Roadmap of Infinite Results

Jiří Srba

Department of Computer Science, University of Aalborg
Selma Lagerlöfs Vej 300, 9220 Aalborg East, Denmark
srba@cs.aau.dk

16. 10. 2013

Abstract. This paper provides a comprehensive summary of equivalence checking results for infinite-state systems. References to the relevant papers will be updated continuously according to the development in the area. The most recent version of this document is available from the web-page http://users-cs.au.dk/srba/roadmap/.

1 Introduction

The growing interest in verification of infinite-state systems during the last decade led to the situation where many new results and novel approaches were invented. The first attempt to map the fundamental techniques and results for the equivalence checking problems was done by Moller in his overview paper "Infinite Results" [46], followed by the paper "More Infinite Results" [7] by Burkart and Esparza focusing on the model checking problems.

A large survey of equivalence and model checking techniques "Verification on Infinite Structures" appeared in the handbook of process algebra [4] due to Burkart, Caucal, Moller and Steffen. Yet another overview paper "Equivalence-Checking with Infinite-State Systems: Techniques and Results" [35] by Kučera and Jančar contains some recent techniques for simulation and bisimulation checking.

Although these comprehensive survey papers provide a valuable overview of proof techniques, the state-of-the-art advances so rapidly that many papers contain outdated information even before they are published.

The main objective of the presented work is to offer a complete overview of known decidability and complexity results for equivalence checking in the most studied classes of infinite-state systems. The most recent version of this document is available from the web-page http://users-cs.au.dk/srba/roadmap/ and we

would like to encourage you

to send notifications about improvements of the results presented in this paper to the author. Any such improvement will be promptly incorporated into this document and the updated version will appear at the URL mentioned above.

We hope that the overview we provide will stimulate further research on fundamental equivalence checking problems for infinite-state systems and it will eventually lead towards a definitive closing of all the gaps in the mosaic of infinite results.

2 Basic Definitions

In this section we introduce the classes of infinite-state processes by means of process rewrite systems (PRS). Process rewrite systems are an elegant and universal approach defined by Mayr [41] and they contain many of the formalisms studied in the context of equivalence checking.

Let $\mathcal{C}onst$ be a set of *process constants*. The classes of process expressions called 1 (process constants plus the empty process), $\mathcal{P}$ (parallel process expressions), $\mathcal{S}$ (sequential process expressions), and $\mathcal{G}$ (general process expressions) are defined by the following abstract syntax

$$
\begin{array}{rlll}
1: & E & ::= & \epsilon \mid X \\
\mathcal{P}: & E & ::= & \epsilon \mid X \mid E \| E \\
\mathcal{S}: & E & ::= & \epsilon \mid X \mid E.E \\
\mathcal{G}: & E & ::= & \epsilon \mid X \mid E \| E \mid E.E
\end{array}
$$

where 'ϵ' is the *empty process*, X ranges over $\mathcal{C}onst$, the operator '.' stands for a sequential composition and '$\|$' stands for a parallel composition. Obviously, $1 \subset \mathcal{S}$, $1 \subset \mathcal{P}$, $\mathcal{S} \subset \mathcal{G}$ and $\mathcal{P} \subset \mathcal{G}$. The classes $\mathcal{S}$ and $\mathcal{P}$ are incomparable and $\mathcal{S} \cap \mathcal{P} = 1$.

We do not distinguish between process expressions related by a *structural congruence*, which is the smallest congruence over process expressions such that the following laws hold:

- '.' is associative,
- '$\|$' is associative and commutative, and
- 'ϵ' is a unit for '.' and '$\|$'.

Let $\alpha, \beta \in \{1, \mathcal{S}, \mathcal{P}, \mathcal{G}\}$ such that $\alpha \subseteq \beta$ and let $\mathcal{A}ct$ be a set of *actions*. An (α, β)-PRS [41] is a finite set

$$\Delta \subseteq \big(\alpha \smallsetminus \{\epsilon\}\big) \times \mathcal{A}ct \times \beta$$

of *rewrite rules*, written $E \xrightarrow{a} F$ for $(E, a, F) \in \Delta$.

An (α, β)-PRS determines a labelled transition system where *states* are process expressions from the class β (modulo the structural congruence), $\mathcal{A}ct$ is the set of *labels*, and the *transition relation* is the least relation satisfying the following SOS rules (recall that '$\|$' is commutative):

$$\frac{(E \xrightarrow{a} E') \in \Delta}{E \xrightarrow{a} E'} \qquad \frac{E \xrightarrow{a} E'}{E.F \xrightarrow{a} E'.F} \qquad \frac{E \xrightarrow{a} E'}{E \| F \xrightarrow{a} E' \| F}$$

