

Backward Bisimulation in Markov Chain Model Checking

Jeremy Sproston and Susanna Donatelli, *Member, IEEE Computer Society*

Abstract—Equivalence relations can be used to reduce the state space of a system model, thereby permitting more efficient analysis. We study backward stochastic bisimulation in the context of model checking continuous-time Markov chains against Continuous Stochastic Logic (CSL) properties. While there are simple CSL properties that are not preserved when reducing the state space of a continuous-time Markov chain using backward stochastic bisimulation, we show that the equivalence can nevertheless be used in the verification of a practically significant class of CSL properties. We consider an extension of these results to Markov reward models and Continuous Stochastic Reward Logic. Furthermore, we identify the logical properties for which the requirement on the equality of state-labeling sets (normally imposed on state equivalences in a model-checking context) can be omitted from the definition of the equivalence, resulting in a better state-space reduction.

Index Terms—Markov processes, model checking, temporal logic, verification.

1 INTRODUCTION

MODEL checking [19] is a verification method which automatically establishes whether a formal description of the system satisfies a desired property, based on a (partial or exhaustive) exploration of the state space. Examples of properties which are verified by model checking are absence of deadlock or reachability of goal states. Typically, the system is described in terms of high-level languages such as Petri nets or process algebras, the underlying semantics of which take the form of labeled transition systems, whereas the property to be verified is specified in terms of a propositional or temporal logic, such as CTL [18] or LTL [35]. A number of model-checking tools have been developed, such as SMV [17] and SPIN [29], and have been used successfully to verify properties of hardware and software systems.

Systems which exhibit nontrivial probabilistic or stochastic behavior, such as randomized algorithms or fault-tolerant systems, are more appropriately modeled using formalisms such as stochastic Petri nets [1] or stochastic process algebra [26], [24], [6], the underlying semantics of which take the form of Markov chains. In particular, model checking algorithms for continuous-time Markov chains (CTMCs) have been developed, where the property to be verified is described in terms of the extension of the temporal logic CTL called Continuous Stochastic Logic (CSL) [3], [9]. The logic CSL provides a formal way to describe potentially complex properties which refer to the performance of a stochastic system. It includes a probabilistic operator, which can refer to the probability with which a certain property is satisfied, and a steady-state operator,

which can refer to the probabilities of the system being in certain states in equilibrium. Model-checking tools such as PRISM [31] and E⁺MC² [25] have been used to analyze the CSL properties of stochastic systems in application areas such as fault-tolerant systems, manufacturing systems, and biological processes. Extending CTMCs with rewards results in Markov reward models, which permit the computation of classical performance measures such as throughput and the mean number of clients in a system's queue. Accordingly, CSL has been extended to obtain the logic Continuous Stochastic Reward Logic (CSRL), the formulae of which are model checked on Markov reward models [7].

The practical applicability of model checking, whether in the traditional or stochastic setting, is often limited by the excessive size of the system's state space. One method to reduce the size of the state space is to use a notion of state equivalence in order to combine suitably equivalent states into a single superstate. In the context of Markov chains, *lumpability* [33] is a well-established notion of state equivalence. Given a state equivalence, the resulting, reduced state space, called a *quotient* (for example, the lumped state space in the case of Markov chains), can then be used for analysis. In this paper, we study the use of equivalence relations to reduce the state space of CTMCs. Forward bisimulation of stochastic systems, which is related to the notion of ordinary (or strong) lumpability of Markov chains, has been widely studied [33], [13], [26], [24], [6], [10]. Instead, our focus is on *backward stochastic bisimulation*, which is related to exact lumpability on Markov chains [36], [12], [14]. In contrast to forward bisimulation on stochastic systems, which is defined according to conditions on the *outgoing* transitions of a state, backward stochastic bisimulation is defined according to the *incoming* transitions of a state and the state's total exit rate. An advantage of backward stochastic bisimulation over its forward counterpart is that, given an appropriate side condition on the initial distribution of the CTMC, each

• The authors are with the Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy. E-mail: {sproston, susi}@di.unito.it.

Manuscript received 7 Oct. 2005; revised 22 Dec. 2005; accepted 12 Jan. 2006; published online 7 Sept. 2006.

Recommended for acceptance by G. Franceschinis, J.-P. Katoen, and M. Woodside.

For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number TSE-0271-1005.

state within each equivalence class has an equal probability, both for the transient and steady-state distributions. This permits us to obtain the transient and steady-state probabilities of the states of the CTMC by computing the corresponding probabilities on the backward stochastic bisimulation quotient.

We study backward stochastic bisimulation in the context of model checking properties of the temporal logic CSL. A useful property that an equivalence for reducing the state space of a system model can have in this context is that all states within a class of the equivalence either satisfy a temporal logic formula or they do not satisfy it. In such a case, it suffices to reason about equivalence classes rather than states, as classes can be used to identify exactly the state sets which satisfy a formula. On the one hand, there are CSL properties that are not preserved in this way when reducing the state space of a CTMC using backward stochastic bisimulation: For example, probabilities of reaching state sets in the future may differ in backward stochastic bisimilar states. We show in this paper that, on the other hand, backward stochastic bisimulation can be used in the verification of a practically significant class of CSL properties, namely, those formulae without nesting of probabilistic or steady-state operators, given an initial distribution of the CTMC.

We consider an extension of these results to Markov reward models and CSRL. Furthermore, in the context of qualitative properties, we extend the results of Buchholz [14] to show that backward stochastic bisimilar states exhibit the same *infinite traces* and, therefore, satisfy the same properties of the linear-time temporal logic LTL. This result makes use of the approach of Lynch and Vaandrager [34] for forward and backward simulation relations of untimed, nonstochastic transition systems. Finally, we also show how forward and backward stochastic bisimulation may be used together in the context of a class of CSL properties in which a steady-state operator is nested within a probabilistic operator.

The forward (or backward) stochastic bisimulation quotient of a CTMC with n states and m transitions can be computed using the $O(m \log n)$ algorithm defined by Derisavi et al. [22]. This quotient is essentially the strongly (or exactly) lumped CTMC.

In Section 2, we recall the definition of CTMCs and, in Section 3, both forward and backward stochastic bisimulation are defined. We consider the qualitative properties of backward stochastic bisimulation in Section 4, whereas, in Section 5 and Section 6, the use of backward stochastic bisimulation in the verification of subclasses of CSL and CSRL properties, respectively, is explored. In Section 7, we study how the subclass of CSL considered in Section 5 may be extended by applying backward stochastic bisimulation to certain parts of the state space of a CTMC and forward stochastic bisimulation elsewhere. In Section 8, we compare the reductions obtained from forward and backward stochastic bisimulation when applied to two case studies. Finally, in Section 9, we conclude the paper. A conference version of this paper appeared as [37]; we extend that version with a proof of Theorem 5.1 and Sections 6 and 8.

2 CONTINUOUS-TIME MARKOV CHAINS

We consider continuous-time Markov chains with a labeling condition which associates with every state a set of atomic propositions which are valid in that state. Let AP be a set of atomic propositions, which will be fixed throughout the remainder of the paper, and let $\mathbb{R}_{\geq 0}$ be the set of nonnegative reals. A *probability distribution* on a set of elements S is a function $\alpha : S \rightarrow [0, 1]$ such that $\sum_{s \in S} \alpha(s) = 1$. We use α_s to denote the probability distribution with probability 1 in the single element s .

Definition 2.1. A continuous-time Markov chain (CTMC) is a quadruple (S, R, p, L) comprised of a finite set S of states, a rate transition function $R : S \times S \rightarrow \mathbb{R}_{\geq 0}$, an initial probability distribution p on S , and a labeling function $L : S \rightarrow 2^{AP}$.

The interpretation of the rate transition function is that $R(s, s') > 0$ if and only if there exists a transition from state s to state s' and that the probability that this transition is triggered within t time units is $1 - e^{-R(s, s')t}$ (that is, the duration of a transition from s to s' is governed by an exponential distribution with rate $R(s, s')$). As in [9], we model self-looping transitions, corresponding to $R(s, s) > 0$, in an explicit manner, in order to retain the standard interpretation of temporal logic formulae. In our context, we may distinguish as different two CTMCs that have the same infinitesimal generator (and the same transient and steady-state distributions), but have different self-looping behavior.

Let the *exit rate* $E(s)$ for the state $s \in S$ be defined by $E(s) = \sum_{s' \in S} R(s, s')$. A state s is called *absorbing* if and only if $E(s) = 0$. If $R(s, s') > 0$ and $t \in \mathbb{R}_{\geq 0}$, then we say that there exists a transition of duration t from state s to state s' (where t is chosen according to the exponential distribution with rate $R(s, s')$). Such a transition is denoted by $s \xrightarrow{t} s'$. An infinite path is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots$ of transitions. A finite path is a sequence $s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots s_{n-1} \xrightarrow{t_{n-1}} s_n$ of transitions such that s_n is absorbing. Let $Path^C$ be the set of paths of C . Let α be a probability distribution on the set S of states. The probability measure given by α using the standard cylinder set construction is denoted by $Prob_\alpha^C$ [9]: Then, for a set of paths $\Omega \subseteq Path^C$, the probability of C exhibiting the paths in Ω after commencing from the starting distribution α is $Prob_\alpha^C(\Omega)$. Often, we let the starting distribution be the initial distribution of the CTMC (that is, $\alpha = p$) or let the starting distribution assign probability 1 to a single state; in the latter case, we write $Prob_s^C$ rather than $Prob_{\alpha_s}^C$. For any infinite path $\omega = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots$ and any $i \in \mathbb{N}$, let $\omega(i) = s_i$, the $(i + 1)$ st state of ω , let $\delta(\omega, i) = t_i$, and, for $t \in \mathbb{R}_{\geq 0}$ and i , the smallest index such that $t \leq \sum_{j=0}^i t_j$, let $\omega @ t = \omega(i)$, the state occupied at time t . For any finite path $\omega = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots s_l$, the state $\omega(i)$ and duration $\delta(\omega, i) = t_i$ are only defined if $i < l$ and are defined as in the infinite-path case. We also let $\delta(\omega, l) = \infty$. Furthermore, for $t > \sum_{j=0}^{l-1} t_j$, let $\omega @ t = s_l$; otherwise, $\omega @ t$ is defined as in the infinite-path case.

A *transient probability* is the probability of being in a certain state s at time t given an initial distribution α . In the model-checking context, we can express a transient probability in terms of paths as $\pi^C(\alpha, s, t) = \text{Prob}_\alpha^C\{\omega \in \text{Path}^C \mid \omega @ t = s\}$. The *steady-state probabilities* are used to refer to the long-run average probability of the CTMC being in a state, and are defined by $\pi^C(\alpha, s) = \lim_{t \rightarrow \infty} \pi^C(\alpha, s, t)$. If the CTMC is ergodic, the steady-state distribution does not depend on α , and we write $\pi^C(s)$ rather than $\pi^C(\alpha, s)$. For $S' \subseteq S$, let $\pi^C(\alpha, S', t) = \sum_{s \in S'} \pi^C(\alpha, s, t)$ and let $\pi^C(\alpha, S') = \sum_{s \in S'} \pi^C(\alpha, s)$.

Next, we consider an extension of CTMCs with real-valued rewards (or costs) in each state [30], [7].

Definition 2.2. A Markov reward model (MRM) is a quintuple (S, R, p, L, ρ) comprising a CTMC (S, R, p, L) and a reward structure $\rho : S \rightarrow \mathbb{R}_{\geq 0}$ which associates a real-valued reward (or cost) with each state.

The finite and infinite paths of the CTMC associated with an MRM are used to define the respective sets of finite and infinite paths of the MRM. More precisely, for the MRM $M = (S, R, p, L, \rho)$, where $C = (S, R, p, L)$, we let $\text{Path}^M = \text{Path}^C$. Furthermore, let $\text{Prob}_\alpha^M = \text{Prob}_\alpha^C$ for all distributions α on S , from which it follows that $\text{Prob}_s^M = \text{Prob}_s^C$ for each state $s \in S$. The reward accumulated along a path ω until time $t \in \mathbb{R}_{\geq 0}$, denoted by $y(\omega, t)$, is defined in the following way: If $\omega = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots$ is infinite, then, for $t = \sum_{i=0}^{k-1} t_i + t'$ with $t' \leq t_k$, we define

$$y(\omega, t) = \sum_{i=0}^{k-1} t_i \cdot \rho(s_i) + t' \cdot \rho(s_k).$$

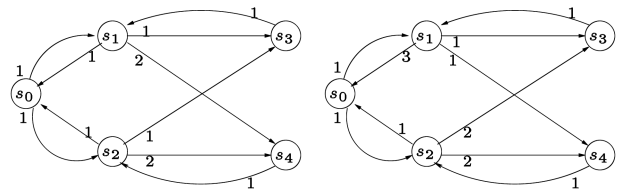
If, however, $\omega = s_0 \xrightarrow{t_0} s_1 \xrightarrow{t_1} \dots s_{n-1} \xrightarrow{t_{n-1}} s_n$ is finite, then we have two cases: If there exists $k < n$ such that $t = \sum_{i=0}^{k-1} t_i + t'$ with $t' \leq t_k$, then we define $y(\omega, t)$ as in the case of infinite paths; otherwise, we let

$$y(\omega, t) = \sum_{i=0}^{n-1} t_i \cdot \rho(s_i) + (t - \sum_{i=0}^{n-1} t_i) \cdot \rho(s_n).$$

3 STOCHASTIC BISIMULATION

We proceed to define the state equivalence relation for CTMCs called *stochastic bisimulation*. The equivalence is studied in two principal forms: the first, *forward* bisimulation, identifies states as equivalent if their outgoing transitions can be related, whereas the second equivalence, *backward* bisimulation, identifies states as equivalent if their incoming transitions can be related. Forward stochastic bisimulation is related to the notion of ordinary lumpability (also called strong lumpability) in Markov chains [33], while backward stochastic bisimulation is related to exact lumpability [36]. Unlike lumpability, we base the equivalences on the rate transition function R of the CTMC and not its infinitesimal generator (hence, our equivalences take self-loops into account).

