that two given processes are equivalent if they are, and that loops if they are not equivalent. This allows us focus only on a procedure which proves that two given processes are not equivalent. Note that Lemma 5 works also for branching equivalence. This will be useful in the next chapter, where we will focus on a semi-decision procedure that checks if two processes are not in branching equivalence.

In Subsection 2.5.1 we show decidability for the class introduced in [53] by proving that the approximants induced by long weak bisimulation expansion stabilize at level ω, i.e. $\overset{\mathrm{L}}{\approx}_\omega = \approx$. Furthermore in Subsection 2.5.2 we reprove decidability for the subclass introduced in [54]. Namely we show that the approximants induced by multiset weak bisimulation expansion stabilize at level ω, i.e. $\overset{\mathrm{M}}{\approx}_\omega = \approx$. In both cases we know due to Section 2.3 that at finite levels the approximants are decidable relations, and hence showing their stabilization at level ω suffices to get a negative decision procedure, due to Lemma 6 below.

Lemma 5. *There is a procedure which stops if two given (not necessarily normed) commutative context-free processes are in weak (resp. branching) equivalence, and loops otherwise.*

Proof. We will exploit the fact that both equivalences are substitutive, i.e. both equivalences are congruences. For the positive semi-decision procedure we use a standard semi-linear representation, knowing that each congruence, including $\approx$ and $\simeq$, is a semi-linear, and thus Presburger-definable, subset of $\mathbb{N}^k \times \mathbb{N}^k$ [21, 32] (where k is the number of variables). The algorithm guesses a Presburger formula ψ with $2k$ free variables and then checks validity of a Presburger formula ϕ_ψ that says that the set defined by ψ is a weak/branching bisimulation containing the input pair of processes. The formula ϕ_ψ is constructed similarly as in Section 2.1.1, both for weak equivalence and for branching equivalence. $\Box$

The next lemma has a generic flavor as it says about an arbitrary hierarchy of decidable approximants.

Lemma 6. *Consider a subclass $\mathcal{C}$ of commutative context-free processes and suppose that some weak/branching expansion induces a hierarchy of approximants that stabilizes at level ω in the class $\mathcal{C}$. Suppose also that all finite approximants are decidable. Then there is a procedure which stops if two given processes from C are not weak (resp. branching) equivalent, and loops otherwise.*

Proof. Observe that if the approximation hierarchy $(R_\kappa)_{\kappa \in Ord}$ induced by some weak (resp. branching) bisimulation expansion stabilizes at ω then α being not equivalent to β implies that there is a finite ordinal κ such that a pair (α, β) is not in approximant relation at level κ. Thus for the negative semi-decision procedure it suffices to check, for consecutive finite approximants R_n, $n \in \mathbb{N}$, if the pair (α, β) belongs to R_n. If some approximant does not contain the pair (α, β) then the procedure stops. $\Box$

Simplified grammars. Before embarking on the stabilization proofs, we will assume from now on that grammars are in a convenient special form, expressively equivalent to the general case: no death variables and no redundancy.

A variable A is *death variable* if it is equivalent to the empty process, $A \approx \varepsilon$. One may safely remove every death variable from a grammar, replacing its every occurrence in transition rules of other variables by ε. For example, a transition rule $B \xrightarrow{a} AC$ would be replaced by $B \xrightarrow{a} C$, and $C \longrightarrow A$ by $C \longrightarrow \varepsilon$. Death variables may be easily detected, as these are exactly those variables that exhibit no observable transition rules.

Recall that we only work with normed processes. For the second simplification we need a notion of *norm* of a process.

Definition 18 (norm). *By* norm *of a process α, denoted $|\alpha|$, we mean the smallest possible number of observable transitions that appears in some sequence to the empty process. Formally, $|\alpha|$ is the length of the shortest word $a_1 \dots a_n \in \mathbb{A}ct^*$ such that*

$$\alpha \Longrightarrow \xrightarrow{a_1} \Longrightarrow \dots \Longrightarrow \xrightarrow{a_n} \Longrightarrow \varepsilon.$$

For unnormed processes we define their norm as infinity.

Look at that, the norm is *weak* in the sense that silent transitions do not count. Norm is additive, i.e. $|(\alpha||\beta)| = |\alpha| + |\beta|$, and invariant under the equivalences investigated by us, namely

$$\text{if } \alpha \approx \beta \text{ then } |\alpha| = |\beta|.$$

Indeed, if $|\alpha| < |\beta|$ then Spoiler wins weak bisimulation game with a sequence of moves from α to ε that witnesses the norm of α.

A transition is $\alpha \xrightarrow{\zeta} \beta$ is *norm preserving* if $|\alpha| = |\beta|$; in such case we write $\alpha \xrightarrow{\zeta}{}_0 \beta$ to point out that the change of norm caused by the transition in 0. In particular we will write $\longrightarrow_0$ for silent transitions, and $\Longrightarrow_0$ for the transitive and reflexive closure of $\longrightarrow_0$.

For variables X, Y such that $X \Longrightarrow_0 Y \Longrightarrow_0 X$ we have $X \approx Y$ by Lemma 4. We say that X is *redundant* because of Y, and vice versa. One can easily detect redundant variables and therefore we can unify them. That is, we can assume wlog. that the commutative context-free grammars do not contain redundant variables.

Sequences of responses. We write $\alpha \sqsubseteq \beta$ if there is some γ such that $\alpha||\gamma = \beta$ ($\sqsubseteq$ is thus the multiset inclusion of processes). Consider a pair or processes related by an ω-approximant, $\alpha \overset{X}{\approx}_\omega \beta$, for $X \in \{L, M, W\}$. By definition of weak bisimulation approximant game, for any Spoiler's move from α to α', labelled by ζ, say, there is a sequence

$$\beta_1, \beta_2, \beta_3, \dots \tag{2.9}$$

of processes such that $\alpha' \overset{X}{\approx}_i \beta_i$ for every $i \in N$. Clearly the sequence is not unique; for instance, β_k could be safely replaced with β_l for $l > k$. By Dickson's Lemma we know that there is an infinite subsequence that is non-decreasing with respect to $\sqsubseteq$:

$$\beta_{n_1} \sqsubseteq \beta_{n_2} \sqsubseteq \beta_{n_3} \sqsubseteq \dots \qquad \text{for } n_1 < n_2 < n_3 < \dots.$$

Moreover, as $n_i > i$, we have $\alpha' \overset{X}{\approx}_i \beta_{n_i}$ for every $i \in N$. Thus there is always some sequence (2.9) that is non-decreasing with respect to multiset inclusion. Such sequences are called *sequences of responses*.

2.5.1 Pure generators

Recall that we have assumed that there are no redundant variables. This assumption guarantees that the set of all pairs (X, Y) of variables satisfying

$$X \Longrightarrow_0 Y||\alpha \quad \text{for some process } \alpha$$

is an asymmetric and transitive relation, and hence a partial ordering on variables. Fix in the following any extension $>$ of this partial order; thus $X > Y$ whenever $X \Longrightarrow_0 Y||\alpha$.

Definition 19. *A* generator *is a variable X that allows a sequence $X \Longrightarrow_0 X||\alpha$ for some process α, in which case we say that X* generates *α. Call a generator X* pure *if $X \Longrightarrow_0 \alpha$ implies that $\alpha = \alpha'||X$ for some α': pure generators cannot vanish silently.*

Stirling showed decidability of weak equivalence for normed processes with only pure generators using a tableaux approach [53]. One motivation for this subclass is that it still allows for infinite branching and that approximants induced by weak bisimulation expansion do not converge at level ω, i.e. $\approx \neq \approx_\omega$. In this section we show that $\approx = \overset{\text{L}}{\approx}_\omega$ and thus due to lemma 6 conclude decidability.

Fix in the sequel a normed commutative context-free grammar, with variables $X_1 > X_2 > \ldots > X_k$, and with pure generators only.

Lemma 7. *For every process α, the set $Succ = \{\alpha' | \alpha \Longrightarrow_0 \alpha'\}$ can be partitioned into finitely many equivalences classes with respect to weak equivalence.*

Proof. From Lemma 4 one concludes that if $\alpha \Longrightarrow_0 \alpha||\beta$ then $\alpha \approx \alpha||\beta$. In other words if α contains variables which allow to generate β, then α and $\alpha||\beta$ are in the same equivalence class with respect to the weak equivalence relation.

Above remark allows us to restrict ourselves to the subset $Succ'$ of $Succ$ of processes which are obtained without use of *generating moves*[2]. Observe that $Succ'$ and $Succ$ have the same number equivalence classes as generators cannot vanish along $\Longrightarrow_0$ moves. Our goal is to show that $Succ'$ is finite which immediately implies the claim of the lemma.

Every derivation of $\alpha \Longrightarrow_0 \alpha'$ is a sum of derivations from variables belonging to α. If we prove that in silent norm preserving transitions without generating moves, we can only derive finitely many processes from each variable occurrence, then we will also prove that $Succ'$ is finite. We will show that this is indeed the case for all variables by induction over the assumed order $<$.

Checking the induction assumption goes as follows: from the smallest variable X_k using silent norm preserving transitions *without generating* we can derive only two processes, namely X_k or ε.

[2]Generating moves are those via *generating transitions* i.e transitions induced by *generating transition rules* of a form $X \longrightarrow X||\alpha$ for some process α.

To make an induction step assume $c_i > 0$ bounds the number of possible silent norm preserving derivations from any variable in $X_i \dots X_k$ and consider the variable X_{i-1}. We want to prove that c_{i-1} is bounded. In case X_{i-1} is a deadlock variable, i.e. $X_{i-1} \longrightarrow X_{i-1}$ is the only applicable rule, we can trivially bound the number of its derivations by 1. Otherwise, because we forbid generating moves we must have that any rule $X_{i-1} \longrightarrow {}_0\delta$ produces a multiset $\delta \in \{X_i \dots X_k\}^{\otimes}$. The fact that there are only finitely many rules that rewrite variable X_{i-1} implies that we can bound the number of its silent norm preserving derivations by

$$d \cdot c_i^l + 1,$$

where d is the number of rules for X_{i-1} and l is the maximal size of any right hand side of a rule rewriting X_{i-1}. □

Theorem 3. $\approx = \overset{L}{\approx}_\omega$ *for normed commutative context-free processes with only pure generators.*

Proof. For two non-equivalent processes $\alpha \not\approx \beta$ we define *equivalence level* of α and β as the largest ordinal κ such that $\alpha \overset{L}{\approx}_\kappa \beta$. This is well defined as the smallest ordinal κ such that $\alpha \overset{L}{\not\approx}_\kappa \beta$ is never a limit ordinal.

Assume towards a contradiction that we have $\alpha \overset{L}{\approx}_\omega \beta \overset{L}{\not\approx}_{\omega+1} \alpha$. Wlog. assume an optimal initial move (i.e. a move that strictly decreases equivalence level) $\alpha \overset{\zeta}{\Longrightarrow} \alpha'$ of Spoiler in the long weak bisimulation approximant game from the position (ω, α, β), and a sequence of responses $B' = \beta'_0, \beta'_1, \dots$ (recall that wlog. it is non-decreasing wrt. multiset inclusion.)

By Lemma 7, the set $Succ = \{\alpha'' | \alpha' \Longrightarrow_0 \alpha''\}$ contains finitely many equivalence classes with respect to weak equivalence relation. Let the set $Succ'$ be a finite set of representants of those classes in $Succ$.

This allows us to define a function $f : B' \rightarrow Succ'$ that maps $\beta'_i \in B'$ to an element in $Succ'$ that maximises their equivalence level κ:

$$\beta'_i \overset{L}{\approx}_\kappa f(\beta'_i) \text{ and } \forall_{\gamma \in Succ'} \forall_{\lambda \in Ord}\, \beta'_i \overset{L}{\approx}_\lambda \gamma \implies \kappa \geqslant \lambda.$$

This function is well defined because the set $Succ'$ is finite.

Now consider an infinite subsequence $B(\gamma)$ of B' that contains all elements which f maps to the process $\gamma \in Succ'$. By the pigeonhole principle such a subsequence exists.

Take two elements $\beta'_i \sqsubseteq \beta'_j$ of $B(\gamma)$ for arbitrary large $i < j$. We have

(1) $\gamma \overset{L}{\approx}_j \beta'_j$.

(2) β'_i and β'_j have the same norm.

The first fact holds because $\alpha' \in Succ'$ and equivalence levels of pairs of processes (β'_i, γ) and (β'_j, γ) are larger or equivalent to equivalence levels of pairs of processes (β'_i, α') and (β'_j, α), respectively. This is due to the definition of the function f.

To see why the second observation is true note that $|\alpha| \neq |\beta|$ implies

$$\alpha \overset{\mathrm{L}}{\not\approx}_{\min\{|\alpha|,|\beta|\}+1} \beta$$

as Spoiler only needs to decrease the smaller process to a deadlock which cannot be mimicked by Duplicator on the other process because the norms differ. We know

$$\beta'_i \overset{\mathrm{L}}{\approx}_i \alpha' \overset{\mathrm{L}}{\approx}_j \beta'_j, \text{ so } |\beta'_i| = |\alpha'| = |\beta'_j|$$

as otherwise i and j would be bounded by $|\alpha'| + 1$.

Consider the game from the position (j, α', β'_j) and a silent, norm preserving move $\beta'_j \Longrightarrow_0 \beta'_i$ made by Spoiler, which must be possible due to observation 2) and the fact that $\beta'_i \sqsubseteq \beta'_j$. Now by definition of the subsequence $B(\gamma)$ we deduce that $\alpha' \Longrightarrow_0 \gamma$ is an optimal response for Duplicator. Therefore by 1), we know that $\beta'_i \overset{\mathrm{L}}{\approx}_{j-1} \gamma$ so $\beta'_i \overset{\mathrm{L}}{\approx}_{j-1} \beta'_j$ by transitivity and the fact that $\beta'_j \overset{\mathrm{L}}{\approx}_{j-1} \gamma$. But now we have $\beta'_i \overset{\mathrm{L}}{\approx}_{j-1} \alpha'$ for arbitrarily high j and therefore $\beta'_i \overset{\mathrm{L}}{\approx}_{\omega} \alpha'$ which contradicts the optimality of Spoiler's very first move. □

2.5.2 Unnormed processes over one letter alphabet

Consider the subclass of commutative context-free processes that satisfy the following conditions:

(1) there is only one observable label, i.e. $\mathbb{A}ct_{\varepsilon} = \{\varepsilon, a\}$, and

(2) every variable has positive or infinite norm.

This class has been introduced in [54], where it was shown that for processes of this kind, Jančar and Hirshfeld's conjecture 1 holds, namely:

$$\approx = \overset{\mathrm{L}}{\approx}_{\omega+\omega} \text{ but } \approx \neq \overset{\mathrm{L}}{\approx}_{\omega} .$$

Note that this class is not a subclass of the *totally normed* systems [23] as it allows for variables of infinite norm. To illustrate $\approx \neq \overset{\mathrm{L}}{\approx}_{\omega}$ we use the following context-free grammar:

Example 12.

$$\begin{tikzcd} X \arrow[r] \arrow[d, "a"'] & Y \circlearrowleft A \arrow[d, "a"] & A \arrow[d, "a"] \\ a \circlearrowright Z & \varepsilon & \varepsilon \end{tikzcd}
\qquad
\begin{array}{ll} X \longrightarrow Y & X \xrightarrow{a} Z \\ Y \xrightarrow{a} \varepsilon & Y \longrightarrow Y||A \\ Z \xrightarrow{a} Z & A \xrightarrow{a} \varepsilon \end{array}$$

We show that this class has decidable weak equivalence by showing that approximants induced by *multiset* weak bisimulation expansion converge at level ω. Note that multiset approximants coincide with word approximants since there is only one observable label.

Theorem 4. $\approx = \overset{M}{\approx}_\omega$ *for the subclass of commutative context-free processes with a single observable label and no variables of norm* 0*.*

Proof. First observe that the first restriction implies that all processes with infinite norm must be equivalent and due to norm preservation cannot be equivalent to any process of finite norm. The second restriction guarantees that there are only finitely many different processes for any given finite norm. Whenever two processes have different but finite norms, they are certainly not related by $\overset{M}{\approx}_2$ as Spoiler may rewrite the smaller process to a deadlock in one move without allowing his opponent to do the same on the other process.

Assume towards a contradiction that $\alpha \overset{M}{\approx}_\omega \beta \overset{M}{\not\approx}_{\omega+1} \alpha$. So for an optimal initial move $\alpha \overset{w}{\Longrightarrow} \alpha'$ of Spoiler from the position (ω, α, β) there is a sequence of responses β_i. This sequence has to contain infinitely many of different processes as otherwise our assumption $\beta \overset{M}{\not\approx}_{\omega+1} \alpha$ would be false. By the pigeonhole principle, there must be at least one variable X that grows indefinitely along this sequence. Take two elements $\beta'_i \sqsubset \beta'_j$, $2 < i < j$ from this sequence such that X occurs more often in β'_j. By observation 2) and the fact that β'_i and β'_j have different norms we know that $\beta'_i \overset{M}{\not\approx}_2 \beta'_j$. Because $\beta'_i \overset{M}{\approx}_i \alpha' \overset{M}{\approx}_j \beta'_j$ and $i < j$ holds $\beta'_i \overset{M}{\approx}_i \alpha' \overset{M}{\approx}_i \beta'_j$. From this and the transitivity of $\overset{M}{\approx}_i$ we conclude that $\beta'_i \overset{M}{\approx}_i \beta'_j$ and because $2 < i$ also $\beta'_i \overset{M}{\approx}_2 \beta'_j$ which is a contradiction. $\square$

Chapter 3

Bisimulation equivalence for commutative context-free processes

In this chapter we concentrate on normed commutative context-free processes only. We will investigate both weak equivalence and branching equivalence.

Our main technical result is the proof of the following *bounded response property* (formulated precisely in Theorem 5 in Section 3.1): if Duplicator has a matching response, then he also has a response that leads to a process of size linearly bounded with respect to the other (Spoiler's) process. Importantly, we obtain an effective bound on the linear coefficient, which enables us to prove (Theorem 6) decidability of branching bisimulation equivalence. The proof of Theorem 5 is quite complex and involves a lot of subtle investigations of combinatorics of silent transitions. The main purpose to do that is to eliminate unnecessary silent transitions and to make Duplicators matching responses small.

A major part of the proof works for weak equivalence equally well (and, as we believe, also for any reasonable equivalence that lies between branching and weak equivalence). However, for weak bisimulation we can merely show *existence* of the linear coefficient witnessing the bounded response property, while we are not able to obtain any effective bound. Nevertheless we strongly believe (and conjecture) that a further elaboration of our approach will enable proving decidability of weak bisimulation equivalence. In particular, we actually can reprove (once more) decidability of weak equivalence in the pure-generators subclass defined in Section 2.5.1.

3.1 Decidability via bounded response property

It was known before that branching and weak equivalences are semi-decidable [15]. A brief explanation of the positive procedure can be found in Lemma 5 in Chapter 2 . The main obstacle for a semi-decision procedure for inequivalence is that commutative context-free processes are not finitely branching with respect to branching or weak equivalence: a priori Duplicator has infinitely many possible responses to a Spoiler's move. The main insight of this chapter is that commutative context-free processes are essentially finitely branching, in the following sense. Define the size of a process as its multiset cardinality. For instance,

$$\text{size}(A^4||B^3||C) = 8.$$

Then Duplicator has always a response of size bounded linearly with respect to a Spoiler's process (as formulated in Theorem 5 below).

Formally we define two new expansions, both of them parametrized by constants c. For sufficient large c those expansions define weak/branching bisimulation respectively. Moreover approximants induced by those two expansions stabilize at level ω. The main obstacle to use them as a tool to solve an equivalence problem is to estimate the proper value of the constant c. We know how to do that in case of branching equivalence. To solve weak equivalence, we still need a deeper insight.

Definition 20. *Let $B \subseteq V \times V$. A pair (α, β) satisfies* c-weak bisimulation expansion *wrt. B if and only if for every $\zeta \in \mathbb{A}ct_\varepsilon$:*

if $\alpha \xrightarrow{\zeta} \alpha'$ then $\beta \overset{\zeta}{\Longrightarrow} \beta'$ such that $(\alpha', \beta') \in B$ and size$(\beta') \leqslant c \cdot$ size(α')

if $\beta \xrightarrow{\zeta} \beta'$ then $\alpha \overset{\zeta}{\Longrightarrow} \alpha'$ such that $(\alpha', \beta') \in B$ and size$(\alpha') \leqslant c \cdot$ size(β').

Definition 21. *Let $B \subseteq V \times V$. A pair (α, β) of processes satisfies the* c-branching bisimulation expansion *wrt. B if for every $\zeta \in \mathbb{A}ct_\varepsilon$:*

$$\textit{if } \alpha \xrightarrow{\zeta} \alpha' \textit{ then } \beta \Longrightarrow \bar{\beta} \xrightarrow{\zeta} \beta' \textit{ such that } (\alpha, \bar{\beta}) \in B,\ (\alpha', \beta') \in B \textit{ and}$$

$$\textit{size}(\beta') \leqslant c \cdot \textit{size}(\alpha') \ \wedge\ \textit{size}(\bar{\beta}) \leqslant c \cdot \textit{size}(\alpha);.$$

$$\textit{if } \beta \xrightarrow{\zeta} \beta' \textit{ then } \alpha \Longrightarrow \bar{\alpha} \xrightarrow{\zeta} \alpha' \textit{ such that } (\beta, \bar{\alpha}) \in B,\ (\beta', \alpha') \in B \textit{ and}$$

$$\textit{size}(\alpha') \leqslant c \cdot \textit{size}(\beta') \ \wedge\ \textit{size}(\bar{\alpha}) \leqslant c \cdot \textit{size}(\beta);.$$

In a natural way, using the above expansions we define new equivalences, namely *c-branching equivalence* and *c-weak equivalence*, denoted $\simeq^c$ and $\approx^c$, respectively, as the union of all c-weak/branching bisimulations. The important fact is that there is a strict connection between c-weak/branching equivalences and standard weak/branching equivalences as stated in Theorems 5 and 7.

Furthermore, in a natural way we get new approximants. To do that we use abstract approximants (cf. Definition 11 in Section 1.1.4) and instantiate them with c-weak bisimulation expansion and c-branching bisimulation expansion, respectively. We denote them $\approx^c_\kappa$, $\simeq^c_\kappa$ as one could expect.