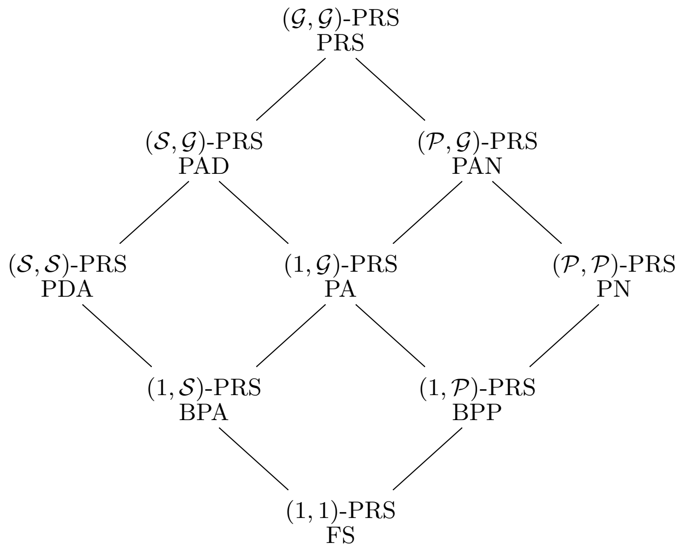

Fig. 1. Hierarchy of process rewrite systems

Many classes of infinite-state systems studied so far — e.g. basic process algebra (BPA), basic parallel processes (BPP), pushdown automata (PDA), Petri nets (PN) and process algebra (PA) — are contained in the hierarchy of process rewrite systems presented in Figure 1. This hierarchy is strict w.r.t. strong bisimilarity and we refer the reader to [41] for further discussions. It is worth mentioning that even the class $(\mathcal{G},\mathcal{G})$-PRS is not Turing powerful since e.g. the reachability problem (i.e. whether from a given process expression we can in finitely many steps reach another given process expression) remains decidable [41].

An (α,β)-*process* is a pair (P,Δ) where Δ is an (α,β)-PRS and $P \in \beta$ is a process expression.

An (α,β)-process (P,Δ) is *normed* iff for all $E \in \beta$ such that $P \longrightarrow^* E$ it is the case that $E \longrightarrow^* \epsilon$. In other words, from every reachable state in (P,Δ) there is a computation ending in the empty process 'ϵ'.

In some papers, the definition of normedness requires only the fact that from every reachable state there is a terminating computation, not necessarily ending in the empty process. However, e.g. for BPA, in order to achieve a reasonable notion of normedness, it is also assumed that every process constant used in the system can perform a transition, i.e., it has at least one rewrite rule associated to it. This alternative definition implies the notion of normedness introduced above. Moreover, the definition we gave becomes more interesting even for the models like PDA, PN, PAD, PAN and PRS where our notion of normedness guarantees deadlock freedom (here the empty process is not understood as a deadlock). This means that e.g. in the case of PDA the stack can always be emptied, and in the case of PN all tokens in places can be removed. Considering simply the possibility of termination without reaching the empty process usually does not restrict the power of the models sufficiently.

We will now introduce the notion of strong and weak bisimilarity [45, 48]. A binary relation R over process expressions is a *strong bisimulation* iff whenever $(E, F) \in R$ then for each $a \in \mathcal{A}ct$:

- if $E \xrightarrow{a} E'$ then $F \xrightarrow{a} F'$ for some F' such that $(E', F') \in R$
- if $F \xrightarrow{a} F'$ then $E \xrightarrow{a} E'$ for some E' such that $(E', F') \in R$.

Processes (P_1, Δ) and (P_2, Δ) are *strongly bisimilar*, written $(P_1, \Delta) \sim (P_2, \Delta)$ iff there is a strong bisimulation R such that $(P_1, P_2) \in R$. Given two processes (P_1, Δ_1) and (P_2, Δ_2) with disjoint sets of process constants contained in Δ_1 and Δ_2, we write $(P_1, \Delta_1) \sim (P_2, \Delta_2)$ iff $(P_1, \Delta_1 \cup \Delta_2) \sim (P_2, \Delta_1 \cup \Delta_2)$.

Assume that $\mathcal{A}ct$ contains a distinguished silent action τ. A *weak transition relation* is defined as follows: $\stackrel{a}{\Longrightarrow} \stackrel{\text{def}}{=} (\xrightarrow{\tau})^* \circ \xrightarrow{a} \circ (\xrightarrow{\tau})^*$ if $a \in \mathcal{A}ct \smallsetminus \{\tau\}$, and $\stackrel{a}{\Longrightarrow} \stackrel{\text{def}}{=} (\xrightarrow{\tau})^*$ if $a = \tau$.

A binary relation R over process expressions is a *weak bisimulation* iff whenever $(E, F) \in R$ then for each $a \in \mathcal{A}ct$:

- if $E \xrightarrow{a} E'$ then $F \stackrel{a}{\Longrightarrow} F'$ for some F' such that $(E', F') \in R$
- if $F \xrightarrow{a} F'$ then $E \stackrel{a}{\Longrightarrow} E'$ for some E' such that $(E', F') \in R$.

Processes (P_1, Δ) and (P_2, Δ) are *weakly bisimilar*, written $(P_1, \Delta) \approx (P_2, \Delta)$ iff there is a weak bisimulation R such that $(P_1, P_2) \in R$. Given two processes (P_1, Δ_1) and (P_2, Δ_2) with disjoint sets of process constants contained in Δ_1 and Δ_2, we write $(P_1, \Delta_1) \approx (P_2, \Delta_2)$ iff $(P_1, \Delta_1 \cup \Delta_2) \approx (P_2, \Delta_1 \cup \Delta_2)$.