Given a CTMC $C = (S, R, p, L)$, a state $s \in S$, and a set of states $C \subseteq S$, let $R(s, C) = \sum_{s' \in C} R(s, s')$ and let $R(C, s) = \sum_{s' \in C} R(s', s)$.



(C_A: forward, not backward) (C_B: backward, not forward)

Fig. 1. CTMCs which can be reduced by only one of forward or backward stochastic bisimulation.

Definition 3.1. Let $C = (S, R, p, L)$ be a CTMC. An equivalence relation \equiv over the set S of states is a forward stochastic bisimulation of C if, for all states $s, s' \in S$ such that $s \equiv s'$, we have $R(s, C) = R(s', C)$ for each equivalence class $C \in S/\equiv$.

Note that forward stochastic bisimilar states s, s' have the same exit rate, that is, $E(s) = E(s')$.

We now consider backward stochastic bisimulation and use a definition adapted from the inverse performance bisimulation of [14]. Our definition of backward stochastic bisimulation includes a condition which stipulates that the exit rates of equivalent states are the same, which is necessary in our context for making the equivalence useful for the verification of CSL path properties (that is, the properties “next” and “until,” see Section 5). As noted above, this condition on exit rates is implicit in the definition of forward stochastic bisimulation.

Definition 3.2. Let $C = (S, R, p, L)$ be a CTMC. An equivalence relation \equiv over the set S of states is a backward stochastic bisimulation of C if, for all states $s, s' \in S$ such that $s \equiv s'$, we have:

1. $R(C, s) = R(C, s')$ for each equivalence class $C \in S/\equiv$ and
2. $E(s) = E(s')$.

Example. In Fig. 1, we illustrate two CTMCs which have the same transition structure, but with different rates (for simplicity, we do not consider the initial distribution or the labeling function). Applying forward stochastic bisimulation to C_A results in the aggregation of states s_1 and s_2 into a single equivalence class and, also, s_3 and s_4 into a single class; however, backward stochastic bisimulation will not result in the aggregation of any states. On the other hand, backward stochastic bisimulation applied to C_B will aggregate s_1 and s_2 into a class and s_3 and s_4 into another class. Forward stochastic bisimulation cannot aggregate any states of C_B .

Two states $s, s' \in S$ of a continuous-time Markov chain C are *forward stochastic bisimilar*, denoted by $s \stackrel{f}{\equiv} s'$, if there exists a forward stochastic bisimulation \equiv which is such that $s \equiv s'$. If the equivalence \equiv is instead a backward stochastic bisimulation, then states can also be identified as *backward stochastic bisimilar*, denoted by $s \stackrel{b}{\equiv} s'$. Note that $\stackrel{f}{\equiv}$ and $\stackrel{b}{\equiv}$ correspond to the coarsest equivalence relations of each type.

We now consider three “static” conditions on states, which can be combined with the two definitions of stochastic equivalence given above to obtain equivalence relations which can be more distinguishing. First, we consider a condition which requires that equivalent states must have the same initial probability (possibly 0); the next two conditions require equality of state-labeling and rewards, respectively, in equivalent states. We formally introduce the definitions for MRMs, but the first two conditions can also be applied to stochastic bisimulation on CTMCs.

Definition 3.3. Let $M = (S, R, p, L, \rho)$ be an MRM. An equivalence relation \equiv over the set S of states satisfies:

- the initial condition if, for all states $s, s' \in S$ such that $s \equiv s'$, we have $p(s) = p(s')$;
- the state-labeling condition if, for all states $s, s' \in S$ such that $s \equiv s'$, we have $L(s) = L(s')$;
- the reward condition if, for all states $s, s' \in S$ such that $s \equiv s'$, we have $\rho(s) = \rho(s')$.

We use the subscripts I, L, and R to denote the initial, state-labeling, and reward conditions, respectively, that an equivalence satisfies. For example, \equiv_I denotes an equivalence relation satisfying the initial condition, \equiv_{LR} satisfies the state-labeling and reward conditions, whereas \equiv_{ILR} satisfies all three conditions. The imposition of a state-labeling is standard when relating temporal logic and bisimulation [5], [2], [28], [9] (and is another factor which distinguishes the notion of lumpability from that of stochastic bisimulation). The reward condition was employed previously in [8]. The initial condition is relevant in the case of the calculation of transient probabilities of a CTMC and steady-state probabilities for nonergodic CTMCs. We combine the notation for stochastic bisimulation and the static conditions of Definition 5 to obtain new equivalences, for example, \equiv_I^f (the coarsest forward stochastic bisimulation satisfying the initial condition) or \equiv_{LR}^b (the coarsest backward stochastic bisimulation satisfying both the state-labeling and reward conditions).

The definition of an equivalence relation on the state space of an MRM can be used to define a *quotient* MRM, the states of which are classes of the equivalence relation. We consider the definition of Buchholz [14]. Again, the definition can be applied to a CTMC by equipping the CTMC with an arbitrary reward function, building the quotient, then by considering only the CTMC of the resulting quotient MRM.

Definition 3.4 [14]. Let $M = (S, R, p, L, \rho)$ be an MRM and \equiv be an equivalence relation on S . The quotient of M is an MRM $M/\equiv = (S/\equiv, R/\equiv, p/\equiv, L/\equiv, \rho/\equiv)$, where:

- S/\equiv is the set of equivalence classes of \equiv ;
- $R/\equiv : S/\equiv \times S/\equiv \rightarrow [0, 1]$ is defined such that, for each pair of classes $C, C' \in S/\equiv$, we have:

$$R/\equiv(C, C') = \frac{\sum_{s \in C} \sum_{s' \in C'} R(s, s')}{|C|};$$

- $p/\equiv : S/\equiv \rightarrow [0, 1]$ is defined by $p/\equiv(C) = \sum_{s \in C} p(s)$ for each $C \in S/\equiv$;

- $L/\equiv : S/\equiv \rightarrow 2^{AP}$ is defined by $L/\equiv(C) = \bigcup_{s \in C} L(s)$ for each $C \in S/\equiv$;
- $\rho/\equiv : S/\equiv \rightarrow \mathbb{R}_{\geq 0}$ is defined by $\rho/\equiv(C) = \frac{\sum_{s \in C} \rho(s)}{|C|}$ for each $C \in S/\equiv$.

For each $C \in S/\equiv$, if \equiv satisfies the state-labeling condition, then $L/\equiv(C) = L(s)$ for each/any state $s \in C$; similarly, if the equivalence satisfies the reward condition, then $\rho/\equiv(C) = \rho(s)$ for each/any state $s \in C$. We note that the labeling function L/\equiv will be used in this paper only in the context of equivalence relations which satisfy the state-labeling condition (hence, L/\equiv could have been equally defined as the intersection, rather than the union, of the label sets of a class's constituent states). In the case in which \equiv is a forward stochastic bisimulation, the definition above collapses to the definitions of [13], [26], [9].

Recall the CTMCs C_A and C_B of Fig. 1. The quotient C_A/\equiv^f of C_A with respect to \equiv^f is comprised of the state set $S/\equiv^f = \{\{s_0\}, \{s_1, s_2\}, \{s_3, s_4\}\}$, and the rate transition function R/\equiv^f defined by

$$R/\equiv^f(\{s_0\}, \{s_1, s_2\}) = 2, R/\equiv^f(\{s_1, s_2\}, \{s_3, s_4\}) = 3, \\ R/\equiv^f(\{s_3, s_4\}, \{s_1, s_2\}) = 1, \text{ and } R/\equiv^f(\{s_1, s_2\}, \{s_0\}) = 1.$$

Similarly, the quotient C_B/\equiv^b of C_B with respect to \equiv^b is comprised of the same state set $S/\equiv^b = S/\equiv^f$ and the rate transition function R/\equiv^b , which is identical to R/\equiv^f except that $R/\equiv^b(\{s_1, s_2\}, \{s_0\}) = 2$. Given the definition of initial and labeling conditions for C_A and C_B , the corresponding conditions of the quotient CTMCs can be defined according to Definition 3.4.

Both the forward and backward stochastic bisimulation equivalence classes of a CTMC can be computed in time $O(|R| \log |S|)$, where $|R|$ is the number of state pairs with positive rate according to R (that is, the number of nonzero entries in the matrix induced by R) [22].

We recall the following result from [14], which relates the transient and steady-state probabilities of the original and quotient CTMCs obtained from backward stochastic bisimulation under the initial condition (or under a combination of conditions, including the initial condition). More precisely, the theorem specifies that each state in an equivalence class has the same probability of being reached after t time units and that this probability can be obtained by calculating the probability of reaching the class after t time units in the quotient system and then dividing by the cardinality of the class.

Theorem 3.1 [14]. Let $C = (S, R, p, L)$ be a CTMC and C/\equiv be its quotient CTMC with respect to $\equiv \in \{\equiv_I^b, \equiv_{IL}^b, \equiv_{IR}^b, \equiv_{ILR}^b\}$. Then, for all classes $C \in S/\equiv$ and for any state $s \in C$, we have:

$$\pi^C(p, s, t) = \frac{\pi^{C/\equiv}(p/\equiv, C, t)}{|C|}.$$

Note that this theorem implies that $\pi^C(p, s) = \pi^{C/\equiv}(p/\equiv, C)/|C|$ for $\equiv \in \{\equiv_I^b, \equiv_{IL}^b, \equiv_{IR}^b, \equiv_{ILR}^b\}$. The theorem describes the main (theoretical) advantage of backward equivalence over forward equivalence, for which the theorem does not hold.

4 QUALITATIVE PROPERTIES

In this section, we briefly consider the qualitative properties of backward stochastic bisimulation. More precisely, we add to the results of Buchholz [14], which described the equivalence of the set of finite behaviors of the unreduced CTMC and that of its quotient CTMC, to show the equivalence of the sets of *infinite behaviors* of the unreduced CTMC and its quotient CTMC.

A *transition system* $T = (S, \Rightarrow, \bar{t}, L)$ is a tuple comprised of a finite set S of states, a transition relation $\Rightarrow \subseteq S \times S$, a set of initial states $\bar{t} \subseteq S$, and a labeling function $L : S \rightarrow 2^{AP}$.

Definition 4.1. *The transition system of a CTMC $C = (S, R, p, L)$ is the tuple $T_C = (S, \Rightarrow, \bar{t}, L)$, where:*

- $\Rightarrow \subseteq S \times S$ is such that $(s, s') \in \Rightarrow$ if and only if $R(s, s') > 0$ and
- $\bar{t} \subseteq S$ is such that $s \in \bar{t}$ if and only if $p(s) > 0$.

An initialized path of T_C is an infinite sequence $s_0 s_1 s_2 \dots$ such that $s_0 \in \bar{t}$ and $(s_i, s_{i+1}) \in \Rightarrow$ for all $i \in \mathbb{N}$. The trace of an initialized path $s_0 s_1 s_2 \dots$ is the sequence $L(s_0)L(s_1)L(s_2)\dots$. The set of traces of a transition system, denoted by $\text{Traces}(T)$, is the set of traces of all initialized paths of T .

We now proceed to define backward bisimulation on transition systems, following [34].

Definition 4.2. *Let $T = (S, \Rightarrow, \bar{t}, L)$ be a transition system. An equivalence relation \equiv over the set S of states is a backward bisimulation of T if, for all states $s, s' \in S$ such that $s \equiv s'$, if $(u, s) \in \Rightarrow$, then there exists $(u', s') \in \Rightarrow$ such that $u \equiv u'$. Furthermore, the equivalence \equiv satisfies the initial condition if $s \equiv s'$ implies that $s \in \bar{t}$ if and only if $s' \in \bar{t}$ and the state-labeling condition if $s \equiv s'$ implies that $L(s) = L(s')$.*

Given a transition system, let \simeq_{IL}^b be the coarsest backward bisimulation which satisfies the initial condition and the state-labeling condition. Given two transition systems, $T_1 = (S_1, \Rightarrow_1, \bar{t}_1, L_1)$ and $T_2 = (S_2, \Rightarrow_2, \bar{t}_2, L_2)$ such that $S_1 \cap S_2 = \emptyset$, we can construct the “union” transition system $T_1 \cup T_2$ in the following way: Let $T_1 \cup T_2 = (S_1 \cup S_2, \Rightarrow_1 \cup \Rightarrow_2, \bar{t}_1 \cup \bar{t}_2, L_{12})$, where $L_{12}(s) = L_1(s)$ if $s \in S_1$ and $L_{12}(s) = L_2(s)$ if $s \in S_2$. Two transition systems $T_1 = (S_1, \Rightarrow_1, \bar{t}_1, L_1)$ and $T_2 = (S_2, \Rightarrow_2, \bar{t}_2, L_2)$ are *backward bisimilar* if, in the union transition system $T_1 \cup T_2$, for each $s_1 \in \bar{t}_1$, there exists an $s_2 \in \bar{t}_2$ such that $s_1 \simeq_{\text{IL}}^b s_2$ (and vice versa).