The following example serves as an illustration what we get using c-expansions:

Example 13. *Consider weak equivalence for the commutative context-free processes from Example 1:*

$$\begin{array}{lllllllllllll} P & \longrightarrow & \varepsilon & \quad & P & \longrightarrow & P||A & \quad & P & \xrightarrow{b} & \varepsilon & \quad & P & \longrightarrow & Q \\ Q & \longrightarrow & \varepsilon & & Q & \longrightarrow & Q||A & & A & \xrightarrow{a} & \varepsilon & & A & \longrightarrow & \varepsilon \end{array}$$

We have $P \not\approx P||Q$ but $P \approx_\omega P||Q$, i.e. $P \approx_n P||Q$ for all $n < \omega$. Indeed, if for instance Spoiler starts with

$$P||Q \xrightarrow{b} Q$$

then Duplicator can respond with $P \Longrightarrow P||A^k \xrightarrow{b} A^k$ for an arbitrarily large k.

On the other hand $\approx$ coincides with $\approx^c$ for $c = 1$, i.e., Duplicator can respond with a process of size at most equal to the size of Spoiler's process. Intuitively, this is due to an observation that two processes are equivalent iff

- *they have the same number of occurrences of P;*
- *Q occurs in both, or in none of them;*
- *in the latter case, the number of occurrences of A is the same.*

Therefore Duplicator, conforming to the size restriction, can keep this invariant. (Thus $\approx$ coincides with $\approx^c$ for any $c \geqslant 1$.)

In consequence $P \not\approx^c P||Q$. In agreement with Lemma 8 it is not true that $P \approx^c_n P||Q$ for all $n \in \mathbb{N}$, for instance for $c = 1$ it holds that $P \not\approx^c_3 P||Q$.

For convenience we use in the sequel a new symbol $\equiv$ to stand for any of the two equivalences, $\simeq$ or $\approx$. Symbol $\equiv^c_\kappa$ stands for any of the two c-approximants.

Lemma 8. *For any $c \in \mathbb{N}$, the approximants stabilize at level ω, i.e. $\equiv^c_\omega = \equiv^c$. Moreover for any $n \in \mathbb{N}$, $\equiv^c_n$ is computable.*

To prove the first statement let's recall that from any position after any Spoiler move Duplicator has only finitely many available responses. Thus we can apply König's Lemma and conclude that if Spoiler wins then the tree of a game is finite. For the second statement observe that for any fixed c, both c-expansions can be expressed in Presburger arithmetic, similarly as it is done for other expansions in Section 2.1.1.

Let the size of a commutative context-free grammar be the sum of sizes of all production rules. Our main technical result is an efficient estimation of the constant c in Definition 21, with respect to the size of a commutative context-free grammar:

Theorem 5 (bounded response property of $\simeq$)**.** *Given a normed commutative context-free grammar, one can compute $c \in \mathbb{N}$ such that branching equivalence coincides with c-branching equivalence, $\simeq \, = \, \simeq^c$, in the labelled transition system induced by the grammar.*

The proof of Theorem 5 is deferred to Sections 3.2–3.4. The theorem leads us directly to decidability:

Theorem 6. *Branching equivalence $\simeq$ is decidable over normed commutative context-free processes.*

Proof. The decision procedure starts with computing $c \in \mathbb{N}$, according to Theorem 5, such that branching equivalence coincides with c-branching equivalence. Then we run two semi-decision procedures (along the lines of Section 2.5): the positive one for branching equivalence and the negative one for c-branching equivalence.

For the positive side we use the decision procedure from Lemma 5. For the negative side, we observe that due to Lemma 8 and Theorem 5, the assumptions of Lemma 6 are satisfied. $\square$

For weak equivalence we obtain a result weaker than Theorem 5, as we are not able to prove that the coefficient c is computable:

Theorem 7 (bounded response property of $\approx$)**.** *For every normed commutative context-free grammar, there is $c \in \mathbb{N}$ such that weak equivalence coincides with c-weak equivalence, $\approx \, = \, \approx^c$, in the labelled transition system induced by the grammar.*

Theorem 7 follows, similarly as Theorem 5, from our results in Sections 3.2–3.4. We note that Theorem 7 does not imply decidability of weak equivalence.

3.1.1 Proof strategy

The rest of Chapter 3 is devoted to the proofs of Theorems 5 and 7. Consider a fixed normed commutative context-free grammar from now on. In Section 3.2 for a given process α, and with respect to an investigated equivalence, we define a notion of normal form $\text{nf}(\alpha)$. Moreover we provide linear lower and upper bounds on its size:

$$\text{size}(\alpha) \leqslant \text{size}(\text{nf}(\alpha)) \leqslant c \cdot \text{size}(\alpha) \tag{3.1}$$

(the lower bound holds assumed that α is minimal wrt. multiset inclusion in its equivalence class). The results of Section 3.2 apply both to branching and weak equivalence (as well as to other variants of bisimulation that lay between branching and weak equivalence, cf. [17]). However, the content of Section does not provide an effective bound on the linear coefficient c.

The computable estimation of the coefficient c is derived in Section 3.3, in case of branching equivalence. Finally, in Section 3.4 we show how the bounds (3.1) are used to prove Theorem 5. Section 3.4 contains also the proof of Theorem 7.

As observed e.g. in [53], a crucial obstacle in proving decidability are *generating* transition rules of the form $X \longrightarrow X||Y$, as they may be used by Duplicator to reach silently $X||Y^m$ for arbitrarily large m. A great part of our proofs is an analysis of combinatorial complexity of generating transition rules and, roughly speaking, elimination of 'unnecessary' generations.

Weak equivalence. Branching equivalence is more discriminating than weak equivalence. The whole development of Section 3.2 is still valid if weak equivalence is considered in place of branching equivalence. Furthermore, except one single case, the entire proof of estimation of the coefficient in Section 3.3 remains valid too. Interestingly, this single case is obvious in the pure-generators subclass, and thus our proof remains valid for weak equivalence in this subclass. We conjecture that the single missing case is provable for weak equivalence and thus Theorem 5 holds for weak equivalence just as well. This would imply decidability of weak equivalence.

3.2 Normal form by squeezing

The results of this section are quite general and apply equally well to branching equivalence $\simeq$ and to weak equivalence $\approx$ over commutative context-free grammars. (This will not be however the case in later sections.) We will thus continue using the symbol $\equiv$ to stand for either $\simeq$ or $\approx$ in this section. Actually the only place where we need to distinguish between weak and branching equivalence is Lemma 15 that speaks of matching Duplicator's responses. Furthermore, we claim that all the results of Section 3.2 apply equally well to intermediate notions of bisimulation, laying between branching and weak equivalence, as introduced in [17].

In the sequel we will use, sometimes implicitly, the well-known fact that both branching and weak equivalences are substitutive over commutative context-free processes, i.e.

$$\alpha \equiv \beta \implies \alpha||\gamma \equiv \beta||\gamma.$$

For the rest of Section 3.2 fix an arbitrary normed commutative context-free grammar.

3.2.1 Normal forms

In Section 3.2 we develop a framework useful for the proofs of Theorems 5 and Theorem 7, to be given in the following sections.

An important role in our development will be played by normal forms of processes that identify the equivalence classes uniquely. The normal forms are defined using the linear well-founded order $\preceq$ on processes[1], as defined in Definition 25 in Section 3.2.3 below. We prefer to postpone the definition of $\preceq$, in order to avoid inessential technical details at this early stage.

Definition 22 (normal form)**.** *For any process α let nf(α) denote the smallest process with respect to $\preceq$, which is equivalent to α.*

Clearly $\alpha \equiv \mathrm{nf}(\alpha)$ and thus we conclude that bisimulation equivalence is characterized by syntactic equality of normal forms:

Lemma 9. *$\alpha \equiv \beta$ if and only if nf$(\alpha) =$ nf(β).*

The main contribution of Section 3.2 is, roughly speaking, providing lower and upper bound on the size of nf(α), relative to the size of α, (cf. Lemmas 21 and 22 appearing at the end of this section). The technical tool will be an operation called below *squeeze* (defined in Section 3.2.4), which transforms a process α into an equivalent one, squeeze$(\alpha) \equiv \alpha$. We will prove that iterative application of squeeze eventually converges to the normal form:

$$\alpha \equiv \mathrm{squeeze}(\alpha) \equiv \mathrm{squeeze}^2(\alpha) \equiv \ldots \equiv \mathrm{squeeze}^i(\alpha) = \mathrm{nf}(\alpha),$$

for some i depending on α. The estimations on the size of normal form will follow easily from the fact that we will be able to control the increase of size of squeeze at every iteration.

3.2.2 Decreasing transitions

In the sequel we will pay special attention to norm preserving ε-transitions. Moreover we say that a transition $\alpha \xrightarrow{\zeta} \beta$ is *norm reducing* if $|\alpha| = |\beta| + 1$.

Definition 23. *We call the transition $\alpha \xrightarrow{\zeta} \beta$* decreasing *if either $\zeta \in \mathbb{A}ct$ and the transition is norm reducing, or $\zeta = \varepsilon$ and the transition is norm preserving.*

Keep in mind that every variable has a sequence of decreasing transitions leading to the empty process ε.

Lemma 10 (decreasing response)**.** *Whenever $\alpha \equiv \beta$ and $\alpha \xrightarrow{\zeta} \alpha'$ is decreasing then any Duplicator's matching sequence of transitions from β contains exclusively decreasing transitions.*

Proof. Follows from the following simple observations: $\equiv$ is norm preserving; for $a \neq \varepsilon$, the transition relation $\xrightarrow{a}$ may decrease the norm by at most one; the transition relation $\xrightarrow{\varepsilon}$ never decreases the norm. □

[1] We deliberately use the same symbol as for branching pre-order, hoping that no confusion will result. Pre-orders are not investigated at all in Chapter 3.

As we have discussed in Section 2.5, we may assume wlog. that there are no redundant variables, i.e. no two distinct variables X, Y satisfy $X \Longrightarrow_0 Y \Longrightarrow_0 X$. Relying on this assumption, we may define a partial order induced by decreasing transitions.

Definition 24. *For variables X, Y, let $X >_{decr} Y$ if there is a sequence of decreasing transitions leading from X to Y. Let $>$ denote an arbitrary total order extending $>_{decr}$.*

Look at that we do not assume that $X > Y \implies |X| \geqslant |Y|$. Indeed, the order $>$ may be chosen in an arbitrary way. In Section 3.2 it is only relevant to have some *fixed* linear order on variables. In the next sections we will alter between different orders, but only those extending $>_{\mathrm{decr}}$.

In the sequel we assume that there are n variables $\{X_1, \ldots, X_n\}$, ordered:

$$X_1 > X_2 > \ldots > X_n.$$

If $\alpha \in \{X_1, \ldots X_k\}^{\otimes}$ and $\beta \in \{X_{k+1}, \ldots, X_n\}^{\otimes}$, for $k \in \{0, \ldots, n\}$, we say that k *separates* α and β. (Note that there may be more than one k separating a given pair of processes.) If some such k exists, we say that α and β are separated. By $\alpha \cdot \beta$ we mean *concatenation*, that is the composition of processes α, β under the assumption that α and β are separated. Thus formally speaking, concatenation is a partially defined operation, and whenever we write $\alpha \cdot \beta$ we implicitly assume that α and β are separated.

Directly from the definition of $>$ we deduce:

Lemma 11 (decreasing transition)**.** *If a decreasing transition*

$$X_1^{a_1} \cdot \ldots \cdot X_n^{a_n} \xrightarrow{\zeta} X_1^{b_1} \cdot \ldots \cdot X_n^{b_n}$$

is performed by X_k, for $k \in \{1, \ldots, n\}$, then $b_1 = a_1, \ldots, b_{k-1} = a_{k-1}$.

Let's recall Definition of the generator (Definition 19). Note that generating transitions are decreasing.

Lemma 12 (decreasing transition cont.)**.** *If a decreasing transition, as in Lemma 11, is not generating then $b_k = a_k - 1$.*

Following [53], we say that X *generates* Y if $X \Longrightarrow_0 X || Y$. Thus if $X \Longrightarrow_0 X || \bar{\delta}$ then X generates every variable that appears in $\bar{\delta}$. In particular, X may generate itself. Note that each generated variable is of norm 0. More generally, we say that α generates β if $\alpha \Longrightarrow_0 \alpha || \beta$. This is the case precisely iff every variable occurring in β is generated by some variable occurring in α.

As a direct corollary of Lemma 4 we obtain:

Lemma 13. *If α generates β then $\alpha \equiv \alpha || \bar{\beta}$ for any $\bar{\beta} \sqsubseteq \beta$.*

Lemma 13 will be useful in the sequel, as a tool for eliminating unnecessary transitions and thus decreasing the size of a resulting process.

3.2.3 Unambiguous processes

Once we have a fixed ordering on variables, a process $X_1^{a_1} \cdot \ldots \cdot X_n^{a_n}$ may be equivalently presented as a sequence of exponents $(a_1, \ldots, a_n) \in \mathbb{N}^n$. In this perspective, $\sqsubseteq$ is the point-wise order. The sequence presentations induce additionally the lexicographic order on processes, denoted $\preceq$.

Definition 25. *We define the order $\preceq$ on processes as follows:*

$$X_1^{a_1} \cdot \ldots \cdot X_n^{a_n} \prec X_1^{b_1} \cdot \ldots \cdot X_n^{b_n} \text{ iff } \exists k.\ (a_k < b_k \text{ and } \forall i < k.\ a_i = b_i).$$

The same may be written briefly using concatenation: $\alpha \prec \beta$ if $\alpha = \gamma \cdot X_k^a \cdot \alpha'$, $\beta = \gamma \cdot X_k^b \cdot \beta'$, and $a < b$.

For instance, the decreasing non-generating transitions $\alpha \xrightarrow{\zeta} \beta$ always go strictly down the lexicographical order, i.e. $\alpha \succ \beta$.

We will exploit the fact that the order $\preceq$ is total, and thus each equivalence class exhibits the least element. The least process in the equivalence class of α will serve as the normal form of α, denoted nf(α) (cf. Definition 22 in Section 3.2.1).

The sequence presentation allows us to speak naturally of *prefixes* of a process: the k-*prefix* of $X_1^{a_1} \cdot \ldots \cdot X_n^{a_n}$ is the process $X_1^{a_1} \cdot \ldots \cdot X_k^{a_k}$, for $k = 0 \ldots n$.

We now introduce one of the core notions used in the proof: *unambiguous processes* and their *greatest extensions*.

Definition 26 (unambiguous processes)**.** *A process $X_1^{a_1} \cdot \ldots \cdot X_n^{a_n}$, is called* k-unambiguous *if for every $i \in \{1, \ldots, k\}$, $\alpha, \beta \in \{X_{i+1}, \ldots, X_n\}^{\otimes}$ and $b, c \in \mathbb{N}$, if $b \neq c$ and*

$$X_1^{a_1} \cdot \ldots \cdot X_{i-1}^{a_{i-1}} \cdot X_i^b \cdot \alpha \ \equiv\ X_1^{a_1} \cdot \ldots \cdot X_{i-1}^{a_{i-1}} \cdot X_i^c \cdot \beta \tag{3.2}$$

hen $b, c \geqslant a_i$. When $k = n$ we write simply unambiguous.

Note that being k-unambiguous is a property of the k-prefix: a process is k-unambiguous iff its k-prefix is so.

Observe that an unambiguous process is necessarily the least one wrt. $\preceq$ in its equivalence class, as the definition disallows the equivalence (3.2) to hold for $a_i = b > c$. On the other hand, it is not immediately clear whether the opposite implication holds, i.e. whether every equivalence class contains some unambiguous process. In the sequel we will show that this is actually the case.

Example 14. *Consider the following grammar:*

$$\begin{array}{lllllllll} X_1 & \xrightarrow{a} & X_1 & \quad X_2 & \xrightarrow{b} & X_3 & \quad X_3 & \xrightarrow{b} & \varepsilon \\ X_1 & \longrightarrow & \varepsilon & \quad X_2 & \longrightarrow & X_3 & \quad X_3 & \longrightarrow & \varepsilon \end{array}$$

and an order $X_1 > X_2 > X_3$ on variables. We observe that $X_1^2 \approx X_1$, therefore the process X_1^2 is not (1-)unambiguous. On the other hand $X_1 \not\approx \alpha$ for any $\alpha \in \{X_2, X_3\}^{\otimes}$ (because neither X_2 nor X_3 can perform an a transition), so X_1 is unambiguous. Furthermore $X_1 \cdot X_2 \approx X_1 \cdot X_3^2$, hence $X_1 \cdot X_2$ is not (2-)unambiguous. Finally we observe that $X_1 \cdot X_3^2 \not\approx X_1 \cdot X_3$. Therefore $X_1 \cdot X_3^2$ is unambiguous, but also $X_1 \cdot X_3$ is so.

Observe that a prefix of a k-unambiguous process is k-unambiguous as well. Moreover, k-unambiguous processes are downward closed wrt. $\sqsubseteq$: whenever $\alpha \sqsubseteq \beta$ and β is k-unambiguous, then α is k-unambiguous as well.

Directly by Definition 26, if $\gamma = X_1^{a_1} \cdot \ldots \cdot X_{k-1}^{a_{k-1}}$ is $(k-1)$-unambiguous then it is automatically k-unambiguous (in fact j-unambiguous for any $j \geqslant k$). This corresponds to $a_k = 0$. We will be especially interested in the greatest value of a_k possible, as formalized in the definition below.

Definition 27 (the greatest extension)**.** *The* greatest k-extension *of a $(k-1)$-unambiguous process $\gamma \in \{X_1 \ldots X_{k-1}\}^{\otimes}$ is that process among k-unambiguous processes $\gamma \cdot X_k^a$ that maximizes a.*

Clearly the greatest extension does not need exist in general, as illustrated below.

Example 15. *Consider the processes from Example 14. The process X_1 is the greatest 1-extension of the empty process as X_1^2 is not 1-unambiguous. X_1 is also its own greatest 2-extension. Furthermore, X_1 does not have the greatest 3-extension. Indeed, $X_1 X_3^a$ is not equivalent to $X_1 X_3^b$, for $a \neq b$, therefore $X_1 X_3^a$ is 3-unambiguous for any a.*

Definition 28 (unambiguous prefix)**.** *By an* unambiguous prefix *of a process*

$$X_1^{a_1} \cdot \ldots \cdot X_n^{a_n}$$

we mean any k-prefix $X_1^{a_1} \cdot \ldots \cdot X_k^{a_k}$ that is k-unambiguous, for $k = 0 \ldots n$. The maximal unambiguous prefix *is the one that maximizes k.*

Example 16. *For the grammar from Example 14, the maximal unambiguous prefix of $X_1 \cdot X_2^2$ is X_1, and the maximal unambiguous prefix of $X_1^2 \cdot X_2$ is the empty process.*

3.2.4 Squeezes

The following lemma is fundamental for our subsequent development. The rough idea is as follows. For an unambiguous α consider a sequence of decreasing transitions from $\alpha \cdot \beta$, for an arbitrary β. The resulting process is necessarily of the form $\alpha' || \beta'$, where α' is obtained from α by a subsequence of transitions, and β' is obtained from β by the remaining subsequence of transitions. The lemma says that up to equivalence, the same process is reached by the latter subsequence of transitions.

Lemma 14. *Consider a process $\alpha \cdot \beta$, where α is unambiguous, and a sequence of decreasing transitions:*

$$\alpha \cdot \beta \xrightarrow{\zeta_1} \ldots \xrightarrow{\zeta_l} \alpha' || \beta'' \tag{3.3}$$

with α' and β' originating from α and β, respectively, i.e.

$$\alpha \xrightarrow{\zeta_{j_1}} \ldots \xrightarrow{\zeta_{j_k}} \alpha' \qquad \textit{and} \qquad \beta \xrightarrow{\zeta_{i_1}} \ldots \xrightarrow{\zeta_{i_m}} \beta', \tag{3.4}$$

for some sequences $j_1 < \ldots < j_k$ and $i_1 < \ldots < i_m$ of indices that partition the sequence $1, 2, \ldots, l$. Suppose that

$$\alpha' || \beta' \equiv \alpha || \gamma$$

for some γ. Then $\alpha' \equiv \alpha$ and hence $\alpha' || \beta' \equiv \alpha \cdot \beta'$.

Proof. Our goal is to show the following two facts:

- α and β' are separated (in other words, β' contains only variables which are smaller than all variables from α with respect to $>$), and
- $\alpha' \equiv \alpha$.

Indeed, in this case $\alpha \cdot \beta'$ is well defined and $\alpha' || \beta' \equiv \alpha \cdot \beta'$ due to substitutivity.

Let's prove the first item first. Observe that each variable X occurring in the process β' is an effect of a sequence of decreasing transitions originating from some variable Y occurring in the process β, hence $Y \geqslant X$. Thus α and β' are separated since α and β are.

Now it remains to prove the second item, namely $\alpha' \equiv \alpha$. Extend the sequence of transitions (3.3) with

$$\alpha' || \beta' \Longrightarrow_0 \bar{\alpha}' || \beta'$$

induced by a sequence

$$\alpha' \Longrightarrow_0 \bar{\alpha}'$$

leading from α' to a $\sqsubseteq$-minimal process $\bar{\alpha}' \sqsubseteq \alpha'$ which is equivalent to α'. By substitutivity $\bar{\alpha}'||\beta'$ is equivalent to $\alpha'||\beta'$, and thus also to $\alpha||\gamma$:

$$\bar{\alpha}'||\beta' \equiv \alpha||\gamma. \tag{3.5}$$

We claim that $\bar{\alpha}' = \alpha$, which immediately implies $\alpha' \equiv \alpha$.

Towards contradiction, suppose $\bar{\alpha}' \neq \alpha$. Let k be the first coordinate on which $\bar{\alpha}'$ doesn't agree with α. In other words:

$$\alpha = \theta \cdot X_k^a \cdot \omega \qquad \bar{\alpha}' = \theta \cdot X_k^{a'} \cdot \omega' \qquad \text{and} \qquad a \neq a',$$

for some k, a, a' and processes θ, ω, and ω'. We consider two cases.

First suppose $a > a'$. As α and β' are separated, and $a > 0$, we know that all variables appearing in β' are smaller than X_k. Thus $\bar{\alpha}'||\beta'$ may be presented as

$$\bar{\alpha}'||\beta' = \theta \cdot X_k^{a'} \cdot (\omega'||\beta'),$$

Recall that the process $\alpha = \theta \cdot X_k^a \cdot \omega$ is unambiguous. By the very definition of unambiguous processes, as $a' < a$ then the process $\bar{\alpha}'||\beta'$ can not be equivalent to $\alpha||\gamma$, which contradicts (3.5).