3 Studied Problems

In this section we define the basic decidability problems studied in the area of equivalence checking of infinite-state systems.

Strong/Weak Bisimilarity $(\sim/\approx)$
The problem is to decide whether a pair of processes from a given class of systems is strongly/weakly bisimilar. Bisimilarity was originally introduced by Park [48] and Milner [45] and it is perhaps the most studied behavioural equivalence because of many pleasant properties it enjoys.

Strong/Weak Bisimilarity with Finite-State Systems ($\sim$ FS/$\approx$ FS)
The problem is to decide whether a process from a given class of systems is strongly/weakly bisimilar to a given finite-state process. Questions of this nature are interesting because they enable to relate a complex behaviour of infinite-state systems with their finite-state specifications. Moreover, recent development showed that many of these problems become computationally feasible and in some instances they are solvable even in polynomial time.

Strong/Weak Regularity ($\sim$ reg/$\approx$ reg)
The problem is to decide whether for a process from a given class of systems there exists a finite-state process such that the two processes are strongly/weakly bisimilar. The interest in regularity checking is based on the fact that for bisimilarity checking of finite-state processes we already have efficient polynomial time algorithms [25, 47]. A positive answer to the regularity question for a given pair E and F of infinite-state processes, together with the possibility of algorithmic construction of bisimilar finite-state processes provides an immediate answer to bisimilarity checking between E and F.

Remark 1. In the hierarchy of process rewrite systems there is a very close and obvious relationship between strong regularity checking of normed processes and the boundedness problem: a normed process is strongly regular if and only if it has only finitely many (on the syntactical level) reachable states. This property, however, does not hold for weak regularity.

4 Summary of Known Results

This section gives a summary of currently known decidability and complexity results for the systems in the PRS-hierarchy. We consider both unnormed and normed systems and for each decision problem we provide the best complexity bounds achieved so far.

Each box in the tables below contains the information whether the considered problem is decidable or not, and in the positive case we present the best known upper bound in the upper part of the box and lower bound in the lower part.

4.1 BPA (Basic Process Algebra)

	BPA	normed BPA
$\sim$	$\in$ 2-EXPTIME [5, 19] EXPTIME-hard [26]	$\in$ P [13], Rem. 2 P-hard [1]
$\approx$	? EXPTIME-hard [43]	? EXPTIME-hard [43]
$\sim$ FS	$\in$ P [37] P-hard [1]	$\in$ P [13] P-hard [1]
$\approx$ FS	$\in$ P [37] P-hard [1]	$\in$ P [37] P-hard [1]
$\sim$ reg	$\in$ 2-EXPTIME [6, 5] PSPACE-hard [54]	$\in$ NL [33] NL-hard [54]
$\approx$ reg	? EXPTIME-hard [43]	? NP-hard [56, 60]

Remark 2. Recently an $O(n^8 \mathit{polylog}\ n)$ has been described in [38].

4.2 BPP (Basic Parallel Processes)

	BPP	normed BPP
$\sim$	$\in$ PSPACE [18] PSPACE-hard [53]	$\in$ P [14], Rem. 3 P-hard [1]
$\approx$	?, Rem. 4 PSPACE-hard [56]	?, Rem. 4 PSPACE-hard [56]
$\sim$ FS	$\in$ P [28], Rem. 5 P-hard [1]	$\in$ P [14], Rem. 5 P-hard [1]
$\approx$ FS	$\in$ PSPACE [22] P-hard [1]	$\in$ P [37] P-hard [1]
$\sim$ reg	$\in$ PSPACE [27] PSPACE-hard [53]	$\in$ NL [33] NL-hard [53]
$\approx$ reg	? PSPACE-hard [56]	? PSPACE-hard [56]

Remark 3. Recently an $O(n^3)$-algorithm has been described in [24].

Remark 4. At INFINITY'02 Jančar conjectured [17] that the method later published in [18] might be used to show decidability of weak bisimilarity for BPP.

Remark 5. The complexity analysis provided in [28] gives a running time in $O(n^4)$.

4.3 PDA (Pushdown Automata)

	PDA	normed PDA
$\sim$	decidable [50], Rem. 6 nonelementary [2]	decidable [58] nonelementary [2]
$\approx$	undecidable [55]	undecidable [55]
$\sim$ FS	$\in$ PSPACE [36] PSPACE-hard [40]	$\in$ PSPACE [36] PSPACE-hard [40], Rem. 7
$\approx$ FS	$\in$ PSPACE [36] PSPACE-hard [40]	$\in$ PSPACE [36] PSPACE-hard [40], Rem. 8
$\sim$ reg	? EXPTIME-hard [36, 54], Rem. 9	$\in$ P [10], Rem. 10 NL-hard [54]
$\approx$ reg	? EXPTIME-hard [36, 54], Rem. 9	? EXPTIME-hard [36, 54], Rem. 8,9

Remark 6. Additional useful references concerning deterministic PDA are [51], [52] and [59].

