Normed Processes, Unique Decomposition, and Complexity of Bisimulation Equivalences

Sibylle Fröschle[1,3]

Department für Informatik
Universität Oldenburg
26111 Oldenburg, Germany

Sławomir Lasota[2,4]

Institute of Informatics
Warsaw University
02–097 Warszawa, Banacha 2, Poland

Abstract

We propose a decision procedure for a general class of normed commutative process rewrite systems and their induced bisimulation equivalences. Our technique is inspired by the polynomial-time algorithm for strong bisimilarity on normed Basic Parallel Processes (BPP), developed by Hirshfeld, Jerrum and Moller. As part of our framework we present a generic unique decomposition result, which we obtain by building on a characterization by Luttik and van Oostrom. We apply our general technique to derive polynomial-time algorithms for strong bisimilarity on normed BPP with communication and for distributed bisimilarity on all BPP with communication. Moreover, our technique yields a PSPACE upper bound for weak and branching bisimilarity on totally normed BPP.

Keywords: Bisimulation equivalence, distributed bisimulation, weak bisimulation, branching bisimulation, Basic Parallel Processes, unique decomposition.

1 Introduction

The idea of unique prime decomposition is relevant in process theory: given a behavioural equivalence one asks whether the commutative monoid of the equivalence classes of processes, with parallel composition as binary operation, has unique decomposition. This means that every element can be uniquely expressed as a composition of primes; an element is prime if it is not the composition of two non-trivial elements.

[1] This work was partially done while the first author was a postdoc at Warsaw University, funded by the EU Research Training Network GAMES.

[2] Partially supported by the Polish government grant N206 008 32/0810 and by the EU Research Training Network GAMES.

[3] Email: froeschle@informatik.uni-oldenburg.de

[4] Email: sl@mimuw.edu.pl

Unique decomposition results in process theory have helped to construct decision procedures for bisimulation equivalences on various process classes. A classical example is the algorithm of Hirshfeld et al. [6,2], which proves that strong bisimilarity on normed Basic Parallel Processes (BPP) is polynomial-time decidable. Using this algorithm as a subroutine, Lasota has shown that distributed bisimilarity on BPP is also polynomial-time decidable [11]. Hirshfeld et al.'s technique does not extend to strong bisimilarity on the full BPP class. Indeed, it has turned out that this problem is PSPACE-complete [15,8]. The PSPACE decision procedure uses a special technique developed by Jančar, which can also be applied to obtain polynomial-time algorithms for strong bisimilarity on normed BPP and distributed bisimilarity on BPP [9,10].

Weak bisimilarity on BPP is PSPACE-hard [14], and it is not even known whether the notion is decidable; this is a long-standing open problem. For normed BPP, weak bisimilarity remains PSPACE-hard but decidability has been achieved for slightly more restricted classes [5,16]; one of them is the class of totally normed BPP [5]. However, both algorithms seem to have non-elementary complexity. Branching bisimilarity [17], which like weak bisimilarity reflects a notion of observable behaviour, is also decidable on totally normed BPP [5].

Two directions seem important to gear up the techniques towards realistic systems: one is to cover process classes with communication; the second is to obtain better techniques for observational equivalences. For BPP_τ, which extends BPP with CCS-style communication, strong bisimilarity restricted to the normed case as well as distributed bisimilarity have been proved decidable [3]. The algorithms are based on unique decomposition results but they both run in exponential time. Jančar's technique is not suited to processes with communication, and it appears not be applicable to observable semantics either. Thus, we were motivated to take another look at unique decomposition. In this paper we will present a general technique, based on unique decomposition, that applies to a wide range of process semantics.

Our contributions are as follows. (1) We define a general class of *commutative rewrite systems* (*CRS'*) and a general definition of when a CRS is *normed*. With each CRS we associate a notion of bisimilarity. (2) We demonstrate that a wide range of semantics can be captured within our framework. In particular, we show that each of the following equivalences can be expressed as the bisimilarity of a normed CRS: strong bisimilarity on normed BPP_τ, distributed bisimilarity on all BPP_τ, weak and branching bisimilarity on totally normed BPP. (3) We prove that the notions of bisimilarity induced by normed CRS', and hence our four concrete equivalences, satisfy unique decomposition. To our knowledge, no unique decomposition result for weak or branching bisimilarity on a BPP class has been achieved before. (4) We exhibit a general procedure which, given an *effectively represented* normed CRS, decides its associated notion of bisimilarity. (5) We instantiate this general technique to obtain the following concrete results: strong bisimilarity on normed BPP_τ and distributed bisimilarity on all BPP_τ are polynomial-time decidable; weak and branching bisimilarity on totally normed BPP are decidable in PSPACE. To our knowledge all of these results are new. The result on weak bisimilarity is particularly interesting: no elementary upper bound has been achieved before for this equivalence on a reasonably large infinite-state class.

It is standard to define classes of infinite-state processes in terms of rewrite systems [2,13]. For example, BPP are captured by what is commonly referred to as $(1, \mathcal{P})$-PRS [13]. Our notion of CRS generalizes $(1, \mathcal{P})$-PRS in two ways. Firstly, we allow processes to communicate; the type of communication will be specified by means of a synchronization algebra, a concept that originates from the semantics of concurrency [18]. Secondly, we allow a process to evolve into a tuple of processes rather than only one. This ensures we can capture generalizations of transition systems such as that used by distributed bisimilarity but it is also crucial to capture branching bisimilarity.

Our unique decomposition result is achieved with the help of an important tool of Luttik and van Oostrom [12]: a partial commutative monoid has unique decomposition iff it has a *decomposition order*. The concept of decomposition order captures the algebraic properties that typically underly unique decomposition proofs in process theory. Each normed CRS induces a notion of *norm-reducing* bisimulation. We will show that congruences which are norm-reducing bisimulations have a natural decomposition order, and hence by [12] they satisfy unique decomposition. Not only will this result imply unique decomposition for bisimilarity but it will also provide the backbone of our general decision procedure.

Our decision procedure is an extension of Hirshfeld et al.'s technique for strong bisimilarity on normed BPP [6,2]. In their algorithm bisimilarity is approximated from above, in a small number of steps, by a sequence of norm-preserving congruences which have unique decomposition. We will make prominent what is underlying but hidden in [6]: each approximation is obtained as the greatest norm-reducing bisimulation that contains a refinement of the previous approximation. Combined with our general unique decomposition result this makes the technique applicable to a wide range of processes. Furthermore, we generalize [6] in that our notion of norm-reducing transition is non-strict.

It is interesting that we can capture the four concrete cases as instantiations of the same type of normed CRS. On the one hand, this makes concrete the connection between strong bisimilarity on normed BPP and distributed bisimilarity on all BPP that has already become apparent from the similarity in their decision procedures [11,9,10]. On the other hand, it shows that this connection extends to BPP_τ. Finally, and perhaps most interestingly, it highlights a new connection between observable transitions and communication. Altogether, our framework brings to light structural similarities between a priori very different notions. Thus, we hope our general technique will be applicable to a wide range of process semantics beyond those studied in this paper.

The remainder of the paper is structured according to our contributions. In Section 2 we introduce our setting of (normed) CRS'. In is intentionally presented in an abstract and general way. Four concrete instantiations of this setting, corresponding to the above-mentioned four bisimulation equivalences, are given in detail in the following Section 3. For the sake of clarity of presentation, we prefer to keep distinct the abstract definitions, and their concrete examples. However, simultaneously with reading Section 2 the Reader is encouraged to inspect Section 3 to get some concrete intuitions about the necessarily abstract notions. Section 4 prepares a tool, the unique decomposition theorem, which we use later. In Section 5, being the core part of the paper, we outline the algorithm. It is again presented in a generic, abstract setting of CRS'. Then in Section 6 we explain how this general framework applies to the four bisimulation equivalences from Section 3. The last section contains conclusions and future plans.

2 Normed Commutative Rewrite Systems

For technical convenience we assume a special action $*$, the *idle action*, which can always happen. A *synchronization algebra* is a pair Act = (Act, $\bullet$) where Act is a set of actions such that $* \notin \text{Act}$, and $\bullet : (\text{Act} \cup \{*\})^2 \to \text{Act} \cup \{*\}$ is a binary, commutative, associative partial operation which satisfies:

(S1) $a \bullet b = *$ iff $a = * = b$;

(S2) $a \bullet * = * \bullet a = a$.

Axiom (S1) is standard. By imposing (S2) we restrict our attention to communication schemes where all actions can occur asynchronously. Typically, we let $a, b, \ldots$ range over actions including $*$.

2.1 Commutative Rewrite Systems

By $V^{\otimes}$ we denote the free commutative monoid over a finite set V. Its elements are multisets of elements from V.

A *commutative rewrite system* (*CRS*) is a triple $P = (V, \text{Act}, \Gamma)$ where

- V is a finite set of variables,
- Act is a finite synchronization algebra, and
- Γ is a finitely presentable set of rewrite rules of the form $(X, a, \alpha_1, \ldots, \alpha_m)$ where $X \in V$, $a \in \text{Act}$, and $\alpha_i \in V^{\otimes}$. $m \geq 1$ is fixed throughout Γ.

Γ is assumed to be *finitely presentable* but not necessarily finite; in fact in the most interesting cases Γ will be infinite or exponential wrt. the actual size of presentation of Γ. We do not assume here anything about the way Γ is presented; this will be made clear in each concrete case. Typically, a presentation will be a finite BPP definition in normal form [3,2]. All the complexity considerations in the following sections will be with respect to the size of the presentation of Γ.

Variables are thought to represent *elementary processes*, and multisets of variables are interpreted as parallel compositions of elementary processes. The elements of $V^{\otimes}$ are thus thought to represent the *processes* of P. Let X, $Y, \ldots$ range over V, and α, $\beta, \ldots$ over $V^{\otimes}$. We write $\alpha\beta$ for the composition of α and β. The *empty process* (empty multiset) is denoted by ϵ.

The rewrite rules of Γ specify the behaviour of elementary processes: if $(X, a, \alpha_1, \ldots, \alpha_m) \in \Gamma$ then X can perform action a and thereby evolve into the tuple of processes $(\alpha_1, \ldots, \alpha_m)$. The case $m > 1$ will turn out to be useful for expressing, e.g., distributed and branching bisimilarity. We impose the natural assumption that the idle transitions are determined, in the sense that for all $X \in V$:

(C1) there exists exactly one rule for the idle action $(X, *, \alpha_1, \ldots, \alpha_m) \in \Gamma$, and for this rule $\alpha_m = X$.

Note that all derivatives may change due to an idle transition, except the last one. The rules for the idle action will be called *idle rules* in the sequel.

We define a *transition relation* for all processes inductively as follows:

(i) if $(X, a, \alpha_1, \dots, \alpha_m) \in \Gamma$ then $X \overset{a}{\twoheadrightarrow} (\alpha_1, \dots \alpha_m)$;

(ii) if $\alpha \overset{a}{\twoheadrightarrow} (\alpha_1, \dots, \alpha_m)$, $\beta \overset{b}{\twoheadrightarrow} (\beta_1, \dots, \beta_m)$, and $a \bullet b$ is defined then we have $\alpha\beta \overset{a \bullet b}{\twoheadrightarrow} (\alpha_1\beta_1, \dots, \alpha_m\beta_m)$.