As the last remaining case, suppose $a < a'$. Because in the sequence of transitions $\alpha \Longrightarrow \bar{\alpha}'$ we use only decreasing transitions, and because k is the first coordinate on which $\bar{\alpha}'$ differs from α, by Lemmas 11 and 12 we deduce that X_k was created via some generating transition. Thus $\theta \cdot X_k^a$ generates X_k, and by Lemma 13 we conclude that

$$\theta \cdot X_k^{a'} \equiv \theta \cdot X_k^a.$$

This means that $\bar{\alpha}' = \theta \cdot X_k^{a'} \cdot \omega' \equiv \theta \cdot X_k^a \cdot \omega'$ and $\bar{\alpha}' \sqsupset \theta \cdot X_k^a \cdot \omega'$ which is in contradiction with $\sqsubseteq$-minimality of $\bar{\alpha}'$. □

The following simple example illustrates the reasoning in the proof above.

Example 17. *Consider the grammar from Example 1 with an order $P > Q > A$. Clearly process P is the only one which can perform transition labelled with b and no other variable can reach P via a sequence of transitions, thus P is unambiguous. Instantiate Lemma 14 with $\alpha = P$, $\beta = Q$, and a sequence of decreasing transitions*

$$P \cdot Q \Longrightarrow P \cdot Q \cdot A^9 \xrightarrow{a} P \cdot Q \cdot A^8 \longrightarrow P \cdot A^8,$$

with all A's produced using the transition rule $P \longrightarrow P||A$. Clearly the assumption of Lemma 14 holds as $P \cdot A^8 \sqsupseteq P$.

Then Lemma 14 says that $P \cdot A^8 \equiv P$. Indeed, this must be true as P generates A. This simple example has an advantage of being general enough: in Lemma 14, all *variable occurrences in α' that do not belong to α are actually generated by α.*

Lemma 15. *Under the assumptions of Lemma 14, if the sequence of transitions* (3.3) *is a matching Duplicator's move (which means in particular that all symbols ζ_i are ε, except possibly one), then the subsequence of transitions originating from β:*

$$\alpha \cdot \beta \xrightarrow{\zeta_{i_1}} \ldots \xrightarrow{\zeta_{i_m}} \alpha \cdot \beta' \tag{3.6}$$

is also a matching Duplicator's move.

Proof. For weak equivalence it is sufficient that the last process $\alpha \cdot \beta'$ in (3.6) is equivalent to the last process in (3.3). In case of branching equivalence we need to inspect also the second last process in (3.6). We know however that the sequence in (3.3), being a matching Duplicator's response, has the form:

$$\alpha \cdot \beta \Longrightarrow \xrightarrow{\zeta} \alpha' || \beta' \tag{3.7}$$

(i.e. $\zeta_1 = \ldots = \zeta_{l-1} = \varepsilon$). Recall that α' is equivalent to α, by Lemma 14. As the last transition in (3.7) is the only one in the sequence that may change equivalence class, and $\alpha' || \beta'$ is equivalent to $\alpha \cdot \beta'$, the process obtained by transitions originating from β, we claim that the last transition necessarily originates from β, i.e.

$$\alpha \cdot \beta \Longrightarrow \alpha' || \beta'' \xrightarrow{\zeta} \alpha' || \beta'$$

for some β''. Thus restricting to only transitions originating from β we obtain

$$\alpha \cdot \beta \Longrightarrow \alpha \cdot \beta'' \xrightarrow{\zeta} \alpha \cdot \beta'.$$

The sequence is necessarily a matching Duplicator's response as $\alpha \cdot \beta''$ is equivalent to $\alpha' || \beta''$. □

A direct conclusion from Lemmas 14 and 15 is the following result that speaks about an interplay between composition and concatenation with an unambiguous process. Consider an unambiguous γ and suppose that the following two processes are equivalent:

$$\gamma \cdot \alpha \equiv (\gamma || \beta). \tag{3.8}$$

Then for any decreasing Spoiler's move from the right process originating from β, there is a matching Duplicator's move from the left one that only engages α. The precise formulation follows.

Lemma 16. *Let γ be a k-unambiguous process and let α, β be arbitrary processes satisfying* (3.8). *Then for any decreasing transition $\beta \xrightarrow{\zeta} \beta'$, giving rise to a Spoiler's* *move*

$$\gamma||\beta \xrightarrow{\zeta} \gamma||\beta'$$

there is a sequence of decreasing transitions:

$$\alpha \xrightarrow{\zeta_1} \dots \xrightarrow{\zeta_n} \alpha'$$

that gives rise to a matching Duplicator's move

$$\gamma \cdot \alpha \xrightarrow{\zeta_1} \dots \xrightarrow{\zeta_n} \gamma \cdot \alpha',$$

as required by the definition of branching or weak bisimulation expansion.

Proof. Consider a matching Duplicator's move (all transitions are necessarily decreasing by Lemma 10):

$$\gamma \cdot \alpha \xrightarrow{\zeta_1} \dots \xrightarrow{\zeta_n} \gamma' \cdot \alpha'. \tag{3.9}$$

As the move is matching, we know that

$$\gamma' \cdot \alpha' \equiv \gamma||\beta'.$$

The process γ is unambiguous and $\gamma||\beta' \sqsupseteq \gamma$, which allows us to apply Lemma 14, to obtain a subsequence of (3.9)

$$\alpha \xrightarrow{\zeta_{i_1}} \dots \xrightarrow{\zeta_{i_m}} \bar{\alpha}$$

such that $\gamma \cdot \bar{\alpha} \equiv \gamma' \cdot \alpha'$. Then by Lemma 15 we learn that the subsequence is a matching Duplicator's move. □

The lemma to follow applies Lemma 16 to specially chosen of unambiguous processes, namely to the greatest k-extensions $\gamma \cdot X_k^a$ of unambiguous processes γ.

Lemma 17 (squeezing out)**.** *Suppose γ is a $(k-1)$-unambiguous process with the greatest k-extension $\gamma \cdot X_k^a$. Then for some process δ it holds:*

$$\gamma \cdot X_k^{a+1} \equiv \gamma \cdot X_k^a \cdot \delta. \tag{3.10}$$

Proof. By δ, δ', etc. we denote below processes from $\{X_{k+1} \dots X_n\}^{\otimes}$.

As a is the maximal extension of γ, there is some $b > a$ and some processes δ, δ' such that

$$\gamma \cdot X_k^b \cdot \delta \equiv \gamma \cdot X_k^a \cdot \delta'.$$

Consider an arbitrary sequence of decreasing transitions

$$X_k^b \cdot \delta \xrightarrow{\zeta_1} \ldots \xrightarrow{\zeta_m} X_k^{a+1}.$$

By Lemma 16 applied to the unambiguous process $\gamma \cdot X_k^a$, there is a sequence of matching (necessarily decreasing) transitions

$$\delta' \xrightarrow{\psi_1} \ldots \xrightarrow{\psi_l} \delta'',$$

for some δ'', such that

$$\gamma \cdot X_k^{a+1} \equiv \gamma \cdot X_k^a \cdot \delta''.$$

This completes the proof. □

Definition 29. *If a $(k-1)$-unambiguous process $\gamma \in \{X_1 \ldots X_{k-1}\}^{\otimes}$ has the greatest k-extension, say $\gamma \cdot X_k^a$, then any $\delta \in \{X_{k+1} \ldots X_n\}^{\otimes}$ satisfying* (3.10) *is called a γ-*squeeze *of X_k.*

By the very definition, X_k has a γ-squeeze only if γ has the greatest k-extension. Lemma 17 shows the opposite: if γ has the greatest k-extension then X_k has a γ-squeeze, that may depend in general on γ and k. The squeeze is however not uniquely determined and in fact X_k may admit many different γ-squeezes. In the sequel assume that for each $(k-1)$-unambiguous $\gamma \in \{X_1 \ldots X_{k-1}\}^{\otimes}$ and X_k, some γ-squeeze of X_k is chosen; this squeeze will be denoted by $\delta_{k,\gamma}$.

Example 18. *Consider one more time the grammar from Example 1, with the order $P > Q > A$. Observe that $Q^2 \equiv Q$ which means that Q^2 is not an unambiguous process and thus can be squeezed. If we fix $\gamma = P^3$, say, then one possible γ-squeeze of Q is the empty one: $P^3 \cdot Q^2 \equiv P^3 \cdot Q \cdot \varepsilon$, but there are many others, for instance $P^3 \cdot Q^2 \equiv P^3 \cdot Q \cdot A^n$, for any $n \geqslant 0$. The same squeezes are fine for any other $\gamma \in \{P\}^{\otimes}$.*

Definition 30 (squeezing step)**.** *For a given process α, assuming it is not n-unambiguous, let γ be its maximal unambiguous prefix. Thus there is $k \leqslant n$ such that*

$$\alpha = \gamma \cdot X_k^a \cdot \delta,$$

$\gamma \in \{X_1 \ldots X_{k-1}\}^{\otimes}$, $\delta \in \{X_{k+1} \ldots X_n\}^{\otimes}$, and $\gamma\, X_k^a$ is not k-unambiguous. Note that a is surely greater than 0. We define squeeze(α) by

$$\textit{squeeze}(\alpha) = \gamma \cdot X_k^{a-1} \cdot \delta_{k,\gamma} \cdot \delta.$$

Otherwise, i.e. when α is n-unambiguous, for convenience put squeeze$(\alpha) = \alpha$.

By Lemma 17 and by substitutivity of $\equiv$ we conclude that $\alpha \equiv \text{squeeze}(\alpha)$ and if α is not unambiguous then $\text{squeeze}(\alpha) \prec \alpha$.

3.2.5 Bounds on normal forms

For an arbitrary partial order $\trianglelefteq$ on processes, a process α is called $\trianglelefteq$-minimal if it is minimal with respect to $\trianglelefteq$ in its equivalence class (in other words, there is no $\beta \triangleleft \alpha$ with $\beta \equiv \alpha$). In the sequel in this section we will often refer to $\sqsubseteq$-minimal processes, and to $\preceq$-minimal ones. Clearly in every equivalence class there is exactly one $\preceq$-minimal process, the normal form of processes from that class. In the following sections we will use the notion of $\trianglelefteq$-minimality also for other orders than $\preceq$ and $\sqsubseteq$.

For a process α, by a $\trianglelefteq$-minimization of α we mean any $\trianglelefteq$-minimal process β with $\beta \equiv \alpha$ and $\beta \trianglelefteq \alpha$. In particular, if α is $\trianglelefteq$-minimal then it is its own minimization, in fact the unique one. Clearly, if the order is well-founded then every process has some $\trianglelefteq$-minimization. All orders considered in this chapter are refinements of the lexicographical order $\preceq$ and are thus well-founded.

Due to Lemma 17 we learn that every equivalence class contains an unambiguous processes:

Lemma 18. *A process α is unambiguous if and only if it is $\preceq$-minimal.*

Proof. One implication does not refer to squeezes. Suppose α is not the least process in its equivalence class. That is, for some $i \leqslant n$ we have $\alpha = \gamma \cdot X_i^a \cdot \bar{\alpha}$ and there is some $\beta = \gamma \cdot X_i^b \cdot \bar{\beta} \equiv \alpha$ with $b < a$. Thus, according to the definition, α is not unambiguous.

The other implication is easily provable building on the development of this section. If α is not unambiguous then it is not the least one in its equivalence class wrt. $\preceq$ as $\alpha \equiv \text{squeeze}(\alpha)$ and $\text{squeeze}(\alpha) \prec \alpha$.

$\square$

It follows that the squeezing step, applied in a systematic manner sufficiently many times on a process α, leads to the normal form process $\text{nf}(\alpha)$.

Lemma 19 (normal form via squeezing)**.** *Let α be an arbitrary process. Then consecutive applications of the squeezing step eventually stabilize at nf(α), i.e. for some $m \geqslant 0$, squeeze$^m(\alpha) =$ nf(α).*

Finally we formulate lower and upper bounds on the size of $\text{nf}(\alpha)$, with respect to the size of α, that will be crucial for the proof of Theorem 5. The first one, stated in Lemma 21, applies uniquely to $\sqsubseteq$-minimal processes. The following lemma is the technical preparation:

Lemma 20. *If α is $\sqsubseteq$-minimal then size$(\alpha) \leqslant$ size$(\bar{\alpha})$, for any $\sqsubseteq$-minimization $\bar{\alpha}$ of squeeze(α).*

Proof. If α is unambiguous the proof is trivial, therefore assume otherwise. According to Definition 30, let $\alpha = \gamma \cdot X_k^a \cdot \delta$ and let

$$\text{squeeze}(\alpha) = \gamma \cdot X_k^{a-1} \cdot (\delta_{k,\gamma} || \delta). \tag{3.11}$$

Consider any $\bar{\alpha} \sqsubseteq \text{squeeze}(\alpha)$ such that $\bar{\alpha} \equiv \text{squeeze}(\alpha)$. First we observe that γ is necessarily a $(k-1)$-prefix of $\bar{\alpha}$ as α is $(k-1)$-unambiguous and $\alpha \equiv \bar{\alpha}$. Therefore

$$\bar{\alpha} = \gamma \cdot X_k^{b-1} \cdot (\bar{\delta}_{k,\gamma} || \bar{\delta})$$

for some $b \leqslant a$ and $\bar{\delta}_{k,\gamma} \sqsubseteq \delta_{k,\gamma}$ and $\bar{\delta} \sqsubseteq \delta$. We observe that $\bar{\delta}_{k,\gamma}$ is necessarily non-empty, as α is $\sqsubseteq$-minimal and $\alpha \equiv \bar{\alpha}$. For size$(\alpha) \leqslant$ size$(\bar{\alpha})$ it is thus sufficient to demonstrate that

$$b = a \ \text{ and } \ \bar{\delta} = \delta.$$

Towards a contradiction assume the opposite, i.e. either $b < a$, or $\bar{\delta} \sqsubset \delta$. As $\alpha \equiv \bar{\alpha}$, i.e.,

$$\gamma \cdot X_k^a \cdot \delta \equiv \gamma \cdot X_k^{b-1} \cdot (\bar{\delta}_{k,\gamma} || \bar{\delta}),$$

knowing that $a > b - 1$ we deduce that the process $\gamma \cdot X^b$ may not be k-unambiguous. Thus we may apply squeeze(_) to $\gamma \cdot X^b \cdot \bar{\delta}$ to obtain

$$\text{squeeze}(\gamma \cdot X_k^b \cdot \bar{\delta}) = \gamma \cdot X_k^{b-1} \cdot (\delta_{k,\gamma} || \bar{\delta}).$$

By Lemma 4 applied to

$$\begin{aligned} &\text{squeeze}(\alpha) = \gamma \cdot X_k^{a-1} \cdot (\delta_{k,\gamma} || \delta) \Longrightarrow_0 \\ &\gamma \cdot X_k^{b-1} \cdot (\delta_{k,\gamma} || \bar{\delta}) \Longrightarrow_0 \gamma \cdot X_k^{b-1} \cdot (\bar{\delta}_{k,\gamma} || \bar{\delta}) = \bar{\alpha} \end{aligned}$$

we deduce squeeze$(\alpha) \equiv \gamma \cdot X_k^{b-1} \cdot (\delta_{k,\gamma} || \bar{\delta})$, i.e.,

$$\text{squeeze}(\alpha) \ \equiv \ \text{squeeze}(\gamma \cdot X_k^b \cdot \bar{\delta}).$$

Since always $\alpha \equiv \text{squeeze}(\alpha)$ we obtain

$$\alpha = \gamma \cdot X_k^a \cdot \delta \equiv \ \text{squeeze}(\alpha) \ \equiv \ \text{squeeze}(\gamma \cdot X_k^b \cdot \bar{\delta}) \ \equiv \ \gamma \cdot X_k^b \cdot \bar{\delta},$$

with either $b < a$ or $\bar{\delta} \sqsubset \delta$, thus contradicting the $\sqsubseteq$-minimality of α. This completes the proof. □

Lemma 21 (lower bound)**.** *If α is $\sqsubseteq$-minimal then size(nf(α)) $\geqslant$ size(α).*

Proof. Lemma is a corollary of Lemma 20, once one observes that the same normal form is obtained by consecutive applications of the following non-deterministic modification of the squeezing step:

- the minimization-squeezing step:

  replace α by any $\sqsubseteq$-minimization of squeeze(α).

Indeed, as minimization preserves the equivalence class, the unambiguous process obtained at the end, starting from a process α, is necessarily nf(α). $\Box$

Contrarily to Lemma 21, the upper bound holds for all processes.

Lemma 22 (upper bound)**.** *There is a constant c, depending only on the grammar, such that size(nf(α)) $\leqslant c \cdot$ size(α) for any process α.*

Proof. Let α be an arbitrary process. We claim that the size of nf(α) is bounded by:

$$\text{size}(\text{nf}(\alpha)) \leqslant \text{size}(\alpha) \cdot \text{size}(\delta_{k_1,\gamma_1}) \cdot \ldots \cdot \text{size}(\delta_{k_n,\gamma_n}) \tag{3.12}$$

for some unambiguous processes $\gamma_1 \ldots \gamma_n$. Indeed, let γ_k be the $(k-1)$-unambiguous process witnessing the squeezing step for X_k (if any). The size of the process, during all squeezing steps for X_k, increases at most size(δ_{k_k,γ_k}) times.

However, in general, there may be infinitely many different processes $\delta_{k,\gamma}$ used in the squeezing steps for different processes α, as there may be in general infinitely many unambiguous processes γ. We will argue that for the purpose of estimating the size of nf(α) for all processes α, it is sufficient to take into account only a finite subset of unambiguous processes. We will rely on the following simple observation. Let $\gamma, \gamma' \in \{X_1 \ldots X_{k-1}\}^{\otimes}$, for some $k \leqslant n$, be both $(k-1)$-unambiguous and $\gamma \sqsubseteq \gamma'$, respectively. Let the greatest k-extensions of γ and γ' be $\gamma \cdot X_k^a$ and $\gamma \cdot X_k^{a'}$. The exponents necessarily satisfy $a \geqslant a'$. The crucial observation is that whenever $a = a'$ then every γ-squeeze, like $\delta_{k,\gamma}$, is also a γ'-squeeze. Indeed:

$$\gamma \cdot X_k^{a+1} \cdot \delta \;\equiv\; \gamma \cdot X_k^a \cdot \delta_{k,\gamma} \cdot \delta \;\text{ implies }\; \gamma' \cdot X_k^{a+1} \cdot \delta \;\equiv\; \gamma' \cdot X_k^a \cdot \delta_{k,\gamma} \cdot \delta,$$

since $\equiv$ is substitutive. In other words: one may safely assume $\delta_{k,\gamma'} = \delta_{k,\gamma}$ whenever $\gamma \sqsubseteq \gamma'$ and $a \leqslant a'$.

Now we easily obtain the estimation. For every $k \in \{1 \ldots n\}$, consider all pairs (γ, a), where $\gamma \in \{X_1 \ldots X_{k-1}\}^{\otimes}$ is any $(k-1)$-unambiguous process that exhibits the greatest extension γX_k^a (note that only such processes γ witness a squeezing step). Choose those among them that are minimal wrt. $\sqsubseteq$ on the first coordinate, and wrt. $\leqslant$ on the second one. By Dickson's Lemma there are only finitely many such minimal pairs. The set of all processes $\delta_{k,\gamma}$, for all chosen minimal pairs (γ, a), jointly for all k, has an element which is maximal wrt. size; denote this maximal size by s. The size of any process δ_{k_i,γ_i} in (3.12) is dominated by s and thus we obtain:

$$\text{size}(\text{nf}(\alpha)) \leqslant \text{size}(\alpha) \cdot s^n \tag{3.13}$$

which completes the proof by putting $c = s^n$. Keep in mind that c only depends on a grammar, and does not depend on a process α. □

Concerning the upper bound, in the following section we demonstrate a sharper result, with the constant c estimated effectively. However, the estimation will be only shown for branching equivalence.

3.3 Effective bound on normal forms

In this section we only consider branching equivalence $\simeq$. In particular, the notion of normal form is understood with respect to $\simeq$. Fix an arbitrary normed commutative context-free grammar and denote its size by d.

Contrarily to the previous section, where the linear order $>$ on variables was fixed, in this section we consider all linear orders on variables that extend $>_{\text{decr}}$ (cf. Definition 24); such orders we call briefly *admissible*. Note however that the whole development of Section 3.2 strongly depends on the choice of $>$. In particular, the normal form of a process may change if one changes the order. Thus in this section we will have to be careful enough to explicitly specify the order we use, whenever we apply any notation or result of Section 3.2.

Concerning the notation, we will use indexed variable names $X_1, X_2, \ldots, X_n$ as in Section 3.2, assuming that the indexing is consistent with a currently used admissible order:

$$X_1 > X_2 > \ldots > X_n.$$

The following lemma is the main result of this section. The lemma will be used in Section 3.4 for the proof of Theorem 5.

Lemma 23 (upper bound)**.** *For every admissible order $>$, and for every process α, size(nf(α)) $\leqslant d^{n-1} \cdot$ size(α).*

Lemma 23 is a direct corollary of Lemma 24 which says that whatever admissible order is chosen, squeezing does not increase a weighted measure of size, defined as:

$$d\text{-size}(X_1^{a_1} \ldots X_n^{a_n}) = a_1 \cdot d^{n-1} + a_2 \cdot d^{n-2} + \ldots + a_{n-1} \cdot d + a_n.$$

(The measure of size clearly depends on the choice of $>$.)

Lemma 24. *For every $k \in \{1 \ldots n\}$, for every admissible order $>$ and $(k-1)$-unambiguous $\gamma \in \{X_1 \ldots X_{k-1}\}^{\otimes}$ that has the greatest k-extension, the variable X_k has a*

γ-squeeze δ with d-size$(\delta) \leqslant d$-size(X_k).

(Keep that even the variable X_k, as well as the property of being $(k-1)$-unambiguous, depend on the choice of $>$.) Indeed, whatever an admissible order $>$ is chosen, Lemma 24 together with Lemma 19 imply d-size(nf(α)) $\leqslant$ d-size(α) and then Lemma 23 follows:

$$\text{size}(\text{nf}(\alpha)) \leqslant d\text{-size}(\text{nf}(\alpha)) \leqslant d\text{-size}(\alpha) \leqslant d^{n-1} \cdot \text{size}(\alpha).$$

All the rest of this section is devoted to the proof of Lemma 24.

3.3.1 Proof of Lemma 24

The proof is by induction on $n-k$. The induction basis is for $k=n$. Whatever an admissible order is chosen, if $k=n$ then it trivially holds that d-size$(\delta) \leqslant d$-size(X_k), as the only possible γ-squeeze δ of X_n is the empty process, whose weighted size is 0.

For the induction step, fix some k and an admissible order $>$, assuming that the lemma holds for all greater values of k, for all admissible orders.

