Institutions
Markets
Technologies
IMT
INSTITUTE
FOR ADVANCED
STUDIES
LUCCA

#09
2013

IMT LUCCA CSA TECHNICAL
REPORT SERIES 09
August 2013

RA **Computer Science and Applications**

Dimming Relations for the Efficient Analysis of Concurrent Systems via Action Abstraction

Rocco De Nicola
Giulio Iacobelli
Mirco Tribastone

IMT LUCCA CSA TECHNICAL REPORT SERIES #09/2013

Piazza San Ponziano 6, 55100 Lucca

Research Area
Computer Science and Applications

Dimming Relations for the Efficient Analysis of Concurrent Systems via Action Abstraction

Rocco De Nicola
IMT Institute for Advanced Studies Lucca

Giulio Iacobelli
Department for Informatics, Ludwig Maximilians University of Munich

Mirco Tribastone
Department for Informatics, Ludwig Maximilians University of Munich

Dimming Relations for the Efficient Analysis of Concurrent Systems via Action Abstraction

Rocco De Nicola[1], Giulio Iacobelli[2], and Mirco Tribastone[2]

[1] IMT — Institute for Advanced Studies Lucca, Italy
rocco.denicola@imtlucca.it
[2] Department for Informatics
Ludwig Maximilians University of Munich, Germany
{iacobelli,tribastone}@pst.ifi.lmu.de

Abstract. We study models of concurrency based on labelled transition systems where abstractions are induced by a partition of the action set. We introduce *dimming relations*, i.e., notions of behavioural equivalence which are able to relate two models if they can match each other's actions whenever they are in the same partition block. We show applicability to a number of situations of practical interest which are apparently heterogeneous but exhibit similar behaviors although manifested via different actions. Dimming relations make the models more homogeneous by collapsing such distinct actions into the same partition block. With our examples, we show how these abstractions permit reducing the state-space complexity from exponential to polynomial in the number of concurrent processes.

1 Introduction

Behavioural relations are powerful techniques to reason about models based on labelled transition systems. The classic notion of bisimulation relates two states P and Q such that any action performed by P can be matched by the same action from Q (and vice versa). Weak bisimulation exploits the fact that some behaviour is abstracted away by hiding the identity of certain actions that can be considered uninteresting at some desired level of detail [17]. Alternatively, in some situations, by exploiting specific structural properties, e.g. symmetries, it is possible to study the properties of a given model by considering another one with a smaller state-space size, thus reducing the cost of the analysis.

We propose an approach that lies between these two extremes. We study behavioural relations that take a coarse-grained view of a model, based on partitioning the set of actions that it can exhibit. Our first notion relates P and Q whenever Q is able to match some action a performed by P *with any action* in the same partition block of a (and vice versa). We call this *dimmed bisimulation* to highlight the fact that, based on the action partition employed, the modeller is able to see and reason about the original system at different levels of details and thus with different levels of accuracy. Clearly, dimmed bisimulation is less discriminating than bisimulation. For instance, using standard notation, let us

consider $a_1.\mathbf{0} + a_2.\mathbf{0}$: a simple process that offers a choice between two actions, a_1 and a_2, and then stops. This will be dimmed bisimilar to $a_1.\mathbf{0}$ if both a_1 and a_2 belong to the same partition block; evidently, $a_1.\mathbf{0} + a_2.\mathbf{0}$ is not strongly bisimilar to $a_1.\mathbf{0}$. This simple example shows that it is possible to obtain a more compact description when the modeller is content with reasoning at the level of partition representatives instead of considering detailed concrete actions. Indeed, we will present more substantial examples where this abstraction will be more significant. To provide some intuition behind this approach, let $\|$ denote a generic parallel operator and consider $a_1.\mathbf{0} \parallel a_2.\mathbf{0} \parallel \cdots \parallel a_n.\mathbf{0}$, e.g., a model of n independent threads of execution, each performing a distinct computation a_i, $1 \leq i \leq n$, and then stopping. The state space size of this (concrete) system is 2^n; however, allowing all a_i to be actions of the same partition block permits a dramatic reduction of the (abstract) state space to only $n+1$ states, where each of them tracks the number of processes that have not become inert ($\mathbf{0}$) yet.

An analogous state-space reduction could have been obtained by assuming that the actions in the same partition block represent unnecessary detail in the model. Thus, in $a_1.\mathbf{0} + a_2.\mathbf{0}$ one would replace a_1 and a_2 with *internal* τ actions, yielding $\tau.\mathbf{0}+\tau.\mathbf{0}$, which is bisimilar to $\tau.\mathbf{0}$. However, there are two major drawbacks with this approach. Firstly, in such a reduced form it is not possible to keep track of the concrete actions from which the internal ones are originated. Secondly, sometimes this abstraction may not be possible, for instance when a_1 and a_2 are intended to be used for synchronising with other processes.

More closely related to our behavioural relations is turning one action, say a_2, into another one. For instance, this could be formally obtained by considering the process $(a_1.\mathbf{0} + a_2.\mathbf{0})[f]$ where f is the relabelling function such that $f(a_1) = a_1$ and $f(a_2) = a_1$. Indeed, we provide a characterisation of dimmed bisimulation based on the existence of an appropriate f such that $P[f]$ is bisimilar, in the classical sense, to $Q[f]$. Thus, dimmed bisimulation is able to relate processes that behave bisimilarly after an appropriate relabelling of their actions.

A highly desirable property of any behavioural relation for process algebra is establishing that it is preserved by its operators. This is not only of theoretical interest, but it is very useful in practice in order to infer relations in a compositional fashion, starting from the simplest constituting communicating processes, for which they are typically not difficult to establish. Indeed, this will be the preferred way in which we will treat our worked examples. Here we study a CSP-like process algebra (see, e.g., [19]) where the parallel operator is parametrised by a synchronisation action set. We find that dimmed bisimulation is preserved if certain syntactic conditions are met. For instance, for parallel composition the action set must enjoy the so-called property of *singleton coherence*. Roughly speaking, this requires that the partitions of the actions in the synchronisation set be singletons. That is, dimmed bisimulation can be used compositionally if all non-trivial partition blocks consist of actions that are performed independently.

Unfortunately, the restriction of singleton coherence can rule out the possibility of compositional reasoning for some models of practical interest. For instance, using again the simple example discussed above, $a_1.\mathbf{0} + a_2.\mathbf{0}$ could not be placed

in any context where both a_1 and a_2 are synchronisation actions. However, this appears to be an attractive feature, because it represents a typical modelling pattern whereby a process is able to offer two (or more) options of different kinds to the environment. In order to capture this situation, we introduce *dimmed simulation* (whereby, analogously to the classic notion of simulation, any action performed by P can be matched by Q with any other action in the same partition block, but not vice versa). It turns out that this is preserved by parallel composition under the much more relaxed assumption of *block coherence*, essentially requiring that a whole partition block must be included in a synchronisation set if there is at least one action of the partition block in it.

We will show the applicability of our dimming relations by studying a number of models which exhibit some form of *heterogeneity*, realised by expressing analogous behaviour with distinct actions.

We will study a simple model of fork/join synchronisation mechanism. This can be relevant, for instance, to capture certain aspects of novel programming paradigms such as MapReduce (see, e.g., [8]) in order to support large-scale concurrent and distributed systems. Here, a large computational task is handled by a master process, which forks several independent children threads. Each thread is responsible for a distinct sub-task (modelled, for instance, by a distinct action type); when all thread finish, the control passes again to the master process which finalises the whole task.

In a producer/consumer system mediated by a buffer, the existence of different classes of items may be captured by having the buffer expose distinct pairs of *put* and *get* actions for each item class. In these cases, the state-space sizes of such models grow exponentially with the number of components in the model.

Our dimming relations allow us to consider processes that can be more easily analysed by making them more *homogeneous*, provided that the modeller accepts a coarser-grained view of the system induced by action partitioning. For example, the modeller may be content with ensuring that the fork/join model only captures that *some* sub-task has been completed by *some* thread; or that some action *put* will enable some action *get* in any buffer place, without necessarily wanting to know *which* specific thread has completed or which specific item class has been handled. In all our examples, it will turn out that it is possible to obtain simpler dimmed (bi-)similar processes with state-space sizes of polynomial complexity rather than exponential.

Paper outline. Section 2 introduces our process algebra of interest and discusses dimmed bisimulation. It presents its characterisation with respect to action relabelling and the compositionalilty properties. Section 3 applies these results to a number of case studies. Section 4 studies dimmed simulation, similarly to dimmed bisimulation. Section 5 is concerned with *weak* extensions of both dimmed relations. Section 6 discusses related work, while Section 7 briefly concludes. Unless otherwise stated, all proofs are given in the appendix.

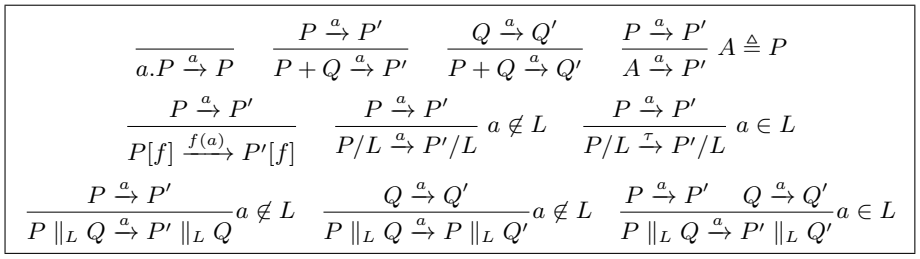

Fig. 1. Process algebra semantics.

2 Dimmed Bisimulation

We carry out our investigation in the context of a process algebra with CSP-style semantics. However, with appropriate changes our ideas of dimmed relations carry over to other synchronisation operators such as the binary one in CCS [17].

Definition 1 (Process Algebra Syntax). *Let $\mathcal{A}$ be a set of actions and $\mathcal{L} = \mathcal{A} \cup \{\tau\}$, with $\tau \notin \mathcal{A}$, be the set of labels. Our process algebra has the syntax*

$$P ::= \mathbf{0} \mid a.P \mid P + P \mid K \mid P[f] \mid P \parallel_L P \mid P/L$$

where $a \in \mathcal{L}$, $K \in \mathcal{K}$, with $\mathcal{K}$ the set of constants and $K \triangleq P$, $f : \mathcal{L} \to \mathcal{L}$ is a relabelling function with $f(\tau) = \tau$, and $L \subseteq \mathcal{A}$. Let $\mathcal{P}$ be the set of processes.

We use standard syntax, thus: τ is the internal action; $\mathbf{0}$ is the inert process, that does nothing; $a.P$ denotes *prefixing*, a process that can perform an a-action and become P; $P + P$ offers a *choice* between behaviours; K is a *constant*, used for recursion; $P[f]$ is a process to which a *relabelling* of its actions is applied (where τ cannot be relabelled); $P \parallel_L P$ is the generalised *parallel* operator, whereby the two operands are required to synchronise only over the actions that are in the set L; P/L models *hiding*, whereby an action performed by P is made internal if it is in L. These rules are captured by the structured operational semantics in Fig. 1.

Notation 1 *Throughout the paper, we let $\mathfrak{F} = \{\mathcal{F}_1, \ldots, \mathcal{F}_m\}$ denote a partition of $\mathcal{A}$. If $a \in \mathcal{A}$ we let $[a]_{\mathfrak{F}}$ denote the partition block $\mathcal{F}_i$ such that $a \in \mathcal{F}_i$; when $\mathfrak{F}$ is clear from the context, we omit the subscript and simply write $[a]$. We extend the notation $[\cdot]$ to τ by setting $[\tau] = \{\tau\}$.*

Definition 2. *Let $\mathfrak{F}$ be a partition of $\mathcal{A}$. We say that $P \xrightarrow{[a]} P'$ iff there exists $b \in [a]$ such that $P \xrightarrow{b} P'$.*

According to this definition and to the previous notation, if $P \xrightarrow{\tau} P'$ then we write $P \xrightarrow{[\tau]} P'$. Intuitively, Definition 2 gives us the *dimmed behaviour* of a process P: if it can make a *concrete* action b then we say that it can make an

abstract action $[a]$, which essentially stands for saying that "P can make *any* of the actions of the partition block to which b belongs." Our notion of dimmed bisimulation allows processes to match each other's actions so long as they are in the same partition block.

