RWTH Aachen

Department of Computer Science
Technical Report

Compositional Abstraction for Stochastic Systems

Joost-Pieter Katoen, Daniel Klink, and Martin Neuhäußer

ISSN 0935–3232 · Aachener Informatik Berichte · AIB-2009-15

RWTH Aachen · Department of Computer Science · June 2009

Compositional Abstraction for Stochastic Systems[⋆]

Joost-Pieter Katoen[1,2], Daniel Klink[1], and Martin R. Neuhäußer[1,2]

[1] RWTH Aachen University, Germany and [2] University of Twente, The Netherlands

Abstract. We propose to exploit three-valued abstraction to stochastic systems in a compositional way. This combines the strengths of an aggressive state-based abstraction technique with compositional modeling. Applying this principle to interactive Markov chains yields abstract models that combine interval Markov chains and modal transition systems in a natural and orthogonal way. We prove the correctness of our technique for parallel and symmetric composition and show that it yields lower bounds for minimal and upper bounds for maximal timed reachability probabilities.

1 Introduction

To overcome the absence of hierarchical, compositional facilities in performance modeling, several efforts have been undertaken to integrate performance aspects, most notably probability distributions, into compositional modeling formalisms. Resulting formalisms are, among others, extensions of the Petri box calculus [27], Statecharts [3], and process algebras [17, 13]. To bridge the gap towards classical performance and dependability analysis, compositional formalisms for continuous-time Markov chains (CTMCs) have received quite some attention. Nowadays, these formalisms are also used intensively in, e.g., the area of systems biology [4].

An elegant and prominent semantic model in this context are interactive Markov chains [12, 14]. They extend CTMCs with nondeterminism, or viewed differently, enrich labeled transition systems with exponential sojourn times in a fully orthogonal and simple manner. They naturally support the specification of phase-type distributions, i.e., sojourn times that are non-exponential, and facilitate the compositional integration of random timing constraints in purely functional models [14]. In addition, bisimulation quotienting can be done in a compositional fashion reducing the peak memory consumption during minimization. This has been applied to several examples yielding substantial state-space reductions, and allowing the analysis of CTMCs that could not be analyzed without compositional quotienting [14, 9, 10].

This paper goes an important step further by proposing a framework to perform more aggressive abstraction of interactive Markov chains (IMCs) in a compositional manner. We consider state-based abstraction that allows to represent any (disjoint) group of concrete states by a single abstract state. This flexible abstraction mechanism generalizes bisimulation minimization (where "only" bisimilar states are grouped) and yields an overapproximation of the IMC under consideration. This abstraction is a natural mixture of abstraction of labeled

---

[⋆] The research has been funded by the DFG Research Training Group 1298 (AlgoSyn), the NWO project QUPES (612.000.420) and the EU FP7 project Quasimodo.

transition systems by modal transition systems [26, 25] and abstraction of probabilities by intervals [8, 21]. Abstraction is shown to preserve simulation, that is to say, abstract models simulate concrete ones. Here, simulation is a simple combination of refinement of modal transition systems [25] and probabilistic simulation [20]. It is shown that abstraction yields lower bounds for minimal and upper bounds for maximal timed reachability probabilities.

Compositional aggregation is facilitated by the fact that simulation is a precongruence with respect to TCSP-like parallel composition and symmetric composition [15] on our abstract model. Accordingly, components can be abstracted prior to composing them. As this abstraction is coarser than bisimulation, a significantly larger state-space reduction may be achieved and peak memory consumption is reduced. This becomes even more advantageous when components that differ only marginally are abstracted by the same abstract model. In this case, the symmetric composition of these abstract components may yield huge reductions compared to the parallel composition of the slightly differing concrete ones. A small example shows this effect, and shows that the obtained bounds for timed reachability probabilities are rather exact.

Several abstraction techniques for (discrete) probabilistic models have been developed so far. However, compositional ones that go beyond bisimulation are rare. Notable exceptions are Segala's work on simulation preorders for probabilistic automata [28] and language-level abstraction for PRISM [23]. Note that compositional abstractions have been proposed in other settings such as traditional model checking [29, 30] and for timed automata [2]. Compositional analysis techniques for probabilistic systems have been investigated in [6, 31]. Alternative abstraction techniques have, e.g., been studied in [7, 5, 24].

Outline. Section 2 gives some necessary background. In section 3 and 4, AIMCs are introduced for which we investigate parallel and symmetric composition in section 5. Section 6 shows how to consistently abstract components. In section 7 we focus on the computation of time-bounded reachability probabilities.

2 Preliminaries

Let X be a finite set. For $Y, Y' \subseteq X$ and function $f : X \times X \to \mathbb{R}$ let $f(Y, Y') = \sum_{y \in Y, y' \in Y'} f(y, y')$ (for singleton sets, brackets may be omitted). Function $f(x, \cdot)$ is given by $x' \mapsto f(x, x')$ for all $x \in X$; further, by $f[y \mapsto x]$ we denote the function that agrees with f except at $y \in X$ where it equals x. Function f is a *distribution on* X iff $f : X \to [0, 1]$ and $f(X) = \sum_{x \in X} f(x) = 1$. The support of a distribution f is $\mathit{supp}(f) = \{x \in X \mid f(x) > 0\}$ and the set of all distributions on X is denoted by $\mathit{distr}(X)$. Let $\mathbb{B}_2 = \{\bot, \top\}$ be the two-valued truth domain.

Interactive Markov chains, a formalism for compositional modeling systems embracing nondeterministic and stochastic behavior, have been thoroughly investigated in [12]. They can be seen as an extension of transition systems with exponentially distributed delays and probabilism. We consider a restricted form, where all delays are exponentially distributed with the same exit rate. These *uniform* IMCs have been successfully adopted for the performability analysis of STATEMATE models [11] by specifying random time constraints as CTMCs that are composed with the functional behavior as in [14]. As CTMCs can simply be transformed into weakly bisimilar uniform ones, uniform IMCs result.

Definition 1 (Uniform IMC). *A uniform interactive Markov chain (IMC) is a tuple* $(S, A, \mathbf{L}, \mathbf{P}, \lambda, s_0)$ *where*

- S *is a non-empty finite set of states with initial state* $s_0 \in S$*,*
- $A = A_e \uplus A_i$ *is a non-empty finite set of* external *and* internal *actions,*
- $\mathbf{L} : S \times A \times S \rightarrow \mathbb{B}_2$ *is a two-valued labeled transition relation,*
- $\mathbf{P} : S \times S \rightarrow [0, 1]$ *is a transition probability function such that for all* $s \in S$ *it holds* $\mathbf{P}(s, S) = 1$*,*
- $\lambda \in \mathbb{R}^+$ *is a uniform exit rate.*

A *Markovian transition* leads from state s to state s' (denoted $s \dashrightarrow s'$) iff $\mathbf{P}(s, s') > 0$; intuitively, if $s \dashrightarrow s'$, the probability to take this transition equals $\mathbf{P}(s, s')$ whereas the residence time in state s is exponentially distributed with rate λ. We require $\mathbf{P}(s, S) = 1$ to exclude deadlock states; this can easily be achieved by adding Markovian self-loops to states without Markovian transitions. Similarly, an *interactive transition* leads from s to s' via action a (denoted $s \xrightarrow{a} s'$) iff $\mathbf{L}(s, a, s') = \top$. *External* actions $a \in A_e$ allow synchronization with the environment whereas *internal* actions $\tau \in A_i$ happen instantaneously and autonomously. The *maximal progress assumption* [12] states that whenever internal transitions exist in the current state, the system nondeterministically moves along one of these transitions ignoring all other Markovian and external transitions. This ensures that internal actions cannot be delayed.

Example 1. As a running example, we consider the IMC model of a worker, depicted in Fig. 1, where $\lambda = 10$. The work cycle starts in s_0 where the quality of a piece of raw material has to be determined. One out of ten pieces is flawed and cannot be used to craft a *premium* product. In that case (s_1) the worker will only be able to make a *value* product, which may take several work steps.

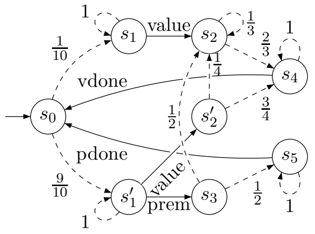

Fig. 1. An IMC.

If the raw material is flawless, the worker decides for *value* or *premium.* For a *premium* product (s_3), everything has to be done smoothly in the first attempt, however, if the result is not perfect, with some corrections, a value product will be made. If the worker decides for *value* (s'_2), chances that no corrections are necessary are better than for the case that the raw material was flawed.

We call an IMC *closed* if all actions are internal. On the one hand, closing a system by turning external actions to internal ones prevents any further interaction, on the other hand it allows for quantitative analysis [18].

3 Abstract Interactive Markov Chains

In this paper, we aim at abstracting an IMC by collapsing disjoint sets of concrete states into single abstract ones. In contrast to bisimulation quotienting where bisimilar states are grouped, here groups of states can (in principle) be chosen arbitrarily. In fact, we abstract an IMC along two lines: We use must- and may-transitions as introduced for modal transition systems [26] to abstract from differences in the states' available nondeterministic choices. Further, instead of only considering fixed transition probabilities, we follow the approach taken in interval Markov chains [8, 21] and allow to specify intervals of transition probabilities. The combination of these two ingredients yields:

Definition 2 (Abstract IMC). *An abstract IMC is a tuple* $(S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ *where* S, A, λ *and* s_0 *are as before, and*

- $\mathbf{L} : S \times A \times S \to \mathbb{B}_3$ *is a* three-valued *labeled transition relation, and*
- $\mathbf{P}_l, \mathbf{P}_u : S \times S \to [0, 1]$ *are transition probability bound functions such that* $\mathbf{P}_l(s, S) \leq 1 \leq \mathbf{P}_u(s, S)$ *for all* $s \in S$.

Here $\mathbb{B}_3 := \{\bot, ?, \top\}$ is the complete lattice with the usual ordering $\bot < ? < \top$ and meet ($\sqcap$) and join ($\sqcup$) operations. The labeling $\mathbf{L}(s, a, s')$ identifies the transition "type": $\top$ indicates must-transitions, ? may-transitions, and $\bot$ the absence of a transition. Note that any IMC is an AIMC without may-transitions for which $\mathbf{P}_l = \mathbf{P}_u = \mathbf{P}$. Further, any interval Markov chain is an AIMC without must- and may-transitions. The requirement $\mathbf{P}_l(s, S) \leq 1 \leq \mathbf{P}_u(s, S)$ ensures that in every state s, a distribution μ over successor states can be chosen such that $\mathbf{P}_l(s, s') \leq \mu(s') \leq \mathbf{P}_u(s, s')$ for all $s' \in S$. This can be achieved by equipping such states with a Markovian $[1, 1]$ self-loop, without altering the model's behavior: if state s has an outgoing internal interactive transition, the maximal progress assumption guarantees that it still takes priority; otherwise, the self-loop neither alters its synchronization capabilities nor its sojourn time.

Example 2. Figure 2 (middle) depicts an example abstract model (AIMC) of a worker, similar to the one in Fig. 2 (left). It abstracts from the difference between the raw material quality represented by the states s_1 and s'_1 in Fig. 2 (left). Instead, the *premium* choice is modeled as a *may*-transition, i.e., it is possible to decide for *premium* in state u_1 but this possibility may be omitted. In state u_2, the probability that no further working step is necessary varies from $\frac{2}{3}$ to $\frac{3}{4}$. We abbreviate point intervals of the form $[p, p]$ and simply write p.

Closing. AIMCs are (like IMCs and transition systems) subject to interaction. In order to carry out a quantitative analysis of such "open" models, one typically considers a closed variant, i.e., a variant that is behaviorally the same, but can no longer interact. This corresponds to the hiding operation in process algebras where external actions are turned into internal (τ)-actions. We keep slightly more information: the distributions in case of a Markovian transition, and the target state id for interactive transitions. This facilitates a transformation of an AIMC into a continuous-time MDP as described later on.

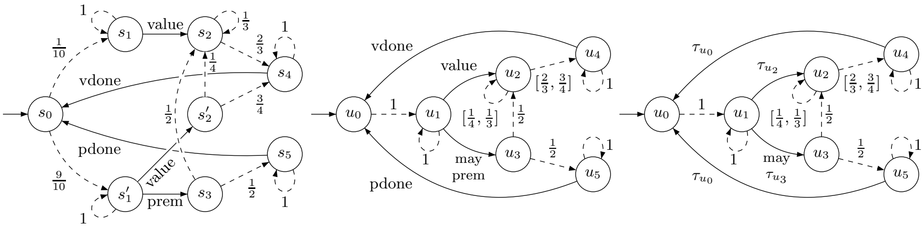

Fig. 2. An open IMC (left), an open AIMC (middle) and its closed version (right).

Definition 3 (Closed AIMC). *An AIMC* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ *induces the closed AIMC* $\mathcal{M}_\tau = (S, A_\tau, \mathbf{L}_\tau, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ *where* $A_\tau = \bigcup_{s \in S} A_s^I \cup A_s^M$ *and*

$$\begin{aligned} A_s^I &= \{\tau_{s'} \mid \exists s' \in S.\ \exists a \in A.\ \mathbf{L}(s, a, s') \neq \bot\}\,, \\ A_s^M &= \{\tau_\mu \mid \exists \mu \in \mathit{distr}(S).\ \forall s' \in S.\ \mathbf{P}_l(s, s') \leq \mu(s') \leq \mathbf{P}_u(s, s')\}\,, \\ \mathbf{L}_\tau(s, \tau, s') &= \begin{cases} \bigsqcup_{a \in A} \mathbf{L}(s, a, s') & \text{if } \tau = \tau_{s'} \\ \bot & \text{otherwise.} \end{cases} \end{aligned}$$

In general, the sets A_s^M are uncountable as the range $[\mathbf{P}_l(s, s'), \mathbf{P}_u(s, s')]$ is dense. A key aspect in our approach is how to deal with these uncountable sets of distributions. We will show in Section 4 that it suffices to consider only a *finite* subset for the analysis.

Example 3. Fig. 2 (right) illustrates the closed induced AIMC of Fig. 2 (middle).