Then fix a $(k-1)$-unambiguous process $\gamma \in \{X_1 \ldots X_{k-1}\}^{\otimes}$, assuming that γ has the greatest k-extension, say $\gamma \cdot X_k^a$. The assumption guarantees existence of some γ-squeeze of X_k, that is a process δ satisfying

$$\gamma \cdot X_k^{a+1} \simeq \gamma \cdot X_k^a \cdot \delta. \tag{3.14}$$

The proof is split into three cases:

- $a > 0$,
- $a = 0$ and X_k has a γ-squeeze δ such that $X_k \Longrightarrow_0 \delta$,
- $a = 0$ and X_k has no γ-squeeze δ such that $X_k \Longrightarrow_0 \delta$.

Case 1: $a > 0$ In this case we will not refer to the induction assumption at all.

The idea behind that proof is based on the fact that $a > 0$, and thus, roughly speaking, X_k does not vanish during squeezing. From this we deduce that variables generated by X_k do not appear in some squeeze δ of X_k. Bounding the number of occurrences of other variables in δ is an easy conclusion from Claim 1, formulated below.

Claim 1. *The variable X_k has a γ-squeeze η such that $\gamma \cdot X_k^{a+1} \Longrightarrow_0 \gamma \cdot X_k^a \cdot \eta$.*

Proof. Choose an arbitrary γ-squeeze of X_k, say δ. Consider the pair (3.14) and an arbitrary non-generating decreasing transition rule $X_k \xrightarrow{\zeta} \omega$ (due to normedness assumption every variable has such a transition rule). The transition rule gives rise to a Spoiler's move

$$\gamma \cdot X_k^a \cdot \delta \xrightarrow{\zeta} \gamma \cdot X_k^{a-1} \cdot (\delta || \omega),$$

matched by some sequence of transitions of the form

$$\gamma \cdot X_k^{a+1} \Longrightarrow_0 \alpha \xrightarrow{\zeta} \alpha'.$$

We claim that

$$\alpha = \gamma \cdot X_k^a \cdot \eta \qquad \text{and} \qquad \alpha' = \gamma \cdot X_k^{a-1} \cdot \eta',$$

for some processes η and η'. The claim follows due to the equivalences

$$\alpha \simeq \gamma \cdot X_k^a \cdot \delta \qquad \text{and} \qquad \alpha' \simeq \gamma \cdot X_k^{a-1} \cdot (\delta || \omega),$$

using the fact that $\gamma \cdot X_k^a$ is k-unambiguous. Thus η is a γ-squeeze of X_k:

$$\gamma \cdot X_k^a \cdot \eta \simeq \gamma \cdot X_k^a \cdot \delta \simeq \gamma \cdot X_k^{a+1},$$

and $\gamma \cdot X_k^{a+1} \Longrightarrow_0 \gamma \cdot X_k^a \cdot \eta$ as required. □

Remark 2. *Actually it follows easily that $X_k \Longrightarrow_0 \eta$. We will however not need this property in the remaining part of the proof.*

Consider the sequence of transitions $\gamma \cdot X_k^{a+1} \Longrightarrow_0 \gamma \cdot X_k^a \cdot \eta$ and assume that all transitions originating from γ precede all transitions originating from X_k^{a+1}. Distinguish the very first transition of X_k, say $X_k \longrightarrow_0 \phi$, that decreases the exponent from $a+1$ to a:

$$\gamma \cdot X_k^{a+1} \Longrightarrow_0 \gamma \cdot X_k^{a+1} \cdot \theta \longrightarrow_0 \gamma \cdot X_k^a \cdot (\phi || \theta) \Longrightarrow_0 \gamma \cdot X_k^a \cdot \eta. \tag{3.15}$$

Note that by Lemma 4 we have:

$$\gamma \cdot X_k^{a+1} \simeq \gamma \cdot X_k^a \cdot (\theta || \phi). \tag{3.16}$$

Furthermore, as $\gamma \cdot X_k^{a+1}$ generates θ, namely $\gamma \cdot X_k^{a+1} \Longrightarrow_0 \gamma \cdot X_k^{a+1} \cdot \theta$, and $a > 0$, we observe that $\gamma \cdot X_k^a$ generates θ as well, and hence

$$\gamma \cdot X_k^a \simeq \gamma \cdot X_k^a \cdot \theta.$$

This allows us to obtain, using substitutivity and the equation (3.16), a γ-squeeze of X_k of size at most d:

$$\gamma \cdot X_k^{a+1} \simeq \gamma \cdot X_k^a \cdot (\theta || \phi) \;\simeq\; \gamma \cdot X_k^a \cdot \phi.$$

Knowing that $\phi \in \{X_{k+1} \dots X_n\}^{\otimes}$ and $\text{size}(\phi) < d$, we easily deduce the required bound on the weighted size of ϕ:

$$d\text{-size}(\phi) \leqslant d \cdot d^{n-k-1} = d^{n-k} = d\text{-size}(X_k).$$

The proof of Case 1 is thus completed.

Case 2.1: $a = 0$ and X_k has a γ-squeeze δ such that $X_k \Longrightarrow_0 \delta$ This is the only case that we are not able to adapt to weak equivalence.

Recall that we have a fixed admissible order $>$, for which we should provide an estimation on the size of a γ-squeeze δ of X_k. The idea of the solution for this case is to do a proof for an admissible order $>'$ different than $>$, knowing that both orders are *k-consistent*, which means that they agree on k greatest elements. For instance, the two orders on the set $\{A, B, C, D, E\}$:

$$A > B > C > D > E \qquad \text{and} \qquad A >' B >' D >' E >' C$$

are 2-consistent but not 3-consistent. The estimation on the size of a γ-squeeze δ will transfer easily to the original order $>$, as we will actually prove that $\text{size}(\delta) \leqslant d$. Our proof will base on the simple observation that if two orders are k-consistent then for $l \leqslant k$, l-unambiguous prefixes with respect to both orders are the same, as well as squeezes of X_l.

The modified order, denoted $>'$, is any one that satisfies the following conditions:

(1) $>'$ is k-consistent with $>$, and

(2) all variables generated by X_k are smaller with respect to $>'$ than all variables not generated by X_k.

Look at that, the second condition is satisfiable: whenever Y is generated by X_k and Z is not, then there exists no sequence of decreasing transitions from Y to Z, thus $Y >_{\text{decr}} Z$ is impossible. From now on we work with the order $>'$, so indexing of variables X_i, squeezes, normal forms, etc. are implicitly understood to be defined with respect to that order.

To make the notation more readable, we will constantly use symbols α, α', etc. for processes containing exclusively variables smaller than X_k that are not generated by X_k, and symbols β, β', etc. for those containing exclusively variables generated by X_k.

Let δ be a γ-squeeze of X_k such that $X_k \Longrightarrow_0 \delta$. In the sequence of transitions $X_k \Longrightarrow_0 \delta$, distinguish the transition that makes X_k disappear, induced by a transition rule $X_k \longrightarrow_0 \omega$, say. We may thus write:

$$\gamma \cdot X_k \Longrightarrow_0 \gamma \cdot X_k \cdot \hat{\beta} \longrightarrow_0 \gamma \cdot (\omega || \hat{\beta}) \Longrightarrow_0 \gamma \cdot \delta. \tag{3.17}$$

for some process $\hat{\beta}$. Let $\mathrm{nf}(\gamma \cdot X_k) = \gamma \cdot \alpha \cdot \beta$. As the first step we prove the following:

Lemma 25. *nf*$(\gamma \cdot \omega) = \gamma \cdot \alpha \cdot \bar{\beta}$ *for some process* $\bar{\beta}$.

Proof. As the first step we compute $\mathrm{nf}(\gamma \cdot (\omega || \hat{\beta}))$. Knowing that δ is a γ-squeeze of X_k, we may apply Lemma 4 to (3.17) to obtain

$$\gamma \cdot X_k \simeq \gamma \cdot (\omega || \hat{\beta})$$

and thus

$$\mathrm{nf}(\gamma \cdot (\omega || \hat{\beta})) = \mathrm{nf}(\gamma \cdot X_k) = \gamma \cdot \alpha \cdot \beta.$$

Now we claim that normal forms of the processes

$$\gamma \cdot \omega \qquad \text{and} \qquad \gamma \cdot (\omega || \hat{\beta}) \tag{3.18}$$

differ only on variables generated by X_k, which immediately proves the lemma. Indeed the processes themselves (3.18) differ only on variables generated by X_k (i.e. variables appearing in $\hat{\beta}$). Recall that the order on variables has been chosen so that the variables generated by X_k are smaller than other variables. As normal form is obtained by consecutive squeezing (cf. Lemma 19), the normal forms of processes (3.18) may only differ on variables generated by X_k, as required. □

Recall that $\mathrm{nf}(\gamma \cdot X_k) = \gamma \cdot \alpha \cdot \beta$ and consider branching bisimulation game from the pair of processes

$$\gamma \cdot X_k \simeq \gamma \cdot \alpha \cdot \beta.$$

Suppose the first Spoiler's move is $\gamma \cdot X_k \longrightarrow_0 \gamma \cdot \omega$, answered by a matching Duplicator's move:

$$\gamma \cdot \alpha \cdot \beta \Longrightarrow_0 \tau' \longrightarrow_0 \tau.$$

The sequence of transitions satisfies assumptions of Lemma 14: $\gamma \cdot \alpha$ is unambiguous and

$$\tau \simeq \gamma \cdot \omega \simeq \gamma \cdot \alpha \cdot \bar{\beta}$$

for some $\bar{\beta}$ (the latter equivalence follows by Lemma 25). We apply Lemma 14 together with Lemma 15 and deduce that Duplicator has a matching move engaging only variables generated by X_k, thus of the form:

$$\gamma \cdot \alpha \cdot \beta \Longrightarrow_0 \gamma \cdot \alpha \cdot (\beta' || Y) \longrightarrow \gamma \cdot \alpha \cdot (\beta' || \theta). \tag{3.19}$$

Finally we use (3.19) to provide a γ-squeeze of X_k of size at most d. Recall that (3.19) is a matching Duplicator's move, i.e.

$$\gamma \cdot \alpha \cdot (\beta' || Y) \simeq \gamma \cdot X_k \qquad \gamma \cdot \alpha \cdot (\beta' || \theta) \simeq \gamma \cdot \omega. \tag{3.20}$$

Using these two equivalences we derive the following sequence of equivalences:

$$\gamma \cdot X_k \simeq \gamma \cdot X_k \cdot \theta \simeq \gamma \cdot \alpha \cdot (\beta' || Y || \theta) \simeq \gamma \cdot \alpha \cdot (\beta' || \theta || Y) \simeq \gamma \cdot (\omega || Y).$$

The first one is due to the fact that θ is generated by X_k. The second one is by substitutivity, using the first equivalence in (3.20). The third one is simply commutativity of composition, and the last one is by substitutivity again, this time using the second equivalence in (3.20).

The process ω, being the right-hand side of a transition rule, is of size smaller than d. Hence $\omega || Y$ is a γ-squeeze of size at most d. This completes the proof of Case 2.1.

Case 2.2: $a = 0$ **and** X_k **has no** γ**-squeeze** δ **such that** $X_k \Longrightarrow_0 \delta$ We start with a lemma that only holds under assumptions of Case 2.2:

Lemma 26. *No $\sqsubseteq$-minimal γ-squeeze of X_k contains a variable generated by X_k.*

Proof. Suppose the contrary, namely

$$\gamma \cdot X_k \simeq \gamma \cdot (\delta' || Y), \tag{3.21}$$

with $\delta' || Y \in \{X_{k+1} \dots X_n\}^{\otimes}$, Y generated by X_k, and $\delta' || Y$ being $\sqsubseteq$-minimal. Consider branching bisimulation game for the pair (3.21), and suppose Spoiler performs an arbitrary sequence of silent decreasing transitions $Y \Longrightarrow_0 \varepsilon$ from Y to the empty process ε, giving rise to the sequence of Spoiler's moves

$$\gamma \cdot (\delta' || Y) \Longrightarrow_0 \gamma \cdot \delta' \tag{3.22}$$

from the right process. Observe that due to $\sqsubseteq$-minimality of $\delta' || Y$, the Spoiler's transitions surely change the equivalence class, i.e.

$$\gamma \cdot (\delta' || Y) \not\simeq \gamma \cdot \delta' \tag{3.23}$$

as otherwise $\delta' \sqsubset \delta' || Y$ would be a γ-squeeze of X_k smaller than $\delta' || Y$.

By Lemma 16 we know that there is a sequence of matching Duplicator's responses to (3.22) that does not engage γ at all:

$$\gamma \cdot X_k \Longrightarrow_0 \gamma \cdot \omega \tag{3.24}$$

(i.e. $X_k \Longrightarrow_0 \omega$). Knowing $\gamma \cdot \omega \simeq \gamma \cdot \delta'$, by (3.23) we deduce

$$\gamma \cdot X_k \not\simeq \gamma \cdot \omega. \tag{3.25}$$

Thus X_k does not appear in ω, as otherwise (3.24) and (3.25) would be in contradiction with Lemma 13.

As $\gamma \cdot \omega \simeq \gamma \cdot \delta'$, we may substitute $\gamma \cdot \omega$ in place of $\gamma \cdot \delta'$ in (3.21), to obtain a γ-squeeze of X_k

$$\gamma \cdot X_k \simeq \gamma \cdot (\omega || Y),$$

such that $X_k \longrightarrow_0 X_k \cdot Y \Longrightarrow_0 \omega || Y$. This is in contradiction with the assumption that no γ-squeeze is reachable from X_k by $\Longrightarrow_0$. Thus the lemma is proved. □

Recall that we have a fixed admissible order $>$, for which we should provide an estimation on the weighted size of a γ-squeeze δ of X_k. As in Case 2.1, we will use in the proof an admissible order $>'$ different than $>$, namely an arbitrary admissible order $>'$ fulfilling the following conditions:

(1) $>'$ is k-consistent with $>$,

(2) all variables generated by X_k are smaller with respect to $>'$ than all variables not generated by X_k, and

(3) the orders $>$ and $>'$ coincide on variables not generated by X_k.

The first two conditions are exactly as before, the last one is added. Similarly as before, the conditions are satisfiable. Moreover, the estimation on the weighted size of a γ-squeeze with respect to $>'$ easily transfers to the original order $>$. Indeed, by Lemma 26, a $\sqsubseteq$-minimal squeeze contains only variables not generated by X_k, and these variables are placed higher in the order $>'$ than in the order $>$, thus their contribution to the weighted size with respect to the order $>'$ is not smaller than with respect to $>$.

From now on we work with the order $>'$ instead of $>$, and thus indexing of variables, squeezes, normal forms, etc. are implicitly understood to be defined with respect to that order.

Let $\text{nf}(\gamma \cdot X_k) = \gamma \cdot \alpha$. By Lemma 26 we know that no variable appearing in α is generated by X_k. We are aiming at showing that the $d\text{-size}(\alpha) \leqslant d\text{-size}(X_k)$.

Case 2.2 is the only one which requires referring to the induction assumption. We will invoke the induction assumption for variables smaller than X_k, and the same admissible order $>'$ on variables. To this aim, we will start by considering branching bisimulation game starting with a decreasing non-generating Spoiler's move, as outlined below.

Let X_m be the the smallest variable occurring in α, i.e. $\alpha = \alpha'||X_m$ (note that X_m may occur in α'). Consider the branching bisimulation game for

$$\gamma \cdot X_k \simeq \gamma \cdot \alpha$$

and the first Spoiler's move from the right process induced by some decreasing non-generating transition rule of X_m, say $X_m \xrightarrow{\zeta} \omega$:

$$\gamma \cdot (\alpha'||X_m) \xrightarrow{\zeta} \gamma \cdot \alpha' \cdot \omega.$$

By Lemma 16 we know that there is a matching Duplicator's response that does not engage γ. As no γ-squeeze of X_k is reachable from X_k by $\Longrightarrow_0$, the response has necessarily the following form

$$\gamma \cdot X_k \Longrightarrow_0 \gamma \cdot X_k \cdot \eta \xrightarrow{\zeta} \gamma \cdot (\sigma||\eta),$$

where η is generated by X_k and X_k disappears in the last transition:

$$X_k \Longrightarrow_0 X_k \cdot \eta \quad \text{and} \quad X_k \xrightarrow{\zeta} \sigma.$$

Indeed, otherwise the second last process in the sequence forming a matching Duplicator's move would be a γ-squeeze of X_k, forbidden by the assumption of Case 2.2. We have $\gamma \cdot (\sigma||\eta) \simeq \gamma \cdot \alpha' \cdot \omega$ and thus also

$$\text{nf}(\gamma \cdot (\sigma||\eta)) \ = \ \text{nf}(\gamma \cdot \alpha' \cdot \omega). \tag{3.26}$$

Now we are going to deduce from equality (3.26) how the weighted sizes of nf$(\gamma \cdot \sigma)$ and nf$(\gamma \cdot \alpha')$ are related, in order to conclude that the weighted size of α is as required.

Let's inspect the m-prefix of the left processes in (3.26). Process η can not contribute to the m-prefix of the normal form. Indeed, η contains only variables generated by X_k, which are necessarily smaller than X_m, as X_m is not generated by X_k, since it appears in nf$(\gamma \cdot X_k)$. Thus if we restrict to the m-prefixes we have the equality

$$m\text{-prefix}(\text{nf}(\gamma \cdot (\sigma||\eta))) \ = \ m\text{-prefix}(\text{nf}(\gamma \cdot \sigma)). \tag{3.27}$$

Similarly, let's inspect the m-prefix of the right process in (3.26). Again, ω can not contribute to the m-prefix of the normal form, thus if we restrict to the m-prefixes we have the equality

$$m\text{-prefix}(\text{nf}(\gamma \cdot \alpha' \cdot \omega)) \ = \ m\text{-prefix}(\text{nf}(\gamma \cdot \alpha')).$$

As $\gamma \cdot \alpha$ is the normal form, $\gamma \cdot \alpha'$ is unambiguous and thus we have nf$(\gamma \cdot \alpha') = \gamma \cdot \alpha'$.

From this we derive that

$$m\text{-prefix}(\text{nf}(\gamma \cdot \alpha' \cdot \omega)) = m\text{-prefix}(\text{nf}(\gamma \cdot \alpha')) = m\text{-prefix}(\gamma \cdot \alpha') = \gamma \cdot \alpha' \quad (3.28)$$

Combine the equalities (3.27), (3.28) and (3.26) to obtain:

$$\gamma \cdot \alpha' \ = \ m\text{-prefix}(\text{nf}(\gamma \cdot \sigma)) \quad (3.29)$$

and to observe that

$$d\text{-size}(\gamma \cdot \alpha') \ = \ d\text{-size}(m\text{-prefix}(\text{nf}(\gamma \cdot \sigma))) \ \leqslant \ d\text{-size}(\text{nf}(\gamma \cdot \sigma)). \quad (3.30)$$

Now we will use induction assumption for the variables smaller than X_k, to derive that every variable X_i smaller than X_k (i.e. for $i > k$) has a γ-squeeze of weighted size smaller than X_i. As normal form is obtained via a sequence of squeezes, and γ is unambiguous, the induction assumption implies that

$$d\text{-size}(\text{nf}(\gamma \cdot \sigma)) \ \leqslant \ d\text{-size}(\gamma \cdot \sigma)$$

which leads to the following estimation:

$$d\text{-size}(\text{nf}(\gamma \cdot \sigma)) \ \leqslant \ d\text{-size}(\gamma) + d\text{-size}(\sigma) \ \leqslant \ d\text{-size}(\gamma) + \text{size}(\sigma) \cdot d^{n-k-1} \quad (3.31)$$

The inequalities (3.30) and (3.31) jointly imply:

$$d\text{-size}(\gamma \cdot \alpha') \ \leqslant \ d\text{-size}(\gamma) + \text{size}(\sigma) \cdot d^{n-k-1},$$

and removing γ from both sides of the inequality, we get:

$$d\text{-size}(\alpha') \leqslant \ \text{size}(\sigma) \cdot d^{n-k-1} \ \leqslant \ (d-1) \cdot d^{n-k-1}.$$

Recalling that $\alpha = \alpha' \cdot X_m$:

$$\begin{aligned} d\text{-size}(\alpha) = d\text{-size}(\alpha' \cdot X_m) \ &\leqslant \ \text{size}(\sigma) \cdot d^{n-k-1} \ + \ d^{n-m} \\ &\leqslant \ (d-1) \cdot d^{n-k-1} \ + \ d^{n-k-1} \ = \ d^{n-k} \ = \ d\text{-size}(X_k) \end{aligned}$$

which is the required bound. As Case 2.2 is the last one, we have thus completed the proof of Lemma 24.

Remark 3. *Lemma 24 is formulated for $\simeq$ but the major part of the proof either works for weak equivalence directly, or may be adapted. The only case that we can not adapt to weak equivalence is Case 2.1. Importantly, under the pure-generators restriction the proof of this subcase is straightforward.*

Consider a γ-squeeze $\gamma \cdot X \approx \gamma \cdot \delta$ such that $X \Longrightarrow_0 \delta$. We can easily produce a new squeeze with size bounded by d. Indeed, the variable X may not be a generator, and thus must vanish in the very first transition of the sequence $X \Longrightarrow_0 \delta$:

$$\gamma \cdot X \longrightarrow_0 \gamma \cdot \omega \Longrightarrow_0 \gamma \cdot \delta,$$

due to a non-generating transition rule $X \longrightarrow \omega$. By Lemma 4 we deduce $\gamma \cdot X \approx \gamma \cdot \omega$, which means that already ω is a γ-squeeze of X of size at most d.

We claim that our proof, after slight adaptations in Cases 1 and 2.2, shows decidability of weak equivalence in the pure-generators subclass.

3.4 Proof of the bounded response property

This section contains finally the proofs of two main results announced in Section 3.1, namely Theorem 5 and Theorem 7. The two theorems state the bounded response property for branching and weak equivalences, respectively; moreover the former one claims a response of an effectively bounded size.

3.4.1 Proof of Theorem 7

In case of weak equivalence $\approx$, the bounded response property follows easily from the estimations given in Lemmas 21 and 22. Fix a normed commutative context-free grammar and an admissible order on variables. Consider $\alpha \approx \beta$, a Spoiler's move $\alpha \xrightarrow{\zeta} \alpha'$ and a matching Duplicator's response:

$$\beta \Longrightarrow_0 \xrightarrow{\zeta} \Longrightarrow_0 \beta',$$

with $\alpha' \approx \beta'$. Extend the Duplicator's response with

$$\beta \Longrightarrow_0 \xrightarrow{\zeta} \Longrightarrow_0 \beta' \Longrightarrow_0 \bar{\beta}',$$

for an arbitrary $\sqsubseteq$-minimal $\bar{\beta}'$. Then using Lemma 21, Lemma 22, and the equality $\mathrm{nf}(\alpha') = \mathrm{nf}(\bar{\beta}')$, we obtained the required bound:

$$\mathrm{size}(\bar{\beta}') \leqslant \mathrm{size}(\mathrm{nf}(\bar{\beta}')) = \mathrm{size}(\mathrm{nf}(\alpha')) \leqslant c \cdot \mathrm{size}(\alpha'),$$

for a constant c from Lemma 22.