Remark 7. The reduction from [40] (Theorem 8) uses unnormed processes but can be modified to work also for the normed case. An important observation is that the stack size of the PDA from Theorem 8 is bounded by the number of variables in the instance of quantified boolean formula from which the reduction is done.

Remark 8. Lemma 3 in [55] gives a polynomial time reduction from weak bisimilarity between two pushdown processes (and between a pushdown process and a finite-state process) to the normed instance of the problem. The reduction moreover preserves the property of being weakly regular.

Remark 9. In [36] a polynomial time reduction from the acceptance problem of alternating linear-bounded automata to strong bisimilarity of normed PDA is provided. Even though there are infinitely many reachable configurations in the constructed PDAs, one can observe that only a fixed part from the top of the stack is relevant for the construction. Hence it is possible to ensure that the PDAs are strongly regular and Theorem 2 from [54] can be applied.

Remark 10. Strong regularity of normed PDA is equivalent to the boundedness problem (Remark 1). Boundedness (even for unnormed PDA) is decidable in polynomial time using the fact that the set of all reachable configurations of a pushdown process is a regular language L [3] and a finite automaton A recognizing L can be constructed in polynomial time (see e.g. [10]). The check whether A generates a finite language can also be done in polynomial time.

4.4 PA (Process Algebra)

	PA	normed PA
$\sim$	? EXPTIME-hard [26]	2-NEXPTIME [12] P-hard [1]
$\approx$	undecidable [57]	? EXPTIME-hard [43]
$\sim$ FS	coNEXPTIME [11] P-hard [1]	coNEXPTIME [11] P-hard [1]
$\approx$ FS	decidable [22] P-hard [1]	decidable [22] P-hard [1]
$\sim$ reg	? PSPACE-hard [53]	$\in$ NL [34] NL-hard [53]
$\approx$ reg	? EXPTIME-hard [43]	? PSPACE-hard [56]

4.5 PN (Petri Nets)

	PN	normed PN
$\sim$	undecidable [16]	undecidable [16], Rem. 11
$\approx$	undecidable [16]	undecidable [16], Rem. 11
$\sim$ FS	decidable [23] EXPSPACE-hard [39], Rem. 12	decidable [23] P-hard [1]
$\approx$ FS	undecidable [20]	? EXPSPACE-hard [39], Rem. 12
$\sim$ reg	decidable [20] EXPSPACE-hard [39], Rem. 13	$\in$ EXPSPACE [49], Rem. 1 EXPSPACE-hard [39], Rem. 13
$\approx$ reg	undecidable [20]	? EXPSPACE-hard [39], Rem. 13

Remark 11. The technique for proving undecidability of strong bisimilarity for Petri nets from [16] can be slightly modified to ensure that the constructed nets are normed. Essentially, it is enough to add extra transitions which enable to remove all tokens from places. Moreover, whenever such an extra transition is fired the two nets are forced to become bisimilar.

Remark 12. The problem whether a given place p of a PN can ever become marked is EXPSPACE-hard (follows from Lipton's construction [39], for a more accessible proof see e.g. [9]). We can now easily see that this problem is reducible in polynomial time to strong nonbisimilarity between PN and FS. All transitions in a given Petri net N are assigned the same label 'a' and we add one more place q (initially marked) and an extra transition labelled by 'a' which takes a token from the place q and returns it back. Moreover we add another transition labelled by 'b' which can be fired whenever there is a token in the place p. Let P be a finite-state process defined by $P \xrightarrow{a} P$. The following property is immediate: the place p can become marked iff N is not strongly bisimilar to P. This reduction works also for normed PN and weak bisimilarity: we take our modified net N and for each its place we add one extra transition labelled by 'τ' such that the transition takes a token from the place and removes it. To the finite-state process P we add the rewrite rule $P \xrightarrow{\tau} \epsilon$. The net N is now normed and moreover it is weakly bisimilar to P iff the place p can never become marked.

Remark 13. Regularity of normed PN is equivalent to the boundedness problem (Remark 1). Boundedness of PN is decidable in EXPSPACE, more precisely in space $2^{cn \log n}$ for some constant c [49]. Moreover, the boundedness problem of normed PN is EXPSPACE-hard because it can be easily seen to be polynomially equivalent to the boundedness problem of general (unnormed) PN and this problem is EXPSPACE-hard [39] (see also [9]).