From the m-ary steps we derive single-derivative steps as follows: $\alpha \xrightarrow{a} \alpha'$ iff $\alpha \overset{a}{\twoheadrightarrow} \alpha_1, \dots, \alpha_m$ with $\alpha_m = \alpha'$. This means we single out the mth derivate to play a special role. The transition relation $\alpha \xrightarrow{a} \alpha'$ is extended to finite words $w \in \mathrm{Act}^*$, written $\alpha \xrightarrow{w} \alpha'$, in a standard way.

Fact 2.1 *Let a, b, and c range over* $\mathrm{Act} \cup \{*\}$.

(i) $\alpha \xrightarrow{*} \alpha$.

(ii) *If $\alpha \xrightarrow{a} \alpha'$, $\beta \xrightarrow{b} \beta'$, and $a \bullet b$ is defined then $\alpha\beta \xrightarrow{a \bullet b} \alpha'\beta'$.*

(iii) *If $\alpha\beta \xrightarrow{c} \gamma$ then there are some a, b, α', β' such that $c = a \bullet b$, $\gamma = \alpha'\beta'$, $\alpha \xrightarrow{a} \alpha'$, and $\beta \xrightarrow{b} \beta'$.*

2.2 Normed Commutative Rewrite Systems

The *norm* of a process α, denoted by $|\alpha|$, is the smallest weight necessary to reach from α the empty process by a sequence of transitions. Formally, we define:

$$|\alpha| = \min\{W(w) : \alpha \xrightarrow{w} \epsilon,\ w \in \mathrm{Act}^*\} \quad \text{if } \alpha \xrightarrow{w} \epsilon \text{ for some } w \in \mathrm{Act}^*,$$
$$|\alpha| = \infty \quad \text{otherwise.}$$

Note that we allow $|\alpha| = 0$ even if $\alpha \neq \varepsilon$, whereas in normed and totally normed BPP (cf. Section 3) $\alpha \neq \varepsilon$ always implies $|\alpha| > 0$. Norm is additive wrt. composition of processes:

Proposition 2.2 *if α and β have finite norms then $|\alpha\beta| = |\alpha| + |\beta|$.*

Proof. If $\alpha \xrightarrow{w} \epsilon$ and $\beta \xrightarrow{v} \epsilon$ then we can derive $\alpha\beta \xrightarrow{wv} \epsilon$ by applying Axiom (S2), Fact 2.1(i), and Fact 2.1(ii). Hence, $|\alpha\beta| \leq |\alpha| + |\beta|$. On the other hand, if $\alpha\beta \xrightarrow{w} \epsilon$ then by Fact 2.1(iii) we obtain $\alpha \xrightarrow{u} \epsilon$ and $\beta \xrightarrow{v} \epsilon$ for some u, v such that $W(u) + W(v) = W(w)$ by Axiom (S4). Thus, $|\alpha\beta| \geq |\alpha| + |\beta|$. □

A *weighted synchronization algebra* is a triple $(Act, \bullet, W)$ where $(Act, \bullet)$ is a synchronization algebra and $W : Act \cup \{*\} \to \mathbb{N}_0$ is a *weight function* such that:

(S3) $W(*) = 0$;

(S4) $a \bullet b = c$ implies $W(c) = W(a) + W(b)$.

Axiom (S3) ensures the idle action has weight 0. (S4) expresses that the weight of an action does not depend on how it arises from synchronization. The weight of a sequence $w = a_1 \dots a_k$ is $W(w) = W(a_1) + \dots + W(a_k)$.

A *normed CRS* is a triple $(V, \mathrm{Act}, \Gamma)$ satisfying the definition of CRS, such that Act is a *weighted* synchronization algebra, and the following holds:

(N1) each process $\alpha \neq \epsilon$ has finite and positive norm: $0 < |\alpha| < \infty$.

We will impose a second condition (N2). For its formulation we need some more definitions. Let $(V, \mathrm{Act}, \Gamma)$ be a normed CRS as defined so far. A transition $\alpha \xrightarrow{a} \alpha'$ is *norm-reducing*, denoted by $\alpha \xrightarrow{a}_{\mathrm{n-r}} \alpha'$, iff it can be extended to a sequence from α to ϵ of minimal weight: $\alpha \xrightarrow{a}_{\mathrm{n-r}} \alpha'$ iff $\alpha \xrightarrow{a} \alpha' \xrightarrow{w} \epsilon$ for some $w \in \mathrm{Act}^*$ with $W(aw) = |\alpha|$. We extend norm-reducing transitions to sequences $v \in \mathrm{Act}^*$ as one could expect: $\alpha \xrightarrow{v}_{\mathrm{n-r}} \alpha'$ iff $\alpha \xrightarrow{v} \alpha' \xrightarrow{w} \epsilon$ for some $w \in \mathrm{Act}^*$ with $W(vw) = |\alpha|$. Due to additivity of norm we obtain the analogue of Fact 2.1 for $\longrightarrow_{\mathrm{n-r}}$.

Without loss of generality we may assume that the variables of V are given as a sequence $X_1, \ldots, X_k$, ordered by non-decreasing norm: $|X_1| \leq |X_2| \leq \ldots \leq |X_k|$.

Remark 2.3 Norm-reducing transitions are non-strict in that they do not lead to a strict decrease in norm: if $\alpha \xrightarrow{a}_{\mathrm{n-r}} \alpha'$ then $|\alpha'| \leq |\alpha|$, but not necessarily $|\alpha'| < |\alpha|$. This is a more general setting than usually considered.

If norm-reducing transitions were strict then a norm-reducing transition from a variable X_i would always lead to a multiset of variables of index strictly smaller than i. We will assume that every normed CRS satisfies a weakening of this property:

(N2) $X_i \xrightarrow{a}_{\mathrm{n-r}} \alpha$ implies $\alpha \in \{X_1, \ldots, X_i\}^{\otimes}$.

(N2) cannot be derived from (N1) in general, but it is satisfied immediately or modulo bisimilarity by all the concrete normed CRS' we will study. With (N2) we obtain the crucial properties of norm-reducing transitions.

Proposition 2.4 *If* $X_i \xrightarrow{a}_{\mathrm{n-r}} \alpha$ *then* $\alpha \in \{X_1, \ldots, X_{i-1}\}^{\otimes}$ *or* $\alpha = X_i$.

This is easily deduced from (N2) because $|\alpha| \leq |X_i|$.

2.3 Bisimulations and Norm-Reducing Bisimulations

In this section we work relative to a CRS $P = (V, \mathrm{Act}, \Gamma)$. We naturally associate a notion of bisimilarity with P, defined as follows.

Definition 2.5 Given a relation $R \subseteq V^{\otimes} \times V^{\otimes}$, we say a pair $(\alpha, \beta) \in V^{\otimes} \times V^{\otimes}$ *satisfies expansion in* R iff:

(i) whenever $\alpha \overset{a}{\twoheadrightarrow} (\alpha_1, \ldots, \alpha_m)$ for some $a, \alpha_1, \ldots, \alpha_m$ then there exist $\beta_1, \ldots, \beta_m$ such that $\beta \overset{a}{\twoheadrightarrow} (\beta_1, \ldots, \beta_m)$ and all pairs $(\alpha_1, \beta_1), \ldots, (\alpha_m, \beta_m)$ are in R;

(ii) the symmetric condition holds.

We denote the set of pairs (α, β) that satisfy expansion in R by $\exp(R)$. A relation $R \subseteq V^{\otimes} \times V^{\otimes}$ is a *bisimulation* iff each pair of R satisfies expansion in R, in other words, iff $R \subseteq \exp(R)$. We denote the greatest bisimulation by $\approx$ and refer to it as *bisimilarity*. If P is not clear from the context we also use the terms P-bisimulation and P-bisimilarity; the latter is denoted by $\approx_P$.

A *congruence* is any equivalence $\equiv$ that is compositional: if $\alpha \equiv \alpha'$ and $\beta \equiv \beta'$ then $\alpha\beta \equiv \alpha'\beta'$. The following fact is proved in a routine way:

Proposition 2.6 *(i) Bisimilarity is a congruence. (ii) If $\equiv$ is a congruence then the relation* $\exp(\equiv)$ *is a congruence too.*

Assume P to be normed. Then there is also a notion of *norm-reducing* bisimulation associated with P. There is an important difference between bisimulations and norm-reducing bisimulations: the former are defined over m-ary steps $\twoheadrightarrow$ while the latter over single-derivative steps $\longrightarrow$ only.

Definition 2.7 Given a relation $R \subseteq V^{\otimes} \times V^{\otimes}$, we say a pair $(\alpha, \beta) \in V^{\otimes} \times V^{\otimes}$ satisfies *norm-reducing expansion* in R, written $(\alpha, \beta) \in \mathrm{n-r-exp}(R)$, iff:

(i) whenever $\alpha \xrightarrow{a}_{\mathrm{n-r}} \alpha'$ for some a, α' then there exists β' such that $\beta \xrightarrow{a} \beta'$ and $(\alpha', \beta') \in R$;

(ii) the symmetric condition holds.

A relation $R \subseteq V^{\otimes} \times V^{\otimes}$ is a *norm-reducing bisimulation* iff $R \subseteq \mathrm{n-r-exp}(R)$.

Note an asymmetry in the definition: the response to the norm-reducing moves are unrestricted. Due to this, each bisimulation is a norm-reducing bisimulation. We say that a relation $R \subseteq V^{\otimes} \times V^{\otimes}$ is *norm-preserving* iff $|\alpha| = |\beta|$ for each $(\alpha, \beta) \in R$.

Proposition 2.8 *Each norm-reducing bisimulation, and thus each bisimulation, is norm-preserving.*

Proof. Let R be a norm-reducing bisimulation and let $(\alpha, \beta) \in R$. Assume $|\alpha| < |\beta|$ and consider a sequence $\alpha \xrightarrow{w} \epsilon$ of minimal weight. Observe that $\alpha \xrightarrow{w} \epsilon$ is norm-reducing, $\alpha \xrightarrow{w}_{\mathrm{n-r}} \epsilon$. There must be some β' with $\beta \xrightarrow{w} \beta'$ and $(\epsilon, \beta') \in R$. But for any such β' we have necessarily $|\beta'| > 0$ and consequently there is a transition $\beta' \xrightarrow{a}_{\mathrm{n-r}} \beta''$ for some a and β'', to which ϵ has no response. Hence, necessarily $|\alpha| = |\beta|$. □

As a consequence we obtain: in each norm-reducing bisimulation, every (necessarily norm-reducing) transition will be matched by a transition that is *norm-reducing* itself. We will assume this property in the sequel without further mentioning.

Given a congruence $\equiv$, the greatest norm-reducing bisimulation contained in $\equiv$ always exists, and is given by the set-theoretic union of all norm-reducing bisimulations contained in $\equiv$. We denote this relation by $\mathrm{gnrb}(\equiv)$.