4 Nondeterminism

In a closed AIMC, we classify states according to the type of outgoing transitions: the state space S is partitioned into the sets of *Markovian states* S_M, *hybrid states* S_H and *may states* S_{MH}. A state is *Markovian* iff only Markovian transitions leave that state; a state is *hybrid* iff it has emanating Markovian and must-transitions. Further, states in S_{MH} only have outgoing Markovian and may-transitions but no must-transitions. By assumption, any state has at least one outgoing Markovian transition; hence, deadlock states do not exist.

According to this state classification, three sources of nondeterminism occur in AIMCs: If multiple must-transitions exist in a state $s \in S_H$, that is, if $\mathbf{L}(s, a, s') = \mathbf{L}(s, b, s'')$ for some $a, b \in A_s^I$ and $s' \neq s''$, the decision which transition to take is nondeterministic. Due to the maximal progress assumption, nondeterminism only occurs between internal transitions.

May-transitions induce the second indefinite behavior: If $\mathbf{L}(s, a, s') = ?$ for some $a \in A_s^I$ and $s, s' \in S$, the existence of the may-transition to s' is nondeterministically resolved: In the positive case, the behavior is that of a hybrid state (i.e. the may-transition is treated as a must-transition). Otherwise, the may-transition will considered to be missing; if further must-transitions exist, the state is treated as a hybrid state, otherwise, it becomes a Markovian state.

The third type of nondeterminism occurs in Markovian states $s \in S_M$ of an AIMC: The abstraction yields transition probability intervals (formalized by $\mathbf{P}_l$ and $\mathbf{P}_u$) which induce a generally uncountable set of distributions that conform to these intervals. Selecting one of these distributions is nondeterministic. Note that in the special case of IMCs, the successor-state distribution is uniquely determined as $\mathbf{P}_l = \mathbf{P}_u$. Hence, IMCs do not exhibit this type of nondeterminism.

To formalize this intuition, let $A(s)$ be the set of *enabled actions* in state s. Formally, define $A(s) = A_s^I$ if $s \in S_H$, $A(s) = A_s^M$ if $s \in S_M$ and $A(s) = A_s^I \cup A_s^M$ if $s \in S_{MH}$. Each action $\tau \in A(s)$ represents a distribution over the successors of state s. We define (for arbitrary $\tau \in A_\tau$) the distribution $\mathbf{T}(\tau)$ such that $\mathbf{T}(\tau_\mu) = \mu$ if $\tau = \tau_\mu$ is a Markovian transition and $\mathbf{T}(\tau_s) = \{s \mapsto 1\}$ if $\tau = \tau_s$ is an internal action; further, we extend this notion to sets of actions: for $B \subseteq A_\tau$ let $\mathbf{T}(B) = \bigcup_{\tau \in B} \mathbf{T}(\tau)$. We use normalization as in [8] to restrict the intervals such that only valid probability distributions arise.

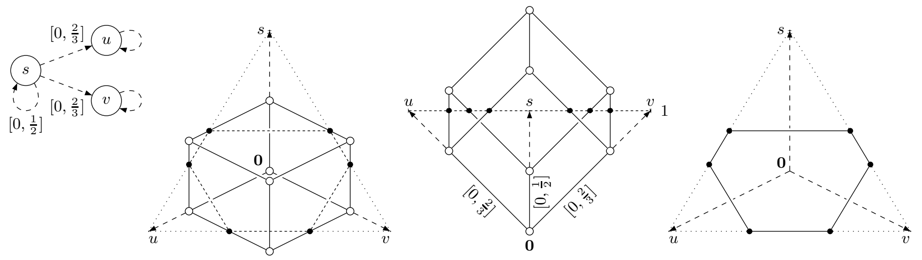

Fig. 3. Finite representation of infinitely many distributions.

Normalization. An AIMC $\mathcal{M}$ is called *delimited*, if for any state, every possible selection of a transition probability can be extended to a distribution, i.e., if for any $s, s' \in S$ and $p \in [\mathbf{P}_l(s, s'), \mathbf{P}_u(s, s')]$, we have $\mu(s') = p$ for some $\mu \in \mathbf{T}_{\mathcal{M}}(A_s^M)$. An AIMC $\mathcal{M}$ can be normalized, yielding the delimited AIMC $\eta(\mathcal{M})$ where $\mathbf{T}_{\eta(\mathcal{M})}(A_s^M) = \mathbf{T}_{\mathcal{M}}(A_s^M)$ for all $s \in S$. Formally, $\eta(\mathcal{M}) = (S, A, \mathbf{L}, \tilde{\mathbf{P}}_l, \tilde{\mathbf{P}}_u, \lambda, s_0)$ and $\eta(\mathbf{P}_l, \mathbf{P}_u) = (\tilde{\mathbf{P}}_l, \tilde{\mathbf{P}}_u)$ where for all $s, s' \in S$:

$$\begin{aligned}\tilde{\mathbf{P}}_l(s, s') &= \max\{\mathbf{P}_l(s, s'), 1 - \mathbf{P}_u(s, S \setminus \{s'\})\} \quad \text{and} \\ \tilde{\mathbf{P}}_u(s, s') &= \min\{\mathbf{P}_u(s, s'), 1 - \mathbf{P}_l(s, S \setminus \{s'\})\}.\end{aligned}$$

Example 4. The AIMC in Fig. 3 (left) is delimited. Selecting $\frac{2}{3}$ for the transition from s to u yields a non-delimited AIMC with $\mathbf{P}_l(s, \cdot) = (0, \frac{2}{3}, 0)$ and $\mathbf{P}_u(s, \cdot) = (\frac{1}{2}, \frac{2}{3}, \frac{2}{3})$. Applying normalization results in new upper bounds $(\frac{1}{3}, \frac{2}{3}, \frac{1}{3})$ and a delimited AIMC: for any probability $p \in [0, \frac{1}{3}]$ to take the self-loop, the probability to take the transition to v can be chosen as $\frac{1}{3} - p$ and vice versa.

Schedulers. In order to maximize (or minimize) the probability to reach a set of goal states B within a given time bound t (denoted $\Diamond^{\leq t} B$), we use schedulers which resolve the nondeterministic choices in the underlying AIMC. If the AIMC is in a state $s \in S$, a scheduler selects an enabled action $\tau \in A(s)$ to continue with. As shown in [1], schedulers that take the system's (time abstract) *history* into account yield better decisions than *positional* schedulers which only rely on the current state. A scheduler is *randomized*, if it may not only choose a single action but a distribution over all enabled actions in the current state.

Note that for Markovian states $s \in S_M$, the set A_s^M is generally uncountable as it consists of all distributions μ that obey the transition probability intervals of Markovian transitions emanating state s. Therefore, we reduce A_s^M to finitely many actions as follows: Consider the cube in Fig. 3. It represents all combinations of values that can be chosen from the three probability intervals $[0, \frac{1}{2}]$, $[0, \frac{2}{3}]$ and $[0, \frac{2}{3}]$ of the AIMC in Fig. 3 (left). The set *distr*(S) is represented by the dotted triangle. Hence, all points in the intersection of the cube and the triangle are valid distributions. For randomized schedulers, the six bold vertices spanning the intersection (right) serve as a finite representation of A_s^M: Every distribution $\mu \in \mathbf{T}(A_s^M)$ can be constructed as a convex combination of the six *extreme* distributions which span the intersection.

Definition 4 (Extreme distributions). *Let* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ *be a delimited AIMC,* $s \in S$ *and* $S' \subseteq S$*. We define* $\mathit{extr}(\mathbf{P}_l, \mathbf{P}_u, S', s) \subseteq \mathit{distr}(S)$ *such that* $\mu \in \mathit{extr}(\mathbf{P}_l, \mathbf{P}_u, S', s)$ *iff either* $S' = \emptyset$ *and* $\mu = \mathbf{P}_l(s, \cdot) = \mathbf{P}_u(s, \cdot)$ *or one of the following conditions holds:*

- $\exists s' \in S' : \mu(s') = \mathbf{P}_l(s, s') \wedge \mu \in \mathit{extr}(\eta(\mathbf{P}_l, \mathbf{P}_u[(s, s') \mapsto \mu(s')]), S' \setminus \{s'\}, s)$
- $\exists s' \in S' : \mu(s') = \mathbf{P}_u(s, s') \wedge \mu \in \mathit{extr}(\eta(\mathbf{P}_l[(s, s') \mapsto \mu(s')]), \mathbf{P}_u, S' \setminus \{s'\}, s)$

A distribution $\mu \in \mathit{extr}(\mathbf{P}_l, \mathbf{P}_u, S, s)$ *is called* extreme*.*

Lemma 1. *Let* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ *be an AIMC and* $s \in S$*. For any* $\mu \in \mathit{distr}(S)$ *with* $\mathbf{P}_l(s, s') \leq \mu(s') \leq \mathbf{P}_u(s, s')$ *for all* $s' \in S$*, there exists* $\bar{\mu} \in \mathit{distr}(\mathit{extr}(\mathbf{P}_l, \mathbf{P}_u, S, s))$ *such that for all* $s' \in S$

$$\mu(s') = \textstyle\sum_{\mu' \in \mathit{extr}(\mathbf{P}_l, \mathbf{P}_u, S, s)} \bar{\mu}(\mu')\mu'(s').$$

For randomized schedulers, we thus may replace the uncountable sets A_s^M in the induced closed AIMC by finite sets $A_s^{M,\mathit{extr}} = \{\tau_\mu \mid \mu \in \mathit{extr}(\mathbf{P}_l, \mathbf{P}_u, S, s)\}$. We use A_s^{extr} to denote the set $A_s^{M,\mathit{extr}} \cup A_s^I$; further, let $A^{\mathit{extr}} = \bigcup_{s \in S} A_s^{\mathit{extr}}$.

Paths. A *timed path* in a closed AIMC $\mathcal{M}_\tau$ is an infinite alternating sequence $\sigma = s_0 \tau_0 t_0 s_1 \tau_1 t_1 \ldots$ of states, internal actions and the states' residence times. A *path fragment* in $\mathcal{M}_\tau$ is a finite alternating sequence $\sigma = s_0 \tau_0 t_0 s_1 \ldots \tau_{n-1} t_{n-1} s_n$. Time-abstract paths (path fragments) are alternating sequences of states and actions only. The set of timed paths in $\mathcal{M}_\tau$ is denoted $\mathit{Paths}_{\mathcal{M}_\tau}$ whereas the set of timed path fragments of length n is denoted $\mathit{Pathf}^n_{\mathcal{M}_\tau}$; further, let $\mathit{Pathf}^\star_{\mathcal{M}_\tau} = \bigcup_{n=0}^\infty \mathit{Pathf}^n_{\mathcal{M}_\tau}$ be the set of all path fragments. In the following, we omit $\mathcal{M}_\tau$ whenever it is clear from the context; further, we denote the sets of time-abstract paths and path fragments by adding subscript *abs*.

By $\sigma[i]$ we denote the $(i{+}1)$-st state on the path, i.e. for $\sigma = s_0 \tau_0 t_0 s_1 \tau_1 t_1 \ldots$, we set $\sigma[i] = s_i$. By $\sigma@t$ we denote the state occupied at time t, i.e. $\sigma@t = s_i$ where i is the smallest index such that $t < \sum_{j=0}^{i} t_j$. For finite path $\sigma = s_0 \tau_0 t_0 \cdots \tau_{n-1} t_{n-1} s_n$, we define $\mathit{last}(\sigma) = s_n$ to denote the last state on σ.

We consider history-dependent randomized schedulers that choose from the set of extreme distributions and from interactive transitions:

Definition 5 (Extreme scheduler). *Let* $\mathcal{M}_\tau$ *be a closed AIMC. An* extreme scheduler *on* $\mathcal{M}_\tau$ *is a function* $D : \mathit{Pathf}^\star_{\mathit{abs}} \to \mathit{distr}\left(A^{\mathit{extr}}\right)$ *with* $\mathit{supp}(D(\sigma)) \subseteq A^{\mathit{extr}}_{\mathit{last}(\sigma)}$ *for all* $\sigma \in \mathit{Pathf}^\star_{\mathit{abs}}$*.*

Let $\mathcal{D}(\mathcal{M}_\tau)$ denote the set of extreme schedulers for $\mathcal{M}_\tau$. For $D \in \mathcal{D}(\mathcal{M}_\tau)$ and history $\sigma \in \mathit{Pathf}^\star_{\mathit{abs}}$, let the distribution over all successor states be given by $\sum_{\tau \in A^{\mathit{extr}}} D(\sigma)(\tau) \cdot \mathbf{T}(\tau)(s)$ for all $s \in S$.

Probability measure. We are interested in the infimum and supremum of probability measures on measurable sets of paths over all schedulers in $\mathcal{D}(\mathcal{M}_\tau)$. In the same fashion as for IMCs [18, p.53], for AIMCs the probability measure $\mathit{Pr}^\omega_{s,D}$ w.r.t. initial state s in $\mathcal{M}_\tau$ and $D \in \mathcal{D}(\mathcal{M}_\tau)$ can be inductively defined via *combined transitions* and *measurable schedulers*.

5 Composing AIMCs

We consider *parallel* and *symmetric* composition of AIMCs and show that the latter typically yields more compact models which are bisimilar to the parallel composition of identical components. These operators are defined in a TCSP-like manner, i.e., they are parameterized with a set of external actions that need to be performed simultaneously by all involved components. To define this multi-way synchronization principle, let for finite set X, the function $\mathbf{I} : X \times X \to \{\bot, \top\}$ be given by $\mathbf{I}(x, x') = \top$ iff $x = x'$. Similarly, let $\mathbf{1} : X \times X \to \{0, 1\}$ be defined by $\mathbf{1}(x, x') = 1$ iff $x = x'$. In the sequel of this paper, we assume that any AIMC is delimited unless stated otherwise.