3.4.2 Proof of Theorem 5

From now on we focus on branching equivalence $\simeq$ only. Compared to weak equivalence, the case of branching equivalence is slightly more subtle. As previously, consider a fixed grammar and a fixed admissible order on variables.

Before showing how Theorem 5 follows from Lemmas 21 and 23, we will need a definition and a lemma. Define a partial order $\trianglelefteq$ as a refinement of the lexicographical order $\preceq$:

$$\alpha \trianglerighteq \beta \text{ iff } \alpha \succeq \beta \text{ and } \alpha \Longrightarrow_0 \beta.$$

In the sequel we will consider $\trianglelefteq$-minimal processes (cf. definition of $\trianglelefteq$-minimality and $\trianglelefteq$-minimization in Section 3.2). Due to the following apparent inclusions of partial orders:

$$\sqsubseteq \quad \subseteq \quad \trianglelefteq \quad \subseteq \quad \preceq$$

we have the following dependencies between minimal processes:

$$\preceq\text{-minimal} \quad \Longrightarrow \quad \trianglelefteq\text{-minimal} \quad \Longrightarrow \quad \sqsubseteq\text{-minimal}. \tag{3.32}$$

Lemma 27. *If α is $\trianglelefteq$-minimal and $\alpha \Longrightarrow_0 \beta \simeq \alpha$ then $\alpha \sqsubseteq \beta$.*

Proof. We will show that

$$\beta = \alpha || \delta \quad \text{for some } \delta \text{ generated by } \alpha. \tag{3.33}$$

For the sake of contradiction assume the contrary and consider the shortest sequence of transitions $\alpha \Longrightarrow_0 \beta$ such that $\beta \simeq \alpha$ and β fails to satisfy (3.33). Consider the last transition, say

$$\alpha || \delta \longrightarrow_0 \beta,$$

performed necessarily by a variable, say X, that appears in α but not in δ. This last transition has the following form

$$\alpha || \delta \longrightarrow_0 \alpha' || \delta,$$

due to a transition $\alpha \longrightarrow_0 \alpha'$. As the latter transition is necessarily decreasing and non-generating, $\alpha' \prec \alpha$. Recall that $\alpha \simeq \alpha' || \delta$ and $\alpha \preceq \alpha' || \delta$. By Lemma 13 we know that those variables in δ that are generated by a variable different than X may be safely cancelled via a sequence of silent transitions $\Longrightarrow_0$ while preserving the equivalence class. Hence

$$\alpha \Longrightarrow_0 \alpha' \cdot \delta' \simeq \alpha, \tag{3.34}$$

where all variables appearing in $\delta' \sqsubseteq \delta$ are generated by X, and thus smaller than X wrt. $<$. Knowing that $\alpha' \prec \alpha$ we obtain

$$\alpha' \cdot \delta' \prec \alpha. \tag{3.35}$$

The facts (3.34) and (3.35) jointly contradict $\trianglelefteq$-minimality of α. $\square$

From now on, the remaining part of Section 3.4 is devoted to proving Theorem 5, using Lemmas 21 and 23 together with Lemma 27.

Consider $\alpha \simeq \beta$, a Spoiler's move $\alpha \xrightarrow{\zeta} \alpha'$ and a Duplicator's response:

$$\beta \Longrightarrow_0 \beta_1 \xrightarrow{\zeta} \beta_2, \tag{3.36}$$

with $\alpha \simeq \beta_1$ and $\alpha' \simeq \beta_2$. We will show that Duplicator has a matching response where β_1 and β_2 are of size bounded by $c \cdot \text{size}(\alpha)$ and $c \cdot \text{size}(\alpha')$, respectively, where $c = (d^{n-1} + d^n + d)$, n is the number of variables and d is the size of the grammar[2].

We can not simply extend this response analogously as for weak equivalence, and we have to estimate the size of the process β_2 resulting from the last transition. The basic idea of the proof is essentially to eliminate some unnecessary generation done by the transitions $\beta \Longrightarrow_0 \beta_1$, without affecting executability of the transition $\beta_1 \xrightarrow{\zeta} \beta_2$.

As the first step we observe that without loosing generality we could assume that the response (3.36) starts in a $\trianglelefteq$-minimal process. Indeed, if we prove our claim in this case, then we easily obtain a matching response, of required size bound, from an arbitrary β by adjoining at the beginning of a matching response from $\bar{\beta}$,

$$\bar{\beta} \Longrightarrow_0 \beta_1 \xrightarrow{\zeta} \beta_2, \tag{3.37}$$

a sequence of transitions $\beta \Longrightarrow_0 \bar{\beta}$, for some $\trianglelefteq$-minimization $\bar{\beta}$ of β, thus obtaining

$$\beta \Longrightarrow_0 \bar{\beta} \Longrightarrow_0 \beta_1 \xrightarrow{\zeta} \beta_2. \tag{3.38}$$

As $\beta \simeq \bar{\beta}$, we know that the response from β is really matching: $\alpha \simeq \beta_1$ and $\alpha' \simeq \beta_2$.

Thus from now on we consider a pair $\alpha \simeq \bar{\beta}$, with $\bar{\beta}$ a $\trianglelefteq$-minimal process, instead of arbitrary $\alpha \simeq \beta$, together with a matching Duplicator's response (3.37). Note that by Lemma 27 we know that $\bar{\beta} \sqsubseteq \beta_1$.

As the second step, we extend (3.38) with any sequence $\beta_2 \Longrightarrow_0 \bar{\beta}_2$ leading to a $\sqsubseteq$-minimal process $\bar{\beta}_2 \sqsubseteq \beta_2$ which is equivalent to β_2. Our knowledge by now may be outlined with the following diagram (the subscript in $\Longrightarrow_0$ is omitted):

$$\begin{tikzcd}
\bar{\beta} \arrow[rr, Rightarrow, "\simeq", "\sqsubseteq"'] & & \beta_1 \arrow[d, "\zeta"] \\
\bar{\beta}_2 & & \beta_2 \arrow[ll, Rightarrow, "\simeq"', "\sqsubseteq"]
\end{tikzcd}$$

Both left-most processes in the diagram are size bounded. Indeed, as both $\bar{\beta}$ and $\bar{\beta}_2$ are

[2]Note that α' is at most d times larger that α.

$\sqsubseteq$-minimal, Lemma 21 applies:

$$\text{size}(\bar{\beta}) \leqslant \text{size}(\text{nf}(\beta)) = \text{size}(\text{nf}(\alpha)) \text{ and } \text{size}(\bar{\beta}_2) \leqslant \text{size}(\text{nf}(\beta_2)) = \text{size}(\text{nf}(\alpha')).$$

Then applying Lemma 23 to α and α' we obtain:

$$\text{size}(\bar{\beta}) \leqslant \text{size}(\alpha) \cdot d^{n-1} \text{ and } \text{size}(\bar{\beta}_2) \leqslant \text{size}(\alpha') \cdot d^{n-1}. \tag{3.39}$$

As the third and the last step of the proof, we claim that β_1 and β_2 may be replaced by processes of size bounded, roughly, by the sum of sizes of $\bar{\beta}$ and $\bar{\beta}_2$.

Claim 2. *There are some processes $\beta_1' \simeq \beta_1$ and $\beta_2' \simeq \beta_2$ such that*

$$\bar{\beta} \Longrightarrow_0 \beta_1' \xrightarrow{\zeta} \beta_2' \tag{3.40}$$

and

$$\mathit{size}(\beta_1'), \mathit{size}(\beta_2') < \mathit{size}(\bar{\beta}) + \mathit{size}(\bar{\beta}_2) + d. \tag{3.41}$$

The claim is sufficient for Theorem 5 to hold, by inequalities (3.39). Thus to complete the proof we only need to demonstrate the claim. The idea underlying the proof of the claim is illustrated in the following diagram:

$$\begin{tikzcd}
\bar{\beta} \arrow[rr, Rightarrow, "\simeq", "\sqsubseteq"'] \arrow[dr, dotted, no head, "\sqsubseteq"'] & & \beta_1 \arrow[dl, Rightarrow, dashed, "\simeq"', "\sqsubseteq"] \arrow[d, "\zeta"] \\
& \beta_1' \arrow[d, "\zeta"'] & \beta_2 \arrow[dl, Rightarrow, "\simeq"', "\sqsubseteq"] \\
\bar{\beta}_2 & \beta_2' \arrow[l, Rightarrow, "\simeq"', "\sqsubseteq"] &
\end{tikzcd}$$

We will use now an intuitive colouring argument. Let us colour uniquely every variable occurrence in β_1 and let every transition preserve the colour of the left-hand side variable of a transition rule that is used. Obviously at most size($\bar{\beta}_2$) of these colours will still be present in $\bar{\beta}_2$, name them *surviving colours*. Suppose the $\beta_1 \xrightarrow{\zeta} \beta_2$ transition be induced by a transition rule $X \xrightarrow{\zeta} \delta$ and colour the occurrence of X in β_1 involved in this transition, say, brown.

Let $\beta_1' \sqsubseteq \beta_1$ contain sufficiently many variables occurrences from β_1 so that $\bar{\beta} \sqsubseteq \beta_1'$ and all occurrences coloured a surviving colour or brown are included. Clearly

$$\bar{\beta} \sqsubseteq \beta_1' \sqsubseteq \beta_1.$$

One easily observes that after firing the brown transition $X \xrightarrow{\zeta} \delta$ from β_1' we get a process β_2' such that

$$\bar{\beta}_2 \sqsubseteq \beta_2' \sqsubseteq \beta_2,$$

because all surviving coloured variables are still present. We now only need to check that all the requirements are satisfied by β_1' and β_2'.

By Lemma 4 we have $\beta_1' \simeq \beta_1$ and $\beta_2' \simeq \beta_2$. Clearly there is a sequence $\beta_1 \Longrightarrow_0 \beta_1'$, that simply cancels superfluous variable occurrences, hence the condition (3.40) is fullfilled.

Finally we obtain the size estimation $\text{size}(\beta_1') \leqslant \text{size}(\bar{\beta}) + \text{size}(\bar{\beta}_2) + 1$ as in β_1' there can be at most $\text{size}(\bar{\beta}_2) + 1$ surviving and brown coloured variables occurrences, except for those that come from $\bar{\beta}$. This easily implies the required size estimation for $\text{size}(\beta_2')$. Thus the required condition (3.41) is shown to hold.

Chapter 4

Simulation pre-order over one-counter automata

In this chapter we investigate weak and branching pre-orders over one-counter automata (recall that we restrict ourselves to automata without zero test). Whenever we write 'automaton' below we mean a one-counter automaton without zero tests.

Observe that we may safely assume in a weak simulation game that Spoiler's automaton[1] has no ε-transition rules (cf, Lemma 28). Indeed, the crucial fact is that every Spoiler's silent move can be matched by a Duplicator's silent self-loop.

The main result to be shown in this section is the following:

Theorem 8. *The weak pre-order is effectively semilinear, and thus decidable, for one-counter automata without zero tests.*

The proof of Theorem 8 is split into two main parts, treated in detail in Sections 4.1 and 4.2, respectively. The first part is elimination of silent transitions in Duplicator's automaton; in Section 4.1 (Theorem 10) we show how to replace Duplicator's automaton N by an ω-automaton[2] N'. The ω-automata have no ε-transition rules. On the other hand, they may have ω-transition rules that increase the counter arbitrarily . The second part of the proof of Theorem 8 focuses on proving decidability of simulation pre-order[3] between an automaton and an ω-automaton (Theorem 11 in Section 4.2).

The latter result bases on yet another non-standard hierarchy of approximants. Decidability comes from the very surprising result that the hierarchy of these approximants always stabilizes at ω. This is in contrast with the approximants induced by

[1]We can silently assume that there are two copies of the automaton, one which belongs to Spoiler and one for Duplicator.

[2]We use here a short name ω-automata, instead of the more appropriate but long expression 'one-counter automata with ω-transitions but without ε-transition rules'. These automata should not be confused with automata over ω-words.

[3]Recall that in systems without ε-transitions, weak and branching simulation coincide. In the literature a term *strong simulation pre-order* is often used.

standard weak simulation expansions, which do not stabilize at level ω, as we have shown in Section 2.3. As an immediate corollary we obtain:

Corollary 1. *The weak simulation equivalence $\preccurlyeq \cap \preccurlyeq^{-1}$ is effectively semilinear, and thus decidable, for one-counter automata without zero tests.*

After proving Theorem 8, in Section 4.3 we show that the hierarchy of approximants induced by weak simulation expansion stabilizes actually at level $\omega \cdot \omega$. Finally in Section 4.4 we sketch how, using similar methods, one can compute branching pre-order over one-counter automata. Thus we prove:

Theorem 9. *Branching pre-order $\preceq$ and branching simulation equivalence $\preceq \cap \preceq^{-1}$ are semilinear, and thus decidable, over one-counter automata without zero tests.*

4.1 Reduction to ω-automata

From now on we can assume wlog. that Spoiler's automaton has no ε-transitions rules, due to the following lemma:

Lemma 28. *For two given automata $M = (Q_M, \mathbb{A}ct, \delta_M)$ and $N = (Q_N, \mathbb{A}ct, \delta_N)$, one can compute an automaton $M' = (Q_M, \mathbb{A}ct \cup \{\mu\}, \delta'_M)$ without silent transitions rules, and an automaton $N' = (Q_N, \mathbb{A}ct \cup \{\mu\}, \delta'_N)$, such that for all $(p, m, q, n) \in Q_M \times \mathbb{N} \times \mathbb{Q}_N \times \mathbb{N}$,*

$$pm \preccurlyeq_\kappa qn \text{ wrt. } M, N \iff pm \preccurlyeq_\kappa qn \text{ wrt. } M', N'.$$

(Note that we explicitly specify which pair of automata is involved in the game. In the sequel we will continue using this notation whenever a risk of confusion appears.)

Proof. Automaton M' is obtained from M by renaming ε labels to μ. Automaton N' is obtained from N by adding in each state $q \in Q_N$ a μ-labeled self-loop $q \xrightarrow{\mu,0} q$.

Consider weak simulation approximants game from a position (κ, pm, qn) wrt. M, N, and the same game wrt. M', N'. If Duplicator wins in the game wrt. M', N' then his strategy may be used in the game wrt. M, N (implicit silent self-loops are used by Duplicator). On the other hand, if Duplicator wins the game wrt. M, N, then the winning strategy may be modified to always match a silent move of Spoiler with a silent self-loop, and thus postponing choices of Duplicator. The modified winning strategy may be used in the game wrt. M', N'. □

As a corollary we deduce:

$$pm \preccurlyeq qn \text{ wrt. } M, N \iff pm \preccurlyeq qn \text{ wrt. } M', N'.$$

The same fact holds for branching pre-order too.

Due to Lemma 28 we may get rid of silent transition rules in Spoiler's automaton. The aim of the remaining part of Section 4.1 is to get rid of silent transition rules in the Duplicator's automaton too. The price to pay will be the shift to a more general class of ω-automata, as defined below.

The ω-automata are like one-counter automata, with two differences. First, there exist dedicated transition rules with symbolic effect ω, which allow to arbitrarily increase the counter in a single transition. Second, we assume that ω-automata do not use ε-transition rules. As we prove in this section, checking weak pre-order between two automata can be reduced to checking the simulation pre-order between an automaton without silent transition rules, and an ω-automaton.

Definition 31. *An* ω-automaton $N = (Q, Act, \delta)$ *is given by a finite set of control-states* Q*, a finite set of actions* $\mathbb{A}ct$ *and transition rules* $\delta \subseteq Q \times \mathbb{A}ct \times \{-1, 0, 1, \omega\} \times Q$*. It induces an LTS, with processes* $Q \times \mathbb{N}$*, that allows a transition* $pm \xrightarrow{a} p'm'$ *if either* $(p, a, d, p') \in \delta$ *and* $m' = m + d \in \mathbb{N}$*, or* $(p, a, \omega, p') \in \delta$ *and* $m' > m$*.*

Every automaton without silent transition rules is clearly an ω-automaton. Unlike the former one, the latter one can induce an infinitely branching labelled transition system, since each ω-transition (p, a, ω, p') introduces transitions[4] $pm \xrightarrow{a} p'm'$ for any two naturals $m' > m$.

It is easily verified that both one-counter automata and ω-automata satisfy the following monotonicity properties.

Proposition 3 (Monotonicity of $\preccurlyeq$)**.** $pm \xrightarrow{\zeta} p'm'$ *implies* $p(m + d) \xrightarrow{\zeta} p'(m' + d)$ *for all* $d \in \mathbb{N}$*. Moreover,* $pm \preccurlyeq qn$ *implies* $pm' \preccurlyeq qn'$ *for* $m' \leqslant m$*,* $n' \geqslant n$*.*

The following theorem justifies our focus on weak simulation game, where Duplicator plays on processes of an ω-automaton.

Theorem 10. *Checking weak pre-order between two automata can be reduced to checking the simulation pre-order between an automaton without* ε*-transition rules and an* ω*-automaton. Formally, for two automata* M *and* N*, with states* Q_M *and* Q_N *respectively, one can effectively construct* M' *an automaton without* ε*-transition rules with states* $Q_{M'} \supseteq Q_M$*, and an* ω*-automaton* N' *with states* $Q_{N'} \supseteq Q_N$*, such that for each pair* $(p, q) \in Q_M \times Q_N$ *of original control states and any ordinal* κ *the following hold:*

(1) $pm \preccurlyeq qn$ *wrt.* M, N *iff* $pm \preccurlyeq qn$ *wrt.* M', N'*;*

(2) if $pm \preccurlyeq_\kappa qn$ *wrt.* M, N *then* $pm \preccurlyeq_\kappa qn$ *wrt.* M', N'*.*

The idea of the proof is to look for counter-increasing cyclic paths via ε-labelled transitions in the control graph and to introduce ω-transition rules accordingly. For any

[4]Transition in LTS induced by ω-transition rules are called ω-transitions.

path that reads a single visible action and visits a "generator" state that is a part of a silent cycle with positive effect, we add an ω-transition rule. For all of the finitely many non-cyclic paths that read a single visible action we introduce direct transition rules. The remaining part of Section 4.1 is devoted to the detailed presentation of the proof.

4.1.1 Proof of Theorem 10

The reduction will be done in two steps. First (Lemma 30) we reduce weak simulation for one-counter automata to simulation between a one-counter automaton and yet another auxiliary model, called *guarded ω-automaton*. The latter differs from ω-automaton in that each transition rule may change the counter by more than one and is guarded by an integer, i.e. can only be applied if the current counter value exceeds the *guard* attached to it. In the second step (Lemma 31) we normalize the effects of all transition rules to $\{-1, 0, 1, \omega\}$ and eliminate all integer guards and thereby construct an ordinary ω-automaton for Duplicator.

Before we start recall that wlog. we assume that every process s allows a silent loop $s \xrightarrow{\varepsilon} s$. Thus even if ε-loops are not defined explicitly in an automaton, Duplicator can use them during the game.

Definition 32. *A* path *in an automaton (resp. ω-automaton) $N = (Q, \mathbb{A}ct, \delta)$ is a sequence*

$$\pi = (s_0, \zeta_0, d_0, t_0)\ (s_1, \zeta_1, d_1, t_1)\ \dots\ (s_k, \zeta_k, d_k, t_k) \in \delta^*$$

of transition rules, where $s_{i+1} = t_i$ for all $i < k$. We call π cyclic *if $s_i = t_j$ for some $0 \leqslant i < j \leqslant k$ and write ${}^i\pi$ for its prefix of length i. A cyclic path is a* loop *if $s_i \neq s_j$ for all $0 \leqslant i < j < k$. Define the* effect $\Delta(\pi)$ *and* guard $\Gamma(\pi)$ *of a path π by*

$$\Delta(\pi) = \sum_{i=0}^{k} d_i \quad \textit{and} \qquad\qquad \Gamma(\pi) = -\min\{\Delta({}^i\pi) | i \leqslant k\}$$

where $n + \omega = \omega + n = \omega$ for every $n \in \mathbb{N}$. The guard $\Gamma(\pi)$ denotes the minimal counter value that is needed to traverse the path π while maintaining a non-negative counter value along all intermediate processes.
Lastly, fix a homomorphism obs : $\delta^* \to \mathbb{A}ct^*$, that maps paths to their observable action sequences*: $obs((s, \varepsilon, d, t)) = \varepsilon$ and $obs((s, a, d, t)) = a$ for $a \neq \varepsilon$.*

Definition 33 (Guarded ω-automata)**.** *A* guarded ω-automaton *$G = (Q, \mathbb{A}ct, \delta)$ is given by finite sets $Q, \mathbb{A}ct$ of states and actions and a transition relation $\delta \subseteq Q \times \mathbb{A}ct \times \mathbb{N} \times \mathbb{Z} \cup \{\omega\} \times Q$. It defines a transition system over the stateset $Q \times \mathbb{N}$ where $qn \xrightarrow{a} q'n'$ iff there is a transition rule $(q, a, g, d, q') \in \delta$ with*

(1) $n \geqslant g$ and

(2) $n' = n + d \in \mathbb{N}$ *or* $d = \omega$ *and* $n' > n$.

Specifically, G is a ω-*automaton* if for all transition rules $g = 0$ and $d \in \{-1, 0, 1, \omega\}$. The next construction establishes the connection between automata and ω-automata in the context of weak pre-order.

Lemma 29. *For an automaton $N = (Q, \mathbb{A}ct, \delta)$ we can effectively construct a guarded ω-automaton $G = (Q, \mathbb{A}ct, \delta_G)$ with the same state space Q, such that for all $a \in \mathbb{A}ct$,*

(1) whenever $qn \overset{a}{\Longrightarrow} q'n'$ *in* N*, there is a* $n'' \geqslant n$ *such that* $qn \overset{a}{\longrightarrow} q'n''$ *in* G*;*

(2) whenever $qn \overset{a}{\longrightarrow} q'n'$ *in* G*, there is a* $n'' \geqslant n$ *such that* $qn \overset{a}{\Longrightarrow} q'n''$ *in* N.