It follows from the work of Lynch and Vaandrager concerning backward simulations of (image-finite) transition systems [34] that the sets of traces of (finite-state) backward bisimilar transition systems are equal, as stated formally by the following theorem.

Theorem 4.1 [34]. *Let T_1 and T_2 be transition systems. If T_1 and T_2 are backward bisimilar, then $\text{Traces}(T_1) = \text{Traces}(T_2)$.*

We now show that T_C and $T_{C/\simeq_{\text{IL}}^b}$ are backward bisimilar. Once we have established this result, from Theorem 4.1, we will have that the set of traces of T_C and $T_{C/\simeq_{\text{IL}}^b}$ are equal. Our proof is similar to that of Buchholz [14], which

considered the transition-by-transition correspondence of T_C and $T_{C/\simeq_{\text{IL}}^b}$ in order to show equivalence of finite traces.

Proposition 4.1. *Let C be a CTMC. Then, T_C and $T_{C/\simeq_{\text{IL}}^b}$ are backward bisimilar.*

Proof. Let $C = (\tilde{S}, \tilde{R}, \tilde{p}, \tilde{L})$, $C/\simeq_{\text{IL}}^b = (\tilde{S}, \tilde{R}, \tilde{p}, \tilde{L})$, $T_C = (S, \Rightarrow, \bar{t}, L)$, and $T_{C/\simeq_{\text{IL}}^b} = (\tilde{S}, \widetilde{\Rightarrow}, \tilde{\bar{t}}, \tilde{L})$. We show that, for $s \in S$ and $C \in \tilde{S}$, if $s \in C$, then $s \simeq_{\text{IL}}^b C$.

The first task is to show that $s \in C$ implies $L(s) = \tilde{L}(C)$, which follows immediately from the state-labeling condition and the definition of \tilde{L} .

The second task is to show that $s \in C$ implies that $s \in \bar{t}$ if and only if $C \in \tilde{\bar{t}}$. On the one hand, $s \in \bar{t}$ implies that $p(s) > 0$. Then, as $s \in C$ and as $\tilde{p}(C) = \sum_{s \in C} p(s)$, we have that $\tilde{p}(C) > 0$ and, therefore, $C \in \tilde{\bar{t}}$. Hence, $s \in \bar{t}$ implies $C \in \tilde{\bar{t}}$. On the other hand, $\tilde{p}(C) > 0$ implies $p(s) > 0$ because \simeq_{IL}^b satisfies the initial condition and, therefore, $p(s)$ is equal for all states in C . Hence $C \in \tilde{\bar{t}}$ implies $s \in \bar{t}$.

The third task is to show that $s \in C$ implies that: $(s', s) \in \Rightarrow$ implies $(C', C) \in \widetilde{\Rightarrow}$ for the class C' such that $s' \in C'$. From $(s', s) \in \Rightarrow$, we must have $R(s', s) > 0$, which, by Definition 3.4, implies that $\tilde{R}(C', C) > 0$, which in turn gives us $(C', C) \in \widetilde{\Rightarrow}$ by the definition of $T_{C/\simeq_{\text{IL}}^b}$.

The fourth task is to show that $s \in C$ implies that: $(C', C) \in \widetilde{\Rightarrow}$ implies $(s', s) \in \Rightarrow$ for some $s' \in C'$. From $(C', C) \in \widetilde{\Rightarrow}$, we have $\tilde{R}(C', C) > 0$ by the definition of $T_{C/\simeq_{\text{IL}}^b}$. Now, because C is a \simeq_{IL}^b -equivalence class, we know that $R(C', u) = R(C', u')$ for all states $u, u' \in C$. Hence, by Definition 3.4, applied for \simeq_{IL}^b , if $\tilde{R}(C', C) > 0$, then $R(C', u) > 0$ for all states $u \in C$. Then, we know that $R(s', s) > 0$ for some $s' \in C'$; hence, $(s', s) \in \Rightarrow$ and we are done. \square

Corollary 1. *Let C be a CTMC. Then, $\text{Traces}(T_C) = \text{Traces}(T_{C/\simeq_{\text{IL}}^b})$.*

Because trace equivalence implies equivalence of linear-time temporal logic (LTL) properties, these results mean that we can use the quotient CTMC C/\simeq_{IL}^b in place of the unreduced CTMC C for qualitative model checking of such properties. We note that trace sets may include traces exhibited with probability 0 and, therefore, our results are oriented toward traditional model checking of linear properties over all system traces, rather than probabilistic model checking over the set of traces exhibited with probability 1 [38], [21].

5 CONTINUOUS STOCHASTIC LOGIC

5.1 Syntax and Semantics

We now recall the syntax and semantics of Continuous Stochastic Logic (CSL) [3], [9].

Definition 5.1. *The syntax of CSL is defined as follows:*

$$\Phi ::= a \mid \Phi \wedge \Phi \mid \neg \Phi \mid \mathcal{P}_{\triangleright \lambda}(X^I \Phi) \mid \mathcal{P}_{\triangleright \lambda}(\Phi U^I \Phi) \mid \mathcal{S}_{\triangleright \lambda}(\Phi),$$

where $a \in AP$ is an atomic proposition, $I \subseteq \mathbb{R}_{\geq 0}$ is a nonempty interval, $\bowtie \in \{<, \leq, \geq, >\}$ is a comparison operator, and $\lambda \in [0, 1]$ is a probability.

A formula of CSL refers to a property which is either satisfied or not satisfied in a given state of a CTMC. We often refer to CSL formulae as *state formulae*. Instead, the subformulae $X^I\Phi$ and $\Phi_1 U^I \Phi_2$ are interpreted as being satisfied or not satisfied for a path of the CTMC. These *path formulae* are interpreted as follows: $X^I\Phi$ is true for a path if the state reached after the first transition along the path satisfies Φ and the duration of this transition lies in the interval I ; the formula $\Phi_1 U^I \Phi_2$ is true along a path if Φ_2 is true at some state along the path, the time elapsed before reaching this state lies in I , and Φ_1 is true along the path until that state. The *probabilistic quantifier* \mathcal{P} is used to refer to the probability of satisfying a path formula, while the *steady-state quantifier* \mathcal{S} refers to the steady-state probability of satisfying a CSL subformula. Examples of CSL formulae, taken from [9], are the following: The formula $\mathcal{S}_{\leq 10^{-5}}(a)$ is satisfied in state s if the probability of being in a state labeled by the atomic proposition a in steady-state is not greater than 0.00001, having started execution from s (obviously, the truth value is not influenced by s if the CTMC is ergodic). The formula $\mathcal{P}_{\leq 0.01}(aU^{[10,20]}b)$ is satisfied in state s if the probability of being in a b -labeled state after between 10 and 20 time units have elapsed, while remaining in a -labeled states before that point, is not greater than 0.01 for system behaviors leaving s . The formula $\mathcal{S}_{\geq 0.9}(\mathcal{P}_{< 0.2}(\text{true}U^{[0,10]}a))$ is satisfied in state s if, in equilibrium with probability at least 0.9, the probability over all possible paths of reaching an a -labeled state, within 10 time units, is at most 0.2. Finally, the formula $\mathcal{P}_{\geq 0.5}(\neg aU^{[10,20]}\mathcal{S}_{\geq 0.8}(b \vee c))$ is satisfied in state s if, with probability at least 0.5, we will reach, from s , a state between 10 and 20 time units (while avoiding a -states) in which the probability of being in a b or c -labeled state in equilibrium is at least 0.8.

Next, we recall the satisfaction relation \models for which $s \models \Phi$ indicates that the CSL formula Φ is satisfied in state s . Given the satisfaction relation \models , let $\text{Sat}(\Phi) = \{s \in S \mid s \models \Phi\}$.

Definition 5.2 [9]. For $C = (S, R, p, L)$ and state $s \in S$, the satisfaction relation \models is defined as follows:

$$\begin{aligned} s &\models a && \text{iff } a \in L(s) \\ s &\models \Phi_1 \wedge \Phi_2 && \text{iff } s \models \Phi_1 \text{ and } s \models \Phi_2 \\ s &\models \neg\Phi && \text{iff } s \not\models \Phi \\ s &\models \mathcal{S}_{\bowtie\lambda}(\Phi) && \text{iff } \pi^C(\alpha_s, \text{Sat}(\Phi)) \bowtie \lambda \\ s &\models \mathcal{P}_{\bowtie\lambda}(\varphi) && \text{iff } \text{Prob}_s^C\{\omega \in \text{Path}^C \mid \omega \models \varphi\} \bowtie \lambda \\ \omega &\models X^I\Phi && \text{iff } \omega(1) \text{ is defined and } \\ &&& \omega(1) \models \Phi \wedge \delta(\omega, 0) \in I \\ \omega &\models \Phi_1 U^I \Phi_2 && \text{iff } \exists t \in I. \omega @ t \models \Phi_2 \text{ and } \\ &&& \forall t' \in [0, t). \omega @ t' \models \Phi_1. \end{aligned}$$

Normally, CSL formulae are evaluated on the states of a CTMC, that is, the output of the CSL model-checking algorithm is the set of states of the CTMC which satisfy the formula. However, we can also consider the interpretation of CSL formulae, not only on individual states of a CTMC, but also on the entire CTMC, taking into account its initial distribution function. For example, we can say that a CTMC

satisfies the CSL formula $\mathcal{P}_{\geq\lambda}(\varphi)$ if and only if the probability of the φ -satisfying paths, weighted by the probabilities of their starting states given by the initial distribution, is $\geq \lambda$. Such an interpretation of CSL on CTMCs makes sense only for formulae for which the outermost operator is a probabilistic or steady-state quantifier \mathcal{P} or \mathcal{S} . Furthermore, in the sequel, it will be useful to reason about the sublogic of CSL which does not feature arbitrary nesting of the probabilistic quantifier \mathcal{P} .

Observe that we can construct a parse tree for a CSL formula, the leaves of which are the formulae of length 1 (atomic propositions) and the root of which is the formula itself. For a CSL formula Φ , we use the convention that the set of its *subformulae* includes Φ itself, whereas its *strict subformulae* is the set of subformulae of Φ minus Φ itself.

Definition 5.3. For $\mathcal{Q} \in \{\mathcal{P}, \mathcal{S}\}$, a \mathcal{Q} -outermost formula of CSL is a formula for which the outermost operator (at the root of the parse tree of the formula) is \mathcal{Q} . A formula Φ of CSL is \mathcal{Q} -nesting free if \mathcal{Q} does not appear in any of the strict subformulae of Φ .

Observe that all of the examples of CSL formulae listed above are \mathcal{P} or \mathcal{S} -outermost. The first two are \mathcal{P} and \mathcal{S} -nesting-free, whereas the third is \mathcal{S} -nesting-free and the fourth is \mathcal{P} -nesting-free. We recall that the CSL variant of [11] for model checking semi-Markov processes is \mathcal{P} and \mathcal{S} -nesting-free.

By considering the initial distribution of a CTMC, we can reason about whether a CTMC satisfies a \mathcal{S} or \mathcal{P} -outermost CSL formula.

Definition 5.4. For $C = (S, R, p, L)$ and \mathcal{S} or \mathcal{P} -outermost CSL formula $\mathcal{S}_{\bowtie\lambda}(\Phi)$ or $\mathcal{P}_{\bowtie\lambda}(\varphi)$, the satisfaction relation \models is defined as follows:

$$\begin{aligned} C &\models \mathcal{S}_{\bowtie\lambda}(\Phi) && \text{iff } \pi^C(p, \text{Sat}(\Phi)) \bowtie \lambda \\ C &\models \mathcal{P}_{\bowtie\lambda}(\varphi) && \text{iff } \text{Prob}_p^C\{\omega \in \text{Path} \mid \omega \models \varphi\} \bowtie \lambda. \end{aligned}$$

5.2 Stochastic Bisimulation and CSL

In this section, we identify which of the stochastic bisimulation relations defined in Section 3 can be used to define quotient CTMCs for verifying CSL formulae.