Definition 6 (Parallel composition). *Let $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ and $\mathcal{M}' = (S', A', \mathbf{L}', \mathbf{P}'_l, \mathbf{P}'_u, \lambda', s'_0)$ be AIMCs. The* parallel composition *of $\mathcal{M}$ and $\mathcal{M}'$ w.r.t. synchronization set $\bar{A} \subseteq A_e \cap A'_e$ is defined by:*

$$\mathcal{M}||_{\bar{A}}\mathcal{M}' = (S \times S', A \cup A', \mathbf{L}'', \mathbf{P}''_l, \mathbf{P}''_u, \lambda + \lambda', (s_0, s'_0))$$

where for $s, u \in S$ and $s', u' \in S'$:

- $\mathbf{L}''((s, s'), a, (u, u'))$
$$= \begin{cases} (\mathbf{L}(s, a, u) \sqcap \mathbf{I}(s', u')) \sqcup (\mathbf{L}'(s', a, u') \sqcap \mathbf{I}(s, u)) & \text{if } a \notin \bar{A} \\ \mathbf{L}(s, a, u) \sqcap \mathbf{L}'(s', a, u') & \text{if } a \in \bar{A} \end{cases}$$
- $\mathbf{P}''_l((s, s'), (u, u')) = \frac{\lambda}{\lambda+\lambda'} \cdot \mathbf{P}_l(s, u) \cdot \mathbf{1}(s', u') + \frac{\lambda'}{\lambda+\lambda'} \cdot \mathbf{P}'_l(s', u') \cdot \mathbf{1}(s, u)$
- $\mathbf{P}''_u((s, s'), (u, u')) = \frac{\lambda}{\lambda+\lambda'} \cdot \mathbf{P}_u(s, u) \cdot \mathbf{1}(s', u') + \frac{\lambda'}{\lambda+\lambda'} \cdot \mathbf{P}'_u(s', u') \cdot \mathbf{1}(s, u)$

Non-synchronizing actions are interleaved while actions in the set $\bar{A}$ need to be performed simultaneously by the involved components. Due to the memoryless property of exponential distributions, parallelly composed components delay completely independently. This is similar as in Markovian process algebras and for parallel composition of IMCs [12, 14]. The proportion with which one of the components delays, i.e., $\frac{\lambda}{\lambda+\lambda'}$ and $\frac{\lambda'}{\lambda+\lambda'}$ respectively, results from the race between exponential distributions. This justifies the definition of $\mathbf{P}''_l$ and $\mathbf{P}''_u$.

Composing several instances of the same AIMC by parallel composition may lead to excessive state spaces. To alleviate this problem, we adopt the approach of [15] and also consider symmetric composition. To formally define this notion, we use the concept of multisets (or bags). A multiset M over a finite set S is a function $S \to \mathbb{N}$. $M(s)$ is the cardinality of s in M. We use common notations as $s \in M$ iff $M(x) > 0$ and e.g., $M = \{\!|a, a, b|\!\}$ for M over $\{a, b\}$ with $M(a) = 2$ and $M(b) = 1$. For multisets M, M' over S, $M \uplus M' = M''$ is a multiset for which $M''(s) = M(s) + M'(s)$ for all $s \in S$. The same applies to $M \setminus M' = M''$ where $M''(s) = \max(0, M(s) - M'(s))$. A *multiset relation* $R : S \times S \to \mathbb{N}$ is a mapping w.r.t. multisets M, M' over S, iff $R(s, S) = M(s)$ and $R(S, u) = M'(u)$. The set of all *mappings* w.r.t. multisets M, M' is denoted $\Gamma_{M,M'}$.

Definition 7 (Symmetric composition). *For AIMC $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ and $\bar{A} \subseteq A_e$, the* symmetric composition *of $n \in \mathbb{N}^+$ copies of $\mathcal{M}$ is given by:*

$$|||^n_{\bar{A}}\mathcal{M} = (S'', A, \mathbf{L}'', \mathbf{P}''_l, \mathbf{P}''_u, n\lambda, \{\!|\overbrace{s_0, \ldots, s_0}^{n \text{ times}}|\!\})$$

where $S'' = \{M : S \to \mathbb{N} \mid \sum_{s \in S} M(s) = n\}$ and for all $s'', u'' \in S''$:

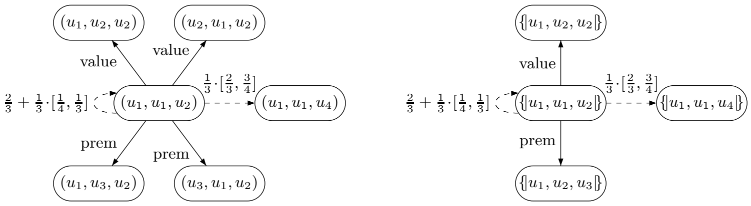

Fig. 5. Fragment of the parallel composition $\mathcal{M}||_\emptyset \mathcal{M}||_\emptyset \mathcal{M}$ (left) and the symmetric composition $|||^3_\emptyset \mathcal{M}$ (right) for open AIMC $\mathcal{M}$ from Fig. 2 (middle).

- $\mathbf{L}''(s'', a, u'') = \begin{cases} \bigsqcup_{s \in s'', u \in u'' : u'' = (s'' \setminus \{\!|s|\!\}) \uplus \{\!|u|\!\}} \mathbf{L}(s, a, u) & \textit{if } a \notin \bar{A} \\ \bigsqcup_{R \in \Gamma_{s'', u''}} \bigsqcap_{s, u \in S : R(s,u) > 0} \mathbf{L}(s, a, u) & \textit{if } a \in \bar{A} \end{cases}$
- $\mathbf{P}''_l(s'', u'') = \begin{cases} \frac{s''(s)}{n} \cdot \mathbf{P}_l(s, u) & \textit{if } s'' \neq u'' \textit{ and } u'' = (s'' \setminus \{\!|s|\!\}) \uplus \{\!|u|\!\} \\ \sum_{s \in S} \frac{s''(s)}{n} \cdot \mathbf{P}_l(s, s) & \textit{if } s'' = u'' \\ 0 & \textit{otherwise} \end{cases}$

The definition of $\mathbf{P}''_u$ *is obtained from* $\mathbf{P}''_l$ *by replacing all instances of* $\mathbf{P}_l$ *by* $\mathbf{P}_u$.

While in parallel compositions states are tuples, in symmetric compositions they are represented by multisets. Transitions, however, are defined in the very same fashion as for parallel composition. Non-synchronized actions of n components are interleaved and in the synchronized case, all components have to simultaneously take the same synchronizing action. For transition probabilities, as all instances of the same component have the same exit rate λ, each component *wins the race* with probability $\frac{1}{n}$.

The application of both composition operators on AIMCs results in another AIMC. Note that this also implies uniformity of the resulting model, cf. [11].

Lemma 2. *Let* $\mathcal{M}$ *and* $\mathcal{M}'$ *be AIMCs,* $\bar{A}$ *the synchronization set and* $n \in \mathbb{N}^+$, *then* $\mathcal{M}||_{\bar{A}} \mathcal{M}'$ *and* $|||^n_{\bar{A}} \mathcal{M}$ *are AIMCs.*

Example 5. Consider AIMC $\mathcal{M}$ in Fig. 4. For state $\{\!|s, s, u|\!\}$ in $|||^3_{\{a\}} \mathcal{M}$, the states reachable with a synchronized must a-transition are $\{\!|s, s, v|\!\}$, $\{\!|s, v, v|\!\}$, $\{\!|v, v, v|\!\}$ and the states reachable with a synchronized may-transition are $\{\!|s, s, s|\!\}$, $\{\!|s, s, v|\!\}$, $\{\!|s, v, v|\!\}$. Note that there are several ways for the system to move to states $\{\!|s, s, v|\!\}$ and $\{\!|s, v, v|\!\}$. In both cases, there exists a must-transition and thus a must a-transition leads from $\{\!|s, s, u|\!\}$ to $\{\!|s, s, v|\!\}$ and $\{\!|s, v, v|\!\}$ respectively.

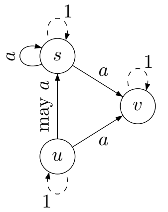

Fig. 4.

Example 6. Modeling three independent (abstract) workers as given in Fig. 2 can be done by both parallel and symmetric composition with an empty synchronization set. As shown in the table on the right, differences in the sizes of the resulting models are significant. Fig. 5 depicts the outgoing transitions of states (u_1, u_1, u_2) and $\{\!|u_1, u_1, u_2|\!\}$ that result from parallel and symmetric composition of three *abstract* workers.

states	IMC	AIMC
1 worker	8	6
3, par. comp.	512	216
3, sym. comp.	120	56

As suggested by Ex. 6, symmetric composition is a more space-efficient way to compose a component several times with itself. While for parallel composition of n identical components the size of the state space is in $\mathcal{O}(|S|^n)$, with symmetric composition, it is in $\mathcal{O}\left(\left(\binom{n-1+|S|}{n}\right)\right)$. The following result shows that symmetric composition yields models that are bisimilar to parallel composition of a component with itself. This generalizes a similar result for IMCs, cf. [15].

Definition 8 (Bisimulation). *Let $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ be an AIMC. An equivalence $R \subseteq S \times S$ is a* bisimulation *on $\mathcal{M}$, iff for any sRs' it holds:*

1. *for all $a \in A$ and $u \in S$ with $\mathbf{L}(s, a, u) \neq \bot$, there exists $u' \in S$ with*
$$\mathbf{L}(s, a, u) = \mathbf{L}(s', a, u') \text{ and } uRu'$$
2. *if for all $a \in A_i$ and all $u \in S$ it holds $\mathbf{L}(s, a, u) \neq \top$, then for all $C \in S/R$:*
$$\mathbf{P}_l(s, C) = \mathbf{P}_l(s', C) \text{ and } \mathbf{P}_u(s, C) = \mathbf{P}_u(s', C)$$

We write $s \approx s'$ if sRs' for some bisimulation R on $\mathcal{M}$ and we write $\mathcal{M} \approx \mathcal{M}'$ for IMCs $\mathcal{M}$ and $\mathcal{M}'$ with initial states s_0 and s'_0, iff $s_0 \approx s'_0$ holds for the disjoint union[1] of $\mathcal{M}$ and $\mathcal{M}'$.

The first condition on may- and must-transitions is standard. The second condition asserts that for state s without outgoing internal must-transitions — which would have priority over Markovian transitions according to the maximal progress assumption— the probability to directly move to an equivalence class (under R) coincides with that of s'. The condition on probabilities is standard, whereas the exception of outgoing internal must-transition originates from IMCs [12, 14]. The main results of this section now follow:

Theorem 1 (Symmetric composition). *Let $\mathcal{M}$ be an AIMC, $\bar{A}$ a synchronization set and $n \in \mathbb{N}^+$, then:*

$$|||^n_{\bar{A}}\mathcal{M} \approx \overbrace{\mathcal{M}||_{\bar{A}} \cdots ||_{\bar{A}}\mathcal{M}}^{n \text{ times}}$$

Lemma 3. *Strong bisimulation $\approx$ is a congruence w.r.t. $||_{\bar{A}}$ and $|||_{\bar{A}}$.*

6 Abstraction

This section describes the process of abstracting (A)IMCs by partitioning the state space, i.e., by grouping sets of concrete states to abstract ones. For state space S and partitioning S' of S, let $\alpha : S \to S'$ map states to their corresponding abstract one, i.e., $\alpha(s)$ denotes the abstract state of s, and $\alpha^{-1}(s')$ is the set of concrete states that map to s'. Abstraction yields an AIMC that covers at least all possible behaviors of the concrete model, but perhaps more. The relationship between the abstraction and its concrete model is formalized by a *strong simulation.* We will define this notion and show that it is a precongruence with respect to parallel and symmetric composition. This result enables a *compositional* abstraction of AIMCs.

---

[1] Note that the union is only defined for two uniform AIMCs with the same exit rate as for different exit rates, the result is not uniform.

Definition 9 (Abstraction). *For an AIMC* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ *and partitioning* S' *of* S*, the abstraction function* $\alpha : S \to S'$ *induces the AIMC* $(S', A, \mathbf{L}', \mathbf{P}'_l, \mathbf{P}'_u, \lambda, \alpha(s_0))$*, denoted by* $\alpha(\mathcal{M})$*, where:*

- $\mathbf{L}'(s', a, u') = \begin{cases} \top & \textit{if } \bigsqcup_{u \in \alpha^{-1}(u')} \mathbf{L}(s, a, u) = \top \textit{ for all } s \in \alpha^{-1}(s') \\ \bot & \textit{if } \bigsqcup_{u \in \alpha^{-1}(u')} \mathbf{L}(s, a, u) = \bot \textit{ for all } s \in \alpha^{-1}(s') \\ ? & \textit{otherwise} \end{cases}$
- $\mathbf{P}'_l(s', u') = \min_{s \in \alpha^{-1}(s')} \sum_{u \in \alpha^{-1}(u')} \mathbf{P}_l(s, u)$
- $\mathbf{P}'_u(s', u') = \min(1, \max_{s \in \alpha^{-1}(s')} \sum_{u \in \alpha^{-1}(u')} \mathbf{P}_u(s, u))$

Lemma 4. *For any AIMC* $\mathcal{M}$*,* $\alpha(\mathcal{M})$ *is an AIMC.*

Example 7. Let $\mathcal{M}$ be the IMC in Fig. 2 (left) and $\mathcal{N}$ be the AIMC in Fig. 2 (middle). Then, $\mathcal{N} = \alpha(\mathcal{M})$ with $\alpha(s_i) = u_i$ for $i \in \{0, \ldots, 5\}$ and $\alpha(s'_i) = u_i$ for $i \in \{1, 2\}$. Consider a worker $\mathcal{M}'$ that is a variant of the one in Fig. 2 (left), say, whose judgement on the quality of raw material is different, i.e. whose $\mathbf{P}(s_0, s_1)$ and $\mathbf{P}(s_0, s'_1)$ differ. For such a worker, we also get $\mathcal{N} = \alpha(\mathcal{M}')$. Symmetric composition of two different workers $\mathcal{M}$ and $\mathcal{M}'$ is not possible. However, replacing both $\mathcal{M}$ and $\mathcal{M}'$ by abstract worker $\mathcal{N}$ enables symmetric composition and yields a compact representation of an abstraction of $\mathcal{M}||_{\bar{A}}\mathcal{M}'$.

The formal relationship between an AIMC and its abstraction is defined in terms of a strong simulation. In fact, the notion defined below combines the concepts of refinement for modal transition systems [25] (items 1a and 1b) with that of probabilistic simulation [19, 20] (item 2).