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

For two states s, t of N, let $D(s, t)$ be the set of *direct* paths from s to t:

$$D(s,t) = \{(q_i, a_i, d_i, q_{i+1})_{i<k} : q_0 = s, q_k = t,\\ \forall_{0 \leqslant i < j \leqslant k}\, q_i = q_j \implies (i = 0 \wedge j = k)\}.$$

Define the subset of *silent direct* paths by $SD(s, t) = \{\pi \in D(s, t) | obs(\pi) = \varepsilon\}$. 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 transition rules in G as follows. Let δ_G contain a transition rule $(q, a, \Gamma(\pi), \Delta(\pi), q')$ for each path $\pi = \pi_1(s, a, d, s')\pi_2 \in \delta^+$ where $\pi_1 \in SD(q, s)$ and $\pi_2 \in SD(s', q')$. This carries over all transition rules of N because the empty path is in $SD(s, s)$ for all states s. Moreover, introduce ω-transition rules in case 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) $\pi_1' \in SD(q, t)$, $\pi_1'' \in SD(t, t)$ and $\pi_1''' \in SD(t, s)$

(2) $\Delta(\pi_1'') > 0$

for some $t \in Q$, then δ_G contains a transition rule $(q, a, \Gamma(\pi_1'\pi_1''), \omega, q')$. 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) $\pi_1 \in SD(q, s)$, $\pi_2' \in SD(s', t)$, $\pi_2'' \in SD(t, t)$ and $\pi_2''' \in SD(t, q')$

(2) $\Delta(\pi_2'') > 0$

add a transition rule (q, 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 q to q' that contains a silent and direct cycle with positive effect, G has an a-labelled ω-transition rule from q to q' with the guard derived from that path.

To prove the first part of the lemma, assume $qn \overset{a}{\Longrightarrow}_N q'n'$. By definition of $\overset{a}{\Longrightarrow}_N$, there must be a path $\pi = \pi_1(s,a,d,s')\pi_2$ with $obs(\pi_1) = obs(\pi_2) = \varepsilon$. Suppose both π_1 and π_2 do not contain loops with positive effect. Then there must be paths $\pi_1' \in SD(q,s), \pi_2' \in SD(s',q')$ with $\Gamma(\pi_i') \leqslant \Gamma(\pi_i)$ and $\Delta(\pi_i') \geqslant \Delta(\pi_i)$ for $i \in \{1,2\}$ that can be obtained from π_1 and π_2 by removing all loops with effects less or equal 0. So G contains a transition rule (q,a,g',d',q') for some $g' \leqslant n$ and $d' \geqslant n'-n$ and hence $qn \xrightarrow{a}_G q'n''$ for $n'' = n + d' \geqslant n'$. Alternatively, either π_1 or π_2 contains a loop 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 counter-increasing loop: If π_1 contains a loop with positive effect, there is a path $\bar{\pi_1} = \pi_1'\pi_1''\pi_1'''$ from q to s, where π_1', π'' and π_1''' are direct and $\Delta(\pi_1'') > 0$ for the loop $\pi_1'' \in SD(t,t)$ for some state t. In this case, G contains a ω-transition rule (q,a,g,ω,q') with $g = \Gamma(\pi_1'\pi_1'')$. Similarly, if π_2 contains the counter-increasing loop, 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 rule (q,a,g,ω,q') in G with $g = \Gamma(\pi_1(s,a,d,s')\pi_2'\pi_2'')$. In both cases, $g \leqslant \Gamma(\pi) \leqslant n''$ and therefore $qn \xrightarrow{a}_G q'i$ for all $i \geqslant n$.

For the second part of the lemma, assume $qn \xrightarrow{a}_G q'n'$. This must be the result of a transition rule $(q,a,g,d,q') \in \delta_G$ for some $g \leqslant n$. In case $d \neq \omega$, there is a path $\pi \in \delta^*$ from q to q' with $\Delta(\pi) = n'-n$, $obs(\pi) = a$ and $\Gamma(\pi) = g$ that witnesses the move $qn \overset{a}{\Longrightarrow}_N q'n'$ in 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 q to q' in N where $\Gamma(\pi) \leqslant n$, 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 q to q' satisfies $\Gamma(\pi') \leqslant \Gamma(\pi) \leqslant n$ and $\Delta(\pi') \geqslant n'-n$. Now π' witnesses the sequence of transitions of a form $qn \overset{a}{\Longrightarrow}_N q'n''$ in N for an $n'' \geqslant n'$. □

Remark 4. *Observe that no transition rule of the automaton G as constructed above has a guard larger than $|Q| * 3 + 1$ and finite effect $> 2|Q| + 1$.*

Lemma 30. *For an automaton $N = (Q, \mathbb{A}ct, \delta)$ one can effectively construct a guarded ω-automaton $G = (Q, \mathbb{A}ct, \delta_G)$ s.t. for any automaton M without silent transition rules, and any two processes pm, qn of M and N, respectively,*

$$(1) \qquad pm \preccurlyeq qn \text{ wrt. } M,N \iff pm \preccurlyeq qn \text{ wrt. } M,G; \tag{4.1}$$

$$(2) \qquad pm \preccurlyeq_\kappa qn \text{ wrt. } M,N \iff pm \preccurlyeq_\kappa qn \text{ wrt. } M,G. \tag{4.2}$$

Proof. Consider the construction from the proof of Lemma 29. Let $\preccurlyeq_{M,N}$ denote weak pre-order wrt. M,N and $\preccurlyeq_{M,G}$ denote the simulation pre-order wrt. M,G.

For the "if" direction we show that $\preccurlyeq_{M,G}$ is a weak simulation wrt. M,N. Assume $pm \preccurlyeq_{M,G} qn$ and $pm \xrightarrow{a}_M p'm'$. That means there is a transition $qn \xrightarrow{a}_G q'n'$ for some $n' \in N$ so that $p'm' \preccurlyeq_G q'n'$. By Lemma 29 (2), $qn \overset{a}{\Longrightarrow}_N q'n''$ for a $n'' \geqslant n'$. Because simulation is monotonic we know that also $p'm' \preccurlyeq_{M,G} q'n''$. Similarly, for the "only if" direction, one can use the first claim of Lemma 29 to check that $\preccurlyeq_{M,N}$ is a (weak) simulation wrt. M, G.

Moreover one round of a game between M and N corresponds to one round of a game between M and G, and other way around. Thus every Spoiler's and Duplicator's strategies can be moved from one simulation game to another. Observe that the same holds for the approximant game as well. Thus the second claim holds. □

Lemma 31. *For an automaton M and a guarded ω-automaton G with states Q_M and Q_G respectively, one can effectively construct an automaton M' and an ω-automaton N' with states $Q_{M'} \supseteq Q_M$ and $Q_{N'} \supseteq Q_G$, respectively, such that for any two processes pm, qn of M and G respectively,*

$$\text{(1)} \qquad pm \preccurlyeq qn \text{ wrt. } M, G \iff pm \preccurlyeq qn \text{wrt. } M', N'. \tag{4.3}$$

$$\text{(2)} \qquad \text{if } pm \preccurlyeq_\kappa qn \text{ wrt. } M, G \text{ then } pm \preccurlyeq_\kappa qn \text{ wrt. } M', N'. \tag{4.4}$$

Proof. We first observe that for any transition rule of the guarded ω-automaton G, the values of its guards are bounded by some constant. The same holds for all finite effects. Let $\Gamma(G)$ be the maximal guard and $\Delta(G)$ be the maximal absolute finite effect of any transition rule of G.

The idea of this construction is to simulate one round of the simulation game with respect to M and G in $k = 2\Gamma(G) + \Delta(G) + 1$ rounds of the simulation game with respect to M' and N'. We will replace each observable[5] original transition of both players by sequences of k transition 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 transform the one-counter automaton $M = (Q_M, \mathbb{A}ct, \delta_M)$ to the one-counter automaton $M' = (Q_{M'}, \mathbb{A}ct', \delta_{M'})$ as follows:

$$\mathbb{A}ct' = \mathbb{A}ct \cup \{b\} \tag{4.5}$$

$$Q_{M'} = Q_M \cup \{p_i | 1 \leqslant i < k, p \in Q_M\} \tag{4.6}$$

$$\delta_{M'} = \{p \xrightarrow{a,d} p'_k | p \xrightarrow{a,d} p' \in \delta_M \wedge a \in \mathbb{A}ct\} \tag{4.7}$$

$$\cup \{p'_i \xrightarrow{b,0} p'_{i-1} | 1 < i < k\} \tag{4.8}$$

$$\cup \{p'_1 \xrightarrow{b,0} p'\}. \tag{4.9}$$

[5]Due to lemmas 28 and 29 ε-transition rules was already removed.

We see that

$$pm \xrightarrow{a}_M p'm' \iff pm \xrightarrow{a}_{M'} p'_{k-1}m' \xrightarrow{b^{k-2}}_{M'} p'_1m' \xrightarrow{b}_{M'} p'm'. \tag{4.10}$$

Now we transform the guarded ω-automaton $G = (Q_G, \mathbb{A}ct, \delta_G)$ to the ω-automaton $N' = (Q_{N'}, \mathbb{A}ct', \delta_{N'})$. Each original transition rule will be replaced by a sequence of k transition rules that test if the current counter value exceeds the guard g and adjust the counter accordingly. The new ω-automaton N' has states

$$Q_{N'} = Q_G \cup \{t_i | 0 \leqslant i < k, t \in \delta_G\}. \tag{4.11}$$

For each original transition rule $t = (q, a, g, d, q') \in \delta_G$, we add the following transition rules to $\delta_{N'}$. First, to test the guard:

$$q \xrightarrow{a,0} t_{k-1}, \tag{4.12}$$

$$t_i \xrightarrow{b,-1} t_{i-1}, \text{ for } k - g < i < k \tag{4.13}$$

$$t_i \xrightarrow{b,+1} t_{i-1}, \text{ for } k - 2g < i < k - g. \tag{4.14}$$

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

$$t_i \xrightarrow{b,+1} t_{i-1}, \text{ for } k - 2g - |d| < i < k - 2g \tag{4.15}$$

$$t_i \xrightarrow{b,0} t_{i-1}, \text{ for } 0 \leqslant i < k - 2g - |d|. \tag{4.16}$$

In case $d < 0$ we add

$$t_i \xrightarrow{b,-1} t_{i-1}, \text{ for } k - 2g - |d| < i < k - 2g \tag{4.17}$$

$$t_i \xrightarrow{b,0} t_{i-1}, \text{ for } 0 \leqslant i < k - 2g - |d|. \tag{4.18}$$

In case $d = \omega$ we add

$$t_i \xrightarrow{b,\omega} t_{i-1}, \text{ for } i = k - 2g \tag{4.19}$$

$$t_i \xrightarrow{b,0} t_{i-1}, \text{ for } 0 \leqslant i < k - 2g. \tag{4.20}$$

Finally, we allow a move to the new state:

$$t_0 \xrightarrow{b,0} p'. \tag{4.21}$$

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

$$qn \xrightarrow{a}_G q'n' \iff qn \xrightarrow{ab^{k-1}}_{N'} q'n'. \tag{4.22}$$

Equations (4.3) and (4.4) now follows from Equations (4.10) and (4.22). This concludes the proof of Lemma 31. □

Theorem 10 follows by the composition of Lemmas 30 and 31.

4.2 Simulation by ω-automata

We have shown how to get rid of silent transitions in checking weak pre-order of one-counter automata, at the price of introducing ω-transitions rules in the Duplicator's automaton. Now we have to solve the simplified problem, i.e. prove the following:

Theorem 11. *The simulation pre-order between a one-counter automaton without silent transition rules and an ω-automaton is effectively semilinear and thus decidable.*

We will start by defining a new notion of approximants promised in the beginning of this chapter, denoted below by $(\preccurlyeq^{\lambda})_{\lambda \in Ord}$. Then we will show the following two facts:

- For every one-counter automaton $M = (Q_M, \mathbb{A}ct, \delta_M)$ and ω-automaton $N = (Q_N, \mathbb{A}ct, \delta_N)$, the approximants $(\preccurlyeq^{\lambda})_{\lambda \in Ord}$ stabilize at a finite level (in Section 4.2.1).
- The finite approximants are effectively semilinear relations (in Section 4.2.2).

4.2.1 Approximants

If not stated otherwise in Section 4.2 we assume that $M = (Q_M, \mathbb{A}ct, \delta_M)$ is a fixed one-counter automaton without ε-transition rules and $N = (Q_N, \mathbb{A}ct, \delta_N)$ is a fixed ω-automaton.

First we define approximants $\preccurlyeq_{\kappa}^{\lambda}$ in two (ordinal) dimensions. The approximants generalize the standard simulation approximants $\preccurlyeq_{\kappa}$ for the simulation pre-order between one-counter automaton and ω-automaton.

From a game-theoretic perspective the subscript κ indicates the number of rounds Duplicator can survive and the superscript λ denotes the number of *ω-transitions*, induced by ω-transition rules, Spoiler needs to allow. E.g., $pm \preccurlyeq_5^2 qn$ if Duplicator can guarantee that no play of the simulation game that contains less than 3 moves via ω-transitions is losing for him in less than 6 rounds.

Definition 34 (approximants)**.** *Define the family of relations $\preccurlyeq_{\kappa}^{\lambda} \subseteq Q_M \times \mathbb{N} \times Q_N \times \mathbb{N}$ indexed by pairs of ordinals κ, λ.*

- $\preccurlyeq_\kappa^0 = \preccurlyeq_0^\lambda = Q_M \times \mathbb{N} \times Q_N \times \mathbb{N}$ *are the full relations, for every* κ *and* λ.
- $pm \preccurlyeq_{\kappa+1}^{\lambda+1} qn$ *if and only if* (pm, qn) *satisfies: for all* $pm \xrightarrow{a} p'm'$ *there is a transition* $qn \xrightarrow{a} q'n'$ *s.t. either*

  (1) $q \xrightarrow{a,\omega} q' \in \delta_N$ *(the move is via an* ω*-transition rule) and* $p'm' \preccurlyeq_\kappa^\lambda q'n'$, *or*

  (2) $q \xrightarrow{a,d} q' \in \delta_N$ *where* $d = n' - n$ *and* $p'm' \preccurlyeq_\kappa^{\lambda+1} q'n'$.
- *For limit ordinals* κ', λ' *we define* $\preccurlyeq_\kappa^{\lambda'} = \bigcap_{\lambda<\lambda'} \preccurlyeq_\kappa^\lambda$ *and* $\preccurlyeq_{\kappa'}^\lambda = \bigcap_{\kappa<\kappa'} \preccurlyeq_\kappa^\lambda$.

Observe that whenever Duplicator has a choice between an ω-transition and a transition induced by ordinary transition rule:

$$q \xrightarrow{a,\omega} q' \text{ and } q \xrightarrow{a,d} q'$$

with the same destination state q'; it is always better for him to chose the ω-transition.

For convenience we introduce additionally two hierarchies indexed by single ordinals:

$$\preccurlyeq^\lambda = \bigcap_{\kappa \in Ord} \preccurlyeq_\kappa^\lambda \qquad\qquad \preccurlyeq_\kappa = \bigcap_{\lambda \in Ord} \preccurlyeq_\kappa^\lambda \ . \tag{4.23}$$

We slightly overload the notation here, which is acceptable as the relations $\preccurlyeq_\kappa$ are exactly the approximants induced by weak simulation expansion. On the other hand $\preccurlyeq^\lambda$ is a special notion derived from the syntactic peculiarity of ω-transitions present in the simulation game when Duplicator plays in an ω-automaton.

Example 19. *Consider an automaton that consists of a single* a*-labelled loop in state* p, *i.e.* $p \xrightarrow{a,0} p$ *and an* ω*-automaton with transition rules* $q \xrightarrow{a,\omega} q' \xrightarrow{a,-1} q'$ *only. We see that for any* $m, n \in \mathbb{N}$, $pm \preccurlyeq_n q'n \not\preccurlyeq_{n+1} pm$. *Moreover,* $pm \preccurlyeq_\omega qn$ *but* $pm \not\preccurlyeq_{\omega+1} qn$ *and* $pm \preccurlyeq^1 qn$ *but* $pm \not\preccurlyeq_{\omega+1}^2 qn$ *and thus* $pm \not\preccurlyeq^2 qn$.

Similarly as before we characterize approximants via game as follows.

Definition 35. *The* approximant game *is played in rounds between Spoiler and Duplicator. Game positions are quadruples* $(\kappa, \lambda, pm, qn)$ *where* pm, qn *are processes of* N *and* N' *respectively, and* κ, λ *are ordinals. In each round that starts in* $(\kappa, \lambda, pm, qn)$*:*

- *Spoiler chooses ordinals* $\kappa' < \kappa$ *and* $\lambda' < \lambda$,
- *Spoiler makes a move* $pm \xrightarrow{a} p'm'$,
- *Duplicator responds by making a move* $qn \xrightarrow{a} q'n'$ *using some transition* t.

If t was an ω-transition the next round starts at the position $(\kappa', \lambda', p'm', q'n')$, Otherwise the next round starts at $(\kappa', \lambda, p'm', q'n')$ (in this case Spoiler's choice of λ' becomes irrelevant). If a player cannot move the other wins and if κ or λ becomes 0, Duplicator wins.

Lemma 32. *If Duplicator wins the approximant game from $(\kappa, \lambda, pm, qn)$ then he also wins the game from $(\kappa', \lambda', pm, qn)$ for any $\kappa' \leqslant \kappa$ and $\lambda' \leqslant \lambda$.*

Proof. If Duplicator has a winning strategy in the game from $(\kappa, \lambda, pm, qn)$ then he can use the same strategy in the game from $(\kappa', \lambda', pm, qn)$ and maintain the invariant that the pair of ordinals in the game position is pointwise smaller than the pair in the original game. Thus Duplicator wins from $(\kappa', \lambda', pm, qn)$. □

Lemma 33. *$pm \preccurlyeq_{\kappa}^{\lambda} qn$ iff Duplicator has a strategy to win the approximant game that starts in $(\kappa, \lambda, pm, qn)$.*

Proof. We show both directions by well-founded induction on pairs of ordinals (κ, λ).

For the "only if" direction we assume $pm \preccurlyeq_{\kappa}^{\lambda} qn$ and show that Duplicator wins the game from $(\kappa, \lambda, pm, qn)$. In the base case of $\kappa = 0$ or $\lambda = 0$ Duplicator directly wins by definition. By induction hypothesis we assume that the claim is true for all pairs pointwise smaller than (κ, λ). Spoiler starts a round by picking ordinals $\kappa' < \kappa$ and $\lambda' < \lambda$ and moves $pm \xrightarrow{a} p'm'$. We distinguish two cases, depending on whether λ is a limit or successor ordinal.

Case 1: λ is a successor ordinal. By Lemma 32 we can safely assume that $\lambda = \lambda' + 1$. By our assumption $pm \preccurlyeq_{\kappa}^{\lambda} qn$, there must be a response $qn \xrightarrow{a} q'n'$ that is either due to an ω-transition and then $p'm' \preccurlyeq_{\kappa'}^{\lambda'} q'n'$ or due to an ordinary transition, in which case we have $p'm' \preccurlyeq_{\kappa'}^{\lambda'+1} q'n'$. 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: λ is a limit ordinal. By $pm \preccurlyeq_{\kappa}^{\lambda} qn$ we obtain $pm \preccurlyeq_{\kappa}^{\gamma} qn$ for all $\gamma < \lambda$. If κ is a successor ordinal then, by Lemma 32, we can safely assume that $\kappa' = \kappa - 1$. Otherwise, if κ is a limit ordinal, then, by Def. 34, we have $pm \preccurlyeq_{\kappa''}^{\gamma} qn$ for all $\kappa'' < \kappa$ and in particular $pm \preccurlyeq_{\kappa'+1}^{\gamma} qn$. So in either case we obtain

$$pm \preccurlyeq_{\kappa'+1}^{\gamma} qn \text{ for all } \gamma < \lambda. \tag{4.24}$$

If there is some ω-transition that allows a response $qn \xrightarrow{a}_{\omega} q'n'$ that satisfies $p'm' \preccurlyeq_{\kappa'}^{\lambda'} q'n'$, 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 (4.24) implies that for every $\gamma < \lambda$ there is a response to some $q'n'$ that uses a non-ω-transition $t(\gamma)$ and that satisfies $p'm' \preccurlyeq_{\kappa'}^{\gamma} q'n'$. Since

λ is a limit ordinal, there exist infinitely many $\gamma < \lambda$. By the pigeonhole principle, that there must be one transition that occurs as $t(\gamma)$ for infinitely many γ. Therefore, a response that uses this transition satisfies $p'm' \preccurlyeq^{\lambda}_{\kappa'} q'n'$. If Duplicator uses this response, the game continues from position $(\kappa', \lambda, p'm', q'n')$ and he wins by induction hypothesis.

For the "if" direction we show that $pm \not\preccurlyeq^{\lambda}_{\kappa} qn$ implies that Spoiler has a winning strategy in the approximant game from $(\kappa, \lambda, pm, qn)$. In the base case of $\kappa = 0$ or $\lambda = 0$ the implication holds trivially since the premise is false. By induction hypothesis we assume that the implication is true for all pairs pointwise smaller than (κ, λ). Observe that if κ or λ are limit ordinals then (by Def. 34) there are successors $\lambda' \leqslant \lambda$ and $\kappa' \leqslant \kappa$ s.t. $pm \not\preccurlyeq^{\lambda'}_{\kappa'} qn$. So without loss of generality we can assume that κ and λ are successors. By the definition of approximants there must be a move $pm \xrightarrow{a} p'm'$ s.t.

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

So if Spoiler chooses $\kappa' = \kappa - 1$, $\lambda' = \lambda - 1$ and moves $pm \xrightarrow{a} p'm'$ then any possible response by Duplicator will take the game to a position $(\gamma, \kappa', p'm', q'n')$ for a $\gamma \leqslant \lambda$. By induction hypothesis Spoiler wins the game. □

Lemma 34. *For all ordinals κ, λ the following properties hold.*

(1) $pm \preccurlyeq^{\lambda}_{\kappa} qn$ implies $pm' \preccurlyeq^{\lambda}_{\kappa} qn'$ for all $m' \leqslant m$ and $n' \geqslant n$

(2) If $\kappa' \geqslant \kappa$ and $\lambda' \geqslant \lambda$ then $\preccurlyeq^{\lambda'}_{\kappa'} \subseteq \preccurlyeq^{\lambda}_{\kappa}$.

(3) There are ordinals $\kappa^\bullet, \lambda^\bullet$ such that $\preccurlyeq_{\kappa^\bullet} = \preccurlyeq_{\kappa^\bullet+1}$ and $\preccurlyeq^{\lambda^\bullet} = \preccurlyeq^{\lambda^\bullet+1}$.