Definition 3 (Dimmed Bisimulation). *Given a partition $\mathfrak{F}$, a binary relation $\mathcal{R}$ over $\mathcal{P}$ is an* $\mathfrak{F}$-dimmed bisimulation *iff whenever $(P,Q) \in \mathcal{R}$ and $a \in \mathcal{L}$:*

- *if $P \xrightarrow{[a]} P'$ then $Q \xrightarrow{[a]} Q'$ and $(P',Q') \in \mathcal{R}$;*
- *if $Q \xrightarrow{[a]} Q'$ then $P \xrightarrow{[a]} P'$ and $(P',Q') \in \mathcal{R}$.*

Two processes P, Q are $\mathfrak{F}$-dimmed bisimilar*, written $P \sim_{\mathfrak{F}} Q$, iff there exists an $\mathfrak{F}$-dimmed bisimulation that relates them.*

Let us stress that the classic notion of bisimulation, hereafter denoted by $\sim$, can be recovered by choosing $\mathfrak{F}$ as the trivial singleton partition (which yields $[a] = \{a\}$ for all $a \in \mathcal{A}$). With an appropriate choice of the action partition, it is possible to find dimmed bisimilar processes that are easier to analyse. For instance, using again the simple process presented in Section 1, it holds that $a_1.\mathbf{0} + a_2.\mathbf{0} \sim_{\mathfrak{F}} a_1.\mathbf{0}$ (and also that $a_1.\mathbf{0} + a_2.\mathbf{0} \sim_{\mathfrak{F}} a_2.\mathbf{0}$) if $\{a_1, a_2\} \in \mathfrak{F}$. This is because $\mathcal{R} = \{(a_1.\mathbf{0} + a_2.\mathbf{0}, a_1.\mathbf{0}), (\mathbf{0},\mathbf{0})\}$ is an $\mathfrak{F}$-dimmed bisimulation. For instance, both $a_1.\mathbf{0}+a_2.\mathbf{0} \xrightarrow{a_1} \mathbf{0}$ and $a_1.\mathbf{0}+a_2.\mathbf{0} \xrightarrow{a_2} \mathbf{0}$ imply that $a_1.\mathbf{0}+a_2.\mathbf{0} \xrightarrow{[a_1]} \mathbf{0}$ and $a_1.\mathbf{0} \xrightarrow{[a_1]} \mathbf{0}$ since $a_1.\mathbf{0} \xrightarrow{a_1} \mathbf{0}$. Please notice that, in this particular case, the dimmed bisimilar process has the same number of states, but fewer transitions. Later in this paper we will provide examples where also the number of states will be reduced.

As expected, similarly to classical bisimulation, the following holds.

Theorem 1. *For any partition $\mathfrak{F}$, the relation $\sim_{\mathfrak{F}}$*

a) is an equivalence relation;
b) is the largest $\mathfrak{F}$-dimmed bisimulation;
c) satisfies the following property: $P \sim_{\mathfrak{F}} Q$ iff, for any $a \in \mathcal{L}$,
- *if $P \xrightarrow{[a]} P'$ then $Q \xrightarrow{[a]} Q'$ and $P' \sim_{\mathfrak{F}} Q'$;*
- *if $Q \xrightarrow{[a]} Q'$ then $P \xrightarrow{[a]} P'$ and $P' \sim_{\mathfrak{F}} Q'$.*

Let us observe that, with $\{a_1, a_2\} \in \mathfrak{F}$, it is also possible to establish the relation $a_1.\mathbf{0}+a_2.\mathbf{0} \sim_{\mathfrak{F}} a_1.\mathbf{0}+a_1.\mathbf{0}$. This intuitively suggests that dimmed bisimulation can be understood as a form of relabelling that turns actions of a partition block into possibly distinct actions of within the same partition block. Indeed, in this simple case we have that $(a_1.\mathbf{0} + a_2.\mathbf{0})[f] \sim (a_1.\mathbf{0} + a_1.\mathbf{0})[f]$ for f such that $f(a_1) = a_1$ and $f(a_2) = a_1$.

Indeed, dimmed bisimulation can be characterised in terms of actions relabelling. We start with identifying functions that relabel within the same block of $\mathfrak{F}$.

Definition 4. *A relabelling function $f : \mathcal{L} \to \mathcal{L}$ is said to be* partition-preserving *(pp) for $\mathfrak{F}$ iff $\forall \mathcal{F} \in \mathfrak{F}$ and $\forall a \in \mathcal{F}$ it holds that $f(a) \in \mathcal{F}$.*

Theorem 2 (Characterisation of Dimmed Bisimulation via relabelling). *Given a partition $\mathfrak{F}$, $P \sim_{\mathfrak{F}} Q$ iff there exists a pp-function f for $\mathfrak{F}$ such that $P[f] \sim Q[f]$.*

We now study whether dimmed bisimulation is preserved by the operators of our process algebra. For hiding and parallel composition, that are parameterized by an action set, we need to impose syntactic restrictions on the actual action sets, which we call *block coherence* and *singleton coherence*, respectively. Roughly speaking, block coherence requires that an element of $\mathfrak{F}$ is either completely in the action set, or completely outside. Singleton coherence, instead, requires that each action belonging to the action set cannot be aggregated with other actions.

Definition 5. *Given a partition $\mathfrak{F}$ and $L \subseteq \mathcal{A}$, we say that L is*

- block coherent *with $\mathfrak{F}$ iff $\forall \mathcal{F} \in \mathfrak{F}$ such that $\mathcal{F} \cap L \neq \emptyset$ we have $\mathcal{F} \subseteq L$.*
- singleton coherent *with $\mathfrak{F}$ iff $\forall \mathcal{F} \in \mathfrak{F}$ such that $\mathcal{F} \cap L \neq \emptyset$ we have $|\mathcal{F}| = 1$.*

For instance, let $\mathfrak{F} = \big\{\{a_1, a_2\}, \{b\}\big\}$. Then $\{b\}$ is singleton coherent with $\mathfrak{F}$, whereas $\{a_1, a_2\}$ is not. However, $\{a_1, a_2\}$ is block coherent with $\mathfrak{F}$. Furthermore, every action set L is singleton coherent with the trivial singleton partition of the action set; finally, the empty set is singleton coherent with any $\mathfrak{F}$.

Theorem 3 (Compositionality for Dimmed Bisimulation). *Let P and Q be two processes such that $P \sim_{\mathfrak{F}} Q$. Then it holds that:*

i) $a.P \sim_{\mathfrak{F}} b.Q$ for any a, b such that $[a] = [b]$;
ii) $P + R \sim_{\mathfrak{F}} Q + S$ for any two processes R and S such that $R \sim_{\mathfrak{F}} S$;
iii) $P[g] \sim_{\mathfrak{F}} Q[g]$ for any function g that is partition preserving for $\mathfrak{F}$;
iv) $P/L \sim_{\mathfrak{F}} Q/L$ for L block coherent with $\mathfrak{F}$;
v) $P \parallel_L R \sim_{\mathfrak{F}} Q \parallel_L S$ if $R \sim_{\mathfrak{F}} S$ and L is singleton coherent with $\mathfrak{F}$.

In general, dimmed bisimulation is not preserved if the above syntactical restrictions are not satisfied. To see this, let us take, for instance, processes $a_1.\mathbf{0}$ and $a_2.\mathbf{0}$, and some partition $\mathfrak{F}$ such that $\{a_1, a_2\} \subseteq [a_1]$. Then it holds, that $a_1.\mathbf{0} \sim_{\mathfrak{F}} a_2.\mathbf{0}$. For relabelling, let $b_1 \notin [a_1]$ and define f such as $f(a_1) = b_1$, $f(a_2) = a_2$, and $f(b_1) = b_1$, whence f is not a pp-function. Then, it holds that $a_1.\mathbf{0}[f] \not\sim_{\mathfrak{F}} a_2.\mathbf{0}[f]$. For hiding, $a_1.\mathbf{0}/L \not\sim_{\mathfrak{F}} a_2.\mathbf{0}/L$ if $L = \{a_1\}$. Finally, dimmed bisimulation is not preserved by parallel composition if the action set is not singleton coherent with $\mathfrak{F}$. For instance, $a_1.\mathbf{0} \parallel_L a_1.\mathbf{0} \not\sim_{\mathfrak{F}} a_2.\mathbf{0} \parallel_L a_1.\mathbf{0}$.

Working with dimmed bisimilar candidates. The following fact establishes a clear relationship between a process P and the process obtained by inserting P in a context making use of a relabelling pp-function. In practice, it allows us to find a nontrivial (i.e., nonidentical) candidate bisimulating process.

Proposition 1. *Let $\mathfrak{F}$ be a partition and f be a pp-function for $\mathfrak{F}$. Then it holds that $P \sim_{\mathfrak{F}} P[f]$.*

Proof. The relation $\{(P', P'[f]) \mid f \text{ is a pp-function for } \mathfrak{F}\}$ is an $\mathfrak{F}$-dimmed bisimulation. □

Notice that, in general, the state space of P is as large as that of $P[f]$. However, in some cases, it may be easier to find a smaller process by studying how the pp-function f distributes over the process algebra operators.

Proposition 2. *Let f be a pp-function for $\mathfrak{F}$. It holds that*

i) $(P[g])[f] \sim_{\mathfrak{F}} P[f]$ *for any function g that is partition preserving for $\mathfrak{F}$;*
ii) $(P/L)[f] \sim_{\mathfrak{F}} (P[f])/L$ *for L block coherent with $\mathfrak{F}$;*
iii) $(P \parallel_L Q)[f] \sim_{\mathfrak{F}} P[f] \parallel_L Q[f]$ *if L is singleton coherent with $\mathfrak{F}$.*

Interestingly, dimmed bisimilarity behaves rather differently than bisimilarity with respect to distributivity. For instance, in *i)* the information of the pp-function g is lost, while bisimilarity uses function composition $f \circ g$; the analogous statement to *ii)* for bisimilarity requires a similar form of coherence for L, i.e. $a \in L$ iff $f(a) \in L$; finally, *iii)* uses a weaker assumption on f than bisimilarity, for which f must be injective, but has a syntactic restriction on the synchronisation set, unlike bisimilarity. In fact, singleton coherence and partition preservation coincide with requiring that f be injective on the synchronised actions.

Let us also remark that, in general, distributivity may not be preserved when the side conditions are not satisfied. For instance:

- $(a_1.\mathbf{0}[g])[f] \not\sim_{\mathfrak{F}} (a_1.\mathbf{0})[f]$ if $g(a_1) = b$, with $b \notin [a_1]$, and $f(a_1) = a_1$ and $f(b) = b$;
- $\big((a_1.\mathbf{0}+a_2.\mathbf{0})/\{a_1\}\big)[f] \not\sim_{\mathfrak{F}} \big((a_1.\mathbf{0}+a_2.\mathbf{0})[f]\big)/\{a_1\}$ if $f(a_2) = a_1$ and $f(a_1) = a_1$, with $\{a_1, a_2\} \in \mathfrak{F}$;
- $(a_1.\mathbf{0} \parallel_{\{a_1,a_2\}} a_2.\mathbf{0})[f] \not\sim_{\mathfrak{F}} (a_1.\mathbf{0})[f] \parallel_{\{a_1,a_2\}} (a_2.\mathbf{0})[f]$, with the same f as above.

Different levels of dimming. So far, all our results have assumed a given fixed partition of the action set. Now we turn to considering what can be said for models with different levels of dimming, induced by different partitions. In general, for any two partitions, establishing dimmed bisimilarity with one partition does not allow us to infer dimmed bisimilarity for the other. For example, $a_1.\mathbf{0}+a_2.\mathbf{0}+b.\mathbf{0} \sim_{\mathfrak{F}_1} a_1.\mathbf{0}+a_2.\mathbf{0}$ for $\mathfrak{F}_1 = \big\{\{a_1, b\}, \{a_2\}\big\}$; but $a_1.\mathbf{0}+a_2.\mathbf{0}+b.\mathbf{0} \not\sim_{\mathfrak{F}_2} a_1.\mathbf{0} + a_2.\mathbf{0}$ if $\mathfrak{F}_2 = \big\{\{a_1, a_2\}, \{b\}\big\}$. However, it turns out that the usual partial order based on *partition refinement* captures the intuitive idea that one partition provides a higher level of abstraction than the other. Formally, given two partitions $\mathfrak{F}_1$ and $\mathfrak{F}_2$, one says that $\mathfrak{F}_1$ is a *refinement* of $\mathfrak{F}_2$, written as $\mathfrak{F}_1 \leq \mathfrak{F}_2$, if every element of $\mathfrak{F}_1$ is a subset of an element of $\mathfrak{F}_2$. In this case, it is also said that $\mathfrak{F}_1$ is *finer* than $\mathfrak{F}_2$ and that $\mathfrak{F}_2$ is *coarser* than $\mathfrak{F}_1$.