Proposition 2.9 *Given a congruence $\equiv$,* $\mathrm{gnrb}(\equiv)$ *is a congruence.*

Proof. That $\mathrm{gnrb}(\equiv)$ is an equivalence follows easily from $\equiv$ being an equivalence. Compositionality of $\mathrm{gnrb}(\equiv)$ is easily proved by showing that the relation $\{(\alpha\beta, \alpha'\beta') : (\alpha, \alpha') \in \mathrm{gnrb}(\equiv), (\beta, \beta') \in \mathrm{gnrb}(\equiv)\}$ is a norm-reducing bisimulation contained in $\equiv$. □

$\mathrm{gnrb}(\equiv)$ has a fix-point characterization which we will need later on:

Proposition 2.10 *Let $\equiv$ be a congruence.* $(\alpha, \beta) \in \mathrm{gnrb}(\equiv)$ *iff $\alpha \equiv \beta$ and* $(\alpha, \beta) \in \mathrm{n-r-exp}(\mathrm{gnrb}(\equiv))$.

Proof. The only-if implication is obvious. For the opposite implication, we show that the relation $\{(\alpha, \beta) : \alpha \equiv \beta \text{ and } (\alpha, \beta) \in \mathrm{n-r-exp(gnrb(\equiv))}\}$ is a norm-reducing bisimulation contained in $\equiv$, using the only-if implication. $\square$

3 Concrete Normed Commutative Rewrite Systems

Finally, we will get to know concrete CRS', and see how their associated notions of bisimilarity coincide with the well-known equivalences: (i) strong bisimilarity on normed BPP_τ, (ii) distributed bisimilarity on all BPP_τ, (iii) weak bisimilarity on totally normed BPP, and (iv) branching bisimilarity on totally normed BPP. We cannot be completely self-contained here; in particular we refer the reader to the literature for the definitions [2,3,5].

In the following, we will need *variables*, ranged over by $X, Y, \ldots$, and *actions* Act, ranged over by $a, b, \ldots$. We will use BPP definitions Δ containing a finite number of rules, typically of the form $X \xrightarrow{a} \alpha$, where α is a finite multiset of variables; in short we write $\Delta = \{X \xrightarrow{a} \alpha\}$. We always assume that if a variable occurs on the right side of a rule it will also appear on the left side of some rule.

In each case below, Δ will be a finite presentation of (will induce) a CRS P; and bisimilarity $\approx_P$ will correspond to the bisimulation equivalence under consideration.

3.1 BPP_τ under Interleaving Semantics

Assume a set of complementary actions $\overline{Act} = \bar{a}, \bar{b}, \bar{c}, \ldots$. Let τ be a special element not contained in Act, the *silent* action. BPP_τ uses CCS-style communication where two complementary actions can synchronize forming τ. The silent action can also occur by itself and any action can occur asynchronously. The corresponding synchronization algebra, Act_{CCS}, is then $(Act \cup \overline{Act} \cup \{\tau\}, \bullet)$ where $\bullet$ is defined by the table on the right. We extend Act_{CCS} by a weight function W, defined by: $W(*) = 0$, $W(a) = 1$ for all $a \in Act \cup \overline{Act}$, and $W(\tau) = 2$.

	$*$	a	$\bar{a}$	b	$\bar{b}$	$\ldots$	τ
$*$	$*$	a	$\bar{a}$	b	$\bar{b}$	$\ldots$	τ
a	a		τ			$\ldots$	
$\bar{a}$	$\bar{a}$	τ				$\ldots$	
b	b				τ	$\ldots$	
$\bar{b}$	$\bar{b}$			τ		$\ldots$	
.	.	.	.	.	.	$\ldots$	.
τ	τ					$\ldots$	

A BPP_τ definition in normal form is a finite set $\Delta = \{X \xrightarrow{a} \alpha\}$ where $a \in Act_{CCS}$. Δ induces a CRS $P = (V, \mathrm{Act}, \Gamma)$ as follows: V and Act are derived from Δ and Act_{CCS} in the obvious way; Γ is Δ with the additional rule $(X, *, X)$ for all $X \in V$. The transition relation $\twoheadrightarrow_P$ coincides with that obtained from the standard operational semantics for BPP_τ definition Δ. This implies:

Fact 3.1 *$\approx_P$ coincides with classical strong bisimilarity on processes of Δ.*

Our notion of norm for processes of P coincides with that typically associated with processes of Δ [2]. Δ is normed iff every non-empty process has finite norm.

Fact 3.2 *If Δ is normed then P is a normed CRS.*

3.2 BPP$_\tau$ under Distributed Semantics

Distributed bisimilarity reflects that whenever a process performs an action it can be thought to evolve into a *local remainder* and a *concurrent remainder.* Any guarded BPP$_\tau$ definition can be transformed (in polynomial-time) into a BPP$_\tau$ definition in distributed normal form [3], that is a finite set $\Delta = \{X \xrightarrow{a} (\alpha_l, \alpha_c)\}$ where $a \in Act_{CCS}$ and the following relation $\prec$ on the variables of Δ is irreflexive: $\prec$ is the transitive closure of the relation $\prec_1$ where $Y \prec_1 X$ iff there is a rule $(X, a, \alpha_l, \alpha_c)$ in Δ such that $Y \in \alpha_c$. Intuitively, α_l and α_c express the *local* and *concurrent* part, respectively, of a derivative of X [3].

Δ induces a CRS $P = (V, \mathrm{Act}, \Gamma)$ as follows: V and Act are derived from Δ and Act_{CCS} in the obvious way; Γ is Δ with the additional rule $(X, *, \epsilon, X) \in \Gamma$ for all $X \in V$. The transition relation $\twoheadrightarrow_P$ coincides with that obtained from the distributed operational semantics for Δ. Thus, we have:

Fact 3.3 *$\approx_P$ coincides with distributed bisimilarity on processes of Δ.*

We assume that the variables of P are ordered according to $\prec$: $X_i \prec X_j$ implies $i < j$. Then it is immediate:

Fact 3.4 *P is a normed CRS.*

3.3 BPP under Weak Semantics

Let Δ be a BPP definition, i.e., a BPP$_\tau$ definition that does not contain any actions of $\overline{Act}$. By $\longrightarrow$ denote the transition relation associated with Δ under the standard operational semantics, or equivalently, the transition relation of the CRS induced by Δ.

Given two processes α, β, we write $\alpha\ (\xrightarrow{\tau})^*\ \beta$ if β can be reached from α by an arbitrary number of τ transitions. Weak bisimilarity abstracts away from silent actions by reflecting the following weak transition relation:

$$\alpha \overset{a}{\Longrightarrow} \beta \text{ iff } \begin{cases} \alpha\ (\xrightarrow{\tau})^*\ \beta & \text{if } a = \tau, \\ \alpha\ (\xrightarrow{\tau})^*\ \beta' \ \xrightarrow{a} \beta''\ (\xrightarrow{\tau})^*\ \beta \text{ for some } \beta', \beta'' & \text{otherwise.} \end{cases}$$

To capture weak bisimilarity as a notion of P-bisimilarity, we need to exhibit a suitable CRS P such that $\twoheadrightarrow_P$ coincides with $\Longrightarrow$. We achieve this by defining the *synchronization algebra of weak actions*, $Act_W = (Act_W, \bullet)$ where Act_W and $\bullet$ are as follows.

Act_W is the set of *weak actions* given by $\{'\tau^* a \tau^{*\prime} : a \in Act\} \cup \{'\tau^{*\prime}\}$. The action names indicate the type of weak transition a weak action is thought to represent; but note that we could have chosen other names just as well. When a process α performs a weak transition then several of α's elementary processes may contribute to it. To reflect this we allow weak actions to occur as synchronized actions: two processes may jointly perform action $'\tau^* a \tau^{*\prime}$ if one process performs $'\tau^* a \tau^{*\prime}$ and the other process performs $'\tau^{*\prime}$. Similarly, two processes may jointly perform action $'\tau^{*\prime}$

if they both perform $'\tau^*{}'$. Formally, $\bullet$ is given by the following table:

	$*$	$'\tau^*{}'$	$'\tau^* a \tau^*{}'$	$'\tau^* b \tau^*{}'$	$\ldots$
$*$	$*$	$'\tau^*{}'$	$'\tau^* a \tau^*{}'$	$'\tau^* b \tau^*{}'$	$\ldots$
$'\tau^*{}'$	$'\tau^*{}'$	$'\tau^*{}'$	$'\tau^* a \tau^*{}'$	$'\tau^* b \tau^*{}'$	$\ldots$
$'\tau^* a \tau^*{}'$	$'\tau^* a \tau^*{}'$	$'\tau^* a \tau^*{}'$			$\ldots$
$'\tau^* b \tau^*{}'$	$'\tau^* b \tau^*{}'$	$'\tau^* b \tau^*{}'$			$\ldots$
.	.	.	.	.	$\ldots$

We associate a weight function W with Act_W, defined by: $W(*) = 0$, $W('\tau^*{}') = 0$, and $W('\tau^* a \tau^*{}') = 1$ for all $a \in Act$.

Δ induces a CRS $P = (V, \mathrm{Act}, \Gamma)$ as follows: V and Act are derived from V and Act_W in the obvious way; Γ is defined by:

$$(X, '\tau^*{}', \alpha) \in \Gamma \quad \text{iff} \quad X \overset{\tau}{\Longrightarrow} \alpha,$$

$$(X, '\tau^* a \tau^*{}', \alpha) \in \Gamma \quad \text{iff} \quad X \overset{a}{\Longrightarrow} \alpha,$$

$$(X, *, X) \in \Gamma.$$

Note here a difference between Γ and its presentation. The set of rules Γ is infinite, in general; however, it is succinctly represented by a finite Δ.

The transition relation $\twoheadrightarrow_P$ indeed captures the weak transition relation of Δ: $\alpha \overset{'\tau^*{}'}{\twoheadrightarrow} \beta$ iff $\alpha \overset{\tau}{\Longrightarrow} \beta$; and, $\alpha \overset{'\tau^* a \tau^*{}'}{\twoheadrightarrow} \beta$ iff $\alpha \overset{a}{\Longrightarrow} \beta$. Thus, we obtain:

Fact 3.5 *$\approx_P$ coincides with weak bisimilarity on processes of Δ.*

Our notion of norm for processes of P coincides with that typically associated with processes of Δ in the context of weak bisimilarity. Δ is totally normed if each process has finite and positive norm. Assume P is induced from a totally normed Δ. Then it is immediate that P satisfies (N1) of the definition of normed CRS. However, condition (N2) is not necessarily obtainable: simply consider two variables X, Y such that $X \overset{\tau}{\Longrightarrow} Y \overset{\tau}{\Longrightarrow} X$. On the other hand, it is easy to check that such X and Y will be weakly bisimilar ($|X| = |Y|$ in particular). Thus, we can transform P into an equivalent (weakly bisimilar) CRS P' by removing one of X or Y, say Y, whenever $X \overset{\tau}{\Longrightarrow} Y \overset{\tau}{\Longrightarrow} X$. This also involves substituting X for Y whenever Y appears on the right side of a rule in Γ. Then the following relation $\preceq$ is a partial order: $X \preceq Y$ iff $|X| < |Y|$ or $Y \overset{\tau}{\Longrightarrow} X$. We assume that the variables of P' are ordered according to $\prec$: $X_i \prec Y_j$ implies $i < j$. Then P' will indeed satisfy (N2): whenever $X_i \xrightarrow{a}_{\mathrm{n-r}} \alpha$ then $\alpha \in \{X_1, \ldots, X_{i-1}\}^{\otimes}$.