(4) $\preccurlyeq \; = \; \bigcap_{\kappa} \preccurlyeq_{\kappa} \; = \; \bigcap_{\lambda} \preccurlyeq^{\lambda}$

The first point states that individual approximants are monotonic in the sense of Proposition 3. Points (2)-(4) imply that both $\preccurlyeq_{\kappa}$ and $\preccurlyeq^{\lambda}$ yield non-increasing sequences of approximants that converge towards weak pre-order. It is easy to see that the approximants $\preccurlyeq_{\kappa}$ do not converge at finite levels, and not even at ω, i.e., $\kappa^\bullet > \omega$ in general. However, we will show that the approximants $\preccurlyeq^{\lambda}$ do converge at a finite level, i.e., $\lambda^\bullet \in \mathbb{N}$ for any pair of automata.

Proof. (1) By Lemma 33 it suffices to observe that Duplicator can reuse a winning strategy in the approximant game from $(\kappa, \lambda, pm, qn)$ to win the game from $(\kappa, \lambda, pm - d_1, qn + d_2)$ for naturals d_1, d_2.

(2) If $pm \preccurlyeq^{\lambda'}_{\kappa'} qn$ then, by Lemma 33, Duplicator wins the approximant game from position $(\kappa', \lambda', pm, qn)$. By Lemma 32 he can also win the approximant game from $(\kappa, \lambda, pm, qn)$. Thus $pm \preccurlyeq^{\lambda}_{\kappa} qn$ by Lemma 33.

(3) By (2) we see that with increasing ordinal index κ the approximant relations $\preccurlyeq_\kappa$ form a decreasing sequence of relations, thus they stabilize for some ordinal $\kappa^\bullet$. The existence of a convergence ordinal for $\preccurlyeq^{\lambda^\bullet}$ follows analogously.

(4) First we observe that $\bigcap_\kappa \preccurlyeq_\kappa = \bigcap_\kappa \bigcap_\lambda \preccurlyeq_\kappa^\lambda = \bigcap_\lambda \bigcap_\kappa \preccurlyeq_\kappa^\lambda = \bigcap_\lambda \preccurlyeq^\lambda$. It remains to show that $\preccurlyeq = \bigcap_\kappa \preccurlyeq_\kappa$.

To show $\preccurlyeq \supseteq \bigcap_\kappa \preccurlyeq_\kappa$, we use $\kappa^\bullet$ from (3) and rewrite the right side to $\bigcap_\kappa \preccurlyeq_\kappa = \preccurlyeq_{\kappa^\bullet} = \preccurlyeq_{\kappa^\bullet+1}$. From Definition 34 we get that $\preccurlyeq_\kappa = \preccurlyeq_\kappa^\gamma$ for $\gamma \geqslant \kappa$ and therefore $\preccurlyeq_{\kappa^\bullet+1}^{\kappa^\bullet+1} = \preccurlyeq_{\kappa^\bullet+1} = \preccurlyeq_{\kappa^\bullet} = \preccurlyeq_{\kappa^\bullet}^{\kappa^\bullet}$. This means $\preccurlyeq_{\kappa^\bullet}^{\kappa^\bullet} = \bigcap_\kappa \preccurlyeq_\kappa$ must be a simulation relation and hence a subset of $\preccurlyeq$.

To show $\preccurlyeq \subseteq \bigcap_\kappa \preccurlyeq_\kappa$, we prove by ordinal induction that $\preccurlyeq \subseteq \preccurlyeq_\kappa$ for all ordinals κ. The base case $\kappa = 0$ is trivial. For the induction step we prove the equivalent property $\not\preccurlyeq_\kappa \subseteq \not\preccurlyeq$. There are two cases.

In the first case, $\kappa = \kappa' + 1$ is a successor ordinal. If $pm \not\preccurlyeq_{\kappa'+1} qn$ then $pm \not\preccurlyeq_{\kappa'+1}^{\kappa'+1} qn$ and therefore, by Lemma 33, Spoiler wins the approximant game from $(\kappa'+1, \kappa'+1, pm, qn)$. Let $pm \xrightarrow{a} p'm'$ 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 $qn \xrightarrow{a} q'n'$ we have $p'm' \not\preccurlyeq_{\kappa'}^{\kappa'} q'n'$. Then also $p'm' \not\preccurlyeq_{\kappa'} q'n'$ and by induction hypothesis $p'm' \not\preccurlyeq q'n'$. From this we obtain that Spoiler wins the simulation game from $(p'm', q'n')$ and thus from (pm, qn). Therefore $pm \not\preccurlyeq qn$, as required.

In the second case, κ is a limit ordinal. Then $pm \not\preccurlyeq_\kappa qn$ implies $pm \not\preccurlyeq_{\kappa'} qn$ for some $\kappa' < \kappa$ and therefore $pm \not\preccurlyeq qn$ by induction hypothesis. $\square$

The following lemma shows an 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.

Lemma 35. *There is a fixed bound $c \in \mathbb{N}$ s.t. the following property holds. Suppose $p \in Q_M,\ q \in Q_N,\ m, n \in \mathbb{N}$ with $m > c$,and $\kappa \in Ord$ satisfy*

$$pm \preccurlyeq_\kappa qn.$$

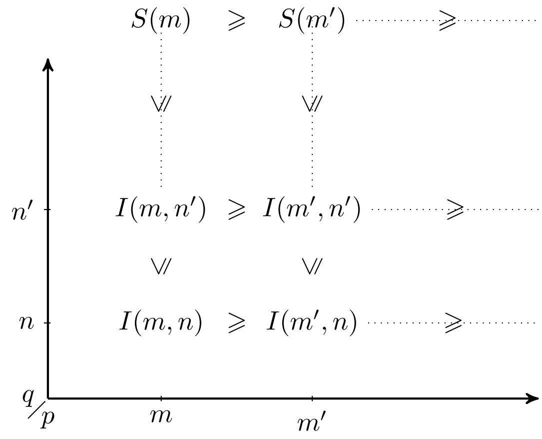
Then for every $m' > m$ there is $n' \in \mathbb{N}$ s.t.

$$pm' \preccurlyeq_\kappa qn'.$$

Proof. It suffices to show the existence of a local bound c for any given pair of states (p, q) that satisfies the condition, since we can simply take the global c to be the maximal such bound over all finitely many pairs (p, q). Consider now a fixed pair (p, q) of states. For $m, n \in \mathbb{N}$, we define the following (sequences of) ordinals.

$$\begin{aligned} I(m,n) &= \text{the largest ordinal } \kappa \text{ with } pm \preccurlyeq_\kappa qn, \text{ or } \kappa^\bullet \\ &\quad \text{if } pm \preccurlyeq qn, \\ I(m) &= \text{the increasing sequence of ordinals } I(m,n)_{n\geqslant 0}, \\ S(m) &= \sup\{I(m)\}. \end{aligned}$$

Observe that $I(m,n)$ 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.



By Lemma 34 (1), we derive that for any $n' > n \in \mathbb{N}$ and $m' > m \in \mathbb{N}$

$$I(m,n') \geqslant I(m,n) \geqslant I(m',n) \tag{4.25}$$

and because of the second inequality also $S(m) \geqslant S(m')$. So the ordinal sequence $S(m)_{m\geqslant 0}$ 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:

$$\forall l > k.\ S(l) = S(k).$$

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.

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

Case 2. For all $l \geqslant k$ 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. Given $m' > m > c = k$ and $pm \preccurlyeq_\kappa qn$ we let $n' := n_{m'}$ and obtain $I(m', n') = S(m') = S(k) \geqslant \kappa$ and thus $pm' \preccurlyeq_\kappa qn'$, 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 $l' > l$ the sequence $I(l')$ has no maximal element. To see this, consider sequences $I(x)$ and $I(x')$ with $x' > x \geqslant k$. If $I(x')$ has a maximal element then so must $I(x)$, by equation (4.25) and $S(x) = S(x') = S(k)$. Given this, we repeat the argument of the Case 1. with $c := l$ and again satisfy the requirement. $\square$

The hierarchy of approximants $\preccurlyeq_\lambda$ stabilizes at a finite level for any pair of automata. This is the first of two key technical results for the proof of Theorem 11.

Lemma 36. *Consider simulation pre-order $\preccurlyeq$ between an automaton without ε-transitions $M = (Q_M, \mathbb{A}ct, \delta_M)$ and an ω-automaton $N = (Q_N, \mathbb{A}ct, \delta_N)$. There exists a constant $\lambda^\bullet \in \mathbb{N}$ s.t. $\preccurlyeq = \preccurlyeq^{\lambda^\bullet}$.*

Proof. We assume the contrary and derive a contradiction. By Lemma 34 (4), the inclusion $\preccurlyeq \subseteq \preccurlyeq^\lambda$ always holds for every ordinal λ. Thus, if $\nexists \lambda^\bullet \in \mathbb{N}.\ \preccurlyeq = \preccurlyeq^{\lambda^\bullet}$ then for every finite $\lambda \in \mathbb{N}$ there are processes $p_0 m_0$ and $q_0 n_0$ s.t. $p_0 m_0 \preccurlyeq^\lambda q_0 n_0$ but $p_0 m_0 \not\preccurlyeq q_0 n_0$. In particular, this holds for the special case of $\lambda = |Q_M \times Q_N|(c+1)$, where c is the constant given by Lemma 35, which we consider in the rest of this proof.

Since $q_0 n_0$ does not simulate $p_0 m_0$, 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* κ_i — the largest ordinal with $p_i m_i \preccurlyeq_{\kappa_i} q_i n_i$ — strictly decreases along rounds of any play. By monotonicity, Lemma 34 (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

$$(p_0 m_0, q_0 n_0)(t_0, t_0')(p_1 m_1, q_1 n_1)(t_1, t_1') \ldots (p_k m_k, q_k n_k)$$

of length k with $p_i = p_j$ and $q_i = q_j$ for some $i < j \leqslant k$ we have $n_j < n_i$ or $m_j > m_i$. 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 κ_i strictly decreases we obtain the following:

$$\text{if } p_i = p_j, q_i = q_j \text{ and } t_{i-1}', t_{j-1}' \in \delta_\omega \text{ then } m_j > m_i. \tag{4.26}$$

Here δ_ω denotes the set of ω-transition, induced by transition rules with symbolic effect ω in the Duplicator's ω-automaton N.

Although Duplicator loses the simulation game between $p_0 m_0$ and $q_0 n_0$, our assumption $p_0m_0 \preccurlyeq^\lambda q_0n_0$ with $\lambda = |Q_M \times Q_N|(c+1)$ implies that Duplicator does not lose with less than λ ω-transitions, regardless of Spoiler's strategy. Thus there always is a prefix of a play along which Duplicator makes use of ω-transitions λ times. Let

$$\pi = (p_0m_0, q_0n_0)(t_0, t_0')(p_1m_1, q_1n_1)(t_1, t_1') \dots (p_km_k, q_kn_k)$$

be such a partial play.

Our choice of $\lambda = |Q_M \times Q_N|(c+1)$ guarantees that some pair (p, q) of control-states repeats at least $c+1$ times directly after Duplicator making a move via an ω-transition. Thus there are indices $i(1) < i(2) < \dots < i(c+1) < k$ s.t. for all $1 \leqslant j \leqslant c+1$ we have $p_{i(j-1)} = p, q_{i(j)} = q$ and $t'_{i(j)-1} \in \delta_\omega$. By observation (4.26) and $m_0 \geqslant 0$ we obtain that $m_{i(x)} \geqslant x$ for $0 \leqslant x \leqslant c+1$. In particular, $c \leqslant m_{i(c)} < m_{i(c+1)}$, i.e. both of Spoiler's counter values after the last two such repetitions must lie above c. This allows us to apply Lemma 35 to derive a contradiction as follows.

Let κ be the simulation level before this repetition: κ is the largest ordinal with $pm_{i(c)} \preccurlyeq_\kappa qn_{i(c)}$. Since $m_{i(c+1)} > m_{i(c)} > c$, Lemma 35 ensures the existence of a natural n' s.t. $pm_{i(c+1)} \preccurlyeq_\kappa qn'$. 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 $n_{i(c+1)}$ 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 $pm_{i(c)} \preccurlyeq_\kappa qn_{i(c)} \not\preccurlyeq_{\kappa+1} pm_{i(c)}$ and $pm_{i(c+1)} \preccurlyeq_\kappa qn_{i(c+1)}$, which contradicts the optimality of Spoiler's strategy. $\square$

4.2.2 Computability of finite approximants

To prove decidability of the simulation pre-order between one-counter automata and ω-automata we will show that for each finite level $k \in \mathbb{N}$ the approximant $\preccurlyeq^k$ is effectively semilinear, i.e. we can compute the semilinearity description of $\preccurlyeq^k$. This yields a decision procedure for simulation pre-order that works as follows. Iteratively compute $\preccurlyeq^k$ for growing k and check after each round if the approximant has converged yet. The convergence test of $\preccurlyeq^k \overset{?}{=} \preccurlyeq^{k-1}$ can easily be done, since the approximants are semilinear sets. Termination of this procedure is guaranteed by Lemma 36, and the limit is the simulation pre-order by Lemma 34 (4).

To start, we recall the following important result by Jančar, Kučera and Moller.

Theorem 12 ([38])**.** *The simulation pre-order $\preccurlyeq$ between processes of two given one-counter automata without ε-transition rules is effectively semilinear.*

The following lemma is the second key technical result underlying the proof of Theorem 11:

Lemma 37. *Given a one-counter automaton without ε-transition rules M and an ω-automaton N, the approximant relations $\preccurlyeq^k$ between them are effectively semilinear sets for all $k \in \mathbb{N}$.*

The rest of Section 4.2.2 is devoted to the proof of the this lemma.

Let $M = (Q_M, \mathbb{A}ct, \delta_M)$ and $N = (Q_N, \mathbb{A}ct, \delta_N)$. We prove the effective semilinearity of $\preccurlyeq^k$ by induction on k.

The base case $\preccurlyeq^0 = Q_M \times \mathbb{N} \times Q_N \times \mathbb{N}$ is trivially effectively semilinear.

For the induction step we proceed as follows. By induction hypothesis $\preccurlyeq^k$ is effectively semilinear. Using this, we reduce the problem of checking $\preccurlyeq^{k+1}$ between M and N to the problem of checking weak simulation $\preccurlyeq$ between two derived one-counter automata without ε-transition rules M' and N', and obtain the effective semilinearity of the relation from Theorem 12. More precisely, the derived one-counter automata M' and N' will contain all control-states of M and N, respectively, and we will have that $pm \preccurlyeq^{k+1} qn$ wrt. M, N iff $pm \preccurlyeq qn$ wrt. M', N'.

Before we describe M' and N' formally, we explain the function of a test gadget used in the construction.

An important observation is that after Duplicator made an ω-move in the approximant game between M and N, the winner of the game from the resulting position depends only on the control-states and Spoiler's counter value, because Duplicator could choose his counter arbitrarily high. Moreover, monotonicity (Lemma 34 (1)) guarantees that there must be a minimal value for Spoiler's counter with which she can win if at all. This yields the following property.

For any pair of states $(p, q) \in Q_M \times Q_N$ there must exist a value $M(p, q) \in \mathbb{N} \cup \{\omega\}$ s.t. for all $m \in \mathbb{N}$

$$(\forall n \in \mathbb{N}.\, pm \not\preccurlyeq^k qn) \iff m \geqslant M(p, q). \tag{4.27}$$

Since, by induction hypothesis, $\preccurlyeq^k$ (and thus also its complement) is effectively semilinear, we can compute the values $M(p, q)$ for all $(p, q) \in Q_M \times Q_N$.

The test gadgets. Given the values $M(p, q)$, we construct test gadgets that check whether Spoiler's counter value is $\geqslant M(p, q)$. For each $(p, q) \in Q_M \times Q_N$ we construct two one-counter automata $S(p, q)$ and $T(p, q)$ with initial states $s(p, q)$ and $t(p, q)$, respectively, such that the following property holds for all $m, n \in \mathbb{N}$.

$$s(p, q)m \not\preccurlyeq t(p, q)n \iff m \geqslant M(p, q). \tag{4.28}$$

The construction of $S(p, q)$ and $T(p, q)$ is very simple. Let $s(p, q)$ be the starting point of a counter-decreasing chain of transition rules labelled with e of length $M(p, q) \in \mathbb{N}$ where the last state of the chain allows to make a move via transition labelled f whereas

$t(p,q)$ is a simple e-loop (where e, f are fresh actions not in $\mathbb{A}ct$). If $M(p,q) = \omega$, making $s(p,q)$ a deadlock suffices. Thus $S(p,q)$ and $T(p,q)$ are one-counter automata which we denote by

$$S(p,q) = (Q_{S(p,q)}, \{e,f\}, \delta_{S(p,q)}) \text{ and } T(p,q) = (Q_{T(p,q)}, \{e\}, \delta_{T(p,q)}). \quad (4.29)$$

Wlog. we assume that their state sets are disjoint from each other and from the original automata M, N.

The construction of the automata M' and N'. Let $M' = (Q'_M, \mathbb{A}ct', \delta'_M)$ and $N' = (Q'_N, \mathbb{A}ct', \delta'_N)$ be one-counter automata constructed as follows.

$\mathbb{A}ct' = \mathbb{A}ct \cup Q_M \times Q_N \cup \{f, e\}$ (where e, f are the actions from the test gadgets).

Spoiler's new automaton M' has states

$$Q'_M = Q_M \cup \bigcup_{p \in Q_M, q \in Q_N} Q_{S(p,q)} \quad (4.30)$$

Duplicator's new automaton N' has states

$$Q'_N = Q_N \cup \{W\} \cup \bigcup_{p \in Q_M, q \in Q_N} Q_{T(p,q)}. \quad (4.31)$$

where W is a new state.

Now we define the transition relations. $\delta'_M = \delta_M \cup \bigcup_{p \in Q_M, q \in Q_N} \delta_{S(p,q)}$ plus the following transition rules for all $p \in Q_M, q \in Q_N$:

$$p \xrightarrow{(p,q),0} s(p,q). \quad (4.32)$$

$\delta'_N = \{q \xrightarrow{a,x} q' \in \delta_N \mid x \neq \omega\} \cup \bigcup_{p \in Q_M, q \in Q_N} \delta_{T(p,q)}$ plus the following transition rules for all $p, p' \in Q_M$ and $q, q' \in Q_N$:

$$\begin{aligned}
q &\xrightarrow{a,0} t(p,q') && \text{if } q \xrightarrow{a,\omega} q' \in \delta_N && (4.33)\\
q &\xrightarrow{(p',q'),0} W && && (4.34)\\
t(p,q) &\xrightarrow{(p,q),0} t(p,q) && && (4.35)\\
t(p,q) &\xrightarrow{(p,q'),0} W && \text{for all } q \neq q' && (4.36)\\
t(p,q) &\xrightarrow{a,0} W && \text{for all } a \in \mathbb{A}ct && (4.37)\\
W &\xrightarrow{a,0} W && \text{for all } a \in \mathbb{A}ct' && (4.38)
\end{aligned}$$

Correctness. We show that for any pair pm, qn of processes of the automata M, N we have $pm \preccurlyeq^{k+1} qn$ if and only if $pm \preccurlyeq qn$ in the newly constructed automata M', N'.

To prove the 'if' direction we assume that $pm \not\preccurlyeq^{k+1} qn$ wrt. M, N and derive that $pm \not\preccurlyeq qn$ wrt. M', N'. By our assumption and Definition 34, there exists some ordinal κ such that $pm \not\preccurlyeq^{k+1}_{\kappa} qn$. By Lemma 33, Spoiler has a winning strategy in the approximant game from the position $(\kappa, k+1, pm, qn)$. The result then follows from the following lemma.

Lemma 38. *For all ordinals κ, control-states $(p, q) \in Q_M \times Q_N$ and naturals $m, n \in \mathbb{N}$: If Spoiler has a winning strategy in the approximant game from the position $(\kappa, k+1, pm, qn)$ then she also has a winning strategy in the simulation game the from position (pm, qn) with respect to M' and N'.*

Proof. To prove lemma, we fix some $p \in Q_M, q \in Q_N$ and proceed by ordinal induction on κ. The base case trivially holds since Spoiler looses from a position $(0, k+1, pm, qn)$.

For the induction step let Spoiler play the same move $pm \xrightarrow{a} p'm'$ for some $a \in \mathbb{A}ct$ in both games according to her assumed winning strategy in the approximant game. Now Duplicator makes his response move in the new game between M', N', which yields two cases. In the first case, Duplicator does not use a transition rule from Equation (4.33). Then his move induces a corresponding move in the approximant game which leads to a new position $(\kappa', k+1, p'm', q'n')$ where $p'm' \not\preccurlyeq^{k+1}_{\kappa'} q'n'$ for some ordinal $\kappa' < \kappa$. Thus, by Lemma 33 and the induction hypothesis, the property holds.

In the second case, Duplicator's response is via a transition rule from Equation (4.33), which leads to a new position $(p'm', t(p'', q')n)$ for some $p'' \in Q_M$. Thus in the approximants game there will exist Duplicator moves to positions $(\kappa', k, p'm', q'n')$ where $n' \in \mathbb{N}$ can be arbitrarily high. We can safely assume that Duplicator chooses $p'' = p'$, since otherwise Spoiler can win in one round by playing $p'm' \xrightarrow{(p',q')} s(p, q)m'$. Now in the following round Spoiler can play $p'm' \xrightarrow{(p',q')} s(p', q')m'$ by Equation (4.32) and Duplicator's only option is to stay in his current state by Equation (4.35). The game thus continues from $(s(p', q')m', t(p', q')n)$. By our assumption Spoiler wins the approximant game from the position $(\kappa, k+1, pm, qn)$. Thus there is some ordinal $\kappa' < \kappa$ such that Spoiler also wins the approximant game from the position $(\kappa', k, p'm', q'n')$ for every $n' \in \mathbb{N}$. Thus, by Lemma 33 and Definition 34, we have $p'm' \not\preccurlyeq^{k}_{\kappa'} q'n'$ and by Lemma 34 (2) $p'm' \not\preccurlyeq^{k} q'n'$ for all $n' \in \mathbb{N}$. By Equation (4.27) we obtain $m' \geqslant M(p', q')$. By the construction of the gadgets and Equation (4.28) we get $s(p', q')m' \not\preccurlyeq t(p', q')n'$, which implies the desired property. □

This concludes the proof of the 'if' direction. Now we prove the 'only if' direction of the correctness property. We assume that $pm \not\preccurlyeq qn$ in the newly constructed automata M' and N' and derive that $pm \not\preccurlyeq^{k+1} qn$ wrt. M, N. To do this, we first show the following property.