Definition 10 (Strong simulation). *For AIMC* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$*,* $R \subseteq S \times S$ *is a* simulation relation*, iff for all* sRs' *the following holds:*

1a. *for all* $a \in A$ *and* $u \in S$ *with* $\mathbf{L}(s, a, u) \neq \bot$ *there exists* $u' \in S$ *with* $\mathbf{L}(s', a, u') \neq \bot$ *and* uRu'*,*
1b. *for all* $a \in A$ *and* $u' \in S$ *with* $\mathbf{L}(s', a, u') = \top$ *there exists* $u \in S$ *with* $\mathbf{L}(s, a, u) = \top$ *and* uRu'*, and*
2. *if for all* $a \in A_i$ *and all* $u \in S$ *it holds* $\mathbf{L}(s, a, u) \neq \top$*, then for all* $\mu \in \mathbf{T}(s)$ *there exists* $\mu' \in \mathbf{T}(s')$ *and* $\Delta : S \times S \to [0, 1]$ *such that for all* $u, u' \in S$*:*
   (a) $\Delta(u, u') > 0 \implies uRu'$ *(b)* $\Delta(u, S) = \mu(u)$ *(c)* $\Delta(S, u') = \mu'(u')$

We write $s \preceq s'$ if sRs' for some simulation R and $\mathcal{M} \preceq \mathcal{M}'$ for AIMCs $\mathcal{M}$ and $\mathcal{M}'$ with initial states s_0 and s'_0, if $s_0 \preceq s'_0$ in the disjoint union of $\mathcal{M}$ and $\mathcal{M}'$.

Let us briefly explain this definition. Item 1a requires that any may- or must-transition of s must be reflected in s'. Item 1b requires that any must-transition of s' must match some must-transition of s, i.e., all required behavior of s' stems from s. Note that this allows a must-transition of s to be mimicked by a may-transition of s'. Finally, condition 2 requires the existence of a weight function Δ [19, 20] that basically distributes μ of s to μ' of s' such that only related states obtain a positive weight (2(a)), and the total probability mass of u that is assigned by Δ coincides with $\mu(u)$ and symmetrically for u' (cf. 2(b), 2(c)). Note that every bisimulation equivalence R is also a simulation relation.

Theorem 2. *For any AIMC* $\mathcal{M}$ *and abstraction function* α*,* $\mathcal{M} \preceq \alpha(\mathcal{M})$*.*

Example 8. Consider AIMCs $\mathcal{M}$ and $\mathcal{N}$ given in Example 7. As $\mathcal{N}$ is an abstraction of $\mathcal{M}$, it follows $\mathcal{M} \preceq \mathcal{N}$.

To be able to compose abstractions while preserving this formal relation, the following result is of interest. It allows to abstract parallel and symmetric compositions of AIMCs in a component-wise manner, to avoid the need for generating the entire state space prior to abstraction.

Theorem 3. *Strong simulation* $\preceq$ *is a precongruence w.r.t.* $||_{\bar{A}}$ *and* $|||_{\bar{A}}$.

7 Timed Reachability

In this section, we show how to analyse closed AIMCs by reducing them to uniform IMCs. As presented in [18], those can be reduced to uniform continuous-time Markov decision processes (CTMDP) for which an efficient algorithm is implemented in MRMC, a state of the art model checker. We analyse two reachability objectives for the running example and show how abstraction and symmetric composition reduce the maximal size of the state space during the construction of the model.

To obtain the induced IMC for an AIMC, we separate the nondeterministic choice for values from the intervals in Markovian states from the actual Markovian behavior, i.e. the delay and the subsequent probabilistic transitions. This is achieved by adding one intermediate state for each extreme distribution.

Definition 11 (Induced IMC). *For closed AIMC* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$, *let* $\theta(\mathcal{M}) = (S \mathbin{\dot\cup} S^{extr}, A^{extr}, \mathbf{L}', \mathbf{P}', \lambda, s_0)$ *where*

- $S^{extr} = \{s_\mu \mid \exists s \in S : \mu \in extr(\mathbf{P}_l, \mathbf{P}_u, S, s)\}$
- $\mathbf{L}'(s, a, s') = \begin{cases} \mathbf{L}(s, a, s') & \text{if } s \in S_H \cup S_{MH},\ a = \tau_{s'} \\ \top & \text{if } s \in S_M \cup S_{MH},\ a = \tau_\mu,\ s' = s_\mu \\ & \text{and } \mu \in extr(\mathbf{P}_l, \mathbf{P}_u, S, s) \\ \bot & \text{otherwise} \end{cases}$
- $\mathbf{P}'(s, s') = \begin{cases} \mu(s') & \text{if } s = s_\mu \in S^{extr} \\ \mathbf{1}(s, s') & \text{otherwise} \end{cases}$

Lemma 5. *For a closed AIMC* $\mathcal{M}$ *it holds that* $\theta(\mathcal{M})$ *is a closed uniform IMC.*

Example 9. Let $\mathcal{M}$ be the symmetric composition of two independent abstract workers as depicted in Fig. 2 (middle). We focus on state $\{\!| s_0, s_2 |\!\}$ in $\mathcal{M}$, cf. Fig. 6 (left). In the corresponding induced IMC $\theta(\mathcal{M})$, there are new states s_μ and $s_{\mu'}$ with outgoing Markovian transitions according to the extreme distributions μ and μ' of $\{\!| s_0, s_2 |\!\}$ with $\mu(\{\!| s_1, s_2 |\!\}) = \frac{1}{2}$, $\mu(\{\!| s_0, s_2 |\!\}) = \frac{1}{6}$, $\mu(\{\!| s_0, s_4 |\!\}) = \frac{2}{6}$ and $\mu'(\{\!| s_1, s_2 |\!\}) = \frac{1}{2}$, $\mu'(\{\!| s_0, s_2 |\!\}) = \frac{1}{8}$, $\mu'(\{\!| s_0, s_4 |\!\}) = \frac{3}{8}$. Additionally, labeled transitions with internal actions τ_μ ($\tau_{\mu'}$ resp.) leading from $\{\!| s_0, s_2 |\!\}$ to the new intermediate states are introduced.

For closed AIMC $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$, we define the set of paths starting in initial state s_0 and visiting a state in $B \subseteq S$ within $t \in \mathbb{R}_{\geq 0}$ time units by $Paths^{\mathcal{M}}(\Diamond^{\leq t} B) = \{\sigma \in Paths_{\mathcal{M}} \mid \sigma[0] = s_0, \exists t' \in [0, t] : \sigma@t' \in B\}$.

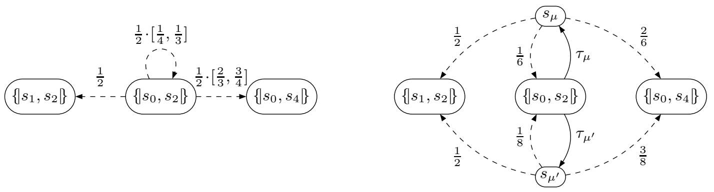

Fig. 6. Fragment of the parallel composition $\mathcal{M}||_{\{\}}\mathcal{M}$ for the AIMC $\mathcal{M}$ from Fig. 2 (left) and the induced IMC detail (right).

Lemma 6. *Let* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ *be a closed AIMC and* $\theta(\mathcal{M})$ *its induced IMC. For all* $B \subseteq S$, $t \in \mathbb{R}_{\geq 0}$ *and* $D \in \mathcal{D}(\mathcal{M})$ *there exists* $D' \in \mathcal{D}(\theta(\mathcal{M}))$ *with* $Pr^{\omega}_{s_0,D}(Paths^{\mathcal{M}}(\Diamond^{\leq t}B)) = Pr^{\omega}_{s_0,D'}(Paths^{\mathcal{M}}(\Diamond^{\leq t}B))$.

For interactive transitions, a corresponding scheduler in the induced IMC chooses exactly as the AIMC scheduler. The choice of a distribution in the AIMC is mimicked by a randomized choice of τ_μ actions (cf. Fig. 6). From this, we obtain:

Theorem 4. *For a closed AIMC* $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$, $B \subseteq S$, $t \in \mathbb{R}_{\geq 0}$*:*

$$\sup_{D\in\mathcal{D}(\mathcal{M})} Pr^{\omega}_{s_0,D}(Paths^{\mathcal{M}}(\Diamond^{\leq t}B)) = \sup_{D\in\mathcal{D}(\theta(\mathcal{M}))} Pr^{\omega}_{s_0,D}(Paths^{\theta(\mathcal{M})}(\Diamond^{\leq t}B))$$
$$\inf_{D\in\mathcal{D}(\mathcal{M})} Pr^{\omega}_{s_0,D}(Paths^{\mathcal{M}}(\Diamond^{\leq t}B)) = \inf_{D\in\mathcal{D}(\theta(\mathcal{M}))} Pr^{\omega}_{s_0,D}(Paths^{\theta(\mathcal{M})}(\Diamond^{\leq t}B))$$

The analysis of time-bounded reachability probabilities for uniform IMCs is investigated in [18] and the core algorithm [1] is implemented in MRMC. Basically, a uniform IMC is reduced to a uniform CTMDP by transformations to so-called *Markov alternating* and *strictly alternating* IMCs. This transformation preserves (weak) bisimulation. The following example relies on this results:

Example 10. Assume the number of machines that are available for crafting value and premium products is limited to two. First, we investigate the probabilities for b out of w workers $\mathcal{M}_1$ to $\mathcal{M}_w$ to be waiting for machines within t time units. Let $\mathcal{P} = (\{m_0, m_1, m_2\}, A, \mathbf{L}, \mathbf{1}, \mathbf{1}, \varepsilon, m_0)$ where in m_i there are i machines in use and let $A = \{\text{value}, \text{prem}, \text{vdone}, \text{pdone}\}$, $\mathbf{L}(m_i, a, m_{i+1}) = \top$ if $a \in \{\text{value}, \text{prem}\}$ for $i \in \{0, 1\}$ and $\mathbf{L}(m_{i+1}, a, m_i) = \top$ if $a \in \{\text{vdone}, \text{pdone}\}$ for $i \in \{0, 1\}$, otherwise $\mathbf{L}(m, a, m') = \bot$. Let $\mathcal{M}_i$ be pairwise distinct variants of workers as described in Ex. 7. Then, $(\mathcal{M}_1||_\emptyset \cdots ||_\emptyset\mathcal{M}_w)||_A\mathcal{P}$ yields an IMC where the measure of interest can be derived by computing probabilities for reaching states $(\bar{s}, m_2)$ with at least b components of $\bar{s}$ being s_1 or s_1'. In contrast, when $\mathcal{M}_1 = \ldots = \mathcal{M}_w = \mathcal{M}$ we can instead compute the probabilities in $\left(|||^w_\emptyset\mathcal{M}\right)||_A\mathcal{P}$ for reaching states (M, m_2) with $M(s_1) + M(s_1') \geq b$. The maximal sizes of the state spaces obtained during the construction of the models are given in Table 1 (left). Let AIMC $\mathcal{N} = \alpha(\mathcal{M}_1) = \ldots = \alpha(\mathcal{M}_w)$ as described in Ex. 7. Then, even for pairwise distinct workers, symmetric composition can be used to obtain

max. size	w=3, b=1	w=4, b=1	w=4, b=2	w=1	w=2	w=3	w=4
IMC, par.	512	4096	4096	352	2816	22528	180224
IMC, sym.	120	330	330	352	1584	5280	14520
AIMC, par.	216	1296	1296	264	1584	9504	57024
AIMC, sym.	56	126	126	264	924	2464	5544

Table 1. Maximal size of the state spaces during construction.

the abstract system $(|||_{\emptyset}^{w}\mathcal{N})||_{A}\mathcal{P}$. While the abstract model of one worker has 6 instead of 8 states, the relative savings during composition are much larger (cf. Table 1). But still, the minimal and maximal probabilities (Fig. 7, left) obtained for w instances of the abstract worker $\mathcal{N}$ (dashed curves) are almost the same as for w copies of the concrete worker $\mathcal{M}$ as shown in Fig. 2 (left) (solid curves).

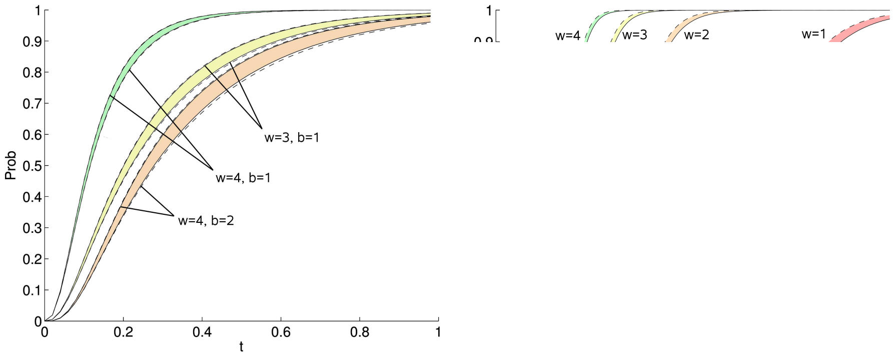

Fig. 7. Minimal and maximal probabilities for b out of w workers having no access to one of 2 machines in t time units (left). Maximal probabilities for w workers and 2 machines to produce 10 value and 3 premium in t time units (right). Curves for concrete workers are solid and dashed for abstract ones.

Secondly, we compute the maximal probabilities for producing 10 value and 3 premium products with w workers within t time units. Note, that minimal probabilities are 0 for all time bounds t, as workers may stall premium production. We define counting AIMC $\mathcal{Q} = (\{n_{v,p} \mid v \in \{0,\ldots,10\}, p \in \{0,\ldots,3\}\}, A, \mathbf{L}, \mathbf{1}, \mathbf{1}, \varepsilon, n_{0,0})$ with $A = \{\text{vdone}, \text{pdone}\}$, $\mathbf{L}(n_{v,p}, \text{vdone}, n_{v+1,p}) = \top$ for $v \in \{0,\ldots,9\}$, $p \in \{0,\ldots,3\}$ and $\mathbf{L}(n_{v,p}, \text{pdone}, n_{v,p+1}) = \top$ for $v \in \{0,\ldots,10\}, p \in \{0,\ldots,2\}$, otherwise $\mathbf{L}(n,a,n') = \bot$. Let concrete and abstract workers $\mathcal{M}$ and $\mathcal{N}$ be given as in Fig. 2. Then, in $\left(|||_{\emptyset}^{w}\mathcal{M}\right) ||_{A}\mathcal{Q}$ and $\left(|||_{\emptyset}^{w}\mathcal{N}\right) ||_{A}\mathcal{Q}$, we compute probabilities to reach any state $(M, n_{10,3})$. As shown in Fig. 7 (right), the maximal probabilities for $w \in \{1,\ldots,4\}$ abstract workers (dashed curves) are rather close to values derived for concrete workers (solid curves). The maximal sizes of the state spaces during construction are given in Table 1 (right).