Fact 3.6 *P' is a normed CRS.*

3.4 BPP under Branching Semantics

Branching bisimilarity abstracts away from silent actions while preserving the branching structure in more detail than weak bisimilarity. It can be captured by considering the following type of observable transitions:

$$\alpha \overset{a}{\Longrightarrow} (\alpha_1, \alpha_2) \text{ iff } \begin{cases} \alpha\ (\xrightarrow{\tau})^*\ \alpha_1 \xrightarrow{a} \alpha_2, \text{ or} \\ \alpha\ (\xrightarrow{\tau})^*\ \alpha_1 = \alpha_2 \text{ and } a = \tau. \end{cases} \tag{1}$$

This is based on the notion of *semi-branching bisimulation.* In [1] it was shown that this variant of bisimulation induced the branching bisimilarity. For a concise formulation, by $\alpha_1 \xrightarrow{(a)} \alpha_2$ we mean $\alpha_1 \xrightarrow{a} \alpha_2$ or $a = \tau$ and $\alpha_1 = \alpha_2$. The following may be easily deduced from [1]:

Proposition 3.7 *A binary relation R over processes is a semi-branching bisimulation if and only if for each $(\alpha, \beta) \in R$ and $a \in Act \cup \{\tau\}$,*

(i) *whenever $\alpha\ (\xrightarrow{\tau})^*\ \alpha' \xrightarrow{(a)} \alpha''$ then $\beta\ (\xrightarrow{\tau})^*\ \beta' \xrightarrow{(a)} \beta''$, for some β', β'', with $(\alpha', \beta') \in R$ and $(\alpha'', \beta'') \in R$;*

(ii) *the symmetric condition holds.*

Accordingly, the set of branching-observable actions is now $\{'\tau^* a' : a \in Act \cup \{\tau\}\} \cup \{'\tau^{*\prime}\}$. Note that we distinguish between $'\tau^{*\prime}$ and $'\tau^*\tau'$, which correspond to the two different cases in (1). By similar considerations as for weak bisimilarity we arrive at the following synchronization algebra.

	$*$	$'\tau^{*\prime}$	$'\tau^*\tau'$	$'\tau^*a'$	$'\tau^*b'$	$\dots$
$*$	$*$	$'\tau^{*\prime}$	$'\tau^*\tau'$	$'\tau^*a'$	$'\tau^*b'$	$\dots$
$'\tau^{*\prime}$	$'\tau^{*\prime}$	$'\tau^{*\prime}$	$'\tau^*\tau'$	$'\tau^*a'$	$'\tau^*b'$	$\dots$
$'\tau^*\tau'$	$'\tau^*\tau'$	$'\tau^*\tau'$				$\dots$
$'\tau^*a'$	$'\tau^*a'$	$'\tau^*a'$				$\dots$
$'\tau^*b'$	$'\tau^*b'$	$'\tau^*b'$				$\dots$
.	.	.	.	.	.	$\dots$

Clearly $'\tau^{*\prime} \bullet {'\tau^{*\prime}} = {'\tau^{*\prime}}$ and $'\tau^{*\prime} \bullet {'\tau^*a'} = {'\tau^*a'}$, but $'\tau^*\tau' \bullet {'\tau^*a'}$ is undefined. Hence necessarily $'\tau^{*\prime} \neq {'\tau^*\tau'}$. On the other hand, branching bisimilarity allows to match a $'\tau^{*\prime}$ step by $'\tau^*\tau'$ and vice versa. This motivates the following definition of Γ:

$$\begin{aligned} (X, {'\tau^{*\prime}}, \alpha_1, \alpha_2) \in \Gamma \quad &\text{iff} \quad X\ (\xrightarrow{\tau})^*\ \alpha_1 = \alpha_2, \\ (X, {'\tau^*a'}, \alpha_1, \alpha_2) \in \Gamma \quad &\text{iff} \quad X\ (\xrightarrow{\tau})^*\ \alpha_1 \xrightarrow{(a)} \alpha_2. \end{aligned} \tag{2}$$

Note that if $(X, {'\tau^{*\prime}}, \alpha_1, \alpha_2) \in \Gamma$ then $(X, {'\tau^*\tau'}, \alpha_1, \alpha_2) \in \Gamma$ as well. Additionally we put $(X, *, X, X) \in \Gamma$. This induces a CRS P and we obtain:

Fact 3.8 *$\approx_P$ coincides with branching bisimilarity on processes of Δ.*

If Δ is totally normed, then P can be easily made normed, similarly as before.

For comparison, we summarize the different formats of idle rules used to characterize the four equivalences:

bisimilarity	idlerules
strongandweak	$(X, *, X)$
distributed	$(X, *, \epsilon, X)$
branching	$(X, *, X, X)$

Note a difference between distributed and branching equivalence: in the former one, the idle move of X 'contributes' the empty process ϵ to the local derivative; in the latter, instead of ϵ, X itself must be used, to faithfully respect definition of semi-branching bisimulation [1].

The observation that the very different kinds of bisimulation equivalences can be captured in a uniform way, was the starting point and motivation for developing the general algorithm presented in Section 5.

4 Unique Decomposition

We now prove our unique decomposition results. As explained in the introduction we build on a theorem by Luttik and van Oostrom that characterizes when a partial commutative monoid has unique decomposition. We first introduce this tool, specialized to the case of commutative monoids.

Let M be a commutative monoid with identity element e. An element p of M is called *indecomposable* if $p \neq e$ and $p = xy$ implies $x = e$ or $y = e$. A *decomposition* of x in M is a finite multiset $p_1^{a_1} \dots p_r^{a_r}$ of indecomposable elements of M such that $x = p_1^{a_1} \dots p_r^{a_r}$. If every element of M has a unique decomposition (up to multiset equality) then we say that M has *unique decomposition.*

A partial order $\preceq \subseteq M \times M$ is a *decomposition order* if

(i) it is *well-founded*: every non-empty subset of M has a $\preceq$-minimal element;

(ii) e is the least element of M wrt. $\preceq$: $e \preceq x$ for all $x \in M$;

(iii) it is *strictly compatible*: for all x, y, $z \in M$, if $x \prec y$ then $xz \prec yz$;

(iv) it is *precompositional*: for all x, y, $z \in M$, $x \preceq yz$ implies $x = y'z'$ for some $y' \preceq y$ and $z' \preceq z$; and

(v) it is *Archimedean*: for all $x, y \in M$, $x^n \prec y$ for all $n \in \mathbb{N}$ implies $x = e$.

Corollary 4.1 (by Corollary 3.15 from [12]) *A commutative monoid has unique decomposition iff it has a decomposition order.*

With the help of this tool we now prove our unique decomposition results. Let $P = (V, \mathrm{Act}, \Gamma)$ be a normed CRS, and $\sim \subseteq V^{\otimes} \times V^{\otimes}$ be a congruence. $V^{\otimes}/\sim$ is the quotient of $V^{\otimes}$ by $\sim$, i.e., the set of equivalence classes of $V^{\otimes}$ wrt. $\sim$. We write $[\alpha]_\sim$ for the equivalence class containing α. Since $\sim$ is a congruence we can define on $V^{\otimes}/\sim$ a binary operation as follows: $[\alpha]_\sim [\beta]_\sim = [\alpha\beta]_\sim$. It is easy to verify that $V^{\otimes}/\sim$ is a commutative monoid under this operation with identity element $[\epsilon]_\sim$.

Theorem 4.2 *Let $\sim$ be a congruence which is a norm-reducing bisimulation. The commutative monoid $V^{\otimes}/{\sim}$ has unique decomposition.*

Assume $\sim$ to be a congruence which is a norm-reducing bisimulation. We can exploit the properties of norm-reducing transitions to exhibit a decomposition order on $V^{\otimes}/{\sim}$. Define a relation $\preccurlyeq$ on $V^{\otimes}/{\sim}$ as follows:

$$[\beta]_{\sim} \preccurlyeq [\alpha]_{\sim} \iff \alpha \sim \alpha' \xrightarrow{w}_{\mathrm{n-r}} \beta' \sim \beta \text{ for some } \alpha',\ \beta',\ \text{and } w \in \mathrm{Act}^*.$$

Proposition 4.3 below establishes that $\preccurlyeq$ is a decomposition order. Theorem 4.2 is then an immediate consequence of Corollary 4.1.

Proposition 4.3 *(i) $\preccurlyeq$ is a partial order. (ii) $\preccurlyeq$ is a decomposition order.*

Proof. (i) Reflexivity is immediate from the definition. Transitivity follows since $\sim$ is a norm-reducing bisimulation. It remains to show that $\preccurlyeq$ is antisymmetric. Assume $[\beta]_{\sim} \preccurlyeq [\alpha]_{\sim}$ and $[\alpha]_{\sim} \preccurlyeq [\beta]_{\sim}$. Hence $\alpha \sim \alpha' \xrightarrow{w}_{\mathrm{n-r}} \beta' \sim \beta \sim \beta'' \xrightarrow{v}_{\mathrm{n-r}} \alpha'' \sim \alpha$ for some w, v, α', α'', β', β''. Since $\sim$ is a norm-reducing bisimulation this allows us to deduce an infinite sequence

$$\alpha' \xrightarrow{w}_{\mathrm{n-r}} \beta' \xrightarrow{v}_{\mathrm{n-r}} \alpha_1 \xrightarrow{w}_{\mathrm{n-r}} \beta_1 \xrightarrow{v}_{\mathrm{n-r}} \alpha_2 \xrightarrow{w}_{\mathrm{n-r}} \beta_2 \ \ldots$$

for some $\alpha_1, \beta_1, \alpha_2, \beta_2, \ldots$ such that for all i, $\alpha \sim \alpha_i$ and $\beta \sim \beta_i$. By an analog of Fact 2.1(iii) for norm-reducing transitions, we can transform the above sequence into one where all the transitions result from rewriting a single variable. By Prop. 2.4 we immediately deduce that the sequence must be constant from some point i onwards: $\alpha_i = \beta_i = \alpha_{i+1} = \beta_{i+1} = \ldots$. This implies $\alpha \sim \beta$ and hence $[\beta]_{\sim} = [\alpha]_{\sim}$ as required.

(ii) We verify that Properties (i) to (v) of the definition of decomposition order are satisfied by $\preccurlyeq$. (i) Well-foundedness: assume that there exists an infinite strictly descending chain wrt. $\preccurlyeq$, and observe that the norm does not increase along the chain. Therefore, this would require infinitely many processes of norm bounded by the norm of the first element – a contradiction. Property (ii) is immediate since for all α, $\alpha \xrightarrow{w}_{\mathrm{n-r}} \epsilon$ for some w. To verify Property (iii) recall that by Axiom (S2) of synchronization algebras every action can occur asynchronously: $a \bullet * = a$. Then strict compatibility and precompositionality are immediate by norm-reducing versions of Fact 2.1(2) and Fact 2.1(iii), respectively. To validate Property (v) observe: for all α such that $\alpha \neq \epsilon$ and β we can always find n such that $|\alpha^n| > |\beta|$. (v) is then immediate since by Prop. 2.8 norm carries over from $V^{\otimes}$ to $V^{\otimes}/{\sim}$. □

Since $\approx_P$ is a congruence and every bisimulation is a norm-reducing bisimulation we immediately obtain:

Corollary 4.4 *The commutative monoid $V^{\otimes}/{\approx_P}$ has unique decomposition.*

Finally, in view of our algorithm we derive a more compact formulation of our unique decomposition results. The formulation is in the style of [6], and makes use of the assumption that decompositions preserve norm, and that variables are given in order of non-decreasing norm.