Proposition 3. *Let $\mathfrak{F}_1$, $\mathfrak{F}_2$ be two partitions of $\mathcal{A}$ such that $\mathfrak{F}_1 \leq \mathfrak{F}_2$ and let P and Q be two processes such that $P \sim_{\mathfrak{F}_1} Q$; then it holds that $P \sim_{\mathfrak{F}_2} Q$.*

Proof. $\mathfrak{F}_1 \leq \mathfrak{F}_2$ entails that $[a]_{\mathfrak{F}_1} \subseteq [a]_{\mathfrak{F}_2}$ for any $a \in \mathcal{A}$. Thus $\mathcal{R} \triangleq \{(P', Q') \mid P' \sim_{\mathfrak{F}_1} Q'\}$ contains (P, Q) and is an $\mathfrak{F}_2$-dimmed bisimulation.

Let us denote by $\mathfrak{F}_0$ the trivial singleton partition, for which it holds that $\mathfrak{F}_0 \leq \mathfrak{F}$ for any $\mathfrak{F}$. Since $\sim = \sim_{\mathfrak{F}_0}$, as a corollary of the above proposition we have the following.

Corollary 1. $\sim$ *implies* $\sim_{\mathfrak{F}}$ *for any partition* $\mathfrak{F}$.

3 Dimmed Bisimulation at Work

In this section we present how to exploit dimmed bisimulation in four examples of typical modelling patterns of distributed systems. In all cases, Corollary 1 will be used in order to find dimmed bisimilar processes that are much easier to analyse than the original ones. Given some process P, the idea is to first find some Q such that $P \sim_{\mathfrak{F}} Q$; then, using well-known algorithms [18], one reduces Q up to bisimulation into R, from which it holds that $P \sim_{\mathfrak{F}} R$. For instance, the relation $a_1.\mathbf{0} + a_2.\mathbf{0} \sim_{\mathfrak{F}} a_1.\mathbf{0}$ can be interpreted as establishing first that $a_1.\mathbf{0} + a_2.\mathbf{0} \sim_{\mathfrak{F}} a_1.\mathbf{0} + a_1.\mathbf{0}$, and then observing that $a_1.\mathbf{0} + a_1.\mathbf{0} \sim a_1.\mathbf{0}$.

Our first example studies a concurrent system with fork/join synchronisation, which could be used, as discussed, for a high-level model of a MapReduce computing task [8].

Example 1 (A Fork/Join System). Let us consider $\mathcal{A} = \{fork, join\} \cup \{w_i \mid 1 \leq i \leq n\}$. Our model consists of a master process, denoted by F, which invokes (*fork*) n worker threads. Each worker thread, W_i, performs a distinct type of computation, modelled as a distinct action w_i. Once all threads finish their task, the master process collects the results (*join*) and repeats the cycle invoking the threads again. The model is as follows.

$$F \triangleq fork.join.F \qquad W_i \triangleq fork.w_i.join.W_i \qquad 1 \leq i \leq n.$$
$$M := F \parallel_L W_1 \parallel_L \cdots \parallel_L W_n, \qquad L = \{fork, join\}.$$

Model M has a state space size that grows as $2^n + 1$. In addition, it cannot be further minimised up to bisimulation. Dimmed bisimulation, on the other hand, yields a simpler model which enjoys linear complexity of the state space size. To show this, let us consider $\mathfrak{F} = \big\{\{fork\}, \{join\}, \{w_i \mid 1 \leq i \leq n\}\big\}$, with which L is singleton coherent. We have that $W_i \sim_{\mathfrak{F}} W_1$ for all $1 \leq i \leq n$. Thus, item *v)* of Theorem 3 may be applied to yield that $M \sim_{\mathfrak{F}} \bar{M}$, with

$$\bar{M} := F \parallel_L \underbrace{W_1 \parallel_L \cdots \parallel_L W_1}_{n \text{ times}}.$$

Now, $\bar{M}$ still has $2^n + 1$ states, but *it can* be minimised up to bisimulation due to the symmetry among the worker threads, which are now copies of the same process. In this case, it is sufficient to just *count* how many worker threads are performing w_1. More precisely, let us consider the process $\bar{W}_0$ defined as

$$\bar{W}_0 \triangleq fork.\bar{W}_1, \qquad \bar{W}_i \triangleq w_1.\bar{W}_{i+1}, \quad \text{for } 1 \leq i \leq n, \qquad \bar{W}_{n+1} \triangleq join.\bar{W}_0.$$

Then it holds that $\underbrace{W_1 \parallel_L \cdots \parallel_L W_1}_{n \text{ times}} \sim \bar{W}_0$, from which, by Proposition 1, it follows that $M \sim_{\mathfrak{F}} F \parallel_L \bar{W}_0$. Hence, given a model with $2^n + 1$ states, we were able to construct a dimmed bisimilar one with only $n + 2$ states.

n_1	n_2	m_1	m_2	k	$\lvert M\rvert$	$\lvert\bar{M}\rvert$	$\lvert M\rvert/\lvert\bar{M}\rvert$
1	1	1	1	1	48	18	2.67
2	2	2	2	1	243	50	5.06
1	1	1	1	10	1056	99	10.67
2	2	2	2	10	5346	275	19.44
1	1	1	1	100	82416	909	90.67
2	2	2	2	100	417213	2525	165.23
3	3	2	2	100	741744	3535	209.83

Table 1. State space sizes for M and $\bar{M}$ in Example 2, denoted by $|M|$ and $|\bar{M}|$, respectively.

In the above example, nontrivial parts of the action set involve actions which are never synchronised. That is, dimmed bisimulation can be inferred *compositionally*, starting from the simplest concurrent processes, because they are composed together over synchronisation action sets that satisfy singleton coherence. By contrast, the following examples can still be related to more compact dimmed bisimilar processes; however, dimmed bisimilarity *cannot* be inferred compositionally because the synchronisation sets are not singleton coherent. Thus a relation must be directly given for the whole composite process under consideration. In Section 4 we will, however, show how compositional reasoning on the same examples can be recovered at the cost of establishing only dimmed simulation.

Our second case study is a model of a producer/consumer system where the interaction is mediated by a buffer of finite capacity (see also [2]).

Example 2 (Multi-class Producer/Consumer). For simplicity, let us consider two classes of producers and two classes of users which share the same buffer. Let m_1 (resp., m_2) be the number of producers of the first (resp., second) class; let n_1 and n_2 be the number of consumers. Finally, let k be the buffer capacity.

A class-i producer, for $i = 1, 2$, is modelled by $P_i \triangleq prod_i.put_i.P_i$; here $prod_i$ models an independent action that describes the production of an item, whereas put_i is a synchronisation action to be performed with the buffer: It can be executed only when there is at least one place available in the buffer. In a similar fashion, the model of a class-i consumer is given by $C_i \triangleq get_i.cons_i.C_i$. In this case, get_i is a synchronisation action with the buffer, whereas the (independent) consumption of an item is modelled by action $cons_i$. Finally, a place in the buffer is modelled by the component $B \triangleq put_1.get_1.B + put_2.get_2.B$. Overall, our model of interest, denoted by M, is given by

$$M := \big(P_1[m_1] \parallel_\emptyset P_2[m_2]\big) \parallel_{L_1} B[k] \parallel_{L_2} \big(C_1[n_1] \parallel_\emptyset C_2[n_2]\big)$$

where $L_1 = \{put_1, put_2\}$, $L_2 = \{get_1, get_2\}$, and $S[l]$ abbreviates $(\underbrace{S \parallel_\emptyset \cdots \parallel_\emptyset S}_{l \text{ times}})$.

A reasonable abstraction is not to insist on keeping the two classes of producers (resp. consumers) distinct.

To this end, let us consider $\mathfrak{F} = \{L_1, L_2, \{prod_1, prod_2\}, \{cons_1, cons_2\}\}$. With this partition, it is possible to show that (cf. Appendix C)

$$M \sim_{\mathfrak{F}} \bar{M}, \qquad \text{with} \quad \bar{M} := P_1[m_1 + m_2] \parallel_{L_1} B_1[k] \parallel_{L_2} C_1[n_1 + n_2].$$

That is, producers and consumers of the second class are dimmed bisimilar to those of the first class; furthermore, a buffer place with two distinct actions is dimmed bisimilar to a buffer with actions of a single representative type. As with Example 1, the state-space reduction achieved can be significant. Indeed, Table 1 compares the state space sizes of M and $\bar{M}$ (after minimisation of both processes up to bisimulation) for different values of n_1, n_2, m_1, m_2, and k.

Our next example is m-out-of-n communication, a typical pattern of interaction that occurs frequently in distributed systems. For instance, it is adopted by certain security protocols, whereby a client performs connections to a subset of the available authentication servers in order to improve resilience to attacks by decentralising the information (e.g., [15]); in quorum consensus protocols, to guarantee availability in distributed databases (e.g., [21]); and in peer-to-peer systems, where among all nodes that are potentially able to serve a request, a peer connects to a subset of them in order to avoid data fragmentation [5].

Example 3 (m-out-of-n Communication). Let us consider an illustrative case of 2-out-of-3 communication, where exactly two processes can make progress in a computation [23]. Each process evolves through two local states, P_i and Q_i, $1 \leq i \leq 3$, where P_i encodes the 2-out-of-3 communication and Q_i performs some local computation whenever P_i turns out to be one of the two processes that is allowed to make progress:

$$\begin{aligned}
P_1 &\triangleq a_{12}.Q_1 + a_{13}.Q_1 + a_{23}.P_1 & \qquad Q_1 &\triangleq b_1.Q_1 + a_{23}.Q_1 \\
P_2 &\triangleq a_{12}.Q_2 + a_{13}.P_2 + a_{23}.Q_2 & \qquad Q_2 &\triangleq b_2.Q_2 + a_{13}.Q_2 \\
P_3 &\triangleq a_{12}.P_3 + a_{13}.Q_3 + a_{23}.Q_3 & \qquad Q_3 &\triangleq b_3.Q_3 + a_{12}.Q_3 \\
M &:= P_1 \parallel_L P_2 \parallel_L P_3 & \qquad L &= \{a_{12}, a_{13}, a_{23}\}.
\end{aligned}$$

The encoding is such that each process P_i is able to react to all synchronisation actions a_{12}, a_{13}, a_{23}; however, P_i does make progress and becomes Q_i only when the synchronisation action a_{kl} is such that $i = k$ or $i = l$. Similarly, Q_i is such that it does not prevent the synchronisation action that does not involve i to happen. For instance, process $Q_1 \parallel_L P_2 \parallel_L P_3$, which is reachable from M, affords an a_{23}-transition to $Q_1 \parallel_L Q_2 \parallel_L Q_3$. In general, constructing processes in this way yields a 2-out-of-n model which has 2^n states.