4.6 PAD

	PAD	normed PAD
$\sim$	? nonelementary [2]	? nonelementary [2]
$\approx$	undecidable [55]	undecidable [55]
$\sim$ FS	decidable [15] PSPACE-hard [40]	decidable [15] PSPACE-hard [40], Rem. 7
$\approx$ FS	decidable [22] PSPACE-hard [40]	decidable [22] PSPACE-hard [40], Rem. 8
$\sim$ reg	? EXPTIME-hard [36, 54], Rem. 9	decidable [44], Rem. 1 NL-hard [54]
$\approx$ reg	? EXPTIME-hard [36, 54], Rem. 9	? EXPTIME-hard [36, 54], Rem. 8,9

4.7 PAN

	PAN	normed PAN
$\sim$	undecidable [16]	undecidable [16], Rem. 11
$\approx$	undecidable [16]	undecidable [16], Rem. 11
$\sim$ FS	decidable [29] EXPSPACE-hard [39], Rem. 12	decidable [44], Rem. 1 P-hard [1]
$\approx$ FS	undecidable [20]	? EXPSPACE-hard [39], Rem. 12
$\sim$ reg	? EXPSPACE-hard [39], Rem. 13	decidable [44], Rem. 1 EXPSPACE-hard [39], Rem. 13
$\approx$ reg	undecidable [20]	? EXPSPACE-hard [39], Rem. 13

4.8 PRS (Process Rewrite Systems)

	PRS	normed PRS
$\sim$	undecidable [16]	undecidable [16], Rem. 11
$\approx$	undecidable [16]	undecidable [16], Rem. 11
$\sim$ FS	decidable [29] EXPSPACE-hard [39], Rem. 12	decidable [44], Rem. 1 PSPACE-hard [40], Rem. 7
$\approx$ FS	undecidable [20]	? EXPSPACE-hard [39], Rem. 12
$\sim$ reg	? EXPSPACE-hard [39], Rem. 13	decidable [44], Rem. 1 EXPSPACE-hard [39], Rem. 13
$\approx$ reg	undecidable [20]	? EXPSPACE-hard [39], Rem. 13

5 Weakly Extended Process Rewrite Systems

In [31], Křetínský, Řehák and Strejček extended the classes from the PRS-hierarchy by adding a finite control-unit with monotonic behaviour. Unlike the finite control-unit with unrestricted behaviour (which makes the model Turing powerful already for state-extended PA processes), the restriction on monotonic behaviour preserves the decidability of reachability. In this section we shall briefly introduce the extension and list the main results regarding the model.

Let Q be a finite set of *control-states* and $\mathcal{A}ct$ a set of *actions*. Let $\alpha, \beta \in \{1, \mathcal{S}, \mathcal{P}, \mathcal{G}\}$ such that $\alpha \subseteq \beta$. A *state-extended* (α, β)*-process rewrite system* ((α, β)-sePRS) is a finite set Δ of rewrite rules of the form $pE \xrightarrow{a} qF$ where $p, q \in Q$, $a \in \mathcal{A}ct$, $E \in \alpha \setminus \{\epsilon\}$ and $F \in \beta$.

Remark 14. Some of the state-extended classes have been studied before, e.g., $(1, \mathcal{P})$-sePRS is also known as the class of *multiset automata* or *parallel pushdown processes*.

A given (α, β)-sePRS Δ generates a labelled transition system $T(\Delta)$ where states are pairs of control-states and process expressions over β (modulo the previously introduced structural congruence), the set of actions is $\mathcal{A}ct$ and the transition relation is given by the following SOS rules (recall that '$\|$' is commutative).

$$\frac{(pE \xrightarrow{a} qF) \in \Delta}{pE \xrightarrow{a} qF} \qquad \frac{pE \xrightarrow{a} qE'}{p(E.F) \xrightarrow{a} q(E'.F)} \qquad \frac{pE \xrightarrow{a} qE'}{p(E\|F) \xrightarrow{a} q(E'\|F)}$$

An (α, β)-sePRS Δ is called a *weakly extended* (α, β)*-process rewrite system* ((α, β)-wPRS) iff there is a partial ordering $\leq$ on Q such that all rewrite rules $pE \xrightarrow{a} qF$ from Δ satisfy that $q \leq p$.

The main result (valid for all the classes from the wPRS hierarchy) is that the reachability problem is decidable [30]. The technique from [30] was later extended in [29], where the authors prove the decidability of the problem "given a formula of Hennessy-Milner logic, is there a reachable state satisfying the formula?". This in particular implies that strong bisimilarity between finite-state processes and $(\mathcal{G}, \mathcal{G})$-wPRS is also decidable [22]. On the other hand, weak bisimilarity on wBPA and wBPP is undecidable [32]. This refines the undecidability border of weak bisimilarity, which is still open both for BPA and BPP.

6 Equivalence Checking across the Models

A polynomial-time algorithm deciding strong bisimilarity between a normed BPA process and a normed BPP process has been recently presented in [21]. It improves the previously known exponential upper bound from [8].

Acknowledgments. I would like to thank Ivana Černá, Petr Jančar, Antonín Kučera, Mogens Nielsen and Paulo Borges Oliva for reading a draft of this paper. My special thanks go to Richard Mayr for his help, suggestions and numerous comments.