8 Conclusion

This paper proposed a novel compositional abstraction technique for continuous-time stochastic systems. This technique allows for aggressive abstractions of single components, enabling the analysis of systems that are too large to be handled when treated as monolithic models. The feasibility of our approach has been demonstrated by the analysis of a production system. Future work includes the application of this technique to realistic applications, counterexample-guided abstraction refinement [16, 22], and the treatment of non-uniform IMCs.

References

1. Baier, C., Hermanns, H., Katoen, J.-P., Haverkort, B. R.: Efficient computation of time-bounded reachability probabilities in uniform continuous-time Markov decision processes. *Theor. Comput. Sci.* **345** (2005) 2–26

2. Berendsen, J., Vaandrager, F. W.: Compositional abstraction in real-time model checking. In: *FORMATS. LNCS*, Vol. 5215. (2008) 233–249
3. Bode, E., Herbstritt, M., Hermanns, H., Johr, S., Peikenkamp, T., Pulungan, R., Wimmer, R., Becker, B.: Compositional performability evaluation for statemate. In: *QEST.* IEEE Computer Society (2006) 167–178
4. Cardelli, L.: On process rate semantics. *Theor. Comput. Sci.* **391** (2008) 190–215
5. Chadha, R., Viswanathan, M., Viswanathan, R.: Least upper bounds for probability measures and their applications to abstractions. In: *CONCUR. LNCS*, Vol. 5201. (2008) 264–278
6. de Alfaro, L., Henzinger, T. A., Jhala, R.: Compositional methods for probabilistic systems. In: *CONCUR. LNCS*, Vol. 2154. (2001) 351–365
7. de Alfaro, L., Roy, P.: Magnifying-lens abstraction for Markov decision processes. In: *CAV. LNCS*, Vol. 4590. (2007) 325–338
8. Fecher, H., Leucker, M., Wolf, V.: Don't know in probabilistic systems. In: *Model Checking Software. LNCS*, Vol. 3925. (2006) 71–88
9. Garavel, H., Hermanns, H.: On combining functional verification and performance evaluation using CADP. In: *FME. LNCS*, Vol. 2391. (2002) 410–429
10. Gilmore, S., Hillston, J., Ribaudo, M.: An efficient algorithm for aggregating PEPA models. *IEEE Trans. Software Eng.* **27** (2001) 449–464
11. Hermanns, H., Johr, S.: Uniformity by construction in the analysis of nondeterministic stochastic systems. *Dependable Systems and Networks* (2007) 718–728
12. Hermanns, H.: *Interactive Markov Chains and the Quest for Quantified Quality. LNCS*, Vol. 2428, Berlin (2002)
13. Hermanns, H., Herzog, U., Katoen, J.-P.: Process algebra for performance evaluation. *Theor. Comput. Sci.* **274** (2002) 43–87
14. Hermanns, H., Katoen, J.-P.: Automated compositional Markov chain generation for a plain-old telephone system. *Sci. Comput. Program.* **36** (2000) 97–127
15. Hermanns, H., Ribaudo, M.: Exploiting symmetries in stochastic process algebras. In: *European Simulation Multiconference.* SCS Europe (1998) 763–770
16. Hermanns, H., Wachter, B., Zhang, L.: Probabilistic CEGAR. In: *CAV. LNCS*, Vol. 5123. (2008) 162–175
17. Hillston, J.: *A Compositional Approach to Performance Modelling.* Cambridge University Press (1996)
18. Johr, S.: *Model Checking Compositional Markov Systems.* PhD thesis, Universität des Saarlandes, Saarbrücken, Germany (2007)
19. Jones, C., Plotkin, G.: A probabilistic powerdomain of evaluations. In: *LICS.* IEEE Computer Society (1989) 186–195
20. Jonsson, B., Larsen, K. G.: Specification and refinement of probabilistic processes. In: *LICS.* IEEE Computer Society (1991) 266–277
21. Katoen, J.-P., Klink, D., Leucker, M., Wolf, V.: Three-valued abstraction for continuous-time Markov chains. In: *CAV. LNCS*, Vol. 4590. (2007) 316–329
22. Kattenbelt, M., Kwiatkowska, M., Norman, G., Parker, D.: Abstraction refinement for probabilistic software. In: *VMCAI. LNCS*, Vol. 5403. (2009)
23. Kattenbelt, M., Kwiatkowska, M. Z., Norman, G., Parker, D.: Game-based probabilistic predicate abstraction in PRISM. *ENTCS* **220** (2008) 5–21
24. Kwiatkowska, M., Norman, G., Parker, D.: Game-based abstraction for Markov decision processes. In: *QEST.* IEEE Computer Society (2006) 157–166
25. Larsen, K. G., Thomsen, B.: A modal process logic. In: *LICS.* IEEE Computer Society (1988) 203–210
26. Larsen, K. G.: Modal specifications. In: *Automatic Verification Methods for Finite State Systems. LNCS*, Vol. 407. (1989) 232–246
27. MaciÃ , H., Valero, V., de Frutos-Escrig, D.: sPBC: A Markovian extension of finite Petri box calculus. *Petri Nets and Performance Models* (2001) 207–216
28. Segala, R., Lynch, N. A.: Probabilistic simulations for probabilistic processes. *Nord. J. Comput.* **2** (1995) 250–273
29. Shoham, S., Grumberg, O.: Compositional verification and 3-valued abstractions join forces. In: *SAS. LNCS*, Vol. 4634. (2007) 69–86
30. Shoham, S., Grumberg, O.: 3-valued abstraction: More precision at less cost. *Inf. Comput.* **206** (2008) 1313–1333
31. Tofts, C. M. N.: Compositional performance analysis. In: *TACAS. LNCS*, Vol. 1217. (1997) 290–305

Appendix

We provide proofs for Theorem 1 and 3. The proof of Theorem 2 follows the lines of the one for Theorem 1 in [21].

Theorem 1 (Symmetric composition). *Let $\mathcal{M}$ be an AIMC, $\bar{A}$ a synchronization set and $n \in \mathbb{N}^+$, then:*

$$|||^n_{\bar{A}}\mathcal{M} \approx \overbrace{\mathcal{M}||_{\bar{A}} \cdots ||_{\bar{A}}\mathcal{M}}^{n \text{ times}}$$

Proof. Let $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$ be an AIMC, $\bar{A} \subseteq A_e$ and $n \in \mathbb{N}^+$. Let

$$|||^n_{\bar{A}}\mathcal{M} = \mathcal{M}' = (S', A', \mathbf{L}', \mathbf{P}'_l, \mathbf{P}'_u, \lambda', s'_0)$$

and

$$\underbrace{\mathcal{M}||_{\bar{A}} \cdots ||_{\bar{A}}\mathcal{M}}_{n \text{ times}} = \tilde{\mathcal{M}}' = (\tilde{S}', \tilde{A}', \tilde{\mathbf{L}}', \tilde{\mathbf{P}}'_l, \tilde{\mathbf{P}}'_u, \tilde{\lambda}', \tilde{s}'_0).$$

As $\lambda' = n\lambda$ and $\tilde{\lambda}' = \lambda + \ldots + \lambda = n\lambda$, the disjoint union $\mathcal{M}^\cup = \mathcal{M}' \cup \tilde{\mathcal{M}}'$ is an AIMC with a set of initial states. Lifting AIMCs to support sets of initial states is trivial and will not be discussed in the following.

We define $R_n \subseteq (S' \cup \tilde{S}') \times (S' \cup \tilde{S}')$ on $\mathcal{M}^\cup$ as the coarsest reflexive and symmetric relation with

$$s' R_n \tilde{s}' \text{ if } s' = \{s_1, \ldots, s_n\} \text{ and } \tilde{s}' = (s_1, \ldots, s_n) \text{ for } s_1, \ldots, s_n \in S.$$

Then, as we show in the following, (a) R_n is a bisimulation relation and (b) the initial states are R_n related, i.e. $s'_0 R_n \tilde{s}'_0$.

a)

We prove that R_n is a strong bisimulation relation: Therefore, assume $s' R_n \tilde{s}'$ for $s' = \{s_1, \ldots, s_n\}$ and $\tilde{s}' = (s_1, \ldots, s_n)$ with $s_1, \ldots, s_n \in S$.

1. If $\mathbf{L}^\cup(s', a, u') = x$ with $x \in \{\top, ?\}$, then we consider two cases:
   - Assume $a \notin \bar{A}$: As $\mathbf{L}^\cup(s', a, u') = x$, $(*)$ there exist $s \in s'$, $u \in u'$ such that $\mathbf{L}(s, a, u) = x$ and there are no $\hat{s} \in s'$, $\hat{u} \in u'$ such that $\mathbf{L}(\hat{s}, a, \hat{u}) > x$. By the definition of symmetric composition, there exists $k \in \{1, \ldots, n\}$ such that

     $$\begin{aligned} s' &= \{s_1, \ldots, s_{k-1}, s, s_{k+1}, \ldots, s_n\} \text{ and} \\ u' &= \{s_1, \ldots, s_{k-1}, u, s_{k+1}, \ldots, s_n\}. \end{aligned}$$

     As $s' R_n \tilde{s}'$, there exists a permutation $\pi \in \mathit{Perm}(\{1, \ldots, n\})$ such that

     $$\tilde{s}' = \left(s_{\pi(1)}, \ldots, s_{\pi(j-1)}, s_{\pi(j)}, s_{\pi(j+1)}, \ldots, s_{\pi(n)}\right).$$

     Then $\pi(j) = k$ for some j and $s_{\pi(j)} = s_k = s$. Applying π to u' yields

     $$\tilde{u}' = \left(s_{\pi(1)}, \ldots, s_{\pi(j-1)}, u, s_{\pi(j+1)}, \ldots, s_{\pi(n)}\right).$$

     From $(*)$ we obtain $\mathbf{L}^\cup(\tilde{s}', a, \tilde{u}') = x$. Further, from the definition of relation R_n, we conclude $\tilde{s}' R_n \tilde{u}'$.

- Assume $a \in \bar{A}$: We first consider $x = \top$. For u' with

  $$\mathbf{L}'(s', a, u') = \bigsqcup\nolimits_{\bar{R} \in \Gamma_{s',u'}} \bigsqcap\nolimits_{s,u \in S^{\cup} : \bar{R}(s,u) > 0} \mathbf{L}(s, a, u) = \top$$

  it holds

  $$\exists \bar{R} \in \Gamma_{s',u'} : \forall s, u \in S^{\cup} : (\bar{R}(s,u) > 0 \implies \mathbf{L}(s,a,u) = \top).$$

  As $s' R_n \tilde{s}'$ we have $s' = \{s_1, \ldots, s_n\}$ and $\tilde{s}' = (s_1, \ldots, s_n)$ for some $s_1, \ldots, s_n \in S$. We will define $\tilde{u}'$ inductively based on some $\bar{R} \in \Gamma_{s',u'}$ with $\forall s, u \in S^{\cup} : (\bar{R}(s,u) > 0 \implies \mathbf{L}(s,a,u) = \top)$: let $\bar{R}_n = \bar{R}$ and for $i \in \{1, \ldots, n\}$, let

  $$\bar{R}_{i-1} = \bar{R}_i[(s_i, u_i) \mapsto \bar{R}_i(s_i, u_i) - 1] \text{ for some } u_i : \bar{R}_i(s_i, u_i) > 0.$$

  Then for all $i \in \{1, \ldots, n\} : \mathbf{L}(s_i, a, u_i) = \top$ as $\bar{R}(s_i, u_i) > 0$. For $\tilde{u}' = (u_1, \ldots, u_n)$, this implies $\tilde{\mathbf{L}}'(\tilde{s}', a, \tilde{u}') = \bigsqcap_{i=1}^{n} \mathbf{L}(s_i, a, u_i) = \top$. Moreover, as by the definition of $\Gamma_{s',u'}$ it holds $\bar{R}(S', u) = u'(u)$ for all $u \in S$, it follows that $u' = \{u_1, \ldots, u_n\}$, i.e. $u' R_n \tilde{u}'$.
  For $x \neq \bot$, it can be argued analogously that for u' with $\mathbf{L}'(s', a, u') \neq \bot$ there exists $\tilde{u}'$ with $u' R_n \tilde{u}'$ and $\tilde{\mathbf{L}}'(\tilde{s}', a, \tilde{u}') \neq \bot$. Together with the result from $x = \top$ it follows that for $x \in \{?, \top\}$ and u' with $\mathbf{L}'(s', a, u') = x$ there exists $\tilde{u}'$ with $u' R_n \tilde{u}'$ and $\tilde{\mathbf{L}}'(\tilde{s}', a, \tilde{u}') = x = \mathbf{L}'(s', a, u')$.