In this form, M cannot be minimised any further up to bisimulation. However, as with the previous examples, dimmed bisimulation offers a means to exploiting symmetries after an appropriate perturbation of the action names. Specifically, let us consider

$$\mathfrak{F} = \{\{a_{12}, a_{13}, a_{23}\}, \{b_1, b_2, b_3\}\}, \quad \bar{P} \triangleq a_{12}.\bar{P} + a_{12}.\bar{Q}, \quad \text{and } \bar{Q} \triangleq b_1.\bar{P} + a_{12}.\bar{Q}.$$

It is possible to show that (cf. Appendix C)

$$P_1 \parallel_L P_2 \parallel_L P_3 \sim_{\mathfrak{F}} \bar{P} \parallel_L \bar{P} \parallel_L \bar{P} := \bar{M}.$$

Now, $\bar{M}$ is bisimilar to a process, $\bar{C}_3$, counting the number of $\bar{P}$-processes:

$$\begin{aligned}
\bar{C}_3 &\triangleq a_{12}.\bar{C}_3 + a_{12}.\bar{C}_2 + a_{12}.\bar{C}_1 + a_{12}.\bar{C}_0 \\
\bar{C}_2 &\triangleq a_{12}.\bar{C}_2 + a_{12}.\bar{C}_1 + a_{12}.\bar{C}_0 + b_1.\bar{C}_3 \\
\bar{C}_1 &\triangleq a_{12}.\bar{C}_1 + a_{12}.\bar{C}_0 + b_1.\bar{C}_2 \\
\bar{C}_0 &\triangleq a_{12}.\bar{C}_0 + b_1.\bar{C}_1.
\end{aligned}$$

In general, for a 2-out-of-n model it is possible to define a process $\bar{C}_n$ in a similar fashion, which reduces the complexity of the state space from 2^n to $n+1$.

We end this section by discussing a classical example in the process algebra literature.

Example 4 (Milner's Cyclers [17]). Milner's cyclers is a model of a scheduler that allows a set of processes $P_1, P_2, \ldots, P_n$ to cyclically perform local computations in succession; that is, process P_i cannot start its computation until P_{i-1} has instructed it to do so. For simplicity, here we study the case for $n = 3$. The model of the three processes is as follows:

$$\begin{array}{lll}
P_1 \triangleq \gamma_1.Q_1 + \gamma_2.P_1 + \gamma_3.P_1 & Q_1 \triangleq \alpha_1.R_1 & R_1 \triangleq \gamma_2.S_1 + \beta_1.T_1 \\
S_1 \triangleq \beta_1.P_1 + \gamma_3.S_1 & T_1 \triangleq \gamma_2.P_1 & \\
P_2 \triangleq \gamma_1.P_2 + \gamma_2.Q_2 + \gamma_3.P_2 & Q_2 \triangleq \alpha_2.R_2 & R_2 \triangleq \gamma_3.S_2 + \beta_2.T_2 \\
S_2 \triangleq \beta_2.P_2 + \gamma_1.S_2 & T_2 \triangleq \gamma_3.P_2 & \\
P_3 \triangleq \gamma_1.P_3 + \gamma_2.P_3 + \gamma_3.Q_3 & Q_3 \triangleq \alpha_3.R_3 & R_3 \triangleq \gamma_1.S_3 + \beta_3.T_3 \\
S_3 \triangleq \beta_3.P_3 + \gamma_2.S_3 & T_3 \triangleq \gamma_1.P_3 &
\end{array}$$

Here, γ_i represents the signal that process i is able to start the computation; its performance is modelled by action α_i; upon completion, the process may signal the start to its successor (hence, to achieve cyclic behaviour R_3 will perform action γ_1), or notify the end of the local computation via β_i, in either order. Process S_i describes the state of a cycler which has already started the computation and let the next cycler go. In that state, it may witness some γ_j-action performed by other cyclers; in this case, the cycler ignores this signal and behaves as S_i again. For instance, S_1 may witness a γ_3-action because the second cycler may let the third one go before the first cycler has finished.

The model is completed by a scheduler, Sc, that will enforce the start of P_1:

$$Sc \triangleq \gamma_1.Sc' \qquad\qquad Sc' \triangleq \gamma_1.Sc' + \gamma_2.Sc' + \gamma_3.Sc'$$

Thus, the overall system is described by

$$M := Sc \parallel_L P_1 \parallel_L P_2 \parallel_L P_3, \quad \text{with } L = \{\gamma_1, \gamma_2, \gamma_3\}.$$

Let us now consider $\mathfrak{F} = \big\{\{\alpha_1, \alpha_2, \alpha_3\}, \{\beta_1, \beta_2, \beta_3\}, \{\gamma_1, \gamma_2, \gamma_3\}\big\}$ and

$$\bar{P} \triangleq \gamma_1.\bar{P} + \gamma_1.\bar{P} \qquad \bar{Q} \triangleq \alpha_1.\bar{R} \qquad \bar{R} \triangleq \gamma_1.\bar{S} + \beta_1.\bar{T}$$
$$\bar{S} \triangleq \beta_1.\bar{P} + \gamma_1.\bar{S} \qquad \bar{T} \triangleq \gamma_1.\bar{P} \qquad \bar{Sc} \triangleq \gamma_1.\bar{Sc}$$

Then, using similar arguments as in Example 3, it holds that $M \sim_{\mathfrak{F}} \bar{Sc} \parallel_L \bar{P} \parallel_L \bar{P} \parallel_L \bar{P}$. In general, Milner's model is of exponential complexity with the number of cyclers (e.g., [13]). Using the same counting-process technique discussed in the previous examples will yield a model of polynomial complexity, by exploiting that $\bar{P}$ is present in multiple identical copies in the dimmed bisimilar process.

4 Dimmed Simulation

As mentioned, single coherence may be an impediment to compositional reasoning for some interesting models of practical relevance. A modelling pattern that *is not* supported is that of *multi-class* systems, where multiple synchronisation actions may be performed by processes that the modeller wishes to keep differentiated. For instance, let $\mathfrak{F} = \big\{\{a_1, a_2\}, \{b_1, b_2\}\big\}$. Even though $a_1.\mathbf{0} + a_2.\mathbf{0} \sim_{\mathfrak{F}} a_1.\mathbf{0}$ and $a_1.b_1.\mathbf{0} + a_2.b_2.\mathbf{0} \sim_{\mathfrak{F}} a_1.b_1.\mathbf{0}$, we *cannot* infer by Theorem 3 that

$$(a_1.\mathbf{0} + a_2.\mathbf{0}) \parallel_L (a_1.b_1.\mathbf{0} + a_1.b_2.\mathbf{0}) \sim_{\mathfrak{F}} a_1.\mathbf{0} \parallel_L a_1.b_1.\mathbf{0}, \qquad \text{with } L = \{a_1, a_2\},$$

because L is not singleton coherent with $\mathfrak{F}$. Similarly to Examples 2, 3, and 4, dimmed bisimilarity does hold. However, since it cannot be proven compositionally, a relation containing the whole processes must be provided.[3]

Fortunately, compositional reasoning can still be applied if one accepts to use dimmed *simulation* instead of dimmed bisimulation.

It has however to be said that simulation, that was introduced by Milner much before bisimulation [16], has been used for establishing interesting properties of systems. For example, in [14] it is argued that "in many cases, neither trace equivalence nor bisimilarity, but similarity is the appropriate abstraction for computer-aided verification".

Definition 6 (Dimmed Simulation). *Given a partition $\mathfrak{F}$, a binary relation $\mathcal{R}$ over $\mathcal{P}$ is an* $\mathfrak{F}$-dimmed simulation *iff whenever $(P, Q) \in \mathcal{R}$ and $a \in \mathcal{L}$:*

- *if $P \xrightarrow{[a]} P'$ then $Q \xrightarrow{[a]} Q'$ and $(P', Q') \in \mathcal{R}$.*

For two processes P and Q, we say that Q $\mathfrak{F}$-dimmed ly simulates *P, written $P \preceq_{\mathfrak{F}} Q$, if there is an* $\mathfrak{F}$-dimmed simulation *which relates them.*

[3] Specifically, consider the relation

$$\mathcal{R} = \big\{\big((a_1.\mathbf{0} + a_2.\mathbf{0}) \parallel_L (a_1.b_1.\mathbf{0} + a_1.b_2.\mathbf{0}), a_1.\mathbf{0} \parallel_L a_1.b_1.\mathbf{0}\big),$$
$$\big(\mathbf{0} \parallel_L b_1.\mathbf{0}, \mathbf{0} \parallel_L b_1.\mathbf{0}\big), \big(\mathbf{0} \parallel_L b_2.\mathbf{0}, \mathbf{0} \parallel_L b_1.\mathbf{0}\big), \big(\mathbf{0} \parallel_L \mathbf{0}, \mathbf{0} \parallel_L \mathbf{0}\big)\big\}.$$

Then, $\mathcal{R}$ is a dimmed bisimulation.

Similarly to dimmed bisimulation, using the trivial singleton partition we recover the standard notion of simulation between processes, hereafter denoted by $\preceq$. We state the next proposition without proof.

Proposition 4. *The following hold:*

i) $\preceq_{\mathfrak{F}}$ *is a preorder;*
ii) $P \sim_{\mathfrak{F}} Q \implies P \preceq_{\mathfrak{F}} Q \ \wedge \ Q \preceq_{\mathfrak{F}} P$.

Mutatis mutandis, the relationship between dimmed simulation and simulation is the same as the relationship between dimmed bisimulation and bisimulation as discussed in Section 2. That is, Theorem 2 carries over. Further, dimmed similarity is preserved by partition refinement (cf. Proposition 3), thus similarity implies dimmed similarity (cf. Corollary 1). Propositions 1 and 2 hold also for dimmed simulation by point ii) of Proposition 4.

The results of Theorem 3 would carry over as well. However, in the case of dimmed simulation it is possible to relax the assumption on singleton coherency, which is needed for the preservation of dimmed bisimulation by parallel composition. Here, instead, we will just require block coherency, as well as a form of *homogeneity* of the two operands of the parallel composition with respect to the synchronisation set. To formally define this notion, we denote by $Act(P)$ the set of all actions that can be performed by process P; for all $a \in \mathcal{L}$,

$$a \in Act(P) \text{ iff } \exists n \geq 1 : P \xrightarrow{a_1} P_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} P_n, a_n = a \wedge a_i \neq a, 1 \leq i \leq n-1.$$

Definition 7 (Homogeneous Processes). *Let P and Q be two processes, $L \subseteq \mathcal{A}$, and $\mathfrak{F}$ a partition. P and Q are said to be* homogeneous for L with $\mathfrak{F}$ *iff*

$$\big|\big(Act(P) \cup Act(Q)\big) \cap [a]\big| \leq 1, \qquad \textit{for all } a \in L.$$

Essentially, we require that both P and Q be able to perform at most one of the synchronisation actions belonging to the same element of $\mathfrak{F}$. For instance, let $P \triangleq a_1.\mathbf{0} + a_2.\mathbf{0}$, $Q \triangleq a_1.Q + a_2.Q$, and $\{a_1, a_2\} \in \mathfrak{F}$. Then P and Q are not homogeneous for $L = \{a_1, a_2\}$ in $\mathfrak{F}$. Let us now consider their respectively dimmed similar processes, $\bar{P} \triangleq a_1.\mathbf{0}$ and $\bar{Q} \triangleq a_1.\bar{Q}$. In this case, instead, it holds that $\bar{P}$ and $\bar{Q}$ are homogeneous for L in $\mathfrak{F}$.