Lemma 39. *Let $(p, q) \in Q_m \times Q_N$. If $pm \not\preccurlyeq qn$ with respect to automata M' and N' then there exists some general ordinal κ' s.t. $pm \not\preccurlyeq^{k+1}_{\kappa'} qn$ with respect to automata M, N.*

Proof. Assume $pm \not\preccurlyeq qn$ with respect to automata M' and N'. Since both M', N' are just one-counter automata non-simulation manifests itself at some finite approximant $\kappa \in \mathbb{N}$, i.e., $pm \not\preccurlyeq_\kappa qn$. We prove the lemma by induction on κ. The base case of $\kappa = 0$ is trivial. For the induction step we consider a move $pm \xrightarrow{a} p'm'$ for some $a \in \mathbb{A}ct$ by Spoiler in both games according to Spoiler's assumed winning strategy in the game between M', N'. It cannot be a Spoiler move $p \xrightarrow{(p,q),0} s(p, q)$ by Equation (4.32), because Duplicator would immediately win via a move by Equation (4.34). Now we consider all (possibly infinitely many) replies by Duplicator in the approximant game between M, N from a position $(\kappa', k{+}1, pm, qn)$ for some yet to be determined ordinal κ'. Our goal is to construct such κ' that the position $(\kappa', k + 1, pm, qn)$ is winning for Spoiler.

These replies fall into two classes.

In the first class, Duplicator's move $qn \xrightarrow{a} q'n'$ is not due to an ω-transition and thus also a possible move in the simulation game between M', N'. From our assumption that Spoiler wins the simulation game from position (pm, qn) in at most κ rounds, it follows that Spoiler wins the simulation game from $(p'm', q'n')$ in at most $\kappa - 1$ rounds. By induction hypothesis, there exists an ordinal κ'' s.t. Spoiler has a winning strategy in the approximant game between M', N' from position $(\kappa'', k{+}1, p'm', q'n')$. There are only finitely many such replies. Thus let κ^0 be the maximal such κ''.

In the second class, Duplicator's move $qn \xrightarrow{a} q'n'$ uses an ω-transition which does not exist in N'. Instead there exists a Duplicator transition $qn \xrightarrow{a,0} t(p'', q')n$ by Equation (4.33). From our assumption that Spoiler wins the simulation game from position (pm, qn) in at most κ rounds, it follows that Spoiler wins the simulation game from $(p'm', t(p'', q')n)$ in at most $\kappa - 1$ rounds. If $p'' \neq p'$ then this is trivially true due to a Spoiler move by Equation (4.32). Otherwise, if $p'' = p'$, then this can only be achieved by a Spoiler move of $p'm' \xrightarrow{(p',q'),0} s(p', q')m'$ in the next round, because for any other Spoiler move Duplicator has a winning countermove by Equations (4.36) or (4.37). In this case Duplicator can only reply with a move $t(p', q')n \xrightarrow{(p',q'),0} t(p', q')n$ by Equation (4.35), and we must have that Spoiler can win in at most $\kappa - 2$ rounds from position $(s(p', q')m', t(p', q')n)$. This implies, by Equation (4.28), that $m' \geqslant M(p', q')$. Then Equation (4.27) yields $\forall n' \in \mathbb{N}.\, p'm' \not\preccurlyeq^k q'n'$. Thus for every $n' \in \mathbb{N}$ there exists some ordinal κ'_n s.t. $p'm' \not\preccurlyeq^k_{\kappa_n} q'n'$. Let κ'' be the smallest ordinal s.t. $\forall n' \in \mathbb{N}.\, \kappa'_n \leqslant \kappa''$.

Each of the finitely many distinct ω-transitions yields such a κ''. Let κ^1 be the maximum of them.

We set $\kappa' := \max(\kappa^0, \kappa^1) + 1$. Then every reply to Spoilers move $pm \xrightarrow{a} p'm'$ in the approximant game from $(\kappa', k+1, pm, qn)$ leads to some position that is winning for Spoiler. So, Spoiler has a winning strategy in the approximant game from $(\kappa', k+1, pm, qn)$ and by Lemma 33, $pm \preccurlyeq_{\kappa'}^{k+1} qn$ wrt. M, N, which concludes the proof of the lemma. □

By Lemma 39 we have $pm \not\preccurlyeq_{\kappa'}^{k+1} qn$ for some ordinal κ' and thus $pm \not\preccurlyeq^{k+1} qn$ wrt. M, N. This concludes the 'only if' direction.

We have constructed one-counter automata M', N' s.t. $pm \preccurlyeq^{k+1} qn$ wrt. M, N if and only if $pm \preccurlyeq qn$ wrt. M', N'. By Theorem 12, $\preccurlyeq^{k+1}$ is effectively semilinear.

4.3 Approximant convergence at level $\omega \cdot \omega$

We show that the hierarchy of approximants $\preccurlyeq_\kappa$ induced by ordinary weak simulation expansion converges at level $\kappa = \omega^2$ on LTSs induced by one-counter automata.

Lemma 40. *When considering approximants between an one-counter automaton and an ω-automaton, we have $\preccurlyeq_{\omega \cdot i} \subseteq \preccurlyeq^i$ for every $i \in \mathbb{N}$.*

Proof. By induction on i. The base case of $i = 0$ is trivial, since $\preccurlyeq^0$ is the full relation. We prove the inductive step by assuming the contrary and deriving a contradiction. Let $pm \preccurlyeq_{\omega \cdot i} qn$ and $pm \not\preccurlyeq^i qn$ for some $i > 0$. Then there exists some ordinal κ s.t. $pm \not\preccurlyeq_\kappa^i qn$. Without restriction let κ be the least ordinal satisfying this condition. If $\kappa \leqslant \omega \cdot i$ then we trivially have a contradiction. Now we consider the case $\kappa > \omega i$. By $pm \not\preccurlyeq_\kappa^i qn$ and Lemma 33, Spoiler has a winning strategy in the approximant game from position (κ, i, pm, qn). Without restriction we assume that Spoiler plays optimally, i.e., wins as quickly as possible. Thus this game must reach some game position $(\kappa' + 1, i, p'm', q'n')$ where $\kappa' \geqslant \omega \cdot i$ is a limit ordinal, such that Spoiler can win from $(\kappa' + 1, i, p'm', q'n')$ but not from $(\kappa', i, p'm', q'n')$. I.e., $p'm' \not\preccurlyeq_{\kappa'+1}^i q'n'$, but $p'm' \preccurlyeq_{\kappa'}^i q'n'$. Consider Spoiler's move $p'm' \xrightarrow{a} p''m''$ according to her optimal winning strategy in the game from position $(\kappa' + 1, i, p'm', q'n')$. Since $p'm' \preccurlyeq_{\kappa'}^i q'n'$ and κ' is a limit ordinal, for every ordinal $\gamma_k < \kappa'$, Duplicator must have some countermove $q'n' \xrightarrow{a} q_k n_k$ s.t. $p''m'' \preccurlyeq_{\gamma_k}^j q_k n_k$, where $j = i-1$ if the move was due to an ω-transition and $j = i$ otherwise. In particular, $\sup_k\{\gamma_k\} = \kappa'$. However, since Spoiler's move $p'm' \xrightarrow{a} p''m''$ was according to her optimal winning strategy from position $(\kappa' + 1, i, p'm', q'n')$, we have that $p''m'' \not\preccurlyeq_{\kappa'}^j q_k n_k$. Therefore, there must be infinitely many different Duplicator countermoves $q'n' \xrightarrow{a} q_k n_k$. 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 $\gamma < \kappa'$ there is some Duplicator countermove $q'n' \xrightarrow{a} q_k n_k$ which is due to an ω-transition s.t. $p''m'' \preccurlyeq^{i-1}_{\gamma_k} q_k n_k$ where $\gamma_k \geqslant \gamma$ (note the $i-1$ index due to the ω-transition). In particular, we can choose $\gamma = \omega \cdot (i-1)$, because $i > 0$ and $\kappa' \geqslant \omega \cdot i$. Then we have $p''m'' \preccurlyeq^{i-1}_{\omega\cdot(i-1)} q_k n_k$, but $p''m'' \not\preccurlyeq^{i-1}_{\kappa'} q_k n_k$. However, from $p''m'' \preccurlyeq^{i-1}_{\omega\cdot(i-1)} q_k n_k$ and the induction hypothesis, we obtain $p''m'' \preccurlyeq^{i-1} q_k n_k$ and in particular $p''m'' \preccurlyeq^{i-1}_{\kappa'} q_k n_k$. Contradiction. □

Theorem 13. *Weak simulation approximants on one-counter automata converge at level ω^2, but not earlier in general.*

Proof. First we show that $\preccurlyeq_{\omega^2}$ is contained in $\preccurlyeq$. Let pm and qn be processes of automata M and N, respectively. Let M', N' be the derived OCA$_{\neg 0}$ and ω-automaton from Theorem 10. Assume $pm \preccurlyeq_{\omega^2} qn$ wrt. M, N. Then, by Theorem 10 (2), $pm \preccurlyeq_{\omega^2} qn$ wrt. M', N'. In particular we have $pm \preccurlyeq_{\omega\cdot\lambda^\bullet} qn$ wrt. M', N', with the $\lambda^\bullet \in \mathbb{N}$ from Lemma 36. From Lemma 40 we obtain $pm \preccurlyeq^{\lambda^\bullet} qn$ wrt. M', N'. Lemma 36 yields $pm \preccurlyeq qn$ wrt. M', N'. Finally, by Theorem 10 (1), we obtain $pm \preccurlyeq qn$ wrt. M, N.

To see that ω^2 is needed in general, consider the following class of simple examples. Let $p \xrightarrow{a} p$ define a simple automaton (actually a finite automaton). For every $i \in \mathbb{N}$ we define an automaton N_i with transition rules $(q_k, a, -1, q_k)$, $(q_{k-1}, \varepsilon, 0, q'_{k-1})$, $(q'_{k-1}, \varepsilon, 1, q'_{k-1})$, and $(q'_{k-1}, a, 0, q_k)$ for all k with $1 \leqslant k \leqslant i$. Then, for the automaton N_i, we have $p \preccurlyeq_{\omega\cdot i} q_0 0$, but $p \not\preccurlyeq q_0 0$. Thus in general $\preccurlyeq \neq \preccurlyeq_{\omega\cdot i}$ for any $i \in \mathbb{N}$.

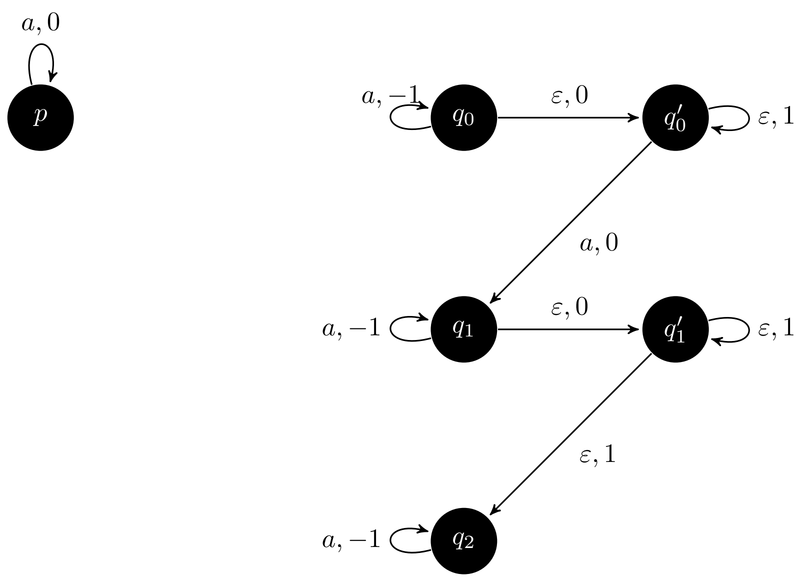


Figure 4.1: Spoiler's automaton and Duplicator's automaton N_2.

□

4.4 Branching pre-order

Here we only sketch how branching pre-order checking may be reduced to weak pre-order checking in the class of one-counter automata, thus providing a sketch of the proof of Theorem 9.

Observe that the branching simulation round can be seen as a sequence of two Spoilers move and two Duplicators moves. In her second move Spoiler decides if she wants to make a rollback of her first move; the second Duplicator's move is then fully determined. The idea of reduction is to encode one round of branching simulation game into two rounds of weak simulation game. We will base our argument on the observation that if Spoiler makes a silent move then Duplicator can safely respond with the silent self-loop, thus remaining in the same process.

We will describe how to transform two given automata M, N into two new automata M', N', with the following properties:

(1) the set of processes of M (resp. N) is included in the set of processes of M' (resp. N');

(2) two processes are related by branching pre-order wrt. M, N if and only if the processes are related by weak pre-order wrt. M', N'.

The alphabet of M' and N' will contain two additional letters r and n, standing for 'rollback' and 'no rollback', respectively. For every transition rule t of M (resp. N) labelled with an observable action $a \in \mathbb{A}ct$, the automaton M' (resp. N') will contain a new states. The transition rules in the new automata are split into four groups:

- silent transition rules are the same as in the original automata;
- for every transition rule $t = (p \xrightarrow{a,d} q)$, where $a \in \mathbb{A}ct$, we introduce a corresponding transition rule which goes from p to state t:

$$p \xrightarrow{a,d} t;$$

- for every state $t = (p \xrightarrow{a,d} q)$ originating in a transition rule, we introduce a transition rule labelled with n

$$t \xrightarrow{n,0} q,$$

  which does not modify the counter;

- for every state $t = (p \xrightarrow{a,d} q)$ originating in a transition rule, we introduce a transition rule labelled with r

$$t \xrightarrow{r,-d} p,$$

  which modifies the counter by $-d$.

Note that rollbacks are only allowed after observable transitions. This does not restrict the power of Spoiler; indeed, in response to a Spoiler's silent move, it is optimal for Duplicator to stay in the same process (using a silent self-loop), and thus rollback is useless. One can easily check that conditions (1) and (2) are fulfilled.

Chapter 5

Bibliography

[1] Parosh Aziz Abdulla and Karlis Cerans. Simulation is decidable for one-counter nets (extended abstract). In *CONCUR*, pages 253–268, 1998.

[2] Jos C. M. Baeten and Jan Willem Bergstra, Jan A.and Klop. Decidability of bisimulation equivalence for processes generating context-free languages. In *PARLE (2)*, pages 94–111, 1987.

[3] Jos C. M. Baeten and Jan Willem Bergstra, Jan A.and Klop. Decidability of bisimulation equivalence for processes generating context-free languages. *J. ACM*, 40(3):653–682, 1993.

[4] Stanislav Böhm, Stefan Göller, and Petr Jančar. Bisimilarity of one-counter processes is PSPACE-complete. In *CONCUR*, pages 177–191, 2010.

[5] Stanislav Böhm, Stefan Göller, and Petr Jančar. Equivalence of deterministic one-counter automata is NL-complete. *CoRR*, abs/1301.2181, 2013.

[6] Olaf Burkart, Didier Caucal, and Bernhard Steffen. An elementary bisimulation decision procedure for arbitrary context-free processes. In *MFCS*, pages 423–433, 1995.

[7] Søren Christensen. *Decidability and Decomposition in process algebras*. PhD thesis, Dept. of Computer Science, University of Edinburgh, UK, 1993.

[8] Søren Christensen, Yoram Hirshfeld, and Faron Moller. Bisimulation equivalence is decidable for Basic Parallel Processes. In *CONCUR*, pages 143–157, 1993.

[9] Søren Christensen, Hans Hüttel, and Colin Stirling. Bisimulation equivalence is decidable for all context-free processes. *Inf. Comput.*, 121(2):143–148, 1995.

[10] Wojciech Czerwiński. *Partially-commutative context-free graphs*. PhD thesis, University of Warsaw, 2012.

[11] Wojciech Czerwiński, Sibylle B. Fröschle, and Sławomir Lasota. Partially-commutative context-free processes. In *CONCUR*, pages 259–273, 2009.

[12] Wojciech Czerwiński, Sibylle B. Fröschle, and Sławomir Lasota. Partially-commutative context-free processes: Expressibility and tractability. *Inf. Comput.*, 209(5):782–798, 2011.

[13] Wojciech Czerwiński, Piotr Hofman, and Sławomir Lasota. Decidability of branching bisimulation on normed commutative context-free processes. In *CONCUR*, pages 528–542, 2011.

[14] Wojciech Czerwiński and Sławomir Lasota. Fast equivalence-checking for normed context-free processes. In *FSTTCS*, pages 260–271, 2010.

[15] Javier Esparza. Petri nets, commutative context-free grammars, and Basic Parallel Processes. *Fundam. Inform.*, 31(1):13–25, 1997.

[16] Rob J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In *CONCUR*, pages 278–297, 1990.

[17] Rob J. van Glabbeek. The linear time - branching time spectrum II. In *CONCUR*, pages 66–81, 1993.

[18] Jan Friso Groote and Hans Hüttel. Undecidable equivalences for basic process algebra. *Inf. Comput.*, 115(2):354–371, 1994.

[19] Ken Higuchi, Etsuji Tomita, and Mitsuo Wakatsuki. A polynomial-time algorithm for checking the inclusion for real-time deterministic restricted one-counter automata which accept by final state. *IEICE Transactions*, 78-D(8):939–950, 1995.

[20] Ken Higuchi, Etsuji Tomita, and Mitsuo Wakatsuki. A polynomial-time algorithm for checking the inclusion for strict deterministic restricted one-counter automata. *IEICE Transactions*, 78-D(4):305–313, 1995.

[21] Yoram Hirshfeld. Congruences in commutative semigroups. Technical report, University of Edinburgh, LFCS report ECS-LFCS-94-291, 1994.

[22] Yoram Hirshfeld. Petri nets and the equivalence problem. In *CSL*, pages 165–174. 1994.

[23] Yoram Hirshfeld. Bisimulation trees and the decidability of weak bisimulations. *Electr. Notes Theor. Comput. Sci.*, 5:2–13, 1996.

[24] Yoram Hirshfeld, Mark Jerrum, and Faron Moller. A polynomial-time algorithm for deciding equivalence of normed context-free processes. In *FOCS*, pages 623–631, 1994.

[25] Yoram Hirshfeld, Mark Jerrum, and Faron Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theor. Comput. Sci.*, 158(1&2):143–159, 1996.

[26] Yoram Hirshfeld, Mark Jerrum, and Faron Moller. A polynomial-time algorithm for deciding bisimulation equivalence of normed Basic Parallel Processes. *Mathematical Structures in Computer Science*, 6(3):251–259, 1996.

[27] Piotr Hofman, Richard Mayr, and Patrick Totzke. Decidability of weak simulation on one-counter nets. In *LICS*, 2013. Accepted.

[28] Piotr Hofman and Patrick Totzke. Approximating weak bisimilarity of basic parallel processes. In *DCM*, pages 99–113, 2012.

[29] Hans Hüttel. Silence is golden: Branching bisimilarity is decidable for context-free processes. In *CAV*, pages 2–12, 1991.

[30] Hans Hüttel. Undecidable equivalences for Basic Parallel Processes. In *TACS*, pages 454–464, 1994.

[31] Dung T. Huynh and Lu Tian. On deciding readiness and failure equivalences for processes. *Inf. Comput.*, 117(2):193–205, 1995.

[32] P. Jančar. Decidability questions for bismilarity of Petri nets and some related problems. In *STACS*, pages 581–592, 1994.

[33] Petr Jančar. Undecidability of bisimilarity for petri nets and some related problems. *Theor. Comput. Sci.*, 148(2):281–301, 1995.

[34] Petr Jančar. Bisimulation equivalence is decidable for one-counter processes. In *ICALP*, pages 549–559, 1997.

[35] Petr Jančar. Strong bisimilarity on Basic Parallel Processes is PSPACE-complete. In *LICS*, pages 218–227, 2003.

[36] Petr Jančar. Bisimilarity on basic process algebra is in 2-EXPTIME (an explicit proof). *CoRR*, abs/1207.2479, 2012.

[37] Petr Jančar and Martin Kot. Bisimilarity on normed Basic Parallel Processes can be decided in time $O(n^3)$. In *AVIS*, 2004.

[38] Petr Jančar, Antonín Kučera, and Faron Moller. Simulation and bisimulation over one-counter processes. In *Proceedings of the 17th Annual Symposium on Theoretical Aspects of Computer Science*, STACS, pages 334–345, 2000.

[39] Petr Jančar and Faron Moller. Simulation of one-counter nets via colouring. Technical report, Uppsala Computing Science, February 1999.

[40] Petr Jančar, Faron Moller, and Zdenek Sawa. Simulation problems for one-counter machines. In *SOFSEM*, pages 404–413, 1999.

[41] Stefan Kiefer. BPA bisimilarity is EXPTIME-hard. *Inf. Process. Lett.*, 113(4):101–106, 2013.

[42] Sławomir Lasota and Wojciech Rytter. Faster algorithm for bisimulation equivalence of normed context-free processes. In *MFCS*, pages 646–657, 2006.

[43] Richard Mayr. Process rewrite systems. *Electr. Notes Theor. Comput. Sci.*, 7:185–205, 1997.

[44] Richard Mayr. Undecidability of weak bisimulation equivalence for 1-counter processes. In *ICALP*, pages 570–583, 2003.

[45] Robin Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.

[46] A. Ponse and Scott A. Smolka, editors. *Handbook of Process Algebra*. Elsevier Science Inc., 2001.

[47] Géraud Sénizergues. The equivalence problem for deterministic pushdown automata is decidable. In *ICALP*, pages 671–681, 1997.

[48] Géraud Sénizergues. Complete formal systems for equivalence problems. *Theor. Comput. Sci.*, 231(2):309–334, 2000.

[49] Jirí Srba. Strong bisimilarity and regularity of Basic Parallel Processes is PSPACE-hard. In *STACS*, pages 535–546, 2002.

[50] Jirí Srba. *Roadmap of Infinite results*, volume Vol 2: Formal Models and Semantics. World Scientific Publishing Co., 2004.

[51] Jirí Srba. Beyond language equivalence on visibly pushdown automata. *Logical Methods in Computer Science*, 5(1), 2009.

[52] Colin Stirling. The joys of bisimulation. In *MFCS*, pages 142–151, 1998.

[53] Colin Stirling. Decidability of weak bisimilarity for a subset of Basic Parallel Processes. In *FoSSaCS*, pages 379–393, 2001.

[54] Jitka Stribrna. *Decidability and complexity of equivalences for simple process algebras*. PhD thesis, University of Edinburgh, 1998.

[55] Patrick Totzke. Trace inclusion for deterministic one-counter nets is NL-Complete. In *EXPRESS/SOS*, 2013. Submitted.

[56] Leslie G. Valiant and Michael S. Paterson. Deterministic one-counter automata. *J. Comput. Syst. Sci.*, 10(3):340–350, June 1975.