5.2.1 Equivalences with the State-Labeling Condition

We start with a review of results concerning forward stochastic bisimulation equivalence with the state-labeling condition \cong_L^f and its relation to CSL. Let $C = (S, R, p, L)$ be a CTMC and Φ be a CSL formula. Then, from [9], for each pair $s, s' \in S$ of states, if $s \cong_L^f s'$, then we have $s \models \Phi$ if and only if $s' \models \Phi$. This result implies that we are able to construct a quotient CTMC C/\cong_L^f on which the standard CSL model-checking algorithms may be applied; then, the set of classes which satisfy a CSL formula in C/\cong_L^f will contain exactly those states of C which satisfy the formula. Also, note that, because the equivalence relation \cong_{IL}^f distinguishes at least as \cong_L^f , if $s \cong_{\text{IL}}^f s'$, then we have $s \models \Phi$ if and only if $s' \models \Phi$. Furthermore, if Φ is a \mathcal{P} or \mathcal{S} -outermost CSL formula, then $C \models \Phi$ if and only if $C/\cong_{\text{IL}}^f \models \Phi$. We also recall that agreement of CSL formulae characterizes forward stochastic

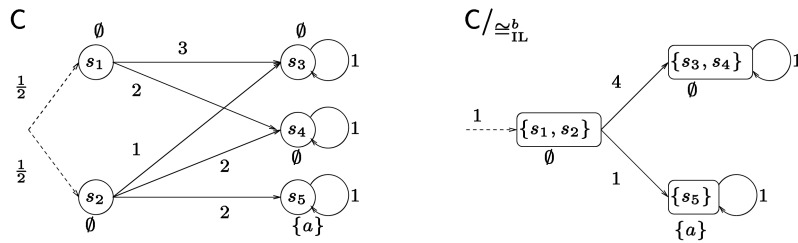


Fig. 2. Backward stochastic bisimulation quotients do not preserve CSL.

bisimulation: More precisely, CTMC states which satisfy the same CSL formulae are \cong_L^f -equivalent [23].

In contrast to the case of forward stochastic bisimulation, backward stochastic bisimulation cannot in general be used to define quotient CTMCs from which results of arbitrary CSL formulae can be derived. This is a consequence of the fact that states satisfying the same CSL formulae are \cong_L^f -equivalent and from the fact that, as indicated by the examples in Fig. 1, the distinguishing powers of forward and backward stochastic bisimulation equivalences are incomparable. Consider the CTMC C and the quotient CTMC C/\cong_{IL}^b , represented graphically in Fig. 2. State labels are written next to their associated states and the initial distributions are denoted by dotted arrows. States s_1 and s_2 of C are backward stochastic bisimilar (under the initial and state-labeling conditions) and form the equivalence class $\{s_1, s_2\}$ of C/\cong_{IL}^b . Now, consider the CSL subformula $\mathcal{P}_{>0}(X^{[0,\infty)}a)$, which requires that an a -labeled state is reachable in one step with positive probability. As s_5 is the only a -labeled state, this property reduces to requiring that s_5 is reachable in one step with positive probability. Consider the case in which this subformula is nested inside another formula, such as $\mathcal{S}_{\geq 0.5}(\mathcal{P}_{>0}(X^{[0,\infty)}a))$ or $b \wedge \mathcal{P}_{>0}(X^{[0,\infty)}a)$. Then, our aim is to obtain the set of states which satisfy the property $\mathcal{P}_{>0}(X^{[0,\infty)}a)$ so that this set can be used when verifying the properties within which $\mathcal{P}_{>0}(X^{[0,\infty)}a)$ is nested. However, note that $\mathcal{P}_{>0}(X^{[0,\infty)}a)$ is satisfied in the class $\{s_1, s_2\}$ of C/\cong_{IL}^b and in the state s_2 of C , but not in s_1 . The problem is that the construction of the class $\{s_1, s_2\}$ does not retain the distinction about the probability of the future behavior from the states within $\{s_1, s_2\}$. Hence, the threshold $\bowtie \lambda$ of a probabilistically quantified path formula could be satisfied by some states of a class and not by others.

This characteristic is also exhibited in the case of the until path operator (for example, $\mathcal{P}_{>0}(\text{true}U^{[0,\infty)}a)$) and the steady-state operator \mathcal{S} . With regard to the latter, consider the formula $\mathcal{S}_{>0}(a)$ and say that this formula is nested within another formula. Then, the formula is false in the state s_1 because the system cannot reach state s_5 , which forms the only bottom strongly connected component in which a holds. However, the formula $\mathcal{S}_{>0}(a)$ is true in the state s_2 and in the class $\{s_1, s_2\}$. In a CTMC that is not strongly connected, the steady-state distribution depends on the probabilities with which its bottom strongly

connected components are reached; as indicated by the previous paragraph, backward stochastic bisimulation does not preserve exact reachability probabilities in equivalent states.

Hence, we are unable to identify the set of states which satisfy arbitrary CSL formulae using the quotient CTMC obtained using backward stochastic bisimulation. However, if we restrict our attention to a subclass of CSL, the picture is different: Backward stochastic bisimulation can be used to define a quotient CTMC on which \mathcal{P} or \mathcal{S} -outermost, \mathcal{P} and \mathcal{S} -nesting-free CSL formulae can be verified. If, for example, we consider the formula $\mathcal{P}_{>0}(X^{[0,\infty)}a)$ in isolation, then the formula will be true when interpreted on both C and on C/\cong_{IL}^b (taking the initial distributions into account). We commence with a lemma which specifies that the probability of any class in the quotient CTMC satisfying a path formula $X^I\Phi$ or $\Phi_1U^I\Phi_2$ is the *average* of the probabilities of satisfying the formula in the constituent states of the class.

Lemma 5.1. *Let Φ_1, Φ_2 be CSL formulae and assume that, for each pair of states $s, s' \in S$, we have $s \cong_{\text{IL}}^b s'$ implies $s \models \Phi_1$ if and only if $s' \models \Phi_1$, and $s \models \Phi_2$ if and only if $s' \models \Phi_2$. Let φ be either $X^I\Phi_1$ or $\Phi_1U^I\Phi_2$. Then, for each class $C \in S/\cong_{\text{IL}}^b$, we have:*

$$\text{Prob}_C^{C/\cong_{\text{IL}}^b}(\varphi) = \frac{\sum_{s \in C} \text{Prob}_s^C(\varphi)}{|C|}.$$

The proof of this lemma is given in the Appendix. The lemma can be used to establish the following result:

Theorem 5.1. *Let $C = (S, R, p, L)$ be a CTMC and Φ be a \mathcal{P} or \mathcal{S} -outermost, \mathcal{P} and \mathcal{S} -nesting-free CSL formula. Then, $C \models \Phi$ if and only if $C/\cong_{\text{IL}}^b \models \Phi$.*

Proof. Let $C = (S, R, p, L)$ be a CTMC and Φ be a \mathcal{P}, \mathcal{S} -nesting-free CSL formula. We aim to show that $C \models \Phi$ if and only if $C/\cong_{\text{IL}}^b \models \Phi$. To aid readability, we write all notation pertaining to the quotient C/\cong_{IL}^b with the overline \sim . For example, we write \tilde{C} instead of C/\cong_{IL}^b , \tilde{S} instead of S/\cong_{IL}^b , and \tilde{R} instead of R/\cong_{IL}^b . Notation which derives from the quotient's components is treated in the same way: For example, we let $\tilde{E}(C) = \sum_{C' \in \tilde{S}} \tilde{R}(C, C')$. For a CSL formula Ψ , we let $\text{Sat}^C(\Psi)$ be the set of states of the CTMC C satisfying Ψ , whereas $\tilde{\text{Sat}}^{\tilde{C}}(\Psi)$ is the set of states of the quotient CTMC \tilde{C} satisfying Ψ .

Case: Φ is a steady-state formula. That is, we consider the case in which Φ is of the form $\mathcal{S}_{\bowtie \lambda}(\Psi)$. As the formula is \mathcal{P}, \mathcal{S} -nesting-free, the subformula Ψ is a formula of the

syntax $\Psi ::= a \mid \Psi \wedge \Psi \mid \neg\Psi$ for $a \in AP$. Our first task is to show that $s \cong_{\text{IL}}^b s'$ implies $s \models \Psi$ if and only if $s' \models \Psi$ for any states $s, s' \in S$. We proceed by induction on the length of subformulae of the formulae Ψ . Consider a subformula $a \in AP$: From the state-labeling condition, we have that $s \cong_{\text{IL}}^b s'$ implies $s \models a$ if and only if $s' \models a$. The cases for the Boolean connectives then follow and our first task is completed.

Our second task is to show that:

$$\pi^C(p, \text{Sat}^C(\Psi)) = \pi^{\tilde{C}}(\tilde{p}, \text{Sat}^{\tilde{C}}(\Psi)).$$

It follows from the first task that, for each class C of \cong_{IL}^b , either $C \subseteq \text{Sat}^C(\Psi)$ or $C \cap \text{Sat}^C(\Psi) = \emptyset$. By the definition of the labeling function of \tilde{C} , we have $C \subseteq \text{Sat}^{\tilde{C}}(\Psi)$ if and only if $C \in \text{Sat}^{\tilde{C}}(\Psi)$. From this fact and from the fact that classes in \tilde{S} are disjoint, we have that $\{C \mid C \in \text{Sat}^{\tilde{C}}(\Psi)\}$ is a partition of $\text{Sat}^{\tilde{C}}(\Psi)$. Then, we have:

$$\begin{aligned} & \pi^C(p, \text{Sat}^C(\Psi)) \\ &= \sum_{s \in \text{Sat}^C(\Psi)} \pi^C(p, s) \\ &= \sum_{C \in \tilde{S} \ \& \ C \subseteq \text{Sat}^C(\Psi)} \sum_{s \in C} \pi^C(p, s) \\ & \text{(because } \{C \mid C \in \text{Sat}^{\tilde{C}}(\Psi)\} \text{ is a partition of } \text{Sat}^{\tilde{C}}(\Psi)) \\ &= \sum_{C \in \tilde{S} \ \& \ C \subseteq \text{Sat}^{\tilde{C}}(\Psi)} \frac{\pi^{\tilde{C}}(\tilde{p}, C)}{|C|} \quad \text{(by Theorem 3.1)} \\ &= \pi^{\tilde{C}}(\tilde{p}, \text{Sat}^{\tilde{C}}(\Psi)). \end{aligned}$$

By the definition of the semantics of the steady-state operator, we then have $C \models \mathcal{S}_{\triangleright\lambda}(\Psi)$ if and only if $\tilde{C} \models \mathcal{S}_{\triangleright\lambda}(\Psi)$ and we are done.

Case: Φ is a probabilistically quantified formula. We now consider the case in which Φ is of the form $\mathcal{P}_{\triangleright\lambda}(\varphi)$, where φ is either of the form $\Psi_1 U^J \Psi_2$ or $X^I \Psi_3$, where Ψ_1 , Ψ_2 , and Ψ_3 are formulae of the syntax $\Psi ::= a \mid \Psi \wedge \Psi \mid \neg\Psi$. Our first task is to show that, for each state pair $s, s' \in S$, we have that $s \cong_{\text{IL}}^b s'$ implies $s \models \Psi$ if and only if $s' \models \Psi$. This fact follows from the analogous part of the proof for steady-state formulae.

Given that this property holds, we now prove that the probabilities of the path formulae φ in C and \tilde{C} agree, hence establishing that $C \models \mathcal{P}_{\triangleright\lambda}(\varphi)$ if and only if $\tilde{C} \models \mathcal{P}_{\triangleright\lambda}(\varphi)$. From the definition of the probability measures Prob_p^C and $\text{Prob}_{\tilde{p}}^{\tilde{C}}$, we have $\text{Prob}_p^C(\varphi) = \sum_{s \in S} p(s) \cdot \text{Prob}_s^C(\varphi)$ and $\text{Prob}_{\tilde{p}}^{\tilde{C}}(\varphi) = \sum_{C \in \tilde{S}} \tilde{p}(C) \cdot \text{Prob}_C^{\tilde{C}}(\varphi)$. From the initial condition, for each state $s \in S$, we have $p(s) = \frac{\tilde{p}(C)}{|C|}$ and, therefore, $\tilde{p}(C) = p(s) \cdot |C|$, for the unique class C of \cong_{IL}^b for which $s \in C$. Hence,

$$\begin{aligned} \text{Prob}_{\tilde{p}}^{\tilde{C}}(\varphi) &= \sum_{C \in \tilde{S}} \tilde{p}(C) \cdot \text{Prob}_C^{\tilde{C}}(\varphi) \\ &= \sum_{C \in \tilde{S}} \tilde{p}(C) \cdot \frac{\sum_{s'' \in C} \text{Prob}_{s''}^C(\varphi)}{|C|} \quad \text{(by Lemma 5.1)} \\ &= \sum_{C \in \tilde{S}} \sum_{s'' \in C} \tilde{p}(C) \cdot \frac{\text{Prob}_{s''}^C(\varphi)}{|C|} \\ &= \sum_{C \in \tilde{S}} \sum_{s'' \in C} p(s'') \cdot |C| \cdot \frac{\text{Prob}_{s''}^C(\varphi)}{|C|} \\ & \quad \text{(by } \tilde{p}(C) = p(s'') \cdot |C| \text{ for } s'' \in C) \\ &= \sum_{C \in \tilde{S}} \sum_{s'' \in C} p(s'') \cdot \text{Prob}_{s''}^C(\varphi) \\ &= \sum_{s'' \in S} p(s'') \cdot \text{Prob}_{s''}^C(\varphi) \\ &= \text{Prob}_p^C(\varphi). \end{aligned}$$

Because $\text{Prob}_{\tilde{p}}^{\tilde{C}}(\varphi) = \text{Prob}_p^C(\varphi)$, by the definition of the semantics of probabilistically quantified CSL operators, we have that $\tilde{C} \models \mathcal{P}_{\triangleright\lambda}(\varphi)$ if and only if $C \models \mathcal{P}_{\triangleright\lambda}(\varphi)$. This establishes the second case of the proof and, hence, we have concluded the proof of Theorem 5.1. \square

5.2.2 Removing the State-Labeling Condition for Basic \mathcal{S} Formulae

We now show how the state-labeling condition of stochastic bisimulation can be weakened for certain CSL formulae, thereby resulting in a potentially smaller quotient CTMC. As indicated in [9], restricting the state-labeling condition to only those atomic propositions that appear in a CSL formula still permits us to obtain a quotient structure which preserves the formula (in the case of forward and backward equivalences). We go further in the case of an outermost \mathcal{S} -operator (which is \mathcal{P} and \mathcal{S} -nesting-free) and show that it suffices to consider backward stochastic bisimulation *without* a state-labeling condition. The result depends heavily on the equiprobability of equivalent states (recall Theorem 3.1) and it therefore applies only to backward stochastic bisimulation. The quotient CTMC that results from such an equivalence is then subject to the following, slightly modified CSL-model-checking algorithm in order to obtain meaningful results concerning the original CTMC.