Theorem 4 (Compositionality for Dimmed Simulation). *Let P, Q be two processes such that $P \preceq_{\mathfrak{F}} Q$. Then it holds that:*

i) $a.P \preceq_{\mathfrak{F}} b.Q$ *for all a, b such that $[a] = [b]$;*
ii) $P + R \preceq_{\mathfrak{F}} Q + S$*, for any R, S such that $R \preceq_{\mathfrak{F}} S$;*
iii) $P[g] \preceq_{\mathfrak{F}} Q[g]$ *for any g pp-function for $\mathfrak{F}$;*
iv) $P/L \preceq_{\mathfrak{F}} Q/L$ *if L is block coherent with $\mathfrak{F}$;*
v) $P \parallel_L R \preceq_{\mathfrak{F}} Q \parallel_L S$*, for R, S such that $R \preceq_{\mathfrak{F}} S$, if L is block coherent with $\mathfrak{F}$ and Q and S are homogeneous for L with $\mathfrak{F}$.*

The conditions required for the last point deserve more explanation. In general, dimmed simulation is not preserved if only L is block coherent with $\mathfrak{F}$ but Q and S *are not* homogeneous for L with $\mathfrak{F}$. To show this take, for instance, $P := a_1.\mathbf{0} + a_2.\mathbf{0}$, $R := P$, $Q := a_1.\mathbf{0}$, and $S := a_2.\mathbf{0}$, with $L = \{a_1, a_2\}$ and $L \in \mathfrak{F}$. Similarly, if the condition of homogeneity is satisfied but L is not block coherent with $\mathfrak{F}$, dimmed simulation may not be preserved either. For instance, take $P := a_1.\mathbf{0}$, $R := P$, $Q := a_2.\mathbf{0}$, $S := Q$, with $L = \{a_1\}$ and $\{a_1, a_2\} \in \mathfrak{F}$. Then, it holds that $P \parallel_L R \xrightarrow{a_1} \mathbf{0} \parallel_L \mathbf{0}$ but $Q \parallel_L S \not\rightarrow$. Finally, we wish to point out the fact that homogeneity is to be satisfied only by the simulating process $Q \parallel_L S$. This is why we have preferred a statement in the form of item *v)* instead of the weaker "$P \preceq_{\mathfrak{F}} Q \implies P \parallel_L R \preceq_{\mathfrak{F}} Q \parallel_L R$ for Q, R homogeneous for L in $\mathfrak{F}$ and L block coherent with $\mathfrak{F}$." Stated in this way, homogeneity for Q and R would imply some form of homogeneity also in the simulated process $P \parallel_L R$. Indeed, R cannot enable two or more alternative synchronisation actions within the same part, thus significantly reducing the class of composite processes $P \parallel_L R$ that can be simulated. This is because, since L must be block coherent with $\mathfrak{F}$, it could in principle contain two actions a_1, a_2 belonging to the same part. But R can enable only one of them, say a_1. Therefore there would be $a_2 \in L$ which is never performed by one of the two operands. This, in turn, would cause a_2 never to be seen at all by $P \parallel_L R$.

In order to show practical usefulness of dimmed simulation, let us now revisit the previous examples and show that they can be simplified *compositionally* via $\preceq_{\mathfrak{F}}$, but not with $\sim_{\mathfrak{F}}$.

Example 2 (continued). It possible to show that $P_2 \sim_{\mathfrak{F}} P_1$. Theorem 3 could be used to show that $P_1[m_1] \parallel_\emptyset P_2[m_2] \sim_{\mathfrak{F}} P_1[m_1 + m_2]$. Similarly, it holds that $B[k] \sim_{\mathfrak{F}} B_1[k]$, where $B_1 \triangleq put_1.get_1.B_1$. However, although $(P_1[m_1] \parallel_\emptyset P_2[m_2]) \parallel_{L_1} B[k] \sim_{\mathfrak{F}} P_1[m_1 + m_2] \parallel_{L_1} B_1[k]$ does hold, this fact *cannot* be inferred compositionally from Theorem 3 because L_1 is not singleton coherent with $\mathfrak{F}$. Thus, a simpler dimmed bisimilar process to M cannot be obtained by congruence.

Instead, a dimmed similar process can indeed be constructed compositionally. Since $\sim_{\mathfrak{F}}$ implies $\preceq_{\mathfrak{F}}$, we have that $P_1[m_1] \parallel_\emptyset P_2[m_2] \preceq_{\mathfrak{F}} P_1[m_1 + m_2]$ and $B[k] \preceq_{\mathfrak{F}} B_1[k]$. Now, $P_1[m_1 + m_2]$ and $B_1[k]$ are homogenous for L_1 and L_1 is block coherent with the chosen $\mathfrak{F}$. Hence, item *v)* of Theorem 4 yields

$$\big(P_1[m_1] \parallel_\emptyset P_2[m_2]\big) \parallel_{L_1} B[k] \preceq_{\mathfrak{F}} P_1[m_1 + m_2] \parallel_{L_1} B_1[k].$$

Analogously, it holds that $C_1[n_1] \parallel_\emptyset C_2[n_2] \preceq_{\mathfrak{F}} C_1[n_1 + n_2]$. Again, we have that $P_1[m_1 + m_2] \parallel_{L_1} B_1[k]$ and $C_1[n_1 + n_2]$ are homogeneous for L_2 and L_2 is block coherent with $\mathfrak{F}$. Therefore we are able to conclude (compositionally) that

$$M \preceq_{\mathfrak{F}} \bar{M}, \qquad \text{with} \quad \bar{M} := P_1[m_1 + m_2] \parallel_{L_1} B_1[k] \parallel_{L_2} C_1[n_1 + n_2].$$

Example 3 (continued). In Section 3 we were able to show that

$$P_1 \parallel_L P_2 \parallel_L P_3 \sim_{\mathfrak{F}} \bar{P} \parallel_L \bar{P} \parallel_L \bar{P} := \bar{M}.$$

However, this cannot be established compositionally because L *is not* singleton coherent with $\mathfrak{F}$, hence Theorem 3 cannot be applied. Instead we show that

$$P_1 \parallel_L P_2 \parallel_L P_3 \preceq_{\mathfrak{F}} \bar{M}$$

since $P_i \preceq_{\mathfrak{F}} \bar{P}$ (in fact, $P_i \sim_{\mathfrak{F}} \bar{P}$) for $i = 1, 2, 3$, and L is block coherent with $\mathfrak{F}$.

Example 4 (continued). Similarly to the previous example, in Section 3 we discussed that

$$Sc \parallel_L P_1 \parallel_L P_2 \parallel_L P_3 \sim_{\mathfrak{F}} \bar{Sc} \parallel_L \bar{P} \parallel_L \bar{P} \parallel_L \bar{P}.$$

However, compositional reasoning was not possible because L is not singleton coherent. Once again, L is instead block coherent for the action partition $\mathfrak{F}$ chosen in this case. Now, it holds that $Sc \preceq_{\mathfrak{F}} \bar{Sc}$ and $P_i \preceq_{\mathfrak{F}} \bar{P}$. This implies that the relation $Sc \parallel_L P_1 \parallel_L P_2 \parallel_L P_3 \preceq_{\mathfrak{F}} \bar{Sc} \parallel_L \bar{P} \parallel_L \bar{P} \parallel_L \bar{P}$ can indeed be proven by repeatedly using point *v)* of Theorem 4.

5 Weak Extensions

So far, we have been able to successfully exploit our dimmed relations in cases where the related processes are, in some informal sense, structurally resembling each other. For instance, in Example 4 of Milner's cyclers, each process P_i is syntactically equal to P_j after a suitable relabelling of the actions. Further state-space compressions may be possible when abstracting away from unnecessary detail in the model. This has been at the basis of any weak notion of behavioural relation considered in the past. Weak extensions of our dimmed relations are similar in spirit.

Given P and P' two processes in $\mathcal{P}$, we shall write, for any action $a \in \mathcal{L}$, $P \stackrel{a}{\Rightarrow} P'$ iff either of the following holds:

- $a \neq \tau$ and there are processes P_1, P_2 such that $P(\stackrel{\tau}{\rightarrow})^* P_1 \stackrel{a}{\rightarrow} P_2(\stackrel{\tau}{\rightarrow})^* P'$;
- $a = \tau$ and $P(\stackrel{\tau}{\rightarrow})^* P'$.

The notation $(\stackrel{\tau}{\rightarrow})^*$ refers to the reflexive and transitive closure of the relation $\stackrel{\tau}{\rightarrow}$. Similarly to Section 2, we define $P \stackrel{[a]}{\Longrightarrow} P'$ iff there exists $b \in [a]$ such $P \stackrel{b}{\Rightarrow} P'$. Based on this, the following definitions of dimmed weak bisimulation and dimmed weak simulation are then natural.

Definition 8 (Dimmed Weak Bisimulation). *Given a partition $\mathfrak{F}$, a binary relation $\mathcal{R}$ over $\mathcal{P}$ is an* $\mathfrak{F}$-dimmed weak bisimulation *iff whenever $(P, Q) \in \mathcal{R}$ and $a \in \mathcal{L}$*

- *if $P \stackrel{[a]}{\longrightarrow} P'$ then $Q \stackrel{[a]}{\Longrightarrow} Q'$ and $(P', Q') \in \mathcal{R}$;*
- *if $Q \stackrel{[a]}{\longrightarrow} Q'$ then $P \stackrel{[a]}{\Longrightarrow} P'$ and $(P', Q') \in \mathcal{R}$.*

Two processes P, Q are $\mathfrak{F}$-dimmed weak bisimilar*, written $P \approx_{\mathfrak{F}} Q$, iff there exists an $\mathfrak{F}$-dimmed weak bisimulation that relates them.*

Definition 9 (Dimmed Weak Simulation). *Given a partition $\mathfrak{F}$, a binary relation $\mathcal{R}$ over $\mathcal{P}$ is an* $\mathfrak{F}$-dimmed weak simulation *iff whenever $(P, Q) \in \mathcal{R}$ and $a \in \mathcal{L}$*

– *if $P \xrightarrow{[a]} P'$ then $Q \stackrel{[a]}{\Longrightarrow} Q'$ and $(P', Q') \in \mathcal{R}$.*

Two processes P, Q are $\mathfrak{F}$-dimmed weak similar*, written $P \precsim_{\mathfrak{F}} Q$, iff there exists an $\mathfrak{F}$-dimmed weak simulation that relates them.*

All the results presented in Section 2 and 4 carry over straightforwardly to these weak variants. (For completeness, they are reported in Appendix A.)

For instance, it holds that $a_1.\tau.\mathbf{0} + a_2.\tau.\tau.\mathbf{0} \approx_{\mathfrak{F}} a_1.\mathbf{0}$, while, clearly, $a_1.\tau.\mathbf{0} + a_2.\tau.\tau.\mathbf{0} \not\approx_{\mathfrak{F}} a_1.\mathbf{0}$. In a variant of Example 1, if the worker threads were additionally given τ-computations of different length, e.g.,

$$\hat{W}_i \triangleq \mathit{fork}.\underbrace{\tau.\cdots.\tau}_{i \text{ times}}.\mathit{join}.\hat{W}_i$$

then it would still hold that $\hat{W}_i \approx_{\mathfrak{F}} W_1$. In a variant of Example 2, a model $\hat{B}$ of a buffer that threats the two kinds of item differently by means of τ-actions, e.g., $\hat{B} \triangleq \mathit{put}_1.\mathit{get}_1.\hat{B} + \mathit{put}_2.\tau.\mathit{get}_2.\hat{B}$ would be such that $\hat{B} \approx_{\mathfrak{F}} B_1$. Our strong notions of dimmed relations almost seem to require a strict structural/syntactic resemblance up to action relabelling for effective applicability. In contrast, their weak extensions may be conveniently applied to practical examples even for less "symmetric" processes, so long as their differences are due to behaviour that may be considered irrelevant at the desired level of abstraction.

6 Related Work

At the very core of our dimmed relations is an abstraction operated at the level of the transition system laid down by the process algebra, which lifts concrete actions a, b, ..., to elements $[a]$, $[b]$, ... of a given partition of the action set. As such, this work is in the general context of abstract interpretation [6], where a concrete model is approximated with a hopefully simpler one that preserves some properties of interest, for example expressed as logical formulae (e.g., [4, 7]). This framework has also been considered in process algebra as early as in [22], where an approximation based on a preorder on the action set of CCS is studied; an analogous preorder has been used for model checking abstractions for μ-calculus [10]. While a preorder can be motivated by situations where certain actions may be considered as carrying more information than others, our approach is fundamentally different because all actions belonging to the same partition block are equal in power.

The present paper is also somewhat related to [9], where the authors consider abstract interpretation to reduce infinite branching to finite branching of value-passing LOTOS based on trace semantics to enable model checking of linear temporal logic. Instead, more work than on action abstraction seems to have

been directed to the dual notion of action *refinement*, where the main idea is that an atomic action is expanded into a process, e.g., a more detailed sequence of actions [11, 1, 12].

7 Conclusion

Dimmed behavioural relations permit trading-off between a detailed knowledge of the action types exhibited by a concrete model under study and a potentially more compact description arising from collapsing several actions together.

From a theoretical standpoint, the characterisation in terms of actions relabelling seems to make justice to this classic process algebra operator, which has been oftentimes neglected in recent developments of this field (see [20] for a discussion). The property of partition-preservation for a relabelling function presented in this paper is less restrictive than injectivity, as required for standard bisimulation results; yet it permits compositional reasoning for our dimmed relations.

From a more pragmatic standpoint, on a number of modelling patterns of practical interest, we showed that our dimmed relations can be effectively employed for a significantly more efficient (as well as more abstract) analysis of heterogeneous systems, when heterogeneity is captured by the presence of analogous but formally distinct behaviours which are told apart by the use of distinct actions.