2. For Markovian transitions we show for $s' = \{s_1, \ldots, s_n\}$ and $\tilde{s}' = (s_1, \ldots, s_n)$ with $s_1, \ldots, s_n \in S$ that for all $C \in S^{\cup}/R_n$:

   $$\mathbf{P}_l^{\cup}(s', C) = \mathbf{P}_l^{\cup}(\tilde{s}', C) \text{ and } \mathbf{P}_u^{\cup}(s', C) = \mathbf{P}_u^{\cup}(\tilde{s}', C).$$

   Note that the condition "$\mathbf{L}^{\cup}(s', a, u') \neq \top$ for all $a \in A_i$ and $u' \in S'$" from the definition of bisimulation is not required for this proof. If $C = \emptyset$, then $\mathbf{P}_l^{\cup}(s', C) = \mathbf{P}_l^{\cup}(\tilde{s}', C) = 0$; hence, in the following we assume $C \neq \emptyset$.
   First, we lift the definition of parallel composition to the n-ary case for the lower bound probability matrix:

   $$\begin{aligned} &\tilde{\mathbf{P}}_l'((s_1, \ldots, s_n), (u_1, \ldots, u_n)) \\ &= \textstyle\sum_{i=1}^{n} \frac{1}{n} \cdot \mathbf{P}_l(s_i, u_i) \cdot \prod_{j \neq i} \mathbf{1}(s_j, u_j) \\ &= \begin{cases} \frac{1}{n} \cdot \mathbf{P}_l(s_i, u_i) & \text{if } \exists i.\ s_i \neq u_i \ \wedge\ \forall j \neq i.\ s_j = u_j \\ \sum_{i=1}^{n} \frac{1}{n} \cdot \mathbf{P}_l(s_i, u_i) & \text{if } \forall j.\ s_j = u_j \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

   Next, we observe for $S^{\cup}/R_n$ that:
   (a) every nonempty $C \in S^{\cup}/R_n$ contains exactly one $s' \in S'$
   (b) $\{s_1, \ldots, s_n\} \in C \in S^{\cup}/R_n$
   $\iff$ for all $\pi \in \mathit{Perm}(\{1, \ldots, n\}) : (s_{\pi(1)}, \ldots, s_{\pi(n)}) \in C$

   From (a) it follows directly that for $s' \in S'$ and nonempty $C \in S^{\cup}/R_n$:

   $$\mathbf{P}_l^{\cup}(s', C) = \mathbf{P}_l'(s', C \cap S') = \mathbf{P}_l'(s', u') \text{ with } \{u'\} = C \cap S'$$

Let $\tilde{s}' = (s_1, \dots, s_n)$, $s' = \{s_1, \dots, s_n\}$, $s = s_i$ and $u' \neq s' \backslash \{s\} \uplus \{u\}$ for all $u \in S$. Then $\mathbf{P}^{\cup}_l(s', [u']_{R_n}) = \mathbf{P}^{\cup}_l(\tilde{s}', [u']_{R_n}) = 0$. Otherwise, i.e. $u' = s' \setminus \{s\} \uplus \{u\}$ for some $u \in S$, we set $C = [u']_{R_n}$ and obtain

$$\begin{aligned}
\mathbf{P}^{\cup}_l(\tilde{s}', C) &= \tilde{\mathbf{P}}'_l(\tilde{s}', C \cap \tilde{S}') \\
&= \textstyle\sum_{\tilde{u}' \in C \cap \tilde{S}'} \tilde{\mathbf{P}}'_l(\tilde{s}', \tilde{u}') \\
&= \begin{cases} \sum_{\tilde{u}' \in C \cap \tilde{S}' : \tilde{u}' = (s_1, \dots, s_{k-1}, u, s_{k+1}, s_n)} \frac{1}{n} \cdot \mathbf{P}_l(s, u) & \text{if } s \neq u \\ \sum_{i=1}^{n} \frac{1}{n} \cdot \mathbf{P}_l(s_i, u_i) & \text{if } s = u \end{cases} \\
&\overset{(*)}{=} \begin{cases} \frac{s'(s)}{n} \cdot \mathbf{P}_l(s, u) & \text{if } s' \neq u' \\ \sum_{i=1}^{n} \frac{1}{n} \cdot \mathbf{P}_l(s_i, u_i) & \text{if } s' = u' \end{cases} \\
&= \mathbf{P}'_l(s', u') = \mathbf{P}^{\cup}_l(s', C)
\end{aligned}$$

where at $(*)$ we use the fact that there are $s'(s)$ positions k in $\tilde{s}'$ where s can be replaced by u such that $\tilde{u}' = (s_1, \dots, s_{k-1}, u, s_{k+1}, \dots, s_n)$.
For the upper bound probability matrix, the proof can be done analogously. Altogether, for all $n \in \mathbb{N}^+$, we showed that R_n as defined earlier is a bisimulation relation.

b)

Now, we show that the initial states are strongly bisimilar. By definition:

$$s'_0 = \{\!|s_0, \dots, s_0|\!\} \qquad \text{and} \qquad \tilde{s}'_0 = (s_0, \dots, s_0)$$

For all $s_1, \dots, s_n \in S_0$, it holds $\{s_1, \dots, s_n\} R_n (s_1, \dots, s_n)$ and thus, obviously,

$$\{\!|s_0, \dots, s_0|\!\} R_n (s_0, \dots, s_0).$$

We conclude by observing that from (a) and (b) it follows $|||^n_{\bar{A}} \mathcal{M} \approx \underbrace{\mathcal{M} ||_{\bar{A}} \dots ||_{\bar{A}} \mathcal{M}}_{n \text{ times}}$.

□

Theorem 3. *Strong simulation $\preceq$ is a precongruence w.r.t.* $||_{\bar{A}}$ *and* $|||_{\bar{A}}$.

Proof. Reflexivity of $\preceq$ follows trivially from the definition. Let $\mathcal{M} = (S, A, \mathbf{L}, \mathbf{P}_l, \mathbf{P}_u, \lambda, s_0)$, $\mathcal{N} = (S', A', \mathbf{L}', \mathbf{P}'_l, \mathbf{P}'_u, \lambda', s'_0)$ and $\mathcal{P} = (S'', A'', \mathbf{L}'', \mathbf{P}''_l, \mathbf{P}''_u, \lambda'', s''_0)$. To argue about simulation of states in different models we have to analyse the disjoint union. For simplicity, in this proof we refrain from explicitly composing the disjoint unions, however, we stress the necessity for all AIMCs involved in a union to have the same exit rates. This will be ensured in the following, as for $\mathcal{M} \preceq \mathcal{N}$ (and $\mathcal{N} \preceq \mathcal{P}$) it follows that $\lambda = \lambda'$ (and $\lambda' = \lambda''$ respectively).

Transitivity: Let $R : S \times S'$ and $R' : S' \times S''$ be simulation relations with $s_0 R s'_0$ and $s'_0 R' s''_0$ respectively. We define $R'' : S \times S''$ with

$$R'' = \{(s, s'') \mid \exists s' \in S : (s, s') \in R, (s', s'') \in R'\}$$

and prove that it is a simulation relation (note that $R'' \supseteq R \cup R'$ due to reflexivity of R and R').

We show that conditions (1a), (1b) and (2) of Def. 10 are fulfilled: for all $s \in S$, $s'' \in S''$ for which there exists $s' \in S'$ with sRs' and $s'R's''$ it holds

1a. By the definition of simulation it holds that

$$\forall a \in A\ \forall u \in S : \mathbf{L}(s, a, u) \neq \bot \implies \exists u' \in S : \mathbf{L}'(s', a, u') \neq \bot \wedge uRu'$$

and

$$\forall a \in A\ \forall u' \in S : \mathbf{L}'(s', a, u') \neq \bot \implies \exists u'' \in S : \mathbf{L}''(s'', a, u'') \neq \bot \wedge u'R'u''.$$

Thus, it follows directly:

$$\forall a \in A\ \forall u \in S : \mathbf{L}''(s'', a, u'') \neq \bot \implies \exists u'' \in S : \mathbf{L}(s, a, u) \neq \bot \wedge uR''u''.$$

1b. As for (1a), by the definition of simulation it holds that

$$\forall a \in A\ \forall u'' \in S : \mathbf{L}''(s'', a, u'') = \top \implies \exists u' \in S : \mathbf{L}'(s', a, u') = \top \wedge u'R'u''$$

and

$$\forall a \in A\ \forall u' \in S : \mathbf{L}'(s', a, u') = \top \implies \exists u \in S : \mathbf{L}(s, a, u) = \top \wedge uRu'.$$

Thus, it follows directly:

$$\forall a \in A\ \forall u'' \in S : \mathbf{L}''(s'', a, u'') = \top \implies \exists u \in S : \mathbf{L}(s, a, u) = \top \wedge uR''u''.$$

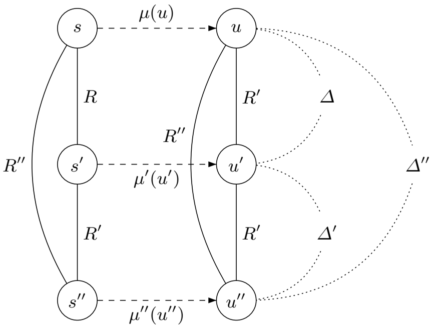

Fig. 8. Transitivity

2. We show that if for all $u \in S$ and all $a \in A_i$ it holds $\mathbf{L}(s, a, u) \neq \top$, then for all $\mu \in \mathbf{T}(s)$ there exists $\mu'' \in \mathbf{T}(s'')$ and $\Delta'' : S \times S'' \to [0, 1]$ such that for all $u \in S$ and $u'' \in S''$: (cf. Fig. 8)
   (a) $\Delta''(u, u'') > 0 \implies uR''u''$
   (b) $\Delta''(u, S'') = \mu(u)$
   (c) $\Delta''(S, u'') = \mu''(u'')$

   First, note that if for all $u \in S$ it holds $\mathbf{L}(s, a, u) \neq \top$ for all $a \in A_i$, then for all $\mu \in \mathbf{T}(s)$ there exists $\mu' \in \mathbf{T}(s')$ and $\Delta : S \times S' \to [0, 1]$ such that for all $u \in S$ and $u' \in S'$:
   (a) $\Delta(u, u') > 0 \implies uRu'$
   (b) $\Delta(u, S') = \mu(u)$
   (c) $\Delta(S, u') = \mu'(u')$

   Second, we observe that for any $s \in S$ and $u \in S$ with $\mathbf{L}(s, a, u) \neq \top$ for some $a \in A_i$, all $s' \in S$ with sRs' may not have a successor $u' \in S$ with $\mathbf{L}(s', a, u') = \top$. Otherwise, if there was some $u' \in S$ with $\mathbf{L}(s', a, u') = \top$, due to condition (1b) in Def. 10 there would exist $u \in S$ with $\mathbf{L}(s, a, u) = \top$ and uRu', leading to a contradiction.
   Hence, for all $s' \in S'$ with sRs', $u' \in S'$ and $a \in A_i$ it holds $\mathbf{L}(s', a, u') \neq \top$ and, as R' is a simulation relation, for all $\mu' \in \mathbf{T}(s')$ there exists $\mu'' \in \mathbf{T}(s'')$ and $\Delta' : S' \times S'' \to [0, 1]$ such that for all $u' \in S'$ and $u'' \in S''$:
   (a) $\Delta'(u', u'') > 0 \implies u'R'u''$
   (b) $\Delta'(u', S'') = \mu'(u')$
   (c) $\Delta'(S', u'') = \mu''(u'')$

   We define $\Delta'' : S \times S'' \to [0, 1]$ such that

   $$\Delta''(u, u'') = \textstyle\sum_{u' \in S' : \mu'(u') > 0} \frac{\Delta(u,u') \cdot \Delta'(u',u'')}{\mu'(u')}$$

   for Δ, Δ' and μ' satisfying the above constraints. For condition (2a), we observe that if $\Delta''(u, u'') > 0$ there exists u' such that $\Delta(u, u') > 0$ and $\Delta'(u', u'')$. Thus, uRu' and $u'R'u''$ implying $uR''u''$.

Further, we show conditions (2b) and (2c) by proving that for all $\mu \in \mathbf{T}(s)$ there exists $\mu'' \in \mathbf{T}(s'')$, such that $\Delta''(u, S'') = \mu(u)$ and $\Delta''(S, u'') = \mu''(u'')$ for all $u \in S$ and $u'' \in S''$:

$$\begin{aligned} \Delta''(u, S'') &= \textstyle\sum_{u' \in S', u'' \in S'' : \mu'(u') > 0} \frac{\Delta(u,u') \cdot \Delta'(u',u'')}{\mu'(u')} \\ &= \textstyle\sum_{u' \in S' : \Delta'(u',S'') > 0} \frac{\Delta(u,u') \cdot \Delta'(u',S'')}{\Delta'(u',S'')} \\ &= \textstyle\sum_{u' \in S' : \Delta'(u',S'') > 0} \Delta(u, u') \\ &\overset{(*)}{=} \textstyle\sum_{u' \in S' : \Delta(S,u') > 0} \Delta(u, u') \\ &= \Delta(u, S') \\ &= \mu(u) \end{aligned}$$

Equation $(*)$ follows from $\Delta'(u', S'') = \mu'(u') = \Delta(S, u')$ for all $u' \in S'$.

$$\begin{aligned} \Delta''(S, u'') &= \textstyle\sum_{u \in S, u' \in S' : \mu'(u') > 0} \frac{\Delta(u,u') \cdot \Delta'(u',u'')}{\mu'(u')} \\ &= \textstyle\sum_{u' \in S' : \Delta(S,u') > 0} \frac{\Delta(S,u') \cdot \Delta'(u',u'')}{\Delta(S,u')} \\ &= \textstyle\sum_{u' \in S' : \Delta(S,u') > 0} \Delta'(u', u'') \\ &\overset{(*)}{=} \textstyle\sum_{u' \in S' : \Delta'(u',S'') > 0} \Delta'(u', u'') \\ &= \Delta'(S', u'') \\ &= \mu''(u'') \end{aligned}$$

This concludes the proof of transitivity.

Now we show that parallel composition does not destroy strong simulation relations. Let $\mathcal{M}||_{\bar{A}}\mathcal{P} = (S \times S'', A \cup A'', \tilde{\mathbf{L}}, \tilde{\mathbf{P}}_l, \tilde{\mathbf{P}}_u, \lambda + \lambda'', (s_0, s_0''))$ and $\mathcal{N}||_{\bar{A}}\mathcal{P} = (S' \times S'', A' \cup A'', \tilde{\mathbf{L}}', \tilde{\mathbf{P}}_l', \tilde{\mathbf{P}}_u', \lambda' + \lambda'', (s_0', s_0''))$. Recall that from $\mathcal{M} \preceq \mathcal{N}$ it follows $\lambda = \lambda'$ and therefore the union of $\mathcal{M}||_{\bar{A}}\mathcal{P}$ and $\mathcal{N}||_{\bar{A}}\mathcal{P}$ that will implicitly be used is valid.