Let Φ be an \mathcal{S} -outermost, \mathcal{P} and \mathcal{S} -nesting-free CSL formula. Then, Φ is of the form $\mathcal{S}_{\triangleright\lambda}(\Psi)$, with Ψ defined by $\Psi ::= a \mid \Psi \wedge \Psi \mid \neg\Psi$. Due to the equiprobability of backward bisimilar states, we can evaluate Φ using \cong_1^b instead of \cong_{IL}^b as follows: We proceed by first calculating the steady-state probabilities on the quotient CTMC C/\cong_1^b . Next, for each class C of \cong_1^b , we obtain $|C \cap \text{Sat}^C(\Psi)|$, the number of states within the class which satisfy the formula Ψ (this number can be computed locally by considering the class C in isolation). Then, the total steady-state probability for the set of states within the class which satisfy Ψ can be obtained by considering the steady-state probability computed for the class on the quotient CTMC, multiplied by the proportion of the states which satisfy Φ to the total number of states in the class, $|C \cap \text{Sat}^C(\Psi)|/|C|$. This process can be repeated for all classes and the sum of the obtained steady-state probabilities over all classes gives the required steady-state probability for

the unreduced CTMC. Hence, we have obtained a model-checking method for \mathcal{S} -outermost, \mathcal{S} , \mathcal{P} -nesting-free steady-state properties on a quotient CTMC which is independent of the state-labeling function. Recalling that the time complexity of model checking the steady-state operator given in [9] is $O(|S|^3)$, we see that the extra time complexity of having to count the number of the states of interest in each equivalence class (that is, $O(|S|)$) does not increase the overall complexity.

6 CONTINUOUS STOCHASTIC REWARD LOGIC

6.1 Syntax and Semantics

We briefly recall the definition of Continuous Stochastic Reward Logic (CSRL) [7]. The syntax of CSRL is identical to that of CSL, except that, in place of the path formulae $X^I\Phi$ and $\Phi_1 U^I \Phi_2$, we have the path formulae $X_J^I\Phi$ and $\Phi_1 U_J^I \Phi_2$, where the interval $J \subseteq \mathbb{R}_{\geq 0}$ represents a bound on the reward accumulated along a path. The formula $X_J^I\Phi$ is satisfied by a path if the state reached after the first transition along the path satisfies Φ , the duration of this transition lies in the interval I , and the reward accumulated before the transition lies in the interval J . Similarly, the formula $\Phi_1 U_J^I \Phi_2$ is satisfied by a path if Φ_2 is satisfied by some state along the path, the accumulated time and reward before this state lie in I and J , respectively, and Φ_1 is satisfied along the path until that state. For example, $\mathcal{P}_{\geq 0.98}(aU_{[7,\infty)}^{[0,30]}b)$ is satisfied by a state if, with probability at least 0.98, the atomic proposition b holds at some point at which no more than 30 time units have elapsed and at least seven units of reward have been accumulated and a holds until that point.

We now recall the formal semantics of CSRL. The semantics for the atomic propositions, Boolean operators, and probabilistic and steady-state quantifiers \mathcal{P} and \mathcal{S} are the same as for CSL, as expressed in Definition 5.2. It therefore suffices to define the semantics of the path formulae. Recall the definition of elapsed time $\delta(\cdot, \cdot)$ and accumulated reward $y(\cdot, \cdot)$ from Section 2.

Definition 6.1. For $M = (S, R, p, L, \rho)$ and a path $\omega \in \text{Path}$, the satisfaction relation \models is defined as follows:

$$\begin{aligned} \omega \models X_J^I\Phi & \quad \text{iff } \omega(1) \text{ is defined and} \\ & \quad \omega(1) \models \Phi \wedge \delta(\omega, 0) \in I \\ & \quad \wedge y(\omega, \delta(\omega, 0)) \in J \\ \omega \models \Phi_1 U_J^I \Phi_2 & \quad \text{iff } \exists t \in I. \omega @ t \models \Phi_2 \text{ and} \\ & \quad (\forall t' \in [0, t). \omega @ t' \models \Phi_1) \wedge y(\omega, t) \in J. \end{aligned}$$

We also consider the operators $\mathcal{E}_J(\Phi)$, $\mathcal{E}_J^t(\Phi)$, and $\mathcal{C}_J^I(\Phi)$ of the extension of CSRL introduced in Baier et al. [7], which we refer to as CSRL $_{\mathcal{E},\mathcal{C}}$. For an MRM M with the initial distribution p , time t , and set of states S' , we consider the instantaneous reward $\rho^M(p, S', t) = \sum_{s \in S'} \pi^M(p, s, t) \cdot \rho(s)$ and the long run reward rate $\rho^M(p, S') = \sum_{s \in S'} \pi^M(p, s) \cdot \rho(s)$. Although the semantics of these operators can be defined on states (see [7]), we instead define their semantics on MRMs. Before doing so, we rewrite Definition 5.3 to reason about classes of CSRL $_{\mathcal{E},\mathcal{C}}$ formulae by considering $\mathcal{Q} \in \{\mathcal{P}, \mathcal{S}, \mathcal{E}, \mathcal{C}\}$ rather than $\mathcal{Q} \in \{\mathcal{P}, \mathcal{S}\}$ in order to refer to, for example, \mathcal{E} -outermost or \mathcal{C} -nesting-free formulae. Then, as in the case of CTMCs, we

can reason about whether an MRM satisfies a \mathcal{P} , \mathcal{S} , \mathcal{E} , or \mathcal{C} -outermost CSRL $_{\mathcal{E},\mathcal{C}}$ formula by considering the initial distribution of a MRM. We omit the rules for $\mathcal{S}_{\bowtie\lambda}(\Phi)$ and $\mathcal{P}_{\bowtie\lambda}(\varphi)$ as they are the same as for CSL (Definition 5.2) with the Markov reward model M substituted for the CTMC C .

Definition 6.2. For $M = (S, R, p, L, \rho)$ and CSRL $_{\mathcal{E},\mathcal{C}}$ formula $\mathcal{E}_J(\Phi)$, $\mathcal{E}_J^t(\Phi)$, or $\mathcal{C}_J^I(\Phi)$, the satisfaction relation \models is defined as follows:

$$\begin{aligned} M \models \mathcal{E}_J(\Phi) & \quad \text{iff } \rho^M(p, \text{Sat}(\Phi)) \in J \\ M \models \mathcal{E}_J^t(\Phi) & \quad \text{iff } \rho^M(p, \text{Sat}(\Phi), t) \in J \\ M \models \mathcal{C}_J^I(\Phi) & \quad \text{iff } \int_I \rho^M(p, \text{Sat}(\Phi), u) du \in J. \end{aligned}$$

6.2 Stochastic Bisimulation and CSRL

6.2.1 Equivalences with the State-Labeling and Reward Conditions

We commence this section by summarizing results concerning forward stochastic bisimulation and the preservation of CSRL $_{\mathcal{E},\mathcal{C}}$ properties. Let $M = (S, R, p, L, \rho)$ be an MRM and Φ be a CSRL $_{\mathcal{E},\mathcal{C}}$ formula. From [8], we have that, for each pair $s, s' \in S$ of states, if $s \cong_{\text{LR}}^f s'$, then $s \models \Phi$ if and only if $s' \models \Phi$. This result implies that we can characterize satisfaction sets for all CSRL $_{\mathcal{E},\mathcal{C}}$ formula using a quotient MRM obtained by the equivalence \cong_{LR}^f . It follows that, if $s \cong_{\text{LR}}^f s'$, then we also have $s \models \Phi$ if and only if $s' \models \Phi$. Furthermore, if Φ is a $\mathcal{P}, \mathcal{S}, \mathcal{E}, \mathcal{C}$ -outermost CSRL $_{\mathcal{E},\mathcal{C}}$ formula, then $M \models \Phi$ if and only if $M / \cong_{\text{LR}}^f \models \Phi$.

As stated by the following theorem, backward stochastic bisimulation can be used to define a quotient CTMC on which $\mathcal{P}, \mathcal{S}, \mathcal{E}$, or \mathcal{C} -outermost, $\mathcal{P}, \mathcal{S}, \mathcal{E}$, and \mathcal{C} -nesting-free CSRL $_{\mathcal{E},\mathcal{C}}$ formulae can be verified, in the same manner as in Theorem 5.1.

Theorem 6.1. Let $M = (S, R, p, L, \rho)$ be an MRM and Φ be a $\mathcal{P}, \mathcal{S}, \mathcal{E}$, or \mathcal{C} -outermost, $\mathcal{P}, \mathcal{S}, \mathcal{E}$, and \mathcal{C} -nesting-free CSRL $_{\mathcal{E},\mathcal{C}}$ formula. Then, $M \models \Phi$ if and only if $M / \cong_{\text{LR}}^b \models \Phi$.

The proof is similar to that of Theorem 5.1, with extra bookwork to handle the introduction of rewards.

6.2.2 Removing the State-Labeling or Reward Condition for Basic \mathcal{E}, \mathcal{C} Formulae

Analogously to the case of CTMCs and CSL, we can obtain a new model-checking algorithm, not only for an outermost \mathcal{S} operator of CSRL, but also for $\mathcal{P}, \mathcal{S}, \mathcal{E}$, and \mathcal{C} -nesting-free formulae of the form $\mathcal{E}_J(\Psi)$, $\mathcal{E}_J^t(\Psi)$, and $\mathcal{C}_J^I(\Psi)$. A feature of the algorithm is that we can dispense with either the reward condition or the state-labeling condition (but not both) when constructing the quotient CTMC using backward stochastic bisimulation.

First, consider the case in which the reward condition is satisfied by the equivalence, but the state-labeling condition is not, that is, we consider the equivalence \cong_{LR}^b . Consider formulae of the form $\mathcal{E}_J(\Psi)$. Note that, from the reward condition and by the construction of the quotient MRM, the reward function $\rho / \cong_{\text{LR}}^b$ of the quotient MRM is such that $\rho / \cong_{\text{LR}}^b(C) = \rho(s)$ for all $s \in C$, for each $C \in S / \cong_{\text{LR}}^b$. Then, as in the case of the steady-state operator in Section 5.2.2, we can obtain, for each class C of \cong_{LR}^b , the proportion of the

number of states of C that satisfy Ψ to the total number of states in C . Multiplying this by the reward rate for the class then gives the total long run reward when in the Ψ -satisfying states of the class. We repeat this process for all classes, then add the results, to obtain the required long-run reward rate for the unreduced CTMC with respect to the states that satisfy Ψ . Formulae of the form $\mathcal{E}_j^t(\Psi)$ and $\mathcal{C}_j^I(\Psi)$ can be treated in the same way, using Theorem 3.1 to consider transient probabilities $\pi^M(p, s, t)$ rather than steady-state probabilities.

Second, we consider the case in which the state-labeling condition is satisfied by the equivalence in question, but the reward condition is not, that is, the equivalence is \cong_{LL}^b . Recall that the formulae $\mathcal{E}_j(\Psi)$, $\mathcal{E}_j^t(\Psi)$, and $\mathcal{C}_j^I(\Psi)$ that we are considering are \mathcal{P} , \mathcal{S} , \mathcal{E} , and \mathcal{C} -nesting-free and, therefore, Ψ is a formulae of the syntax $\Psi ::= a \mid \Psi \wedge \Psi \mid \neg\Psi$, where $a \in AP$. From the state-labeling condition, it follows that, for each class C of \cong_{LL}^b , either $C \subseteq \text{Sat}(\Psi)$ or $C \cap \text{Sat}(\Psi) = \emptyset$. In the case of $\mathcal{E}_j(\Psi)$, from Theorem 3.1 and the fact that $\rho/\cong_{\text{LL}}^b(C)$ is obtained by taking the average of the rewards in the constituent states of C , we know that the total long-run reward rate for all states in the class can be obtained by computing the long-run reward rate for the class in the quotient CTMC. Repeating this for all classes which satisfy Ψ , then summing the result, gives us the long-run reward rate for the unreduced CTMC with respect to the states that satisfy Ψ , which can be used when verifying formulae of the form $\mathcal{E}_j(\Psi)$. The method above can be adapted to $\mathcal{E}_j^t(\Psi)$ or $\mathcal{C}_j^I(\Psi)$ by considering transient probabilities, as in the previous paragraph.