Future work will be concerned with a thorough investigation of a logical characterisation of dimmed bisimulation and simulation. We expect, however, that a straightforward adaptation of Hennessy-Milner logic should characterise the former; instead, an extension of simulation to *ready simulation* should be characterised by a suitably revised class of *denial formulas*, along the lines of [3].

Acknowledgement

Most of this work was done while the first author was visiting LMU in Munich; he would like to thank Martin Wirsing and his group for the excellent scientific and social atmosphere.

References

1. L. Aceto and M. Hennessy. Towards action-refinement in process algebras. *Information and Computation*, 103(2):204–269, 1993.
2. Alessandro Aldini, Marco Bernardo, and Flavio Corradini. *A Process Algebraic Approach to Software Architecture Design*. Springer Publishing Company, 2009.
3. Bard Bloom, Sorin Istrail, and Albert R. Meyer. Bisimulation can't be traced. *J. ACM*, 42(1):232–268, January 1995.
4. Edmund M. Clarke, Orna Grumberg, and David E. Long. Model checking and abstraction. *ACM Trans. Program. Lang. Syst.*, 16(5):1512–1542, September 1994.

5. Bram Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA, June 2003.
6. Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252, New York, NY, USA, 1977. ACM.
7. Dennis Dams, Rob Gerth, and Orna Grumberg. Abstract interpretation of reactive systems. *ACM Trans. Program. Lang. Syst.*, 19(2):253–291, March 1997.
8. Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.
9. Alessandro Fantechi, Stefania Gnesi, and Diego Latella. Towards automatic temporal logic verification of value passing process algebra using abstract interpretation. In *CONCUR '96*, volume 1119 of *LNCS*, pages 563–578. Springer, 1996.
10. Harald Fecher and Michael Huth. Model checking for action abstraction. In *Verification, Model Checking, and Abstract Interpretation*, volume 4905 of *LNCS*, pages 112–126. Springer, 2008.
11. Rob Glabbeek and Ursula Goltz. Equivalence notions for concurrent systems and refinement of actions. In *MFCS*, volume 379 of *LNCS*, pages 237–248. Springer, 1989.
12. Roberto Gorrieri, Arend Rensink, and Mura Anteo Zamboni. Action refinement. In *Handbook of Process Algebra*, pages 1047–1147. Elsevier, 2000.
13. JanFriso Groote and Faron Moller. Verification of parallel systems via decomposition. In *CONCUR*, volume 630 of *LNCS*, pages 62–76. Springer, 1992.
14. Monika Rauch Henzinger, Thomas A. Henzinger, and Peter W. Kopke. Computing simulations on finite and infinite graphs. In *FOCS*, pages 453–462. IEEE Computer Society, 1995.
15. Philip D. MacKenzie, Thomas Shrimpton, and Markus Jakobsson. Threshold password-authenticated key exchange. *J. Cryptology*, 19(1):27–66, 2006.
16. Robin Milner. An algebraic definition of simulation between programs. In D. C. Cooper, editor, *IJCAI*, pages 481–489. William Kaufmann, 1971.
17. Robin Milner. *A Calculus of Communicating Systems.* Springer-Verlag, 1980.
18. R. Paige and R. Tarjan. Three partition refinement algorithms. *SIAM Journal on Computing*, 16(6):973–989, 1987.
19. A.W. Roscoe. *The Theory and Practice of Concurrency.* Prentice Hall, 1997.
20. Davide Sangiorgi. *Introduction to Bisimulation and Coinduction.* Cambridge University Press, 2011.
21. Robert H. Thomas. A majority consensus approach to concurrency control for multiple copy databases. *ACM Trans. Database Syst.*, 4:180–209, June 1979.
22. Bent Thomsen. An extended bisimulation induced by a preorder on actions. M.Sc. thesis, Aalborg University, 1987.
23. Max Tschaikowski and Mirco Tribastone. Generalised Communication for Interacting Agents. In *QEST*, pages 178–188. IEEE Computer Society, September 2012.

A Dimmed Weak Relations: Results

The results of this section are given without proof because they are a standard extension from dimmed strong bisimulation and simulation, which are proven in Appendix B.

Theorem 5. *For any partition* $\mathfrak{F}$*, the relation* $\approx_{\mathfrak{F}}$

a) is an equivalence relation;
b) is the largest $\mathfrak{F}$-dimmed weak bisimulation;
c) satisfies the following property: $P \approx_{\mathfrak{F}} Q$ iff, for any action a,
- *if $P \xrightarrow{[a]} P'$ then $Q \stackrel{[a]}{\Longrightarrow} Q'$ and $P' \approx_{\mathfrak{F}} Q'$;*
- *if $Q \xrightarrow{[a]} Q'$ then $P \stackrel{[a]}{\Longrightarrow} P'$ and $P' \approx_{\mathfrak{F}} Q'$.*

Theorem 6 (Characterisation of Dimmed Weak Bisimulation). *Given a partition $\mathfrak{F}$, $P \approx_{\mathfrak{F}} Q$ iff there exists a pp-function f for $\mathfrak{F}$ such that $P[f] \approx Q[f]$.*

Similarly to weak bisimilarity, dimmed weak bisimilarity turns out not to be a congruence with respect to the choice operator. Indeed, although $\tau.a_1.\mathbf{0} \approx_{\mathfrak{F}} a_2.\mathbf{0}$ if $\{a_1, a_2\} \in \mathfrak{F}$, we have that $\tau.a_1.\mathbf{0} + b.\mathbf{0} \not\approx_{\mathfrak{F}} a_2.\mathbf{0} + b.\mathbf{0}$. However, dimmed weak bisimilarity is preserved by the others operators, and this is addressed in the next theorem.

Theorem 7 (Compositionality for Dimmed Weak Bisimulation). *Let P and Q be two processes such that $P \approx_{\mathfrak{F}} Q$. Then it holds that:*

i) $a.P \approx_{\mathfrak{F}} b.Q$ for all a, b such that $[a] = [b]$;
ii) $P[g] \approx_{\mathfrak{F}} Q[g]$ for any g partition-preserving for $\mathfrak{F}$;
iii) $P \parallel_L R \approx_{\mathfrak{F}} Q \parallel_L S$ if $R \approx_{\mathfrak{F}} S$ and L is singleton coherent with $\mathfrak{F}$.

Proposition 5 (Characterisation of Dimmed Weak Simulation). *Given a partition $\mathfrak{F}$, $P \precsim_{\mathfrak{F}} Q$ iff there exists a pp-function f for $\mathfrak{F}$ such that $P[f] \precsim Q[f]$.*

Theorem 8 (Compositionality for Dimmed Weak Simulation). *Let P, Q be two processes such that $P \precsim_{\mathfrak{F}} Q$. Then it holds that:*

i) $a.P \precsim_{\mathfrak{F}} b.Q$ for all a, b such that $[a] = [b]$;
ii) $P[g] \precsim_{\mathfrak{F}} Q[g]$ for any g pp-function for $\mathfrak{F}$;
iii) $P \parallel_L R \precsim_{\mathfrak{F}} Q \parallel_L S$ for R, S such that $R \precsim_{\mathfrak{F}} S$, if L is block coherent with $\mathfrak{F}$ and Q and S are homogeneous for L with $\mathfrak{F}$.

B Proofs

Proof (Theorem 2). [If $P \sim_{\mathfrak{F}} Q \Rightarrow \exists f$ pp-function for $\mathfrak{F}$ such that $P[f] \sim Q[f]$]: In order to prove this implication, we need to show that there exists a partition-preserving function f and a strong bisimulation relating $P[f]$ and $Q[f]$. Let us define h to be the pp-function for $\mathfrak{F}$ such that for every $\mathcal{F} \in \mathfrak{F}$ and for every $a, b \in \mathcal{F}$, $h(a) = h(b)$. Let us now consider the following relation

$$\mathcal{R} \triangleq \{\big(P'[h], Q'[h]\big) \mid P' \sim_{\mathfrak{F}} Q'\} .$$

Readily, the pair $\big(P[h], Q[h]\big) \in \mathcal{R}$. We shall now prove that $\mathcal{R}$ is a strong bisimulation. For this purpose suffices to show that when $\big(P'[h], Q'[h]\big) \in \mathcal{R}$ and $P'[h] \xrightarrow{a} P''[h]$ there exists a transition $Q'[h] \xrightarrow{a} Q''[h]$ such that $\big(P''[h], Q''[h]\big) \in \mathcal{R}$. If $P'[h] \xrightarrow{a} P''[h]$ then there exists an action $b \in [a]$, with $h(b) = a$ such that

$P' \xrightarrow{b} P''$. The assumption $P' \sim_{\mathfrak{F}} Q'$ guarantees the existence of an action $d \in [b]$ and a transition $Q' \xrightarrow{d} Q''$ such that $P'' \sim_{\mathfrak{F}} Q''$. Thus, $Q'[h] \xrightarrow{h(d)} Q''[h]$. Being the function h defined in such a way that $h(m) = h(l)$ for all $m, l \in \mathcal{F}$ and $[d] = [b]$ we have $h(d) = h(b) = a$, whence, $Q'[h] \xrightarrow{a} Q''[h]$ and $\big(P''[h], Q''[h]\big) \in \mathcal{R}$.

[If $\exists f$ pp-function for $\mathfrak{F}$ such that $P[f] \sim Q[f] \Rightarrow P \sim_{\mathfrak{F}} Q$]: We prove this claim by providing an $\mathfrak{F}$-dimmed strong bisimulation relating P and Q. In order to construct such a relation, let f denote the pp-function for $\mathfrak{F}$ such that $P[f] \sim Q[f]$ and define

$$\mathcal{R} \triangleq \big\{(P', Q') \mid P'[f] \sim Q'[f]\big\} .$$

We are left with proving that $\mathcal{R}$ is indeed an $\mathfrak{F}$-dimmed strong bisimulation. By assumption we have that $(P, Q) \in \mathcal{R}$. Furthermore, due to the fact that the strong bisimilarity is a symmetric relation, to prove that $\mathcal{R}$ is an $\mathfrak{F}$-dimmed strong bisimulation suffices to show that if $(P', Q') \in \mathcal{R}$ and $P' \xrightarrow{a} P''$ then there exists a $b \in [a]$ and a transition $Q' \xrightarrow{b} Q''$ such that $(P'', Q'') \in \mathcal{R}$. If $P' \xrightarrow{a} P''$ then there exists a c, with $c = f(a)$, such that $P'[f] \xrightarrow{c} P''[f]$. Exploiting the fact that $P'[f] \sim Q'[f]$ we know there exists a transition $Q'[f] \xrightarrow{c} Q''[f]$ with $P''[f] \sim Q''[f]$. Furthermore, from the transition $Q'[f] \xrightarrow{c} Q''[f]$ follows that there exists an action b, with $f(b) = c$, such that $Q' \xrightarrow{b} Q''$. The assumption on the function f being partition-preserving guarantees that $[b] = [c]$ and $[a] = [c]$. Thus, given a transition $P' \xrightarrow{a} P''$ we have shown the existence of an action $b \in [a]$ and a transition $Q' \xrightarrow{b} Q''$ such that $(P'', Q'') \in \mathcal{R}$. □

Proof (Theorem 3). We prove each claim separately by providing for each of them the $\mathfrak{F}$–dimmed strong bisimulation relating the corresponding pair.