Assume $\sim$ to be a norm-preserving congruence. An elementary process X_i is *decomposable* wrt. $\sim$ if there exists $\alpha \in \{X_1, \ldots, X_{i-1}\}^{\otimes}$ such that $X_i \sim \alpha$. Otherwise X_i is *prime* wrt. $\sim$. Let $P_1, \ldots, P_r$ be the primes of V wrt. $\sim$, ordered consistently with V itself. We say that $\sim$ has *unique decomposition* if for any processes $\alpha = P_1^{a_1} \dots P_r^{a_r}$ and $\beta = P_1^{b_1} \dots P_r^{b_r}$, $\alpha \sim \beta$ implies $a_1 = b_1, \dots, a_r = b_r$.

Corollary 4.5

(i) *Let $\sim$ be a congruence which is a norm-reducing bisimulation. Then $\sim$ has unique decomposition.*

(ii) *$\approx_P$ has unique decomposition.*

Proof. (i) This follows from Theorem 4.2 by the following observation: if X_i is prime wrt. $\sim$, then $[X_i]_\sim$ is indecomposable in $V^\otimes/\sim$. By Prop. 2.8 norm-reducing bisimulations are norm-preserving. Hence, any proper component of $[X_i]_\sim$ is of norm smaller than X_i, which implies $[X_i]_\sim$ is either prime or representable by variables of index smaller than i.

(ii) is then also obvious. □

5 Decision Procedure

Motivated by concrete CRS' from Section 3 we introduce the following notions. A *presentation* Δ consists of a finite set V of variables, a finite set of actions, and a finite set of rules of the form $X \xrightarrow{a} (\alpha_1, \dots, \alpha_m)$, for some fixed m, where $X \in V$, a is a action, and each $\alpha_i \in V^\otimes$ consists exclusively of variables than appear on the left-hand side of some rule. A *presentation mapping* is a partial function $\mathbb{P}$ that assigns, to a presentation Δ, a normed CRS $P = \mathbb{P}(\Delta)$ over the same variables V. Assume a fixed presentation mapping $\mathbb{P}$ in this section. If $P = \mathbb{P}(\Delta)$ we say that Δ is a presentation of P, or that P is represented by Δ.

In this section we provide our general decision procedure. As explained in Section 1 it is an extension of [6]. We consider the following bisimilarity problem:

Input: A presentation Δ of a normed CRS P, and two variables X, Y.

Question: Decide whether $X \approx_P Y$.

From our generic decision procedure, one obtains the algorithms for the four concrete bisimulation equivalences (cf. Section 6), by providing a particular implementation of a subroutine that we specify in the sequel.

In the following, we first present some preliminaries. In particular, we will define when, wrt. to a given complexity class $\mathcal{T}$, we consider the presentation mapping to be *$\mathcal{T}$-effective*. We will then prove our main result: whenever a presentation mapping is $\mathcal{T}$-effective then the bisimilarity problem can be decided in $\mathcal{T}$. Finally, we provide the proofs that have been left out in the main text.

5.1 Preliminaries

Throughout the section, let $P = (V, \mathrm{Act}, \Gamma)$ be a normed CRS represented by Δ. Recall that the set of rewrite rules Γ can be infinite. However, the input to our algorithm will be Δ, not P itself. The set of variables V is available through Δ while the synchronization algebra Act will not be explicitly needed. All we need is that the presentation mapping is $\mathcal{T}$-effective. As we will see below this ensures a sufficient set of the behaviour of P can be enumerated in $\mathcal{T}$. We impose the following assumption though.

Assumption 1 *We assume that the presentation Δ of P satisfies the following condition. For each variable $X \in V$ we have:*

(i) *The idle rule for X is in Δ. (Recall that the idle rule is unique.)*

(ii) *$X \xrightarrow{a} (\alpha_1, \ldots, \alpha_m)$ is in Δ for at least one rule $(X, a, \alpha_1, \ldots, \alpha_m) \in \Gamma$ such that $X \xrightarrow{a}_{\mathrm{n-r}} \alpha_m$ and $X \neq \alpha_m$.*

The first condition guarantees that idle transitions are available for the computation as a part of input; it is used later on to prove Prop. 5.4(i). The second condition, necessary to prove Prop. 5.1 below, requires availability of at least one non-trivial norm-reducing transition of each X. The conditions are naturally satisfiable for all the concrete cases we will consider in Section 6. In the following, we denote the size of presentation Δ by n.

We assume that processes of P are represented as products $X_1^{a_1} \ldots X_k^{a_k}$, where the exponents $a_1, \ldots, a_k$ are encoded in binary. Often, it will be sufficient to consider *small* processes only. A process α is called *small* if $|\alpha| \leq |X|$ for some variable X. We will make use of the following observations:

Proposition 5.1

(i) *The norms of all variables are computable in time polynomial wrt. n.*

(ii) *The norm of a process α can be computed in time polynomial wrt. n and the size of α.*

(iii) *The norm of a variable, and hence, of a small process is at most exponential wrt. n.*

(iv) *A small process can be represented in space polynomial wrt. n.*

For the purpose of generality, we work in this section with an arbitrary deterministic complexity class $\mathcal{T}$ that includes PTIME (e.g., PTIME, PSPACE, EXPTIME, etc). $\mathcal{T}$ is specified by (1) some complexity bound function f, mapping the size of input to the amount of resource available, and (2) the type of resource f refers to, time or space. Furthermore, we assume that the class $\mathcal{T}$ is determined 'up to a polynomial', in the sense that it is the union of the complexity classes $\mathcal{T}_k$, $k > 0$, corresponding to the bounds $f_k(n) = f(n^k)$. In Section 6 we will instantiate $\mathcal{T}$ with PTIME and PSPACE. We use the following convention: whenever we state that a problem is in $\mathcal{T}$ and do not specify what the input size of the problem is we always mean complexity wrt. n, the size of Δ.

We consider a presentation mapping to be $\mathcal{T}$-*effective* whenever, given Δ as input, we are able to enumerate in $\mathcal{T}$ a subset of the m-ary transitions of P, say $\mathcal{S}$, such that $\mathcal{S}$ fully captures the behaviour of small processes in the following sense: every transition starting from a small process is either contained in $\mathcal{S}$ or it can be obtained as the synchronization of a transition in $\mathcal{S}$ and an idle transition. In the following, for notational convenience, we allow the transition $\epsilon \overset{*}{\twoheadrightarrow} (\epsilon, \ldots, \epsilon)$.

Definition 5.2 We say a set $\mathcal{S}$ of $\twoheadrightarrow$-transitions is *sufficient* for P iff whenever $\alpha \overset{a}{\twoheadrightarrow} (\alpha_1, \ldots, \alpha_m)$ and α is small then there is $\alpha' \overset{a}{\twoheadrightarrow} (\alpha'_1, \ldots, \alpha'_m) \in \mathcal{S}$ and $\beta \overset{*}{\twoheadrightarrow}$

$(\beta_1, \ldots, \beta_m)$ such that $\alpha = \alpha'\beta$, $\alpha_1 = \alpha'_1\beta_1$, …, $\alpha_m = \alpha'_m\beta_m$.

Definition 5.3 We say $\mathbb{P}$ is $\mathcal{T}$*-effective* if every $P = \mathbb{P}(\Delta)$ has a sufficient set that, given Δ as input, can be enumerated in $\mathcal{T}$.

Observe that we do not require to compute and store the sufficient set: we only need to *enumerate* it, i.e., generate each of its transitions in a systematic way for further processing. For the rest of this section we need:

Assumption 2 *Presentation mapping* $\mathbb{P}$ *is* $\mathcal{T}$*-effective.*

Proposition 5.4 *Given an action* $a \in \mathrm{Act}$*, and a small process* α*, the following sets can be enumerated in* $\mathcal{T}$

(i) $\mathrm{succ}_a(\alpha) = \{(\alpha_1, \ldots, \alpha_m) : \alpha \overset{a}{\twoheadrightarrow} (\alpha_1, \ldots, \alpha_m)\}$,

(ii) $\mathrm{n-r-succ}_a(\alpha) = \{\beta : \alpha \xrightarrow{a}_{\mathrm{n-r}} \beta\}$.

5.2 Decision Procedure

The initial insight behind our procedure is this: since $\approx_P$ satisfies unique decomposition it can be represented by a *unique decomposition base* [6].

Definition 5.5 A *(norm-preserving) unique decomposition base* for the variables $V_i = \{X_1, \ldots, X_i\}$ of V is a pair $D = (\Pi, \Gamma)$, where $\Pi \subseteq V_i$ is a set of *primes*, and Γ is a set of pairs (X, α) such that $\alpha \in \Pi^{\otimes}$, $|X| = |\alpha|$, and there is one pair for each variable $X \in V_i \setminus \Pi$. Γ is thought to specify for each non-prime variable X a decomposition into primes. The primes of a decomposition base are, a priori, arbitrarily chosen and not to be confused with the primes wrt. a given congruence. D induces a relation $\equiv_D$ on $V_i^{\otimes}$: $\alpha \equiv_D \beta$ iff the prime decompositions of α and β are equal. For $V_i = V$ we simply call D a unique decomposition base.

Proposition 5.6 *Let* D *be a unique decomposition base.*

(i) $\equiv_D$ *can be decided in polynomial time wrt.* n *and the sizes of the two input processes.*

(ii) *For small processes,* $\exp(\equiv_D)$ *can be decided in* $\mathcal{T}$*.*

Let $\equiv$ be a norm-preserving congruence and recall from Section 2.3 that the greatest norm-reducing bisimulation contained in $\equiv$ exists and is denoted by $\mathrm{gnrb}(\equiv)$. By Prop. 2.9 and Corollary 4.5(i) $\mathrm{gnrb}(\equiv)$ has unique decomposition, and is thus representable by a unique decomposition base D. Our core insight is that due to Prop. 2.4 we can construct D inductively, and thus efficiently.

Using this insight we can approximate bisimilarity from above by a sequence of norm-preserving congruences. We start with the greatest such relation, say $\equiv$, and compute the decomposition base D that represents $\mathrm{gnrb}(\equiv)$. If $\equiv_D$ is a bisimulation then we have already reached bisimilarity. Otherwise we can move strictly closer to bisimilarity by intersecting $\equiv_D$ with $\exp(\equiv_D)$. That is, we consider a new congruence $\equiv'$ defined by $\alpha \equiv' \beta$ iff $\alpha \equiv_D \beta$ and (α, β) satisfies expansion in $\equiv_D$. We continue by computing the unique decomposition base that represents $\mathrm{gnrb}(\equiv')$, and proceed in this fashion until finally we will hit bisimilarity. First of all, we prove the core insight.