Remark. Recall that the ‘‘duality’’ method presented in [7] can be used to convert an MRM into a CTMC, and a CSRL formula with trivial time bounds of $[0, \infty)$ into a CSL formula. Hence, after conversion from an MRM to a CTMC, we are able to apply stochastic bisimulation relations (forward or backward) which do not refer explicitly to rewards (only implicitly as they are encoded in the rates of the resulting CTMC).

7 SELECTIVE APPLICATION OF EQUIVALENCES

In Section 5.2, we saw that backward stochastic bisimulation can only be used in the presence of \mathcal{P} or \mathcal{S} -outermost, \mathcal{P} and \mathcal{S} -nesting-free CSL properties. In those sections, we considered the application of a single equivalence relation to all states of a CTMC; however, we now consider applying *different* equivalence relations to different parts of the state space. Our aim is to apply backward stochastic bisimulation *without* the state-labeling condition as much as possible in the state space, in order to offer a potentially more efficient alternative to forward stochastic bisimulation *with* the state-labeling condition. The approach is based on the observation that the bottom strongly connected components of a CTMC can be treated in a different manner from the rest of the states of the CTMC.

Consider the graph G of the CTMC $C = (S, R, p, L)$, defined such that the vertices are the states of C , and where there is an edge from s to s' if and only if $R(s, s') > 0$. A subgraph B of G is a *bottom strongly connected component* (BSCC) if it is a maximal strongly connected component

such that edges from states within B always point at vertices which are also within B . Let \mathcal{B} be the set of BSCCs of C . Note that a strongly connected CTMC is an ergodic CTMC. When clear from the context, we also use B to denote to the set of states of the BSCC B .

Let $S' \subseteq S$ be a subset of states of C . Then, $C[S'] = (S, R', p, L)$ is the CTMC obtained from C by letting $R'(s', s) = 0$ for each $s' \in S'$, $s \in S$ (that is, all states in S' are made absorbing). The following lemma asserts that entry states of a BSCC may be turned into absorbing states without altering the satisfaction sets of a \mathcal{P} -until formula in three cases: 1) if the second argument of the until formula is not satisfied in all states of the BSCC, 2) if the arguments of the until formula are both satisfied in all states of the BSCC, and 3) if the second argument of the until formula is satisfied in all states of the BSCC and the lower bound of the formula’s time interval is 0. We write \models_C for the CSL satisfaction relation \models interpreted on C .

Lemma 7.1. *Let B be a BSCC of the CTMC C and let $\mathcal{P}_{\triangleright\lambda}(\Phi_1 U^I \Phi_2)$ be a CSL formula. Then, for each state $s \in S \setminus B$, if at least one of the following conditions hold:*

1. $s' \not\models_C \Phi_2$ for all states $s' \in B$,
2. $s' \models_C \Phi_1 \wedge \Phi_2$ for all states $s' \in B$, or
3. $s' \models_C \Phi_2$ for all states $s' \in B$, and $\inf I = 0$,

then we have $s \models_C \mathcal{P}_{\triangleright\lambda}(\Phi_1 U^I \Phi_2)$ if and only if $s \models_{C[B]} \mathcal{P}_{\triangleright\lambda}(\Phi_1 U^I \Phi_2)$.

For example, consider the formula $\mathcal{P}_{\leq 0.01}(aU^{[10,20]}b)$: If the CTMC enters a BSCC in which b is not true for any of its states, then b will never become true in the future. Hence, all states in the BSCC can be made absorbing without affecting the probability of satisfying the until formula. Similarly, if the CTMC enters a BSCC in which a and b are both always true, then the until formula must be satisfied in the BSCC. Finally, in the case of the formula $\mathcal{P}_{\leq 0.01}(aU^{[0,15]}b)$, if a BSCC in which b is always true is entered and the elapsed time is less than 15 on entry (which can be evaluated on the state space outside of the BSCC), then the until formula will be satisfied. We note that related reductions were introduced by Baier et al. [9] (although not in the context of BSCCs).

The conversion to absorbing states described by Lemma 7.1 requires a step to find the BSCCs of the CTMC in question (which has time complexity $O(|S| + |R|)$ [20]), followed by a check of whether Φ_1 and/or Φ_2 hold in the states of the BSCCs (which takes $O(|S|)$ time).

We consider how these reductions can be used in the context of forward and backward stochastic bisimulation, for a subset of CSL properties. In order to simplify the following explanation, we consider formulae of the form $\mathcal{P}_{\triangleright\lambda}(\diamond^I \mathcal{S}_{\triangleright\lambda}(\Psi))$ and $\mathcal{P}_{\triangleright\lambda}(\Psi' U^{[0,t]} \mathcal{S}_{\triangleright\lambda}(\Psi))$, where Ψ, Ψ' are formulae of the syntax $\Psi ::= a \mid \Psi \wedge \Psi \mid \neg\Psi$ and where we use $\diamond^I \Phi$ to abbreviate $\text{true} U^I \Phi$. From the semantics of the steady-state operator, we have the following fact: For any BSCC B , either $s \models \mathcal{S}_{\triangleright\lambda}(\Psi)$ for all states $s \in B$ or $s \not\models \mathcal{S}_{\triangleright\lambda}(\Psi)$ for all states $s \in B$ (see [4]). Hence, in the case of $\mathcal{P}_{\triangleright\lambda}(\diamond^I \mathcal{S}_{\triangleright\lambda}(\Psi))$ and $\mathcal{P}_{\triangleright\lambda}(\Psi' U^{[0,t]} \mathcal{S}_{\triangleright\lambda}(\Psi))$, we know that all BSCCs of a CTMC satisfy at least one of the conditions of Lemma 7.1: Either $\mathcal{S}_{\triangleright\lambda}(\Psi)$ is not satisfied within a BSCC, in which case condition 1) applies, or $\mathcal{S}_{\triangleright\lambda}(\Psi)$ is satisfied

throughout a BSCC, in which case conditions 2) and 3) apply in the case of $\mathcal{P}_{\bowtie \lambda'}(\diamond^I \mathcal{S}_{\bowtie \lambda}(\Psi))$ and $\mathcal{P}_{\bowtie \lambda'}(\Psi'U^{[0,t]} \mathcal{S}_{\bowtie \lambda}(\Psi))$, respectively.

We now consider how the results of Section 5.2.2 can be used to define a model-checking algorithm for formulae such as $\mathcal{P}_{\bowtie \lambda'}(\diamond^I \mathcal{S}_{\bowtie \lambda}(\Psi))$ and $\mathcal{P}_{\bowtie \lambda'}(\Psi'U^{[0,t]} \mathcal{S}_{\bowtie \lambda}(\Psi))$, which exploits backward stochastic bisimulation. For the CTMC C , we consider each of the BSCCs of C in turn, with the aim of determining whether the formula $\mathcal{S}_{\bowtie \lambda}(\Psi)$ holds in the BSCC. For a BSCC $B \in \mathcal{B}$, consider the CTMC $C|_B$ obtained from C by restricting the components of C to states in B . As the resulting CTMC is strongly connected, steady-state probabilities are independent of the initial distribution; hence, we can write $\pi^B(s)$ to denote the steady-state probability of state s within B , instead of $\pi^{C|_B}(\alpha, s)$ (for some distribution α). We also write $Sat^B(\Psi)$ instead of $Sat^{C|_B}(\Psi)$ for simplicity.

Next, observe that the BSCC B is necessarily contained within some BSCC \tilde{B} of $C|_{B/\cong}$: For every finite path $s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_n$ in B , there exists a path $C_0 \rightarrow C_1 \rightarrow \dots \rightarrow C_n$ in \tilde{B} such that $s_i \in C_i$ for each $0 \leq i \leq n$. Note that the fact that we consider the equivalence \cong^b , and not the equivalence \cong_1^b , which depends on the initial condition, suffices because the initial condition is irrelevant to the steady-state distribution in B . Hence, taking such an (arbitrarily chosen) initial distribution into account when constructing the quotient CTMC of B will make no difference to the correspondence of results between the CTMC of B and the resulting quotient CTMC. Then, as in the case of Section 5.2.2, we can obtain the following result:

$$\sum_{s \in B \cap Sat^B(\Psi)} \pi^B(s) = \sum_{C \in \tilde{B}} \pi^{\tilde{B}}(C) \cdot \frac{|C \cap Sat^B(\Psi)|}{|C|}.$$

Hence, by applying backward stochastic bisimulation without the initial and state-labeling conditions to a BSCC, we can obtain $\sum_{s \in B \cap Sat^B(\Psi)} \pi^B(s)$. Then, if $(\sum_{s \in B \cap Sat^B(\Psi)} \pi^B(s)) \bowtie \lambda$, we know that all states within B satisfy $\mathcal{S}_{\bowtie \lambda}(\Psi)$; otherwise, all states within B do not satisfy $\mathcal{S}_{\bowtie \lambda}(\Psi)$. We can repeat this process for all the BSCCs of the CTMC.

It remains to consider whether $\mathcal{S}_{\bowtie \lambda}(\Psi)$ holds in the states of C which lie outside of the BSCCs. Observe the following fact, adapted from [9]:

$$\pi^C(\alpha_s, Sat^C(\Psi)) = \sum_{B \in \mathcal{B}} \left(ProbReach^C(s, B) \cdot \sum_{s' \in B \cap Sat^B(\Psi)} \pi^B(s') \right), \quad (1)$$

where $ProbReach^C(s, S') = Prob_s^C\{\omega \in Path^C \mid \exists i. \omega(i) \in S'\}$ for $S' \subseteq S$. We advocate constructing a *forward* stochastic bisimulation quotient from all of the states *outside* of the BSCCs. The following lemma uses the same reasoning used to show that forward stochastic bisimulation with the state-labeling condition preserves CSL path properties with trivial time bounds of $[0, \infty)$.

Lemma 7.2. *Let C be a CTMC and C/\cong_L^f be its quotient CTMC with respect to forward stochastic bisimulation and the state-labeling condition. Then, for any state $s \in S$, the*

class $C \in S/\cong_L^f$ for which $s \in C$, and an atomic proposition $a \in AP$, we have

$$ProbReach^C(s, Sat^C(a)) = ProbReach^{C/\cong_L^f}(C, Sat^{C/\cong_L^f}(a)).$$

We extend the set of atomic propositions AP with the atomic proposition at_B for each $B \in \mathcal{B}$ and extend the labeling of a state $s \in S$ with at_B if and only if $s \in B$. We also use Lemma 7.1 to make the states of each BSCC absorbing to obtain the CTMC $C[\mathcal{B}]$ (recall that at least one condition of this lemma must hold, given our restriction to the formulae $\mathcal{P}_{\bowtie \lambda'}(\diamond^I \mathcal{S}_{\bowtie \lambda}(\Psi))$ and $\mathcal{P}_{\bowtie \lambda'}(\Psi'U^{[0,t]} \mathcal{S}_{\bowtie \lambda}(\Psi))$). Next, we construct the quotient CTMC $\tilde{C} = C[\mathcal{B}]/\cong_L^f$ using the equivalence \cong_L^f , where the state-labeling condition takes into account the additional labels of the form at_B .

Then, to obtain $\pi^C(\alpha_s, Sat^C(\Psi))$ for $s \in S \setminus \mathcal{B}$, it suffices to use (1), using the values $\sum_{s' \in B \cap Sat^B(\Psi)} \pi^B(s')$ for each $B \in \mathcal{B}$ that we have already computed, but substituting $ProbReach^{\tilde{C}}(C, Sat^{\tilde{C}}(a))$ for $ProbReach^C(s, B)$, where $s \in C$, according to Lemma 7.2. Given the computed values of $\pi^C(\alpha_s, Sat^C(\Psi))$, we can then decide whether $s \models \mathcal{S}_{\bowtie \lambda}(\Psi)$ for each $s \in S$.

Hence, we have obtained a method establishing whether each state of the CTMC satisfies $\mathcal{S}_{\bowtie \lambda}(\Psi)$, first by using backward stochastic bisimulation on the states of the BSCCs, then using forward stochastic bisimulation on the remaining states. A further reduction in the size of the state space can be made by combining all of the states of a BSCC into a single state after they have been made absorbing, as done in [9]. Moreover, if two BSCCs are found to have the same value $\sum_{s \in B \cap Sat^B(\Psi)} \pi^B(s)$, then there is no need to introduce different atomic propositions for each of these BSCCs; this can have the effect of reducing the amount of subdivision that is necessary to distinguish BSCCs when computing the forward stochastic bisimulation relation. Furthermore, the forward stochastic bisimulation quotient CTMC may be reused to establish the states which satisfy the formulae $\mathcal{P}_{\bowtie \lambda'}(\diamond^I \mathcal{S}_{\bowtie \lambda}(\Psi))$ and $\mathcal{P}_{\bowtie \lambda'}(\Psi'U^{[0,t]} \mathcal{S}_{\bowtie \lambda}(\Psi))$: Hence, we can use the quotient \tilde{C} not only for obtaining the satisfaction set of $\mathcal{S}_{\bowtie \lambda}(\Psi)$, but also for resolving the \mathcal{P} part of the overall formula.