i) Let $\mathcal{R}'$ be an $\mathfrak{F}$-dimmed strong bisimulation relating P and Q. Then the relation $\mathcal{R} \triangleq \{(a.P, b.Q)\} \cup \mathcal{R}'$, with $[a] = [b]$, proves the first point.
ii) Let $\mathcal{R}'$ be an $\mathfrak{F}$-dimmed strong bisimulation relating P and Q and $\mathcal{R}''$ be an $\mathfrak{F}$-dimmed strong bisimulation relating R and S. Then the relation $\mathcal{R} \triangleq \{(P + R, Q + S)\} \cup \mathcal{R}' \cup \mathcal{R}''$ proves the second point.
iii) For any g pp-function for $\mathfrak{F}$ the relation $\mathcal{R} \triangleq \{(P'[g], Q'[g]) \mid P' \sim_{\mathfrak{F}} Q'\}$ yields the claim.
iv) To prove this point, we exploit Theorem 2 and show, under the assumptions $P \sim_{\mathfrak{F}} Q$ and L block coherent with $\mathfrak{F}$, the existence of a pp-function f for $\mathfrak{F}$ such that $P/L[f] \sim Q/L[f]$. The assumption $P \sim_{\mathfrak{F}} Q$ assures the existence of a pp-function h for $\mathfrak{F}$ such that $P[h] \sim Q[h]$. We shall now prove, using the assumption that L is block coherent with $\mathfrak{F}$, that the relation

$$\mathcal{R} \triangleq \{(P'/L[h], Q'/L[h]) \mid P'[h] \sim Q'[h]\} ,$$

contains the pair $(P/L[h], Q/L[h])$ and is a strong bisimulation. By assumption, the pair $(P/L[h], Q/L[h]) \in \mathcal{R}$. To show that it is a strong bisimulation it is enough to show that if $(P'/L[h], Q'/L[h]) \in \mathcal{R}$ and $P'/L[h] \xrightarrow{a} E$, then

there exists a transition $Q'/L[h] \xrightarrow{a} T$ such that $(E,T) \in \mathcal{R}$. Let us assume $P'/L[h] \xrightarrow{a} E$; this implies the existence of an action $b \in [a]$, with $h(b) = a$, such that $P'/L \xrightarrow{b} E'$. We distinguish among two cases:

1) $\underline{b \neq \tau}$. By the operational semantics it must hold that $b \notin L$. Let us assume $E' = P''/L$. It must then hold that $P' \xrightarrow{b} P''$; thereby $P'[h] \xrightarrow{h(b)} P''[h]$. By the assumption $P'[h] \sim Q'[h]$, we can infer the existence of a transition $Q'[h] \xrightarrow{h(b)} Q''[h]$ such that $P''[h] \sim Q''[h]$. From $Q'[h] \xrightarrow{h(b)} Q''[h]$ we can conclude there exists a c, with $h(c) = h(b) = a$, such that $Q' \xrightarrow{c} Q''$. Moreover, the partition preserving property of h tells us that $[c] = [b]$ and, in addition, the assumption on L being block coherent with $\mathfrak{F}$ guarantees that $c \notin L$. Therefore, $Q'/L \xrightarrow{c} Q''/L$ from which $Q'/L[h] \xrightarrow{h(c)=a} T$, with $T = Q''/L[h]$, and $(E,T) \in \mathcal{R}$.
2) $\underline{b = \tau}$. In this case we have that $a = \tau$. By the operational semantics either $P' \xrightarrow{\tau} P''$ ($\tau \notin L$) or $P' \xrightarrow{d} P''$, with $d \in L$. The first scenario implies that $P'[h] \xrightarrow{\tau} P''[h]$ which together with the assumption $P'[h] \sim Q'[h]$ assures the existence of a transition $Q'[h] \xrightarrow{\tau} Q''[h]$ such that $P''[h] \sim Q''[h]$. Consequently, $Q' \xrightarrow{\tau} Q''$ wherefrom $Q'/L \xrightarrow{\tau} Q''/L$ and $Q'/L[h] \xrightarrow{\tau} T$, with $T = Q''/L[h]$ and the pair $(E,T) \in \mathcal{R}$. As regards the second case instead, we have $P'[h] \xrightarrow{h(d)} P''[h]$ which, by the assumption $P'[h] \sim Q'[h]$, implies $Q'[h] \xrightarrow{h(d)} Q''[h]$, with $P''[h] \sim Q''[h]$. From $Q'[h] \xrightarrow{h(d)} Q''[h]$ we infer the existence of an action c, with $h(c) = h(d)$, such that $Q' \xrightarrow{c} Q''$. The partition-preserving property of h together with the fact that L is block coherent with $\mathfrak{F}$ yield that $[c] = [d]$ and $c \in L$. Therefore, $Q'/L \xrightarrow{\tau} Q''/L$ from which $Q'/L[h] \xrightarrow{\tau} T$, with $T = Q''/L[h]$ and $(E,T) \in \mathcal{R}$.

v) Albeit the proof of this point, similarly to that of point iv), could be attained by exploiting the result of Theorem 2, we opt for a more direct proof, which only relies on the definition of dimmed strong bisimilarity.
We shall prove, under the assumption that L is singleton coherent with $\mathfrak{F}$, that the relation

$$\mathcal{R} \triangleq \{\big((P' \parallel_L R'), (Q' \parallel_L S')\big) \mid P' \sim_{\mathfrak{F}} Q' \text{ and } R' \sim_{\mathfrak{F}} S'\} ,$$

contains the pair $\big((P \parallel_L R), (Q \parallel_L S)\big)$ and is an $\mathfrak{F}$-dimmed strong bisimulation. The fact that the pair is in $\mathcal{R}$ promptly follows. To show that $\mathcal{R}$ is an $\mathfrak{F}$-dimmed strong bisimulation suffices to prove that for any pair $\big((P' \parallel_L R'), (Q' \parallel_L S')\big) \in \mathcal{R}$ if $(P' \parallel_L R') \xrightarrow{a} E$, then there exists an action $b \in [a]$ and a transition $(Q' \parallel_L S') \xrightarrow{b} T$ such that $(E,T) \in \mathcal{R}$.
Let us assume $(P' \parallel_L R') \xrightarrow{a} E$ and consider a case distinction:

1) $\underline{a \notin L}$. Assume that $E = P'' \parallel_L R'$. By the operational semantics it must hold that $P' \xrightarrow{b} P''$. Exploiting the assumption that $P' \sim_{\mathfrak{F}} Q'$, we can infer the existence of an action $b \in [a]$ and of a transition $Q' \xrightarrow{b} Q''$ such

that $P'' \sim_{\mathfrak{F}} Q''$. Due to the fact that $a \notin L$ and L is singleton coherent with $\mathfrak{F}$, we have that $b \notin L$. Therefore, $(Q' \parallel_L S') \xrightarrow{b} T$, with $T = (Q'' \parallel_L S')$, and $(E, T) \in \mathcal{R}$. The case in which we assume $E = P' \parallel_L R''$ can be proved similarly.

2) $\underline{a \in L}$. By the operational semantics, there must exist transitions $P' \xrightarrow{a} P''$ and $R' \xrightarrow{a} R''$ and $E = P'' \parallel_L R''$. Exploiting the assumptions $P' \sim_{\mathfrak{F}} Q'$ and $R' \sim_{\mathfrak{F}} S'$ we can infer the existence of an action $b \in [a]$ and a transition $Q' \xrightarrow{b} Q''$ such that $P'' \sim_{\mathfrak{F}} Q''$, as well as the existence of an action $c \in [a]$ and a transition $S' \xrightarrow{c} S''$ such that $R'' \sim_{\mathfrak{F}} S''$. The assumption of L being singleton coherent with $\mathfrak{F}$ assures us that $a = b = c$. Hence, from $Q' \xrightarrow{a} Q''$ and $S' \xrightarrow{a} S''$ we have that $(Q' \parallel_L S') \xrightarrow{a} T$, with $T = (Q'' \parallel_L S'')$ and $(E, T) \in \mathcal{R}$.

Proof (Theorem 4). The proofs of the points $i) - iv)$ are similar to those previously shown for the compositionality for dimmed strong bisimulation.

v) In order to prove this point, we shall provide, under the assumption that L is block coherent with $\mathfrak{F}$ and the assumption of homogeneity for Q and S, that the relation

$$\mathcal{R} \triangleq \Big\{ \big((P' \parallel_L R'), (Q' \parallel_L S')\big) \mid P' \preceq_{\mathfrak{F}} Q', R' \preceq_{\mathfrak{F}} S', \text{ and } Q' \text{ and } S' \text{ are homogeneous for } L \Big\},$$

contains the pair $\big((P \parallel_L R), (Q \parallel_L S)\big)$ and is an $\mathfrak{F}$-dimmed strong simulation. The fact that the pair is in $\mathcal{R}$ promptly follows. To show that $\mathcal{R}$ is an $\mathfrak{F}$-dimmed strong simulation, we must prove that for any pair $\big((P' \parallel_L R'), (Q' \parallel_L S')\big) \in \mathcal{R}$ if $(P' \parallel_L R') \xrightarrow{a} E$ then there exists an action $b \in [a]$ and a transition $(Q' \parallel_L S') \xrightarrow{b} T$ with $(E, T) \in \mathcal{R}$.

Let us assume $(P' \parallel_L R') \xrightarrow{a} E$ and distinguish the following two cases:

1) $\underline{a \notin L}$. Assume that $E = P'' \parallel_L R'$. By the operational semantics it must hold that $P' \xrightarrow{a} P''$. Exploiting the assumption that $P' \preceq_{\mathfrak{F}} Q'$, we can infer the existence of an action $b \in [a]$ and a transition $Q' \xrightarrow{b} Q''$ such that $P'' \preceq_{\mathfrak{F}} Q''$. Since $a \notin L$, the fact that L is block coherent with $\mathfrak{F}$ guarantees that $[a] \cap L = \emptyset$, assuring that $b \notin L$. Therefore, $(Q' \parallel_L S') \xrightarrow{b} T$, with $T = (Q'' \parallel_L S')$. The fact that the pair $(E, T) \in \mathcal{R}$ follows by noticing that if Q' and S' are homogeneous for L, then Q'' and S' are still homogeneous for L because $Act(Q'') \subseteq Act(Q')$ since $Q' \xrightarrow{b} Q''$. The case in which we assume $E = P' \parallel_L R''$ can be proved similarly.

2) $\underline{a \in L}$. By the operational semantics, there must exist transitions $P' \xrightarrow{a} P''$ and $R' \xrightarrow{a} R''$ and $E = P'' \parallel_L R''$. Exploiting the assumptions $P' \preceq_{\mathfrak{F}} Q'$ and $R' \preceq_{\mathfrak{F}} S'$, we can infer the existence of an action $b \in [a]$ and a transition $Q' \xrightarrow{b} Q''$ such that $P'' \preceq_{\mathfrak{F}} Q''$, as well as the existence of an action $c \in [a]$ and a transition $S' \xrightarrow{c} S''$ such that $R'' \preceq_{\mathfrak{F}} S''$. If $a \in L$ it must hold that $[a] \cap L \neq \emptyset$; the assumption that L is block coherent

with $\mathfrak{F}$ guarantees then that b, c are both in L. Moreover, due to the homogeneity assumption of Q' and S', it holds that $b = c$. In fact, let us assume toward a contradiction that it is not the case i.e., $b \neq c$. Since $Q' \xrightarrow{b} Q''$ and $S' \xrightarrow{c} S''$ we have that $b \in Act(Q')$ and $c \in Act(S')$; thus it holds that $\{b, c\} \subseteq Act(Q') \cup Act(S')$. Since $\{c, d\} \subseteq [a]$, we conclude that $\big|\big(Act(Q') \cup Act(S')\big) \cap [a]\big| \geq 2$; a contradiction to the hypothesis of homogeneity. Hence, from $Q' \xrightarrow{b=c} Q''$ and $S' \xrightarrow{c=b} S''$ we have that $(Q' \parallel_L S') \xrightarrow{c} T$, with $T = (Q'' \parallel_L S'')$ and $(E, T) \in \mathcal{R}$ because Q'' and S'' are still homogeneous for L for a similar reason to point 1).

C Examples

Example 2. Using the process definitions in the main text, here we show that

$$M \sim_{\mathfrak{F}} \bar{M}, \qquad \text{with} \quad \bar{M} := P_1[m_1 + m_2] \parallel_{L_1} B_1[k] \parallel_{L_2} C_1[n_1 + n_2].$$

We do this by exhibiting a relation $\mathcal{R}$ which contains the pair $(M, \bar{M})$. We proceed by first introducing a *counting function*: Given a matching process P and a model M, it returns the number of sub-terms of M that are syntactically equal to P.