Theorem 5.7 *Let $\equiv$ be a norm-preserving congruence that, for small processes, can be decided in $\mathcal{T}$. Then a unique decomposition base D can be computed in $\mathcal{T}$ such that $\equiv_D = \mathrm{gnrb}(\equiv)$.*

Proof. For each $i \in [1, \dots, k]$ we define, by induction, a unique decomposition base $D_i = (\Pi_i, \Gamma_i)$ for $\{X_1, \dots, X_i\}^{\otimes}$, with $\Pi_i = \{P_1, \dots, P_r\}$, such that:

(1) each P_j is prime wrt. gnrb($\equiv$); and

(2) $(X_j, P_1^{x_1} \dots P_r^{x_r}) \in \Gamma_i$ implies $(X_j, P_1^{x_1} \dots P_r^{x_r}) \in \mathrm{gnrb}(\equiv)$.

Assume there exists a unique decomposition base D_i for $\{X_1, \dots, X_i\}^{\otimes}$ that satisfies these two properties. Then the following three claims will hold (c.f. Section 5.3 for the detailed proofs):

Claim 5.8 $\equiv_{D_i} = \mathrm{gnrb}(\equiv)$ *holds on the set* $\{X_1, \dots, X_i\}^{\otimes}$.

Claim 5.9 *Let α be a composition $P_1^{x_1} \dots P_r^{x_r}$. Then $(X_{i+1}, \alpha) \in \mathrm{gnrb}(\equiv)$ iff $X_{i+1} \equiv \alpha$ and $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\equiv_{D_i} \cup \{(X_{i+1}, \alpha)\})$.*

Claim 5.10 *It can be decided in $\mathcal{T}$ whether there is $\alpha = P_1^{x_1} \dots P_r^{x_r}$ that satisfies the right-hand side of Claim 5.9. If there is such α it can be found in $\mathcal{T}$.*

Claim 5.8 specialized to $i = k$ ensures $\equiv_{D_k} = \mathrm{gnrb}(\equiv)$ as required. Claim 5.9 indicates how to extend D_i to a unique decomposition base D_{i+1} that also satisfies Properties (1) and (2). By Claim 5.9, Property (1), and since gnrb($\equiv$) has unique decomposition there is at most one $\alpha = P_1^{x_1} \dots P_r^{x_r}$ such that $X_{i+1} \equiv \alpha$ and $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\equiv_{D_i} \cup \{(X_{i+1}, \alpha)\})$. If there is such α, declare it to be the decomposition of X_{i+1} in D_{i+1}. Otherwise declare X_{i+1} to be prime in D_{i+1}. It is clear that Properties (1) and (2) are satisfied. By Claim 5.10 the extension of D_i to D_{i+1} can be computed in $\mathcal{T}$. Then overall, D_k can also be computed in $\mathcal{T}$. □

Apart from our core theorem we require two lemmas, which are analogous to insights of [6]. The first lemma gives us the means to efficiently check whether we have reached bisimilarity.

Lemma 5.11 *Let $D = (\Pi, \Gamma)$ be a unique decomposition base.*

(i) *$\equiv_D$ is a bisimulation iff Γ satisfies expansion in $\equiv_D$, that is $\Gamma \subseteq \exp(\equiv_D)$.*

(ii) *It can be checked in $\mathcal{T}$ whether $\equiv_D$ is a bisimulation.*

The second lemma ensures that $\equiv_D \cap \exp(\equiv_D)$ is a congruence as required, and strictly refines $\equiv_D$ whenever we have not reached bisimilarity yet.

Lemma 5.12 *Let D be a unique decomposition base.*

(i) *$\equiv_D \cap \exp(\equiv_D)$ is a norm-preserving congruence that, for small processes, can be decided in $\mathcal{T}$.*

(ii) *If $\equiv_D$ is not a bisimulation then some variable decomposable wrt. $\equiv_D$ is prime wrt. $\equiv_D \cap \exp(\equiv_D)$.*

The algorithm consists of the loop outlined below. In each iteration a given congruence $\equiv$ is refined to a congruence $\equiv_D$ that is represented by a unique decomposition base D. Then $\equiv_D$ is in turn reduced to a new congruence, which provides the value of $\equiv$ for the next iteration. Note that $\equiv$ will not be kept explicitly but be available in terms of its decision procedure. The following two invariants hold:

Invariant 1: $\equiv$ is a norm-preserving congruence that, for small processes, can be decided in $\mathcal{T}$.

Invariant 2: $\equiv$ and $\equiv_D$ both subsume bisimilarity: $\approx \subseteq \equiv$ and $\approx \subseteq \equiv_D$.

(1) Let $\equiv$ be the congruence defined by $\alpha \equiv \beta$ iff $|\alpha| = |\beta|$.

(2) Compute a unique decomposition base D such that $\equiv_D = \operatorname{gnrb}(\equiv)$.
This can be done in $\mathcal{T}$ by using Theorem 5.7.

(3) If $\equiv_D$ is a bisimulation then halt and return D.
The condition can be checked in $\mathcal{T}$ by Lemma 5.11(ii).

(4) Otherwise, redefine $\equiv$ to be the congruence $\equiv_D \cap \exp(\equiv_D)$. Go to step (2).

Claim 5.13 (Invariants) *Invariants 1 and 2 are indeed satisfied.*

After step (1), Invariant 1 holds by the definition of $\equiv$ and because by Prop. 5.1(ii) and (iv) $\equiv$ is computable in polynomial time. Invariant 2 follows since by Prop. 2.8 $\approx$ is norm-preserving. Step (2) clearly preserves Invariant 2: $\approx$ is a norm-reducing bisimulation contained in $\equiv$ while $\equiv_D$ is the greatest such relation. After Step (4), Invariant 1 holds by Lemma 5.12(i). Invariant 2 follows since $\approx \subseteq \equiv_D$ and $\approx = \exp(\approx) \subseteq \exp(\equiv_D)$.

Claim 5.14 (Termination) *The algorithm halts after at most n iterations.*

By definition $\equiv_D \cap \exp(\equiv_D)$ is finer than $\equiv_D$. This implies that if a variable X_i is prime in D it will also be prime in the decomposition base of the next iteration. Lemma 5.12(ii) ensures that on each iteration at least one variable becomes prime that was not prime before. Thus, the number of iterations is indeed bounded by n.

Claim 5.15 (Complexity) *The algorithm works in $\mathcal{T}$.*

This follows from Claim 5.14 and because each iteration can be completed in $\mathcal{T}$.

Claim 5.16 (Correctness) *The unique decomposition base D output by the algorithm represents bisimilarity: $\equiv_D = \approx$.*

The inclusion $\approx \subseteq \equiv_D$ follows by Invariant 2 while the opposite inclusion is immediate by $\equiv_D$ being a bisimulation.

Since $\equiv_D$ can be decided in polynomial time (c.f. Prop. 5.6(i)) altogether we have proved:

Theorem 5.17 *For a $\mathcal{T}$-effective presentation mapping, the bisimilarity problem is decidable in $\mathcal{T}$ (wrt. the size of presentation).*

5.3 Proofs

Proof. [Prop. 5.1] (i) The norms of all variables can be computed in the standard way as the unique solution of the set of linear equations of the form: $|X| = W(a) + |\alpha_m|$, one for each variable X, where α_m is given by the norm-reducing transition $X \xrightarrow{a}_{\mathrm{n-r}} \alpha_m$ obtained from a rule $X \xrightarrow{a} (\alpha_1, \ldots, \alpha_m)$ from Δ as assured by Assumption 1; $|\alpha_m|$ stands for the sum of norms of all variables in α_m. Recall that by Prop. 2.4 in Section 2, the set of equations is ‘acyclic’ and hence has a unique solution.

(ii) follows from (i). (iii) is straightforward considering Assumption 1. (iv) is a consequence of (iii). □

For the proof of Prop. 5.4 we will use Prop. 5.18 below. In the formulation of this proposition we make use of the following observation: given an idle transition $\beta \overset{*}{\twoheadrightarrow} (\beta_1, \ldots, \beta_m)$, for all $j \in [1, \ldots, m]$, β_j is uniquely determined by β: if $\beta = X_1^{a_1} \ldots X_k^{a_k}$ then $\beta_j = \beta_{1,j}^{a_1} \ldots \beta_{k,j}^{a_k}$ where, for $j \in [1, \ldots, m]$, $i \in [1, \ldots k]$, $\beta_{i,j}$ is given by $X_i \overset{*}{\twoheadrightarrow} (\beta_{i,1}, \ldots, \beta_{i,m}) \in \Gamma$. (Recall that for each variable there is exactly one idle rule in Γ. Also recall our convention $\epsilon \overset{*}{\twoheadrightarrow} (\epsilon, \ldots, \epsilon)$.) For two processes α, α' we write $\alpha' \sqsubseteq \alpha$ iff α' is contained in α as a multiset. We write $\alpha \backslash \alpha'$ for α minus α' as multisets.

Proposition 5.18 *Let α be small, and $\mathcal{S}$ be a sufficient set for P.*

(i) $\alpha \overset{a}{\twoheadrightarrow} (\alpha_1, \ldots, \alpha_m)$ *iff there is* $\alpha' \overset{a}{\twoheadrightarrow} (\alpha'_1, \ldots, \alpha'_m) \in \mathcal{S}$ *such that* $\alpha' \sqsubseteq \alpha$, $\alpha'_1 \sqsubseteq \alpha_1$, …, $\alpha'_m \sqsubseteq \alpha_m$ *and* $\alpha \backslash \alpha' \overset{*}{\twoheadrightarrow} (\alpha_1 \backslash \alpha'_1, \ldots, \alpha_m \backslash \alpha'_m)$.

(ii) $\alpha \xrightarrow{a}_{\mathrm{n-r}} \beta$ *iff there is* $\alpha' \overset{a}{\twoheadrightarrow} (\alpha'_1, \ldots, \alpha'_m) \in \mathcal{S}$ *such that* $|\alpha'| = W(a) + |\alpha'_m|$, $\alpha' \sqsubseteq \alpha$, *and* $\beta = (\alpha \backslash \alpha')\, \alpha'_m$.

Proof. (i) The only-if direction follows from the definition of sufficient sets. The if-direction follows from the inductive definition of the m-ary transition relation (c.f. Section 2).

(ii) This is immediate with (i) when considering Axiom (C1) of the definition of CRS’. □

Proof. [Prop. 5.4] (i) We can assume a sufficient set $\mathcal{S}$ that can be enumerated in $\mathcal{T}$. Then we can enumerate $\mathrm{succ}_\alpha(a)$ in $\mathcal{T}$ by stepping through the transitions in $\mathcal{S}$ as indicated by Prop. 5.18(i). Note that by Assumption 1(i) the idle transitions are contained in Δ and therefore directly available for this computation.

(ii) This similarly follows from Prop. 5.18(ii): in addition consider Prop. 5.1(ii).□

Proof. [Prop. 5.6] (i) A polynomial-time algorithm follows directly from the definition of $\equiv_D$ when considering Prop. 5.1(iv) and that for pairs $(X, \alpha) \in \Gamma$, α is small.