We show that $\mathcal{M} \preceq \mathcal{N}$ implies $\mathcal{M}||_{\bar{A}}\mathcal{P} \preceq \mathcal{N}||_{\bar{A}}\mathcal{P}$ for synchronization set $\bar{A}$, i.e. that for initial state (s_0, s_0'') there exists (s_0', s_0'') with $(s_0, s_0'') \preceq (s_0', s_0'')$. Let $\tilde{R} : (S \times S'') \times (S' \times S'')$ such that $(s, s'')\tilde{R}(s', s''')$ iff $s \preceq s'$ and $s'' \preceq s'''$. From $\mathcal{M} \preceq \mathcal{N}$ we know that $s_0 \preceq s_0'$ and due to reflexivity, $s_0'' \preceq s_0''$. Thus, $(s_0, s_0'')\tilde{R}(s_0', s_0'')$.

In the following, we show that for all $(s, s'') \in S \times S''$ and $(s', s''') \in S' \times S''$ with sRs' and $s''R's'''$ for simulation relations R and R', conditions (1a), (1b) and (2) in Def. 10 are fulfilled, i.e. that $\tilde{R}$ is a simulation relation.

1a. This can be shown in a similar fashion as (1b).

1b. For $a \in \bar{A}$ we compute:

$$\begin{aligned}
&\forall (u', u''')\, \exists (u, u'') : \tilde{\mathbf{L}}'((s', s'''), a, (u', u''')) = \top \\
&\qquad\qquad \implies \tilde{\mathbf{L}}((s, s''), a, (u, u'')) = \top \wedge (u, u'')\tilde{R}(u', u''') \\
\iff& \\
&\forall (u', u''')\, \exists (u, u'') : \mathbf{L}'(s', a, u') \sqcap \mathbf{L}''(s''', a, u''') = \top \\
&\qquad\qquad \implies \mathbf{L}(s, a, u) \sqcap \mathbf{L}''(s'', a, u'') = \top \wedge (u, u'')\tilde{R}(u', u''') \\
\iff& \\
&\forall (u', u''')\, \exists (u, u'') : \mathbf{L}'(s', a, u') = \top \wedge \mathbf{L}''(s''', a, u''') = \top \\
&\qquad\qquad \implies \mathbf{L}(s, a, u) = \top \wedge \mathbf{L}''(s'', a, u'') = \top \wedge (u, u'')\tilde{R}(u', u''')
\end{aligned}$$

This follows directly from sRs' and $s''R's'''$ as

$$\begin{matrix}
\forall u'\, \exists u : \mathbf{L}'(s', a, u') = \top \implies \mathbf{L}(s, a, u) = \top \wedge uRu' & \text{and} \\
\forall u'''\, \exists u'' : \mathbf{L}''(s''', a, u''') = \top \implies \mathbf{L}''(s'', a, u'') = \top \wedge u''Ru''' &
\end{matrix}.$$

For $a \notin \bar{A}$ we compute:

$$\begin{aligned}
&\forall (u', u''')\, \exists (u, u'') : \tilde{\mathbf{L}}'((s', s'''), a, (u', u''')) = \top \\
&\qquad\qquad \implies \tilde{\mathbf{L}}((s, s''), a, (u, u'')) = \top \wedge (u, u'')\tilde{R}(u', u''') \\
\iff& \\
&\forall (u', u''')\, \exists (u, u'') : \\
&\qquad\qquad (\mathbf{L}'(s', a, u') \sqcap \mathbf{I}(s''', u''')) \sqcup (\mathbf{L}''(s''', a, u''') \sqcap \mathbf{I}(s', u')) = \top \\
&\qquad\qquad \implies (\mathbf{L}(s, a, u) \sqcap \mathbf{I}(s'', u'')) \sqcup (\mathbf{L}''(s'', a, u'') \sqcap \mathbf{I}(s, u)) = \top \\
&\qquad\qquad\qquad \wedge (u, u'')\tilde{R}(u', u''')
\end{aligned}$$

We investigate the two cases where on the left side of the implication either $(\mathbf{L}'(s', a, u') \sqcap \mathbf{I}(s''', u''')) = \top$ or $(\mathbf{L}''(s''', a, u''') \sqcap \mathbf{I}(s', u')) = \top$.
If $(\mathbf{L}'(s', a, u') \sqcap \mathbf{I}(s''', u'''))$ resolves to $\top$, so does $\mathbf{L}(s, a, u)$ for some $u \in S$. We choose $u'' = s''$, such that $\mathbf{I}(s'', u'') = \top$. For the right side of the implication to be fulfilled, it remains to show that $(u, u'')\tilde{R}(u', u''')$. From the satisfaction of $\mathbf{I}(s''', u''')$ it follows $u''' = s'''$ and together with $u'' = s''$, from $s''R's'''$ we directly get $u''R'u'''$. Further, sRs' implies

$$\forall u'\, \exists u : \mathbf{L}'(s', a, u') = \top \implies \mathbf{L}(s, a, u) = \top \wedge uRu'.$$

Thus, there exist $u \in S$ and $u'' \in S''$ fulfilling the right side of the implication. For the case where $(\mathbf{L}(s''', a, u''') \sqcap \mathbf{I}(s', u'))$ resolves to $\top$, the proof goes along the same lines.

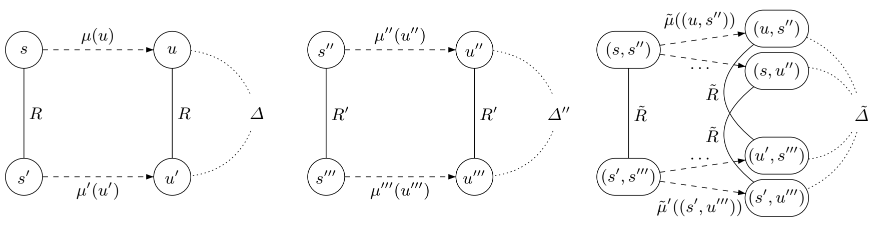

Fig. 9. Compatibility with parallel composition

2. First, note that from sRs' it follows: if for all $u \in S$ it holds $\mathbf{L}(s, a, u) \neq \top$ for all $a \in A_i$, then for all $\mu \in \mathbf{T}(s)$ there exists $\mu' \in \mathbf{T}(s')$ and $\Delta : S \times S' \to [0, 1]$ such that for all $u \in S$ and $u' \in S'$:
   (a) $\Delta(u, u') > 0 \implies uRu'$
   (b) $\Delta(u, S') = \mu(u)$
   (c) $\Delta(S, u') = \mu'(u')$

   Second, from $s''R's'''$ it follows: if for all $u'' \in S''$ it holds $\mathbf{L}(s'', a, u'') \neq \top$ for all $a \in A_i$, then for all $\mu'' \in \mathbf{T}(s'')$ there exists $\mu''' \in \mathbf{T}(s''')$ and $\Delta'' : S'' \times S'' \to [0, 1]$ such that for all $u'' \in S''$ and $u''' \in S''$:
   (a) $\Delta''(u'', u''') > 0 \implies u''R'u'''$
   (b) $\Delta''(u'', S'') = \mu''(u'')$
   (c) $\Delta''(S'', u''') = \mu'''(u''')$

   We show that, if for all $(u, u'') \in S \times S''$ it holds $\tilde{\mathbf{L}}((s, s''), a, (u, u'')) \neq \top$ for all $a \in A_i \cup A'_i$, then for all $\tilde{\mu} \in \mathbf{T}((s, s''))$ there exists $\tilde{\mu}' \in \mathbf{T}((s', s'''))$ and $\tilde{\Delta} : (S \times S'') \times (S' \times S'') \to [0, 1]$ such that for all $(u, u'') \in S \times S''$ and $(u', u''') \in S' \times S''$: (cf. Fig. 9)
   (a) $\tilde{\Delta}((u, u''), (u', u''')) > 0 \implies (u, u'')\tilde{R}(u', u''')$
   (b) $\tilde{\Delta}((u, u''), S' \times S'') = \tilde{\mu}((u, u''))$
   (c) $\tilde{\Delta}(S \times S'', (u', u''')) = \tilde{\mu}'((u', u'''))$

   Given $(s, s'') \in S \times S''$ and $(s', s''') \in S' \times S''$, we define $\tilde{\Delta} : (S \times S'') \times (S' \times S'') \to [0, 1]$ such that:

$$\begin{aligned} \tilde{\Delta}((u, u''), (u', u''')) = & \tfrac{\lambda}{\lambda + \lambda''} \cdot \Delta(u, u') \cdot \mathbf{1}(s'', u'') \cdot \mathbf{1}(s''', u''') \\ & + \tfrac{\lambda''}{\lambda + \lambda''} \cdot \Delta''(u'', u''') \cdot \mathbf{1}(s, u) \cdot \mathbf{1}(s', u') \end{aligned}$$

   Condition (a) follows from this definition as

$$\begin{aligned} \tilde{\Delta}((u, u''), (u', u''')) > 0 \implies & (\Delta(u, u') > 0 \wedge s'' = u'' \wedge s''' = u''') \\ & \vee (\Delta''(u'', u''') > 0 \wedge s = u \wedge s' = u') \\ \implies & (uRu' \wedge s'' = u'' \wedge s''' = u''') \\ & \vee (u''R'u''' \wedge u = u' \wedge s = u \wedge s' = u') \\ \implies & (u, u'')\tilde{R}(u', u''') \end{aligned}$$

   where in the last implication we use the fact that sRs' and $s''R's'''$.

Regarding condition (b), for any $\tilde{\mu} \in \mathbf{T}((s, s''))$ we compute:

$$\begin{aligned}
\tilde{\mu}((u, u'')) &= \tfrac{\lambda}{\lambda+\lambda''} \cdot \mu(u) \cdot \mathbf{1}(s'', u'') + \tfrac{\lambda''}{\lambda+\lambda''} \cdot \mu''(u'') \cdot \mathbf{1}(s, u) \\
&= \textstyle\sum_{u' \in S'} \tfrac{\lambda}{\lambda+\lambda''} \cdot \varDelta(u, u') \cdot \mathbf{1}(s'', u'') \\
&\quad + \textstyle\sum_{u''' \in S''} \tfrac{\lambda''}{\lambda+\lambda''} \cdot \varDelta''(u'', u''') \cdot \mathbf{1}(s, u) \\
&= \textstyle\sum_{u''' \in S''} \Big(\sum_{u' \in S'} \tfrac{\lambda}{\lambda+\lambda''} \cdot \varDelta(u, u') \cdot \mathbf{1}(s'', u'') \cdot \mathbf{1}(s''', u''') \\
&\qquad\qquad + \textstyle\sum_{u' \in S'} \tfrac{\lambda''}{\lambda+\lambda''} \cdot \varDelta''(u'', u''') \cdot \mathbf{1}(s, u) \cdot \mathbf{1}(s', u') \Big) \\
&= \textstyle\sum_{u' \in S', u''' \in S''} \tfrac{\lambda}{\lambda+\lambda''} \cdot \varDelta(u, u') \cdot \mathbf{1}(s'', u'') \cdot \mathbf{1}(s''', u''') \\
&\qquad\qquad + \tfrac{\lambda''}{\lambda+\lambda''} \cdot \varDelta''(u'', u''') \cdot \mathbf{1}(s, u) \cdot \mathbf{1}(s', u') \\
&= \textstyle\sum_{u' \in S', u''' \in S''} \tilde{\varDelta}((u, u''), (u', u''')) \\
&= \tilde{\varDelta}((u, u''), S' \times S'')
\end{aligned}$$

Analogously, for condition (c) we compute for any $\tilde{\mu}' \in \mathbf{T}((s', s'''))$:

$$\begin{aligned}
\tilde{\mu}'((u', u''')) &= \tfrac{\lambda}{\lambda+\lambda''} \cdot \mu'(u') \cdot \mathbf{1}(s''', u''') + \tfrac{\lambda''}{\lambda+\lambda''} \cdot \mu'''(u''') \cdot \mathbf{1}(s', u') \\
&= \textstyle\sum_{u \in S} \tfrac{\lambda}{\lambda+\lambda''} \cdot \varDelta(u, u') \cdot \mathbf{1}(s''', u''') \\
&\quad + \textstyle\sum_{u'' \in S''} \tfrac{\lambda''}{\lambda+\lambda''} \cdot \varDelta''(u'', u''') \cdot \mathbf{1}(s', u') \\
&= \textstyle\sum_{u'' \in S''} \Big(\sum_{u \in S} \tfrac{\lambda}{\lambda+\lambda''} \cdot \varDelta(u, u') \cdot \mathbf{1}(s'', u'') \cdot \mathbf{1}(s''', u''') \\
&\qquad\qquad + \textstyle\sum_{u \in S} \tfrac{\lambda''}{\lambda+\lambda''} \cdot \varDelta''(u'', u''') \cdot \mathbf{1}(s, u) \cdot \mathbf{1}(s', u') \Big) \\
&= \textstyle\sum_{u \in S, u'' \in S''} \tfrac{\lambda}{\lambda+\lambda''} \cdot \varDelta(u, u') \cdot \mathbf{1}(s'', u'') \cdot \mathbf{1}(s''', u''') \\
&\qquad\qquad + \tfrac{\lambda''}{\lambda+\lambda''} \cdot \varDelta''(u'', u''') \cdot \mathbf{1}(s, u) \cdot \mathbf{1}(s', u') \\
&= \textstyle\sum_{u \in S, u'' \in S''} \tilde{\varDelta}((u, u''), (u', u''')) \\
&= \tilde{\varDelta}(S \times S'', (u', u'''))
\end{aligned}$$

Thus, conditions (a) to (c) hold.

We conclude by observing that $\preceq$ is reflexive, transitive and compatible with parallel composition.

For parallel composition, note that parallel and symmetric composition yield bisimilar AIMCs. Hence, $\preceq$ is also a precongruence for symmetric composition.