7 Record of Updates

- 16. 10. 2013: Michael Benedikt, Stefan Göller, Stefan Kiefer and Andrzej S. Murawski proved a nonelementary lower-bound of strong bisimilarity on (normed) pushdown automata [2], improving the previously known EXPTIME-hardness [36].
- 6. 6. 2013: The decidability result [15] for bisimulation checking between PA and FS has been updated to coNEXPTIME [11], a result achieved by Stefan Göller and Anthony Widjaja Lin.
- 6. 4. 2013: Petr Jančar published an explicit proof of 2-EXPTIME upper bound for strong bisimilarity on BPA [19].
- 29. 12. 2012: Stefan Kiefer improved the PSPACE-hardness [54] result for deciding strong bisimulation on BPA to EXPTIME-hardness. The result implies also EXPTIME-hardness of strong bisimilarity on PA.
- 10. 6. 2008: There is a new update in strong bisimilarity for normed BPA, which is decidable in $O(n^8 \mathit{polylog}\, n)$ thanks to Sławomir Lasota and Wojciech Rytter.
- 10. 6. 2008: A section about equivalence checking across the different models has been added and it includes now a polynomial time algorithm to decide strong bismilarity between normed BPA and normed BPP. Thanks to Petr Jančar, Martin Kot and Zdeněk Sawa.
- 21. 6. 2007: There are two new updates in the BPP section: strong bisimilarity between BPP and FS is decidable in time $O(n^4)$ thanks to Martin Kot and Zdeněk Sawa and strong regularity is decidable in PSPACE due to Martin Kot.
- 20. 2. 2006: A new section dealing with weakly extended process rewrite systems (Section 5) has been added. My thanks go to Vojtěch Řehák and Jan Strejček for providing me the references and for their comments.
- 10. 1. 2006: Mojmír Křetínský, Vojtěch Řehák and Jan Strejček showed in [29] the decidability of reachability of a state satisfying a given Hennessy–Milner formula for PRS extended with monotonic control-state unit. This implies, in particular, the decidability of strong bisimilarity between PRS and finite-state systems [22].
- 3. 6. 2004: Petr Jančar and Martin Kot published an $O(n^3)$ algorithm for strong bisimilarity of normed BPP [24].
- 9. 9. 2003: Petr Jančar proved in [18] that strong bisimilarity for unnormed BPP is in PSPACE.
- 9. 9. 2003: The EXPTIME hardness results from [42] were published in proceedings of EXPRESS'03 [43].
- 19. 12. 2002: Richard Mayr [42] improved the complexity lower bounds for weak bisimilarity of normed BPA and for weak regularity of unnormed BPA to EXPTIME.

References

[1] J. Balcazar, J. Gabarro, and M. Santha. Deciding bisimilarity is P-complete. *Formal Aspects of Computing*, 4(6A):638–648, 1992.

[2] M. Benedikt, S. Göller, S. Kiefer, and A.S. Murawski. Bisimilarity of pushdown automata is nonelementary. In *Proccedings of the 28th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS'13)*, pages 488–498. IEEE Computer Society, 2013.

[3] J.R. Büchi. Regular canonical systems. *Arch. Math. Logik u. Grundlagenforschung*, 6:91–111, 1964.

[4] O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification on infinite structures. In J. Bergstra, A. Ponse, and S. Smolka, editors, *Handbook of Process Algebra*, chapter 9, pages 545–623. Elsevier Science, 2001.

[5] O. Burkart, D. Caucal, and B. Steffen. An elementary decision procedure for arbitrary context-free processes. In *Proceedings of the 20th International Symposium on Mathematical Foundations of Computer Science (MFCS'95)*, volume 969 of *LNCS*, pages 423–433. Springer-Verlag, 1995.

[6] O. Burkart, D. Caucal, and B. Steffen. Bisimulation collapse and the process taxonomy. In *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*, volume 1119 of *LNCS*, pages 247–262. Springer-Verlag, 1996.

[7] O. Burkart and J. Esparza. More infinite results. *Bulletin of the European Association for Theoretical Computer Science*, 62:138–159, June 1997. Columns: Concurrency.

[8] Ivana Cerná, Mojmír Křetínský, and Antonín Kučera. Comparing expressibility of normed BPA and normed BPP processes. *Acta Informatica*, 36(3):233–256, 1999.

[9] J. Esparza. Decidability and complexity of Petri net problems — an introduction. In *Lectures on Petri Nets I: Basic Models*, volume 1491 of *LNCS*, pages 374–428. Springer-Verlag, 1998.

[10] J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithms for model checking pushdown systems. In *Proceedings of the 12th International Conference on Computer Aided Verification (CAV'00)*, volume 1855 of *LNCS*, pages 232–247. Springer-Verlag, 2000.

[11] S. Göller and A.W. Lin. The complexity of verifying ground tree rewrite systems. In *Proceedings of the 26th Annual IEEE Symposium on Logic in Computer Science (LICS'11)*, pages 279–288. IEEE Computer Society, 2011.

[12] Y. Hirshfeld and M. Jerrum. Bisimulation equivalence is decidable for normed process algebra. In *Proceedings of 26th International Colloquium on Automata, Languages and Programming (ICALP'99)*, volume 1644 of *LNCS*, pages 412–421. Springer-Verlag, 1999.

[13] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial algorithm for deciding bisimilarity of normed context-free processes. *Theoretical Computer Science*, 158(1–2):143–159, 1996.

[14] Y. Hirshfeld, M. Jerrum, and F. Moller. A polynomial-time algorithm for deciding bisimulation equivalence of normed basic parallel processes. *Mathematical Structures in Computer Science*, 6(3):251–259, 1996.