Although we concentrated our attention on two formulae, $\mathcal{P}_{\bowtie \lambda'}(\diamond^I \mathcal{S}_{\bowtie \lambda}(\Psi))$ and $\mathcal{P}_{\bowtie \lambda'}(\Psi'U^{[0,t]} \mathcal{S}_{\bowtie \lambda}(\Psi))$ for simplicity, the method of this section can be applied to more general formulae. For example, the method can be adapted in a straightforward way to $\mathcal{P}_{\bowtie \lambda'}(\diamond^I (a \wedge \mathcal{S}_{\bowtie \lambda}(\Psi)))$ if the atomic proposition a either holds in all states of a BSCC or it does not hold in all of the BSCC's states. Similarly, the method could be used to verify properties of the form $\mathcal{P}_{\bowtie \lambda'}(\mathcal{S}_{\bowtie \lambda'}(\Psi)U^I \mathcal{S}_{\bowtie \lambda}(\Psi))$.

Furthermore, we can construct the backward stochastic bisimulation quotient only for *some* BSCCs and apply forward stochastic bisimulation with the state-labeling condition to others. This can be useful if some BSCCs satisfy the conditions of Lemma 2 and others do not.

A disadvantage of the approach of this section is that it can only be applied after the CTMC has been constructed and, therefore, cannot be used in, for example, a compositional manner on system subcomponents.

8 BACKWARD STOCHASTIC BISIMULATION IN PRACTICE

In this section, we consider how backward stochastic bisimulation may be applied in practice to system models represented as CTMCs. Our approach was to implement algorithms for computing the forward and backward stochastic bisimulation quotients (using the “splitting” principle of Kanellakis and Smolka [32]) in the open-source probabilistic model-checking tool PRISM [31]. We also extended these algorithms in order to compute quotients of equivalences satisfying the initial and state-labeling conditions.

We applied the implementation principally to two models: a multiserver polling system [1] and the multiprocessor system of Buchholz [14]. The multiserver polling system consists of a number of identical stations at which customers arrive and a number of servers which visit the stations cyclically. When a server is at a station, it can provide service to the station’s customers, who then exit from the system. The polling system has a number of parameters which affect the size of the underlying CTMC: the number of stations, the capacity of the queue of customers at each station, and the number of servers. Note that the basic system, without taking state labeling into account, exhibits symmetry: The characteristics of each of the stations are the same. This means that states which are identical up to the permutation of the identities of the stations are *both* forward *and* backward stochastic bisimilar (without the state-labeling and initial conditions) and, hence, the quotient CTMCs resulting from each of these two relations are identical. Indeed, application of the splitting algorithms to different configurations (in terms of number of stations, the capacity of the stations, and the number of servers) of the polling system resulted in a reduction of the size of the state space roughly proportional to the number of stations. For example, the size of the state space of the configuration with two servers and four stations, each with a capacity of two customers, was reduced from 1,998 to 507, whereas the size of the state space of the configuration with a single server and 10 stations, each with a capacity of one customer, was reduced from 15,360 to 1,536.

Now, assume that we wish to verify a \mathcal{S} -outermost, \mathcal{S} , \mathcal{P} -nesting free formula $\mathcal{S}_{\triangleright\triangleleft\lambda}(\Psi)$, where Ψ (which is of the syntax $\Psi ::= a \mid \Psi \wedge \Psi \mid \neg\Psi$) distinguishes between stations. Examples of formulae such as Ψ could be the atomic proposition $station1_queue > 0$, which indicates that there is at least one customer in the queue of station 1, or $station2_service = 0 \wedge station3_service = 0$, which indicates that stations 2 and 3 are not currently being served. In such cases, backward stochastic bisimulation can offer an advantage over forward stochastic bisimulation: To verify $\mathcal{S}_{\triangleright\triangleleft\lambda}(\Psi)$, forward stochastic bisimulation must be employed in conjunction with the state-labeling condition, whereas, following the results of Section 5.2.2, backward stochastic bisimulation may be employed alone (we do need to take the initial condition into account because the system forms a single BSCC). Indeed, in the case of formulae Ψ which identify the identity of the stations, our experiments established that incorporation of the state-labeling condition results in a

(forward or backward) stochastic bisimulation quotient which is as large as the original state space, thereby offering *no advantage* in the verification process.

We now consider the multiprocessor system example. The system consists of a series of “teams,” each of which is comprised of a number of processors and a number of memory units, and a designated number of additional memory units which can be shared across the teams. All of the components of the system, whether processor or memory unit, fail with a certain rate. When all memory units of a given team have failed, the active processors of the team can use the shared memory units. The overall system is said to fail when either all processors of a certain team have failed or when all memory units available to a certain team (that is, the memory units that belong to the team and the shared memory units) have failed. After system failure, a repair unit is deployed which returns the system to its original state in which all components are operational after a certain amount of time has elapsed. This multiprocessor example was considered in [14] in the case of one shared memory unit and two teams, each comprised of one processor and one memory unit.

The results of applying the splitting algorithms for forward and backward stochastic bisimulation to this example are illustrated in Table 1. For simplicity, we consider the case in which team sizes are always equal, in which the number of processors and memory units within a team is equal, and in which the number of shared memory units is equal to the chosen team size. We also assume that all memory units fail with the same rate and that all processors fail with the same rate, but that the failure rate for memory units and processors differ. The first and second columns of Table 1 are self-explanatory and the third column includes the size of the state space before reduction. The fourth and fifth columns describe the size of the quotient state space, together with the proportion of the size of the quotient state space to that of the original state space, for forward and backward stochastic bisimulation, respectively. It can be observed that the state-space reductions obtained by backward stochastic bisimulation are greater than those for forward stochastic bisimulation. The reason for this difference is that, in contrast to the backward case, forward stochastic bisimulation distinguishes between the failure of memory units in teams and the failure of shared memory units.

Incorporating the state-labeling condition into the equivalence can result in quotient state spaces which are strictly larger than those obtained without this condition, but, in contrast to the symmetry-breaking propositions in the polling example, are smaller than the original state space. For example, in the context of the system configuration with two shared memory units and two teams of size two, taking into account a labeling which distinguishes between those states in which all memory units of team one is down from those in which at least one is up results in a forward quotient of size 177 (instead of 91) and a backward quotient of size 89 (instead of 56). Finally, we note that, for the multiprocessor system, the results of taking the initial condition into account are the same as those when this

TABLE 1
State-Space Reductions of Forward and Backward Stochastic Bisimulation for the Multiprocessor System

Number of teams	Team size	Original	Forward quotient	Backward quotient
2	1	20	10 (50%)	7 (35%)
2	2	201	91 (45.27%)	56 (27.86%)
2	3	928	415 (44.72%)	214 (23.06%)
2	4	2945	1321 (44.86%)	585 (19.86%)
3	1	100	28 (28%)	15 (15%)
3	2	2059	438 (21.38%)	179 (8.74%)
3	3	15966	3078 (19.28%)	900 (5.64%)
4	1	430	53 (12.33%)	24 (5.58%)
4	2	19030	1339 (7.04%)	404 (2.12%)
4	3	259504	14619 (5.63%)	2658 (1.02%)
5	1	1805	114 (6.32%)	55 (3.05%)
5	2	174027	6901 (3.97%)	1477 (0.85%)

condition is not considered (the initial state is always the unique member of a class).

We also applied our algorithm to a number of CTMC models taken from the PRISM Web page.¹ The only examples in which the reduction obtained from forward stochastic bisimulation was at least an order of magnitude better than the backward case were one of the dynamic power management examples and the embedded control system in which the state space was reduced from 3,478 states to 98 classes (compared to 2,910 classes in the case of backward stochastic bisimulation).

9 CONCLUSIONS

We have presented a study of backward stochastic bisimulation in the context of CSL model checking. It has been shown that the choice of the application of backward or forward equivalences is formula-dependent, in the sense that backward stochastic bisimulation cannot be applied to CSL formulae with arbitrary nesting. However, the degree of nesting in practical examples of CSL formulae that have been given in the literature thus far is usually limited; this leads us to believe that the methods in Section 5 and Section 7 are often applicable to a significant class of properties. In Section 8, we showed that backward stochastic bisimulation can outperform forward stochastic bisimulation, particularly in the context of the class of steady-state properties for which the state-labeling condition does not need to be considered. Our work has some parallels with that of [28] in the nonstochastic context, in which it was shown that branching-time temporal logic properties (with nesting of “path quantifiers”) are not preserved by backward bisimulation, but linear-time properties are preserved. We note that the results of this paper can also be applied in the case of discrete-time Markov chains and the temporal logic PCTL [27].

Although, in this paper, we have worked at the CTMC level, the lumpability and stochastic bisimulations considered have also been used successfully for directly producing a quotient CTMC starting from a high-level description of the

stochastic system, as is the case, for example, in PEPA [26] and in the symbolic reachability graph (SRG) construction of Stochastic Well-formed Nets (SWN) [15]. The SRG construction is based on exploiting model symmetries, which subsume both strong *and* exact lumpability (forward *and* backward bisimulation). More recent work on “extended SRGs” is based on a strong (forward) equivalence; as observed in [16], an exact (backward) equivalence could be used instead, if it leads to a smaller number of equivalence classes. However, as in our work, in the current state of research it is not possible to foresee which one of the two works better for a given SWN model.

APPENDIX

PROOF OF LEMMA 5.1

As in the proof of Theorem 5.1, we write the notation pertaining to the quotient $C/\overset{\sim}{\cong}_{\text{ll}}$ with the overline \sim (for example, we write \tilde{C} , \tilde{S} , \tilde{R} instead of $C/\overset{\sim}{\cong}_{\text{ll}}$, $S/\overset{\sim}{\cong}_{\text{ll}}$, and \tilde{R} , respectively). Recall that $Sat^C(\Phi)$ denotes the set of states of the CTMC C satisfying the CSL formula Φ , whereas $Sat^{\tilde{C}}(\Phi)$ denotes the set of states of the quotient satisfying Φ . Furthermore, we let $P(s, s') = \frac{R(s, s')}{E(s)}$ for each $s, s' \in S$ and let $\tilde{P}(C, C') = \frac{R(C, C')}{E(C)}$ for each $C, C' \in \tilde{S}$.

Before proceeding to the proof of Lemma 1, we recall two lemmata from [9].

Lemma A.1 [9]. *Let $s \in S$, $I \subseteq \mathbb{R}_{\geq 0}$ and Φ be a CSL formula.*

Then,

$$Prob_s^C(X^I \Phi) = (e^{-E(s) \cdot \inf I} - e^{-E(s) \cdot \sup I}) \cdot \sum_{s' \models \Phi} P(s, s').$$

For any pair of states $s, s' \in S$ and real value $x \in \mathbb{R}_{> 0}$, let $T(s, s', x) = R(s, s') \cdot e^{-E(s) \cdot x}$. Let $I \ominus x$ denote $\{t - x \mid t \in I \text{ and } t \geq x\}$.

Lemma A.2 [9]. *Let $s \in S$ and Φ_1, Φ_2 be CSL formulae. The function $\vartheta : S \rightarrow [0, 1]$, $\vartheta(s) = Prob_s^C(\Phi_1 U^I \Phi_2)$ is the least fixed point of the operator $F : (S \rightarrow [0, 1]) \rightarrow (S \rightarrow [0, 1])$, where:*

1. <http://www.cs.bham.ac.uk/~dpx/prism/>.

$$F(f)(s, I) = \begin{cases} 1 & \text{if } s \models \neg\Phi_1 \wedge \Phi_2 \text{ and} \\ & a = 0 \\ \int_0^b \sum_{s' \in S} T(s, s', x) \cdot f(s', I \ominus x) dx & \text{if } s \models \Phi_1 \wedge \neg\Phi_2 \\ e^{-E(s) \cdot a} + \int_0^a \sum_{s' \in S} T(s, s', x) \\ \cdot f(s', I \ominus x) dx & \text{if } s \models \Phi_1 \wedge \Phi_2 \\ 0 & \text{otherwise.} \end{cases}$$

Naturally, these lemmata can also be applied to the quotient CTMC \tilde{C} of C . For example, in the case of Lemma A.1, for each class $C \in \tilde{S}$, we have

$$Prob_C^{\tilde{C}}(X^I \Phi) = (e^{-\tilde{E}(C) \cdot \inf I} - e^{-\tilde{E}(C) \cdot \sup I}) \cdot \sum_{C' \models \Phi} \tilde{P}(C, C').$$

We now proceed to the proof of Lemma 5.1.