(ii) This follows from (i) and Prop. 5.4(i). □

Proof. [Claim 5.8] Let $\alpha, \beta \in \{X_1, \ldots, X_i\}^\otimes$. If $\alpha \equiv_{D_i} \beta$ then by Property (2) and $\mathrm{gnrb}(\equiv)$ being a congruence we can conclude $(\alpha, \beta) \in \mathrm{gnrb}(\equiv)$. For the opposite direction, assume $(\alpha, \beta) \in \mathrm{gnrb}(\equiv)$. By Property (2) and $\mathrm{gnrb}(\equiv)$ being a congruence we obtain $(\alpha', \beta') \in \mathrm{gnrb}(\equiv)$ where α' is the prime decomposition of α wrt. D_i, and similarly for β'. Then $\alpha \equiv_{D_i} \beta$ follows by Property (1) and unique decomposition of $\mathrm{gnrb}(\equiv)$. □

Proof. [Claim 5.9] Let $\alpha = P_1^{x_1} \ldots P_r^{x_r}$. For the only-if implication, assume that $(X_{i+1}, \alpha) \in \mathrm{gnrb}(\equiv)$. By Prop. 2.10 this gives us $X_{i+1} \equiv \alpha$ and $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\mathrm{gnrb}(\equiv))$. We will show how the latter implies $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\equiv_{D_i} \cup \{(X_{i+1}, \alpha)\})$.

By Prop. 2.4 we know: if $X_{i+1} \xrightarrow{a}_{\mathrm{n-r}} \beta$ then either $\beta = X_{i+1}$ or $\beta \in \{X_1, \ldots, X_i\}^{\otimes}$; and, if $\alpha \xrightarrow{a}_{\mathrm{n-r}} \gamma$ then $\gamma \in \{X_1, \ldots, X_i\}^{\otimes}$. Thus, using Claim 5.8 we deduce that (X_{i+1}, α) is element of the set:

$$\mathrm{n-r-exp}(\equiv_{D_i} \cup \{(X_{i+1}, \gamma) : \gamma \in \{X_1, \ldots, X_i\}^{\otimes}, (X_{i+1}, \gamma) \in \mathrm{gnrb}(\equiv)\}).$$

By unique decomposition of $\mathrm{gnrb}(\equiv)$ there can only be one $\gamma \in \{X_1, \ldots, X_i\}^{\otimes}$ satisfying $(X_{i+1}, \gamma) \in \mathrm{gnrb}(\equiv)$, and this γ must be α. Hence, we indeed obtain that $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\equiv_{D_i} \cup \{(X_{i+1}, \alpha)\})$.

To prove the if implication, assume $X_{i+1} \equiv \alpha$ and $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\equiv_{D_i} \cup\{(X_{i+1}, \alpha)\})$. By Claim 5.8 we obtain $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\mathrm{gnrb}(\equiv) \cup \{(X_{i+1}, \alpha)\})$. But then $\mathrm{gnrb}(\equiv) \cup \{(X_{i+1}, \alpha)\}$ is a norm-reducing bisimulation contained in $\equiv$, and $(X_{i+1}, \alpha) \in \mathrm{gnrb}(\equiv)$ is immediate because $\mathrm{gnrb}(\equiv)$ is the greatest such relation. □

Proof. [Claim 5.10] We need to investigate whether there is $\alpha = P_1^{x_1} \ldots P_r^{x_r}$ such that $X_{i+1} \equiv \alpha$ and $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\equiv_{D_i} \cup \{(X_{i+1}, \alpha)\})$. Since $|X_{i+1}| > 0$ there must be some norm-reducing transition $X_{i+1} \xrightarrow{a}_{\mathrm{n-r}} \beta$ such that $\beta \in \{X_1, \ldots, X_i\}^{\otimes}$. Then α must be in the following candidate set: $\{P_1^{y_1} \ldots P_r^{y_r} : |P_1^{y_1} \ldots P_r^{y_r}| = |X_{i+1}|, P_1^{y_1} \ldots P_r^{y_r} \xrightarrow{a}_{\mathrm{n-r}} \beta', \text{ and } \beta' \equiv_{D_i} \beta\}$. Below we will show:

Claim 5.19 *The candidate set can be enumerated in $\mathcal{T}$.*

Then, for each process α in the candidate set, check if

- $X_{i+1} \equiv \alpha$, and
- $(X_{i+1}, \alpha) \in \mathrm{n-r-exp}(\equiv_{D_i} \cup \{(X_{i+1}, \alpha)\})$.

The first condition is decidable in $\mathcal{T}$ by assumption since each process in the candidate set is small. The second condition amounts to checking whether for all actions a, the processes from $\mathrm{n-r-succ}_a(X_{i+1})$, and $\mathrm{n-r-succ}_a(\alpha)$ respectively, induce the same set of prime decompositions wrt. D_i. This is computable in $\mathcal{T}$ by Prop. 5.4(ii) and Prop. 5.6(i).

It remains to prove the above claim. Let $\mathcal{S}$ be a sufficient set for P that can be enumerated in $\mathcal{T}$. Observe that by Prop. 5.18(ii) there are only as many possibilities for α as there are transitions in $\mathcal{S}$. Considering this proposition, we can enumerate the candidate set in $\mathcal{T}$ as follows. Let β_p be the prime decomposition wrt. D_i of β. For each transition $\alpha' \overset{a'}{\twoheadrightarrow} (\alpha'_1, \ldots, \alpha'_m) \in \mathcal{S}$ do: check whether (1) $a = a'$, (2) α' is a prime decomposition, and (3) $|\alpha'| - |\alpha'_m| = |X_{i+1}| - |\beta|$. Further check whether the prime decomposition wrt. D_i of α'_m, say α'_{mp}, is contained in β_p (as a multiset). If all these checks are successful, in β_p substitute α' for α'_{mp}, and output the result as an element of the candidate set. □

Proof. [Lemma 5.11] (i) The *iff*-direction is obvious. To prove the *if*-direction suppose $\Gamma \subseteq \exp(\equiv_D)$. Assuming $\alpha \equiv_D \beta$ we will show $(\alpha, \beta) \in \exp(\equiv_D)$. Let γ be the unique decomposition of α and β into primes: γ is obtained from α and β by substituting all their non-prime variables X_i by γ_i where $(X_i, \gamma_i) \in \Gamma$. Since each pair of Γ satisfies expansion in $\equiv_D$, by Prop. 2.6(ii) this must also hold for the pair (α, β).

(ii) This is a consequence of the first part and Prop. 5.6(ii). □

Proof. [Lemma 5.12] (i) To see that $\equiv_D \cap \exp(\equiv_D)$ is a norm-preserving congruence consider Prop. 2.6(ii) and that $\equiv_D$ is a norm-preserving congruence. Decidability in $\mathcal{T}$ follows from Prop. 5.6(i) and (ii).

(ii) Let $D = (\Pi, \Gamma)$ where $\Pi = P_1, \ldots, P_r$. If $\equiv_D$ is not a bisimulation then by Lemma 5.11, $\Gamma \not\subseteq \exp(\equiv_D)$. Let X_i be the variable of smallest index such that $(X_i, P_1^{x_1} \ldots P_r^{x_r}) \in \Gamma$ but $(X_i, P_1^{x_1} \ldots P_r^{x_r})$ does not satisfy expansion in $\equiv_D$. We will show that X_i is prime wrt. the congruence $\equiv_D \cap \exp(\equiv_D)$.

Suppose to the contrary that $X_i \equiv_D \alpha$ and (X_i, α) satisfies expansion in $\equiv_D$, for some $\alpha \in \{X_i, \ldots, X_{i-1}\}^{\otimes}$. By minimality of i, for every X_j appearing in α, if $(X_j, P_1^{y_1} \ldots P_r^{y_r}) \in \Gamma$ then this pair satisfies expansion in $\equiv_D$. Hence, by $X_i \equiv_D \alpha$ and Prop. 2.6(ii) the pair $(\alpha, P_1^{x_1} \ldots P_r^{x_r})$ also satisfies expansion in $\equiv_D$, and by transitivity, so does $(X_i, P_1^{x_1} \ldots P_r^{x_r})$. But this contradicts the choice of X_i. □

6 Results for Concrete Cases

Finally we apply Theorem 5.17 to obtain results for the four concrete process semantics introduced in Section 3.

Theorem 6.1

(i) Strong bisimilarity on normed BPP$_\tau$ is decidable in PTIME.

(ii) Distributed bisimilarity on BPP$_\tau$ is decidable in PTIME.

Proof. As explained in Section 3, the two process semantics are captured by the normed CRS' represented by a BPP$_\tau$ process definition in standard form, and distributed standard form respectively, extended with appropriate idle rules. In each case presentation clearly satisfies Assumption 1. The two presentation mappings are PTIME-effective. A sufficient set of transitions can easily be enumerated in PTIME since according to CCS synchronization at most two non-idle transitions can be synchronized: for each pair of complementarily labelled rules in Γ add a τ transition that represents the synchronization of the two rules. □

Theorem 6.2

(i) Weak bisimilarity on totally normed BPP is decidable in PSPACE.

(ii) Branching bisimilarity on totally normed BPP is decidable in PSPACE.

Let Δ be a totally normed BPP definition. In the following, we exhibit two normed CRS' that characterize the two process semantics, are given by a PSPACE-effective presentation mapping. We will make use of a result by Esparza about communication-free Petri nets. (An analogous result for commutative context-free grammars was first obtained in [7].)

Theorem 6.3 (Theorem 3.2 of [4]) *The reachability problem for communication-free Petri nets is NP-complete.*

It is well-known that communication-free Petri nets exactly correspond to BPP in standard form [4]. Hence, the theorem carries over to processes of Δ: given processes α, β, whether β can be reached from α by some sequence of transitions $\alpha \xrightarrow{a_1} \ldots \xrightarrow{a_n} \beta$ is NP-complete. Further, if we adapt the decision procedure by only considering the τ-labelled rules of Δ we obtain:

Corollary 6.4 *Given two processes α, β, whether $\alpha\,(\xrightarrow{\tau})^*\beta$ can be decided in NP (in the sizes of Δ, α, and β).*

6.1 Branching Bisimilarity

Let P be the normed CRS, represented by Δ, that captures branching bisimilarity on totally normed BPP (c.f. Section 3.4). Extend Δ by the idle rule $(X, *, X, X)$ for each variable X; this ensures Assumption 1 is satisfied. We will show that this presentation mapping is PSPACE-effective. Recall that P is based on a particular notion of semi-branching bisimulation [1], which only considers transitions

$$\beta\ (\xrightarrow{\tau})^*\ \beta_1 \xrightarrow{(a)} \beta_2. \tag{3}$$

This kind of bisimulation requires a response (3) to each move $\alpha \xrightarrow{a} \alpha_2$, whenever a pair (α, β) is related, such that the pairs (α, β_1) and (α_2, β_2) are related too. This implies that $|\alpha| = |\beta_1|$, and therefore $|\beta_1| = |\beta|$. Knowing this, we assume that the additional requirement

$$|X| = |\alpha_1|$$

was imposed on the rewrite rules of P (cf. (2) in Section 3.4). This ensures that if β is small the resulting processes β_1 and β_2 in (3) will remain representable in polynomial space. Using Corollary 6.4 it is then straightforward to enumerate a sufficient set for P.