[15] P. Jančar and A. Kučera. Bisimilarity of processes with finite-state systems. In *Proceedings of the 2nd International Workshop on Verification of Infinite State Systems (INFINITY'97)*, volume 9 of *Electronic Notes in Theoretical Computer Science*, 1997.

[16] P. Jančar. Undecidability of bisimilarity for Petri nets and some related problems. *Theoretical Computer Science*, 148(2):281–301, 1995.

[17] P. Jančar. New results for bisimilarity on basic parallel processes. In *Proceedings of the 4th International Workshop on Verification of Infinite-State Systems (INFINITY'02)*, page 153. Technical report, Faculty of Informatics, Masaryk University Brno, FIMU-RS-2002-04, 2002. Short presentation.

[18] P. Jančar. Strong bisimilarity on basic parallel processes is PSPACE-complete. In *Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science (LICS'03)*, pages 218–227. IEEE Computer Society Press, 2003.

[19] P. Jančar. Bisimilarity on basic process algebra is in 2-exptime (an explicit proof). *Logical Methods in Computer Science*, 9(1):1 – 19, 2013.

[20] P. Jančar and J. Esparza. Deciding finiteness of Petri nets up to bisimulation. In *Proceedings of 23rd International Colloquium on Automata, Languages, and Programming (ICALP'96)*, volume 1099 of *LNCS*, pages 478–489. Springer-Verlag, 1996.

[21] P. Jančar, M. Kot, and Z. Sawa. Normed BPA vs. normed BPP revisited. In *to appear in Proceedings of CONCUR 2008*, LNCS. Springer-Verlag, 2008.

[22] P. Jančar, A. Kučera, and R. Mayr. Deciding bisimulation-like equivalences with finite-state processes. *Theoretical Computer Science*, 258(1–2):409–433, 2001.

[23] P. Jančar and F. Moller. Checking regular properties of Petri nets. In *Proceedings of the 6th International Conference on Concurrency Theory (CONCUR'95)*, volume 962 of *LNCS*, pages 348–362. Springer-Verlag, 1995.

[24] Petr Jančar and Martin Kot. Bisimilarity on normed basic parallel processes can be decided in time $o(n^3)$. In Ramesh Bharadwaj, editor, *Proceedings of the Third International Workshop on Automated Verification of Infinite-State Systems (AVIS'04)*, 2004.

[25] P.C. Kanellakis and S.A. Smolka. CCS expressions, finite state processes, and three problems of equivalence. *Information and Computation*, 86(1):43–68, 1990.

[26] S. Kiefer. BPA bisimilarity is EXPTIME-hard. *Information Processing Letters*, 113(4):101 – 106, 2013.

[27] M. Kot. Regularity of BPP is PSPACE-complete. In Václav Snášel, editor, *Proceedings of the 3rd Ph.D. Workshop of Faculty of Electrical En gineering and Computer Science (WOFEX'05)*, pages 393–398, Ostrava, 2005. VŠB-TUO FEECS.

[28] M. Kot and Z. Sawa. Bisimulation equivalence of a BPP and a finite-state system can be decided in polynomial time. In *Proceedings of the 6th International Workshop on Verification of Infinite-State Systems (INFINITY'04)*, volume 138 of *ENTCS*, pages 49–60. Elsevier Science Publishers, 2005.

[29] M. Křetínský, V. Řehák, and J. Strejček. Reachability of Hennessy–Milner properties for weakly extended PRS. In *Proceedings of the 25th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05)*, volume 3821 of *LNCS*, pages 213–224. Springer-Verlag, 2005.

[30] M. Křetínský, V. Řehák, and J. Strejček. Extended process rewrite systems: Expressiveness and reachability. In *Proceedings of the 15th International Conference on Concurrency Theory (CONCUR'04)*, volume 3170 of *LNCS*, pages 355–370. Springer-Verlag, 2004.

[31] M. Křetínský, V. Řehák, and J. Strejček. On extensions of process rewrite systems: Rewrite systems with weak finite-state unit. In *Proceedings of the 5th International Workshop on Verification of Infinite-State Systems (INFINITY'03)*, volume 98 of *ENTCS*, pages 75–88. Elsevier Science Publishers, 2004.

[32] M. Křetínský, V. Řehák, and J. Strejček. Refining the undecidability border of weak bisimilarity. In *Proceedings of the 7th International Workshop on Verification of Infinite-State Systems (INFINITY'05)*, volume 149 of *ENTCS*, pages 17–36. Elsevier Science Publishers, 2006.

[33] A. Kučera. Regularity is decidable for normed BPA and normed BPP processes in polynomial time. In *Proceedings of the 23th Annual Conference on Current Trends in Theory and Practice of Informatics (SOFSEM'96)*, volume 1175 of *LNCS*, pages 377–384. Springer-Verlag, 1996.

[34] A. Kučera. Regularity is decidable for normed PA processes in polynomial time. In *Proceedings of the 16th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'96)*, volume 1180 of *LNCS*, pages 111–122. Springer-Verlag, 1996.