□

Aachener Informatik-Berichte

This list contains all technical reports published during the past five years. A complete list of reports dating back to 1987 is available from http://aib.informatik.rwth-aachen.de/. To obtain copies consult the above URL or send your request to: Informatik-Bibliothek, RWTH Aachen, Ahornstr. 55, 52056 Aachen, Email: biblio@informatik.rwth-aachen.de

2004-01 * Fachgruppe Informatik: Jahresbericht 2003

2004-02 Benedikt Bollig, Martin Leucker: Message-Passing Automata are expressively equivalent to EMSO logic

2004-03 Delia Kesner, Femke van Raamsdonk, Joe Wells (eds.): HOR 2004 – 2nd International Workshop on Higher-Order Rewriting

2004-04 Slim Abdennadher, Christophe Ringeissen (eds.): RULE 04 – Fifth International Workshop on Rule-Based Programming

2004-05 Herbert Kuchen (ed.): WFLP 04 – 13th International Workshop on Functional and (Constraint) Logic Programming

2004-06 Sergio Antoy, Yoshihito Toyama (eds.): WRS 04 – 4th International Workshop on Reduction Strategies in Rewriting and Programming

2004-07 Michael Codish, Aart Middeldorp (eds.): WST 04 – 7th International Workshop on Termination

2004-08 Klaus Indermark, Thomas Noll: Algebraic Correctness Proofs for Compiling Recursive Function Definitions with Strictness Information

2004-09 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Parameterized Power Domination Complexity

2004-10 Zinaida Benenson, Felix C. Gärtner, Dogan Kesdogan: Secure Multi-Party Computation with Security Modules

2005-01 * Fachgruppe Informatik: Jahresbericht 2004

2005-02 Maximillian Dornseif, Felix C. Gärtner, Thorsten Holz, Martin Mink: An Offensive Approach to Teaching Information Security: “Aachen Summer School Applied IT Security”

2005-03 Jürgen Giesl, Rene Thiemann, Peter Schneider-Kamp: Proving and Disproving Termination of Higher-Order Functions

2005-04 Daniel Mölle, Stefan Richter, Peter Rossmanith: A Faster Algorithm for the Steiner Tree Problem

2005-05 Fabien Pouget, Thorsten Holz: A Pointillist Approach for Comparing Honeypots

2005-06 Simon Fischer, Berthold Vöcking: Adaptive Routing with Stale Information

2005-07 Felix C. Freiling, Thorsten Holz, Georg Wicherski: Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks

2005-08 Joachim Kneis, Peter Rossmanith: A New Satisfiability Algorithm With Applications To Max-Cut

2005-09 Klaus Kursawe, Felix C. Freiling: Byzantine Fault Tolerance on General Hybrid Adversary Structures

2005-10 Benedikt Bollig: Automata and Logics for Message Sequence Charts

2005-11 Simon Fischer, Berthold Vöcking: A Counterexample to the Fully Mixed Nash Equilibrium Conjecture

2005-12 Neeraj Mittal, Felix Freiling, S. Venkatesan, Lucia Draque Penso: Efficient Reductions for Wait-Free Termination Detection in Faulty Distributed Systems

2005-13 Carole Delporte-Gallet, Hugues Fauconnier, Felix C. Freiling: Revisiting Failure Detection and Consensus in Omission Failure Environments

2005-14 Felix C. Freiling, Sukumar Ghosh: Code Stabilization

2005-15 Uwe Naumann: The Complexity of Derivative Computation

2005-16 Uwe Naumann: Syntax-Directed Derivative Code (Part I: Tangent-Linear Code)

2005-17 Uwe Naumann: Syntax-directed Derivative Code (Part II: Intraprocedural Adjoint Code)

2005-18 Thomas von der Maßen, Klaus Möller, John MacGregor, Eva Geisberger, Jörg Dörr, Frank Houdek, Harbhajan Singh, Holger Wußmann, Hans-Veit Bacher, Barbara Paech: Einsatz von Features im Software-Entwicklungsprozess - Abschlußbericht des GI-Arbeitskreises “Features”

2005-19 Uwe Naumann, Andre Vehreschild: Tangent-Linear Code by Augmented LL-Parsers

2005-20 Felix C. Freiling, Martin Mink: Bericht über den Workshop zur Ausbildung im Bereich IT-Sicherheit Hochschulausbildung, berufliche Weiterbildung, Zertifizierung von Ausbildungsangeboten am 11. und 12. August 2005 in Köln organisiert von RWTH Aachen in Kooperation mit BITKOM, BSI, DLR und Gesellschaft fuer Informatik (GI) e.V.

2005-21 Thomas Noll, Stefan Rieger: Optimization of Straight-Line Code Revisited

2005-22 Felix Freiling, Maurice Herlihy, Lucia Draque Penso: Optimal Randomized Fair Exchange with Secret Shared Coins

2005-23 Heiner Ackermann, Alantha Newman, Heiko Röglin, Berthold Vöcking: Decision Making Based on Approximate and Smoothed Pareto Curves

2005-24 Alexander Becher, Zinaida Benenson, Maximillian Dornseif: Tampering with Motes: Real-World Physical Attacks on Wireless Sensor Networks

2006-01 * Fachgruppe Informatik: Jahresbericht 2005

2006-02 Michael Weber: Parallel Algorithms for Verification of Large Systems

2006-03 Michael Maier, Uwe Naumann: Intraprocedural Adjoint Code Generated by the Differentiation-Enabled NAGWare Fortran Compiler

2006-04 Ebadollah Varnik, Uwe Naumann, Andrew Lyons: Toward Low Static Memory Jacobian Accumulation

2006-05 Uwe Naumann, Jean Utke, Patrick Heimbach, Chris Hill, Derya Ozyurt, Carl Wunsch, Mike Fagan, Nathan Tallent, Michelle Strout: Adjoint Code by Source Transformation with OpenAD/F

2006-06 Joachim Kneis, Daniel Mölle, Stefan Richter, Peter Rossmanith: Divide-and-Color

2006-07 Thomas Colcombet, Christof Löding: Transforming structures by set interpretations

2006-08 Uwe Naumann, Yuxiao Hu: Optimal Vertex Elimination in Single-Expression-Use Graphs

2006-09 Tingting Han, Joost-Pieter Katoen: Counterexamples in Probabilistic Model Checking

2006-10 Mesut Günes, Alexander Zimmermann, Martin Wenig, Jan Ritzerfeld, Ulrich Meis: From Simulations to Testbeds - Architecture of the Hybrid MCG-Mesh Testbed

2006-11 Bastian Schlich, Michael Rohrbach, Michael Weber, Stefan Kowalewski: Model Checking Software for Microcontrollers

2006-12 Benedikt Bollig, Joost-Pieter Katoen, Carsten Kern, Martin Leucker: Replaying Play in and Play out: Synthesis of Design Models from Scenarios by Learning

2006-13 Wong Karianto, Christof Löding: Unranked Tree Automata with Sibling Equalities and Disequalities

2006-14 Danilo Beuche, Andreas Birk, Heinrich Dreier, Andreas Fleischmann, Heidi Galle, Gerald Heller, Dirk Janzen, Isabel John, Ramin Tavakoli Kolagari, Thomas von der Maßen, Andreas Wolfram: Report of the GI Work Group "Requirements Management Tools for Product Line Engineering"

2006-15 Sebastian Ullrich, Jakob T. Valvoda, Torsten Kuhlen: Utilizing optical sensors from mice for new input devices

2006-16 Rafael Ballagas, Jan Borchers: Selexels: a Conceptual Framework for Pointing Devices with Low Expressiveness

2006-17 Eric Lee, Henning Kiel, Jan Borchers: Scrolling Through Time: Improving Interfaces for Searching and Navigating Continuous Audio Timelines

2007-01 * Fachgruppe Informatik: Jahresbericht 2006

2007-02 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, Rene Thiemann, and Harald Zankl: SAT Solving for Termination Analysis with Polynomial Interpretations

2007-03 Jürgen Giesl, Rene Thiemann, Stephan Swiderski, and Peter Schneider-Kamp: Proving Termination by Bounded Increase

2007-04 Jan Buchholz, Eric Lee, Jonathan Klein, and Jan Borchers: coJIVE: A System to Support Collaborative Jazz Improvisation

2007-05 Uwe Naumann: On Optimal DAG Reversal

2007-06 Joost-Pieter Katoen, Thomas Noll, and Stefan Rieger: Verifying Concurrent List-Manipulating Programs by LTL Model Checking

2007-07 Alexander Nyßen, Horst Lichter: MeDUSA - MethoD for UML2-based Design of Embedded Software Applications

2007-08 Falk Salewski and Stefan Kowalewski: Achieving Highly Reliable Embedded Software: An empirical evaluation of different approaches

2007-09 Tina Kraußer, Heiko Mantel, and Henning Sudbrock: A Probabilistic Justification of the Combining Calculus under the Uniform Scheduler Assumption

2007-10 Martin Neuhäußer, Joost-Pieter Katoen: Bisimulation and Logical Preservation for Continuous-Time Markov Decision Processes

2007-11 Klaus Wehrle (editor): 6. Fachgespräch Sensornetzwerke

2007-12 Uwe Naumann: An L-Attributed Grammar for Adjoint Code

2007-13 Uwe Naumann, Michael Maier, Jan Riehme, and Bruce Christianson: Second-Order Adjoints by Source Code Manipulation of Numerical Programs

2007-14 Jean Utke, Uwe Naumann, Mike Fagan, Nathan Tallent, Michelle Strout, Patrick Heimbach, Chris Hill, and Carl Wunsch: OpenAD/F: A Modular, Open-Source Tool for Automatic Differentiation of Fortran Codes

2007-15 Volker Stolz: Temporal assertions for sequential and concurrent programs

2007-16 Sadeq Ali Makram, Mesut Günec, Martin Wenig, Alexander Zimmermann: Adaptive Channel Assignment to Support QoS and Load Balancing for Wireless Mesh Networks

2007-17 Rene Thiemann: The DP Framework for Proving Termination of Term Rewriting

2007-18 Uwe Naumann: Call Tree Reversal is NP-Complete

2007-19 Jan Riehme, Andrea Walther, Jörg Stiller, Uwe Naumann: Adjoints for Time-Dependent Optimal Control

2007-20 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf: Three-Valued Abstraction for Probabilistic Systems

2007-21 Tingting Han, Joost-Pieter Katoen, and Alexandru Mereacre: Compositional Modeling and Minimization of Time-Inhomogeneous Markov Chains

2007-22 Heiner Ackermann, Paul W. Goldberg, Vahab S. Mirrokni, Heiko Röglin, and Berthold Vöcking: Uncoordinated Two-Sided Markets

2008-01 * Fachgruppe Informatik: Jahresbericht 2007

2008-02 Henrik Bohnenkamp, Marielle Stoelinga: Quantitative Testing

2008-03 Carsten Fuhs, Jürgen Giesl, Aart Middeldorp, Peter Schneider-Kamp, Rene Thiemann, Harald Zankl: Maximal Termination

2008-04 Uwe Naumann, Jan Riehme: Sensitivity Analysis in Sisyphe with the AD-Enabled NAGWare Fortran Compiler

2008-05 Frank G. Radmacher: An Automata Theoretic Approach to the Theory of Rational Tree Relations

2008-06 Uwe Naumann, Laurent Hascoet, Chris Hill, Paul Hovland, Jan Riehme, Jean Utke: A Framework for Proving Correctness of Adjoint Message Passing Programs

2008-07 Alexander Nyßen, Horst Lichter: The MeDUSA Reference Manual, Second Edition

2008-08 George B. Mertzios, Stavros D. Nikolopoulos: The λ-cluster Problem on Parameterized Interval Graphs

2008-09 George B. Mertzios, Walter Unger: An optimal algorithm for the k-fixed-endpoint path cover on proper interval graphs

2008-10 George B. Mertzios, Walter Unger: Preemptive Scheduling of Equal-Length Jobs in Polynomial Time

2008-11 George B. Mertzios: Fast Convergence of Routing Games with Splittable Flows

2008-12 Joost-Pieter Katoen, Daniel Klink, Martin Leucker, Verena Wolf: Abstraction for stochastic systems by Erlang's method of stages

2008-13 Beatriz Alarcon, Fabian Emmes, Carsten Fuhs, Jürgen Giesl, Raul Gutierrez, Salvador Lucas, Peter Schneider-Kamp, Rene Thiemann: Improving Context-Sensitive Dependency Pairs

2008-14 Bastian Schlich: Model Checking of Software for Microcontrollers

2008-15 Joachim Kneis, Alexander Langer, Peter Rossmanith: A New Algorithm for Finding Trees with Many Leaves

2008-16 Hendrik vom Lehn, Elias Weingärtner and Klaus Wehrle: Comparing recent network simulators: A performance evaluation study

2008-17 Peter Schneider-Kamp: Static Termination Analysis for Prolog using Term Rewriting and SAT Solving

2008-18 Falk Salewski: Empirical Evaluations of Safety-Critical Embedded Systems

2009-03 Alexander Nyßen: Model-Based Construction of Embedded

Real-Time Software - A Methodology for Small Devices

2009-04 Daniel Klünder: Entwurf eingebetteter Software mit abstrakten Zustandsmaschinen und Business Object Notation

2009-05 George B. Mertzios, Ignasi Sau, Shmuel Zaks: A New Intersection Model and Improved Algorithms for Tolerance Graphs

2009-06 George B. Mertzios, Ignasi Sau, Shmuel Zaks: The Recognition of Tolerance and Bounded Tolerance Graphs is NP-complete

2009-07 Joachim Kneis, Alexander Langer, Peter Rossmanith: Derandomizing Non-uniform Color-Coding I

2009-08 Joachim Kneis, Alexander Langer: Satellites and Mirrors for Solving Independent Set on Sparse Graphs

2009-09 Michael Nett: Implementation of an Automated Proof for an Algorithm Solving the Maximum Independent Set Problem

2009-10	Felix Reidl, Fernando Sanchez Villaamil: Automatic Verification of the Correctness of the Upper Bound of a Maximum Independent Set Algorithm
2009-11	Kyriaki Ioannidou, George B. Mertzios, Stavros D. Nikolopoulos: The Longest Path Problem is Polynomial on Interval Graphs
2009-12	Martin Neuhäußer, Lijun Zhang: Time-Bounded Reachability in Continuous-Time Markov Decision Processes
2009-13	Martin Zimmermann: Time-optimal Winning Strategies for Poset Games
2009-14	Ralf Huuck, Gerwin Klein, Bastian Schlich (eds.): Doctoral Symposium on Systems Software Verification (DS SSV'09)

[*] These reports are only available as a printed version.

Please contact biblio@informatik.rwth-aachen.de to obtain copies.