Definition 10. *The counting function of P in M, denoted by $\mathcal{C}(P, M)$, is recursively defined as follows.*

$$\mathcal{C}(P, M) = \begin{cases} 1 & \text{if } M = P = \mathbf{0}, M = P = a.Q, \\ & \quad M = P = Q + R, \text{ or } M = P = K, K \in \mathcal{K}, \\ \mathcal{C}(P, Q) & \text{if } M = Q[f], \\ \mathcal{C}(P, Q) + \mathcal{C}(P, R) & \text{if } M = Q \parallel_L R, \\ \mathcal{C}(P, Q) & \text{if } M = Q/L, \\ 0 & \text{otherwise.} \end{cases}$$

Next, we introduce a notational convenience by removing the static operators from the process representation and by only listing the sequential processes. For instance, $\bar{M}$ is denoted by the tuple

$$(\underbrace{P_1, \ldots, P_1}_{m_1+m_2 \text{ times}}, \underbrace{B_1, \ldots, B_1}_{k \text{ times}}, \underbrace{C_1, \ldots, C_1}_{n_1+n_2 \text{ times}}).$$

In this case it holds that $\mathcal{C}(P_1, \bar{M}) = m_1 + m_2$ and $\mathcal{C}(B_2, \bar{M}) = 0$.

A generic derivative of M, denoted by M', will be in the form

$$M' = (P'_1, \ldots, P'_{m_1}, P''_1, \ldots, P''_{m_2}, B'_1, \ldots, B'_k, C'_1, \ldots, C'_{n_1}, C''_1, \ldots, C''_{n_2}),$$

with

$$\begin{aligned}
P'_{i_1} &\in \{P_1, put_1.P_1\}, && 1 \leq i_1 \leq m_1,\\
P''_{i_2} &\in \{P_2, put_2.P_2\}, && 1 \leq i_2 \leq m_2,\\
B'_j &\in \{B, get_1.B, get_2.B\}, && 1 \leq j \leq k,\\
C'_{l_1} &\in \{C_1, cons_1.C_1\}, && 1 \leq l_1 \leq n_1,\\
C''_{l_2} &\in \{C_2, cons_2.C_2\}, && 1 \leq l_2 \leq n_2.
\end{aligned}$$

Similarly, a generic derivative of $\bar{M}$, denoted by $\bar{M}'$, will be in the form

$$\bar{M}' = (\bar{P}'_1, \ldots, \bar{P}'_{m_1+m_2}, \bar{B}'_1, \ldots, \bar{B}'_k, C'_1, \ldots, \bar{C}'_{n_1+n_2}),$$

with

$$\begin{aligned}
\bar{P}'_{i_1} &\in \{P_1, put_1.P_1\}, && 1 \leq i_1 \leq m_1 + m_2,\\
\bar{B}'_j &\in \{B, get_1.B\}, && 1 \leq j \leq k,\\
\bar{C}'_{l_1} &\in \{C_1, cons_1.C_1\}, && 1 \leq l_1 \leq n_1 + n_2.
\end{aligned}$$

Then let us consider the relation $\mathcal{R}$ constructed as follows.

$$\begin{aligned}
\mathcal{R} = \big\{(M', \bar{M}') \mid\ & \mathcal{C}(P_1, M') + \mathcal{C}(P_2, M') = \mathcal{C}(P_1, \bar{M}'),\\
& \mathcal{C}(put_1.P_1, M') + \mathcal{C}(put_2.P_2, M') = \mathcal{C}(put_1.P_1, \bar{M}'),\\
& \mathcal{C}(B, M') = \mathcal{C}(B_1, \bar{M}'), \mathcal{C}(get_1.B, M') + \mathcal{C}(get_2.B, M') = \mathcal{C}(get_1.B_1, \bar{M}'),\\
& \mathcal{C}(C_1, M') + \mathcal{C}(C_2, M') = \mathcal{C}(C_1, \bar{M}'),\\
& \mathcal{C}(cons_1.C_1, M') + \mathcal{C}(cons_2.C_2, M') = \mathcal{C}(cons_1.C_1, \bar{M}')\big\}.
\end{aligned}$$

This is a dimmed bisimulation in order to show that $M \sim_{\mathfrak{F}} \bar{M}$. Indeed, $(M, \bar{M}) \in \mathcal{R}$. Then, let us consider two processes $(M', \bar{M}') \in \mathcal{R}$. We need to consider all possible transition that are enabled, using the operational semantics.

We focus on the case $M' \xrightarrow{put_1} M''$. It must hold that there must be one i_1, with $1 \leq i_1 \leq m_1$, such that $P'_{i_1} = put_1.P_1$ and one j, with $1 \leq j \leq k$, such that $B'_j = B$. Then, M'' will be in the form

$$M'' = (P'_1, \ldots, \underbrace{P_1}_{\text{pos. } i_1} \ldots, P'_{m_1}, P''_1, \ldots, P''_{m_2}, B'_1, \ldots, \underbrace{get_1.B}_{\text{pos. } j}, \ldots, B'_k, C'_1, \ldots, C'_{n_1}, C''_1, \ldots, C''_{n_2}).$$

That is, M'' is syntactically equal to M' expect for positions i_1 and j. Now, the fact that $P'_{i_1} = put_1.P_1$ and $B'_j = B$ implies that $\mathcal{C}(put_1.P_1, \bar{M}') \geq 1$ and $\mathcal{C}(B_1, \bar{M}') \geq 1$. Suppose that one such $put_1.P_1$ and B_1 are located in positions $\bar{i}_1$ and $\bar{j}$, respectively, with $1 \leq \bar{i}_1 \leq m_1 + m_2$ and $1 \leq \bar{j} \leq k$. Therefore, the exists a transition $\bar{M}' \xrightarrow{put_1} \bar{M}''$, with

$$\bar{M}'' = (\bar{P}'_1, \ldots, \underbrace{P_1}_{\text{pos. } \bar{i}_1}, \ldots, \bar{P}'_{m_1+m_2}, \bar{B}'_1, \ldots, \underbrace{get_1.B}_{\text{pos. } \bar{j}}, \ldots, \bar{B}'_k, C'_1, \ldots, \bar{C}'_{n_1+n_2})$$

which is syntactically equal to $\bar{M}'$ except for positions $\bar{i}_1$ and $\bar{j}$. Since $(M', \bar{M}') \in \mathcal{R}$ by assumption, it also holds that $(M'', \bar{M}'') \in \mathcal{R}$ because the changes have involved the same subterms; that is, exactly one $put_1.P_1$ (resp. B) subterm in M' and $\bar{M}'$ have become P_1 (resp. $get_1.B$) in M'' and $\bar{M}''$.

The vice versa, i.e., assuming that $\bar{M}' \xrightarrow{put_1} \bar{M}''$, is similar. In this case we have that $\bar{M}'$ must be such that there exists one $\bar{i}_1$ such that $\bar{P}'_{\bar{i}_1} = put_1.P_1$ and one $\bar{j}$ such that $\bar{B}'_{\bar{j}} = B_1$, which in turn implies that $\mathcal{C}(put_1.P_1, \bar{M}') \geq 1$ and that $\mathcal{C}(B_1, \bar{M}') \geq 1$. Now, $\bar{M}''$ will be syntactically equal to $\bar{M}'$ expect for positions $\bar{i}_1$ and $\bar{j}$, where it features P_1 and $get_1.B_1$, respectively. Because of the relation $\mathcal{R}$, we have that $\mathcal{C}(put_1.P_1, M') + \mathcal{C}(put_2.P_2, M') \geq 1$ and $\mathcal{C}(B, M') \geq 1$.

We now distinguish the following three cases: i) $\mathcal{C}(put_1.P_1, M') = 0$, hence $\mathcal{C}(put_2.P_2, M') = \mathcal{C}(put_1.P_1, \bar{M}')$; ii) $\mathcal{C}(put_2.P_2, M') = 0$, hence $\mathcal{C}(put_1.P_1, M') = \mathcal{C}(put_1.P_1, \bar{M}')$; iii) $\mathcal{C}(put_1.P_1, M') > 0$ and $\mathcal{C}(put_2.P_2, M') > 0$. The most interesting case is i). Since we have that $\mathcal{C}(put_2.P_2, M') > 1$, then there must be some i_2, with $1 \leq i_2 \leq m_2$ such that $P''_{i_2} = put_2.P_2$. Since $\mathcal{C}(B, M') \geq 1$, then there exists some j, $1 \leq j \leq k$ such that $B'_j = B$. Then, $M' \xrightarrow{put_2} M''$ (recall that $put_2 \in [put_1]$), where

$$M'' = (P'_1, \ldots, P'_{m_1}, P''_1, \ldots, \underbrace{P_2}_{\text{pos. } i_2}, \ldots P''_{m_2}, B'_1, \ldots, \underbrace{get_2.B}_{\text{pos. } j}, \ldots, B'_k, C'_1, \ldots, C'_{n_1}, C''_1, \ldots, C''_{n_2}).$$

M'' will be syntactically equivalent to M', except for positions i_2 and j, where it features P_2 and $get_2.B$. Now, it holds that $(M'', \bar{M}'') \in \mathcal{R}$ because although the affected terms in the transitions from M'' and $\bar{M}''$ are different, the conditions on the equalities between all counting functions do hold.

All other cases regarding the possible transitions from M' and $\bar{M}'$ can be treated in a similar fashion.

Example 3. Using the process definitions in the main text, we wish to show that

$$M := P_1 \parallel_L P_2 \parallel_L P_3 \sim_{\mathfrak{F}} \bar{P} \parallel_L \bar{P} \parallel_L \bar{P} := \bar{M}.$$

To this end, let us consider the process representation as in the previous example, that is $M = (P_1, P_2, P_3)$ and $\bar{M} = (\bar{P}, \bar{P}, \bar{P})$, and the relation $\mathcal{R}$ defined as

$$\begin{aligned}\mathcal{R} = \big\{ \big(&(R_1, R_2, R_3), (S_1, S_2, S_3) \big) \mid \\ &(R_1, R_2, R_3) \in \{P_1, Q_1\} \times \{P_2, Q_2\} \times \{P_3, Q_3\}, \\ &(S_1, S_2, S_3) \in \{\bar{P}, \bar{Q}\} \times \{\bar{P}, \bar{Q}\} \times \{\bar{P}, \bar{Q}\}, \\ &R_i = P_i \iff S_i = \bar{P}, R_i = Q_i \iff S_i = \bar{Q}, 1 \leq i \leq 3 \big\}.\end{aligned}$$

Then, $(M, \bar{M}) \in \mathcal{R}$ and it is easy to check that $\mathcal{R}$ is a dimmed bisimulation. For instance, suppose that $(R_1, R_2, R_3) \xrightarrow{a_{13}} (R'_1, R'_2, R'_3)$. Then, by the operational semantics it must hold that $R_1 = P_1$ and $R_3 = P_3$. There are two cases: i)

$R_2 = P_2$ or ii) $R_2 = Q_2$. Let us consider case i), the other being similar. Then, we have that $(R'_1, R'_2, R'_3) = (Q_1, P_2, Q_3)$. By construction of $\mathcal{R}$ it holds that $(S_1, S_2, S_3) = (\bar{P}, \bar{P}, \bar{P})$. There is a transition $(\bar{P}, \bar{P}, \bar{P}) \xrightarrow{a_{12}} (\bar{Q}, \bar{P}, \bar{Q})$, with $a_{12} \in [a_{13}]$, and it holds that $\big((Q_1, P_2, Q_3), (\bar{Q}, \bar{P}, \bar{Q})\big) \in \mathcal{R}$.

Vice versa, suppose that $(S_1, S_2, S_3) \xrightarrow{a_{12}} (S'_1, S'_2, S'_3)$. We need to separately consider the 8 possible source states that may lead to such a transition. For instance, let us take $(S_1, S_2, S_3) = (\bar{P}, \bar{Q}, \bar{P})$. Then we have that $(S'_1, S'_2, S'_3) = (\bar{Q}, \bar{Q}, \bar{Q})$ and that $(R_1, R_2, R_3) = (P_1, Q_2, P_3)$. This state affords an a_{13}-transition, i.e., $(P_1, Q_2, P_3) \xrightarrow{a_{13}} (Q_1, Q_2, Q_3)$, with $a_{13} \in [a_{12}]$ and we have that $\big((Q_1, Q_2, Q_3), (\bar{Q}, \bar{Q}, \bar{Q})\big) \in \mathcal{R}$. The other cases are treated analogously.

INSTITUTE FOR ADVANCED STUDIES LUCCA