[35] A. Kučera and P. Jančar. Equivalence-checking with infinite-state systems: Techniques and results. In *Proceedings of the 29th Annual Conference on Current Trends in Theory and Practice of Informatics (SOFSEM'02)*, volume 2540 of *LNCS*, pages 41–73. Springer-Verlag, 2002.

[36] A. Kučera and R. Mayr. On the complexity of semantic equivalences for pushdown automata and BPA. In *Proceedings of the 27th International Symposium on Mathematical Foundations of Computer Science (MFCS'02)*, volume 2420 of *LNCS*, pages 433–445. Springer-Verlag, 2002.

[37] A. Kučera and R. Mayr. Weak bisimilarity between finite-state systems and BPA or normed BPP is decidable in polynomial time. *Theoretical Computer Science*, 270:667–700, 2002.

[38] Sławomir Lasota and Wojciech Rytter. Faster algorithm for bisimulation equivalence of normed context-free processes. In *Proceedings of 31st International Symposium on Mathematical Foundations of Computer Science (MFCS'06)*, volume 4162 of *LNCS*, pages 646–657. Springer-Verlag, 2006.

[39] R. Lipton. The reachability problem requires exponential space. Technical Report 62, Department of Computer Science, Yale University, 1976.

[40] R. Mayr. On the complexity of bisimulation problems for pushdown automata. In *Proceedings of IFIP International Conference on Theoretical Computer Science (IFIP TCS'00)*, volume 1872 of *LNCS*. Springer-Verlag, 2000.

[41] R. Mayr. Process rewrite systems. *Information and Computation*, 156(1):264–286, 2000.

[42] R. Mayr. Weak bisimilarity and regularity of BPA is EXPTIME-hard. Technical Report No. 181, Department of Computer Science, Freiburg University, Germany, 2002.

[43] R. Mayr. Weak bisimilarity and regularity of BPA is EXPTIME-hard. In *Proccedings of the 10th International Workshop on Expressiveness in Concurrency (EXPRESS'03)*, pages 160–143, 2003. To appear in ENTCS.

[44] R. Mayr and M. Rusinowitch. Reachability is decidable for ground AC rewrite systems. In *Proceedings of 3rd International Workshop on Verification of Infinite State Systems (INFINITY'98)*, volume TUM-I9825 of *Technical Report, Technishe Universität München*, pages 53–64, 1998.

[45] R. Milner. A calculus of communicating systems. *LNCS*, 92, 1980.

[46] F. Moller. Infinite results. In *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*, volume 1119 of *LNCS*, pages 195–216. Springer-Verlag, 1996.

[47] R. Paige and R. Tarjan. Three partition refinement algorithms. *SIAM Journal of Computing*, 16(6):973–989, 1987.

[48] D.M.R. Park. Concurrency and automata on infinite sequences. In *Proceedings of the 5th GI Conference*, volume 104 of *LNCS*, pages 167–183. Springer-Verlag, 1981.

[49] C. Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6(2):223–231, 1978.

[50] G. Sénizergues. Decidability of bisimulation equivalence for equational graphs of finite out-degree. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science(FOCS'98)*, pages 120–129. IEEE Computer Society, 1998.

[51] G. Sénizergues. L(A)=L(B)? Decidability results from complete formal systems. *Theoretical Computer Science*, 251(1–2):1–166, 2001.

[52] G. Sénizergues. L(A)=L(B)? A simplified decidability proof. *Theoretical Computer Science*, 281(1–2):555–608, 2002.

[53] J. Srba. Strong bisimilarity and regularity of basic parallel processes is PSPACE-hard. In *Proceedings of the 19th International Symposium on Theoretical Aspects of Computer Science (STACS'02)*, volume 2285 of *LNCS*, pages 535–546. Springer-Verlag, 2002.

[54] J. Srba. Strong bisimilarity and regularity of basic process algebra is PSPACE-hard. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming (ICALP'02)*, volume 2380 of *LNCS*, pages 716–727. Springer-Verlag, 2002.

[55] J. Srba. Undecidability of weak bisimilarity for pushdown processes. In *Proceedings of the 13th International Conference on Concurrency Theory (CONCUR'02)*, volume 2421 of *LNCS*, pages 579–593. Springer-Verlag, 2002.

[56] J. Srba. Complexity of weak bisimilarity and regularity for BPA and BPP. *Mathematical Structures in Computer Science*, 13:567–587, 2003.

[57] J. Srba. Undecidability of weak bisimilarity for PA-processes. In *Proceedings of the 6th International Conference on Developments in Laguage Theory (DLT'02)*, volume 2450 of *LNCS*, pages 197–208. Springer-Verlag, 2003.

[58] C. Stirling. Decidability of bisimulation equivalence for normed pushdown processes. *Theoretical Computer Science*, 195(2):113–131, 1998.

[59] C. Stirling. Decidability of DPDA equivalence. *Theoretical Computer Science*, 255(1–2):1–31, 2001.

[60] J. Stříbrná. Hardness results for weak bisimilarity of simple process algebras. In *Proceedings of the MFCS'98 Workshop on Concurrency*, volume 18 of *Electronic Notes in Theoretical Computer Science*. Springer-Verlag, 1998.