Lemma 6.5 *The presentation mapping of branching bisimilarity is PSPACE-effective.*

Proof. Given Δ, one can enumerate the set of all non-idle transitions of P that start from a small process in the following way. Step through every pair of small processes α, β_1. Test whether $|\alpha| = |\beta_1|$ and, using Corollary 6.4, whether $\alpha\,(\xrightarrow{\tau})^*\beta_1$. If both tests are successful then: (1) output $\alpha \overset{'\tau^{*\prime}}{\twoheadrightarrow} (\beta_1, \beta_1)$; (2) output $\alpha \overset{'\tau^*\tau'}{\twoheadrightarrow} (\beta_1, \beta_1)$; (3) for every rule $(X, a, \beta) \in \Delta$ such that $X \sqsubseteq \beta_1$ output $\alpha \overset{'\tau^*a'}{\twoheadrightarrow} (\beta_1, (\beta_1 \setminus X)\beta)$. The enumerated set is clearly sufficient and can be computed in PSPACE. □

6.2 Weak Bisimilarity

The weak bisimilarity case is not as immediate. In particular, the normed CRS exhibited in Section 3 would only lead to an elementary decision procedure. Let Δ be a totally normed BPP definition. We can decide in NP whether there exists a weak transition between two given processes:

Lemma 6.6 *Given $a \in Act \cup \{\tau\}$, and processes α, β, whether $\alpha \overset{a}{\Longrightarrow} \beta$ can be decided in NP (in the sizes of Δ, α, and β).*

Proof. A transition labelled by τ induces a change of norm ≥ 0 and one labelled by $a \in Act$ a change of norm ≥ -1. Hence, we can decide the above problem as follows.

Guess processes α', β' such that $|\alpha'| \leq |\beta| + 1$ and $|\beta'| \leq |\beta|$. Using Corollary 6.4 check whether $\alpha\,(\xrightarrow{\tau})^*\,\alpha' \xrightarrow{a} \beta'\,(\xrightarrow{\tau})^*\beta$. Since α' and β' can be represented in space polynomial in the input, this clearly runs in NP. □

Since weak transitions can bring about an unbounded increase of norm, this lemma does not immediately help us to enumerate a sufficient set in PSPACE. As for branching bisimilarity we need a special characterization of weak bisimilarity, which allows us to only consider transitions with a small increase of norm. We use the concept of *stratified bisimilarity* by Stirling [16].

Any transition $\alpha \overset{a}{\Longrightarrow} \beta$ gives rise to a change of weak norm given by $|\beta| - |\alpha|$. As already observed in the previous proof, the change is greater or equal to -1, when $a \neq \tau$, and greater or equal to 0 otherwise. The application of a rule $(X \xrightarrow{a} \gamma) \in \Delta$ induces a change of norm $|\gamma| - |X|$ since norm is additive. Denote by K the maximal change of norm exhibited by a rule in Δ: $K = \max\{|\gamma| - |X| : X \xrightarrow{a} \gamma \in \Delta\}$. As the maximal norm of a variable is at most exponential in the size of Δ, K is also so. Stirling's insight is that, in the definition of weak bisimulation, it is sufficient to consider only those transitions that give rise to a change of norm not larger than K. We call such transitions *small weak transitions* in the sequel and use the following notation for $-1 \leq k \leq K$:

$$\alpha \overset{a}{\Longrightarrow}_k \beta \quad \text{iff} \quad \alpha \overset{a}{\Longrightarrow} \beta \text{ and } |\beta| - |\alpha| = k.$$

Based on the insight, we will now derive another normed CRS P that captures weak bisimilarity. This new P will be similar to the previous one; however, the presentation mapping will be, in contrast to the previous one, PSPACE-effective. The only new ingredient is the treatment of the indices k.

The new synchronization algebra contains more actions. We define $Act_W = \{{'\tau^* a \tau^{*}}'_k : a \in Act, -1 \leq k \leq K\} \cup \{{'\tau^*}'_k : 0 \leq k \leq K\}$. Similarly to before, the action names indicate the type of weak transition an action is thought to represent; k describes the change of norm induced by a given action. Operation $\bullet$ is also defined similarly to before, but in addition takes care of the k parameter:

$$\begin{array}{lllll} {'\tau^*}'_k \bullet {'\tau^* a \tau^*}'_l & = & {'\tau^* a \tau^*}'_l \bullet {'\tau^*}'_k & = {'\tau^* a \tau^*}'_{k+l} & \text{if } k+l \leq K \\ {'\tau^*}'_k \bullet {'\tau^*}'_l & = & {'\tau^*}'_{k+l} & & \text{if } k+l \leq K \end{array}$$

For all other combinations $\bullet$ is undefined.

Note that in the first line $-1 \leq k + l$ holds since $0 \leq k$. A weight function W is associated with Act_W analogously to before: $W(*) = 0$, $W({'\tau^*}'_k) = 0$, and $W({'\tau^* a \tau^*}'_k) = 1$ for all a and k.

Δ represents a CRS P whose set of variables is V (i.e., the same as in Δ), and whose set of rewrite rules Γ is defined by: for all $X \in V$, $a \in Act$,

$$\begin{array}{rl} (X, {'\tau^*}'_k, \alpha) \in \Gamma & \text{iff} \quad X \overset{\tau}{\Longrightarrow}_k \alpha \\ (X, {'\tau^* a \tau^*}'_k, \alpha) \in \Gamma & \text{iff} \quad X \overset{a}{\Longrightarrow}_k \alpha \\ (X, *, X) \in \Gamma. & \end{array}$$

Since Δ is totally normed, P can easily be made normed (c.f. Section 3). The transition relation $\twoheadrightarrow_P$ indeed captures the small weak transition relation of Δ:

$\alpha \overset{'\tau^{*'}_{k}}{\twoheadrightarrow} \beta$ iff $\alpha \overset{\tau}{\Longrightarrow}_k \beta$; and, $\alpha \overset{'\tau^{*}a\tau^{*'}_{k}}{\twoheadrightarrow} \beta$ iff $\alpha \overset{a}{\Longrightarrow}_k \beta$, $a \in Act$. We obtain:

Fact 6.7 *$\approx_P$ coincides with weak bisimilarity on processes of Δ.*

Remark 6.8 Note that Act_W is of size exponential wrt. the size of Δ, but since we do not consider the synchronization algebra as part of the input this does not affect our complexity result. All we need to show is that the presentation mapping is PSPACE-effective.

To ensure that Assumption 1 is satisfied extend Δ by the rule $(X, *, X)$ for each variable X. We are now able to apply Lemma 6.6 to show:

Lemma 6.9 *The presentation mapping of weak bisimilarity is PSPACE-effective.*

Proof. Given Δ, we can enumerate the set of all non-idle transitions of P starting from a small process in the following way.

For every $a \in Act \cup \{\tau\}$ and every pair of processes α, β such that α is small and $|\beta| \leq |\alpha| + K$, check whether $\alpha \overset{a}{\Longrightarrow} \beta$ using Lemma 6.6. If this is the case compute $k = |\beta| - |\alpha|$; output $\alpha \overset{'\tau^{*'}_{k}}{\twoheadrightarrow} \beta$ if $a = \tau$, and $\alpha \overset{'\tau^{*}a\tau^{*'}_{k}}{\twoheadrightarrow} \beta$ otherwise. The enumerated set is clearly sufficient and can be computed in PSPACE. (Recall that K is at most exponential in the size of Δ.) □

7 Conclusions

We have shown that the idea behind the classical algorithm of Hirshfeld et al. is applicable far beyond bisimilarity on normed BPP; indeed to any process semantics that is commutative and normed in the sense made precise by our notion of normed CRS. With distributed bisimilarity we have seen that normedness can come in many guises. Work in progress suggests that our approach is not restricted to parallel processes but may also be applied to the sequential case, e.g., to normed BPA.

We have phrased the algorithm in a way that allows us to combine it with the theorem of Luttik and van Oostrom, which captures when a partial commutative monoid has unique decomposition. Thus, we were able to put together two important results in process algebra, and demonstrate the applicability of both of them.

Furthermore, we have captured four a priori very different process semantics in a uniform framework and obtained previously unknown complexity results. In the case of distributed bisimilarity on BPP_τ and bisimilarity on normed BPP_τ this has made concrete a connection that has already been indicated by previous work [10,9,3]. On the other hand, we have brought to light unexpected structural similarities between observational equivalences and synchronization.

One motivation of this work was to see how far the classical methods can be extended to more realistic settings with synchronization and observational equivalences. We could extend them but there are limitations of course: we cannot capture forced synchronization and neither can we capture both synchronization and observational equivalences at the same time. The limits are set by the requirements on our synchronization alphabet. Further research is needed to investigate how, by suitable abstractions, the results could still be of use in control flow analysis.

References

[1] Basten, T., *Branching bisimilarity is an equivalence indeed!*, Information Processing Letters **58** (1996), pp. 141–147.

[2] Burkart, O., D. Caucal, F. Moller and B. Steffen, *Verification of infinite structures*, in: *Handbook of Process Algebra*, Elevier, 2001 pp. 545–623.

[3] Christensen, S., "Decidability and Decomposition in process algebras," Ph.D. thesis, Dept. of Computer Science, University of Edinburgh, UK (1993).

[4] Esparza, J., *Petri nets, commutative context-free grammars, and basic parallel processes*, Fundamenta Informaticae **31** (1997), pp. 13–26.

[5] Hirshfeld, Y., *Bisimulation trees and the decidability of weak bisimulations*, Electr. Notes Theor. Comput. Sci. **5** (1996).

[6] Hirshfeld, Y., M. Jerrum and F. Moller, *A polynomial time algorithm for deciding bisimulation equivalence of normed Basic Parallel Processes*, Mathematical Structures in Computer Science **6** (1996), pp. 251–259.

[7] Huynh, D. T., *Commutative grammars: The complexity of uniform word problems*, Information and Control **57** (1983), pp. 21–39.

[8] Jančar, P., *Bisimilarity of Basic Parallel Processes is PSPACE-complete*, in: *Proc. LICS'03*, 2003, pp. 218–227.

[9] Jančar, P. and M. Kot, *Bisimilarity on normed Basic Parallel Processes can be decided in time $O(n^3)$*, in: *Proc. AVIS'04, ENTCS* (2004).

[10] Jančar, P. and Z. Sawa, *On distributed bisimilarity over Basic Parallel Processes*, in: *Proc. AVIS'05, ENTCS* (2005).

[11] Lasota, S., *A polynomial-time algorithm for deciding true concurrency equivalences of Basic Parallel Processes*, in: *MFCS'03*, LNCS **2747**, 2003, pp. 521–530.

[12] Luttik, B. and V. van Oostrom, *Decomposition orders—another generalisation of the fundamental theorem of arithmetic*, Theor. Comp. Sci. **335** (2005), pp. 147–186.

[13] Mayr, R., "Decidability and Complexity of Model Checking Problems for Infinite-State Systems," Ph.D. thesis, Technische Universität München (1998).

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

[15] Srba, J., *Strong bisimilarity of simple process algebras: Complexity lower bounds*, Acta Informatica **39** (2003), pp. 469–499.

[16] Stirling, C., *Decidability of weak bisimilarity for a subset of basic parallel processes*, in: *FOSSACS'01*, LNCS 2030, 2001, pp. 379–393.

[17] van Glabbeek, R. and W. Weijland, *Branching time and abstraction in bisimulation semantics*, Journal of the ACM **43** (1996), pp. 555–600.

[18] Winskel, G., *Synchronization trees*, Theor. Comput. Sci. **34** (1984), pp. 33–82.