Proof. Let Φ_1, Φ_2 be CSL formulae. Recall that the statement of Lemma 5.1 makes the assumption that, for each pair of states $s, s' \in S$, we have $s \cong_{\text{IL}}^b s'$ implies $s \models \Phi_1$ if and only if $s' \models \Phi_1$, and $s \models \Phi_2$ if and only if $s' \models \Phi_2$. Let φ be either $X^I \Phi_1$ or $\Phi_1 U^I \Phi_2$. Recall that our aim is to prove that, for each class $C \in S / \cong_{\text{IL}}^b$:

$$Prob_C^{\tilde{C}}(\varphi) = \frac{\sum_{s \in C} Prob_s^C(\varphi)}{|C|}.$$

We divide the proof into two cases, depending on whether φ is $X^I \Phi_1$ or whether φ is $\Phi_1 U^I \Phi_2$.

Case: $\varphi = X^I \Phi_1$. We show that $Prob_C^{\tilde{C}}(X^I \Phi_1) = \frac{\sum_{s \in C} Prob_s^C(X^I \Phi_1)}{|C|}$ by the following derivation:

$$\begin{aligned} & Prob_C^{\tilde{C}}(X^I \Phi_1) \\ &= (e^{-\tilde{E}(C) \cdot \inf I} - e^{-\tilde{E}(C) \cdot \sup I}) \cdot \sum_{C' \models \Phi_1} \tilde{P}(C, C') \quad (\text{by Lemma A.1}) \\ &= (e^{-\tilde{E}(C) \cdot \inf I} - e^{-\tilde{E}(C) \cdot \sup I}) \cdot \sum_{C' \models \Phi_1} \frac{\tilde{R}(C, C')}{\tilde{E}(C)} \\ &= (e^{-\tilde{E}(C) \cdot \inf I} - e^{-\tilde{E}(C) \cdot \sup I}) \cdot \sum_{C' \models \Phi_1} \sum_{s \in C} \sum_{s'' \in C'} \frac{R(s, s'')}{|C| \cdot \tilde{E}(C)} \\ &= \frac{1}{|C|} \cdot (e^{-\tilde{E}(C) \cdot \inf I} - e^{-\tilde{E}(C) \cdot \sup I}) \cdot \sum_{C' \models \Phi_1} \sum_{s \in C} \sum_{s'' \in C'} \frac{R(s, s'')}{\tilde{E}(C)} \\ &= \frac{1}{|C|} \cdot \sum_{s \in C} (e^{-\tilde{E}(C) \cdot \inf I} - e^{-\tilde{E}(C) \cdot \sup I}) \cdot \sum_{C' \models \Phi_1} \sum_{s'' \in C'} \frac{R(s, s'')}{\tilde{E}(C)} \\ &= \frac{1}{|C|} \cdot \sum_{s \in C} (e^{-E(s) \cdot \inf I} - e^{-E(s) \cdot \sup I}) \cdot \sum_{C' \models \Phi_1} \sum_{s'' \in C'} \frac{R(s, s'')}{E(s)} \\ &\quad (\text{because } \tilde{E}(C) = E(s) \forall s \in C) \\ &= \frac{1}{|C|} \cdot \sum_{s \in C} (e^{-E(s) \cdot \inf I} - e^{-E(s) \cdot \sup I}) \cdot \sum_{s'' \models \Phi_1} \frac{R(s, s'')}{E(s)} \\ & (\text{because } \{C' \mid C' \in \text{Sat}^{\tilde{C}}(\Phi_1)\} \text{ is a partition of } \text{Sat}^{\tilde{C}}(\Phi_1)) \\ &= \frac{1}{|C|} \cdot \sum_{s \in C} (e^{-E(s) \cdot \inf I} - e^{-E(s) \cdot \sup I}) \cdot \sum_{s'' \models \Phi_1} P(s, s'') \\ &= \sum_{s \in C} \frac{Prob_s^C(X^I \Phi_1)}{|C|} \quad (\text{by Lemma A.1}). \end{aligned}$$

Case: $\varphi = \Phi_1 U^I \Phi_2$. Let f_0, f_1, f_2, \dots be the sequence of functions computed to obtain the least-fixpoint function ϑ in Lemma A.2 for C, I, Φ_1 , and Φ_2 , and let $\tilde{f}_0, \tilde{f}_1, \tilde{f}_2, \dots$ be the corresponding sequence of functions computed to obtain the least-fixpoint $\tilde{\vartheta}$ for \tilde{C}, I, Φ_1 , and Φ_2 . Then, by Lemma A.2, our task reduces to proving that, for any interval $J \subseteq \mathbb{R}_{\geq 0}$, for any class $C \in \tilde{S}$, and, for each $i \geq 0$, we have:

$$\tilde{f}_i(C, J) = \frac{\sum_{s \in C} f_i(s, J)}{|C|}.$$

We proceed by induction on $i \geq 0$. Because we are computing the least-fixpoint, we have $\tilde{f}_0(C, J) = 0$ and $f_0(s, J) = 0$ for all intervals $J \subseteq \mathbb{R}_{\geq 0}$, all classes $C \in \tilde{S}$, and all states $s \in S$, and, hence,

$$\tilde{f}_0(C, J) = 0 = \left(\sum_{s \in C} f_0(s, J) \right) / |C|.$$

Now, we consider $i \geq 1$ and assume that we have established $\tilde{f}_j(C, J) = (\sum_{s \in C} f_j(s, J)) / |C|$ for all $j < i$. We consider an arbitrary class $C \in \tilde{S}$ and split the cases of the proof according to the conditions on satisfaction in Lemma A.2.

Subcase 1. $a = 0$ and $C \models \Phi_2$. By the definition of \cong_{IL}^b , we have $s \models \Phi_2$ for each $s \in C$. Furthermore, by the definition of \tilde{f}_i , we have $\tilde{f}_i(C, J) = 1$ and, by the definition of f_i , we have $f_i(s, J) = 1$ for each $s \in C$. Therefore:

$$\frac{\sum_{s \in C} f_i(s, J)}{|C|} = \frac{|C|}{|C|} = 1 = \tilde{f}_i(C, J).$$

Subcase 2. $C \models \Phi_1 \wedge \neg\Phi_2$.

$$\begin{aligned} & \tilde{f}_i(C, J) \\ &= \int_0^b \sum_{C' \in \tilde{S}} \tilde{T}(C, C', x) \cdot \tilde{f}_{i-1}(C', J \ominus x) dx \\ &= \int_0^b \sum_{C' \in \tilde{S}} \tilde{R}(C, C') \cdot e^{-\tilde{E}(C) \cdot x} \cdot \tilde{f}_{i-1}(C', J \ominus x) dx \\ &= \int_0^b \sum_{C' \in \tilde{S}} \tilde{R}(C, C') \cdot e^{-\tilde{E}(C) \cdot x} \cdot \left(\frac{\sum_{s' \in C'} f_{i-1}(s', J \ominus x)}{|C'|} \right) dx \\ &= \int_0^b \sum_{C' \in \tilde{S}} \left(\frac{\sum_{s \in C} \sum_{s'' \in C'} R(s, s'')}{|C|} \right) \cdot e^{-\tilde{E}(C) \cdot x} \\ & \quad \cdot \left(\frac{\sum_{s' \in C'} f_{i-1}(s', J \ominus x)}{|C'|} \right) dx \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{C' \in \tilde{S}} \sum_{s \in C'} \sum_{s'' \in C'} R(s, s'') \cdot \\
&\quad \left(\frac{\sum_{s' \in C'} f_{i-1}(s', J \oplus x)}{|C'|} \right) dx \\
&= \frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{C' \in \tilde{S}} \sum_{s' \in C'} R(C, s'') \cdot \\
&\quad \left(\frac{\sum_{s' \in C'} f_{i-1}(s', J \oplus x)}{|C'|} \right) dx \\
&= \frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{C' \in \tilde{S}} |C'| \cdot R(C, s_{C'}) \cdot \\
&\quad \left(\frac{\sum_{s' \in C'} f_{i-1}(s', J \oplus x)}{|C'|} \right) dx
\end{aligned}$$

for some arbitrary $s_{C'} \in C'$. Observe that we can write that $\sum_{s'' \in C'} R(C, s'') = |C'| \cdot R(C, s_{C'})$ because it follows from the definition of \cong_{IL}^b that $R(C, s'') = R(C, s_{C'})$ for all $s'' \in C'$.

Cancelling the factors of $|C'|$, we then obtain:

$$\begin{aligned}
&\frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{C' \in \tilde{S}} R(C, s_{C'}) \cdot \sum_{s' \in C'} f_{i-1}(s', I \oplus x) dx \\
&= \frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{C' \in \tilde{S}} \sum_{s' \in C'} R(C, s_{C'}) \cdot f_{i-1}(s', J \oplus x) dx \\
&= \frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{C' \in \tilde{S}} \sum_{s' \in C'} R(C, s') \cdot f_{i-1}(s', J \oplus x) dx,
\end{aligned}$$

the last step again following from the fact that $R(C, s') = R(C, s_{C'})$ for all $s' \in C'$. Then, because $\{C' \mid C' \in \tilde{S}\}$ is a partition of S , we have:

$$\begin{aligned}
&\frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{C' \in \tilde{S}} \sum_{s' \in C'} R(C, s') \cdot f_{i-1}(s', J \oplus x) dx \\
&= \frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{s' \in S} R(C, s') \cdot f_{i-1}(s', J \oplus x) dx \\
&= \frac{1}{|C|} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{s' \in S} \sum_{s \in C} R(s, s') \cdot f_{i-1}(s', J \oplus x) dx \\
&= \frac{1}{|C|} \sum_{s \in C} \int_0^b e^{-\tilde{E}(C) \cdot x} \cdot \sum_{s' \in S} R(s, s') \cdot f_{i-1}(s', J \oplus x) dx \\
&= \frac{1}{|C|} \sum_{s \in C} \int_0^b e^{-E(s) \cdot x} \cdot \sum_{s' \in S} R(s, s') \cdot f_{i-1}(s', J \oplus x) dx,
\end{aligned}$$

with the final step relying on the fact that $\tilde{E}(C) = E(s)$ for any $s \in C$. To establish this fact, observe that, from the definition of \tilde{R} , we can derive that $\tilde{E}(C) = \frac{1}{|C|} \sum_{s \in C} E(s)$. Let s_C be some state in C . Then, from the definition of \cong_{IL}^b , we have that $E(s) = E(s_C)$ for all $s \in C$ and, therefore, $\tilde{E}(C) = \frac{1}{|C|} \sum_{s \in C} E(s) = \frac{1}{|C|} \cdot |C| \cdot E(s_C) = E(s_C)$. Hence, we can substitute $E(s)$ for $\tilde{E}(C)$ for all $s \in C$.

From the fact that

$$\begin{aligned}
&\frac{1}{|C|} \sum_{s \in C} \int_0^b e^{-E(s) \cdot x} \cdot \sum_{s' \in S} R(s, s') \cdot f_{i-1}(s', J \oplus x) dx \\
&= \frac{1}{|C|} \sum_{s \in C} f_i(s, J),
\end{aligned}$$

we have established that $\tilde{f}_i(C) = \frac{1}{|C|} \sum_{s \in C} f_i(s, J)$ for $i \geq 1$.

Subcase 3. $C \models \Phi_1 \wedge \Phi_2$. This case follows similarly from Subcase 2 and, therefore, we omit the details. The second operand of the summation is handled exactly as in Subcase 2. Therefore, to establish the result that $\tilde{f}_i(C, J) = \frac{1}{|C|} \sum_{s \in C} f_i(s, J)$, we need to show that $e^{-\tilde{E}(C) \cdot a} = \frac{1}{|C|} \sum_{s \in C} e^{-E(s) \cdot a}$. As we saw toward the end of the part of Subcase 2, we can substitute $\tilde{E}(C)$ for $E(s)$ for all $s \in C$. Hence, we obtain $\frac{1}{|C|} \sum_{s \in C} e^{-E(s) \cdot a} = \frac{1}{|C|} \cdot |C| \cdot e^{-\tilde{E}(C) \cdot a} = e^{-\tilde{E}(C) \cdot a}$, and we are done.

Subcase 4. $C \models \neg\Phi_1 \wedge \neg\Phi_2$. From the definition of \cong_{IL}^b , we have $s \models \neg\Phi_1 \wedge \neg\Phi_2$ for each $s \in C$. By the definition of \tilde{f}_i , we have $\tilde{f}_i(C, J) = 0$ and, by the definition of f_i , we have $f_i(s, J) = 0$ for each $s \in C$. Therefore,

$$\frac{\sum_{s \in C} f_i(s, J)}{|C|} = \frac{0}{|C|} = 0 = \tilde{f}_i(C)$$

and the required result is established for this fourth and final subcase.

By Lemma A.2, the fact that we have

$$\tilde{f}_i(C, I) = \frac{\sum_{s \in C} f_i(s, I)}{|C|}$$

for each $i \geq 0$ implies that

$$\text{Prob}_C^{\tilde{C}}(\Phi_1 U^I \Phi_2) = \frac{\sum_{s \in C} \text{Prob}_s^C(\Phi_1 U^I \Phi_2)}{|C|}.$$

As we have considered the two possible cases of φ , we have established Lemma 5.1. \square

ACKNOWLEDGMENTS

The authors would like to thank David Parker for help with the PRISM implementation of the algorithms for computing the forward and backward stochastic bisimulation quotients. This work was supported in part by the Italian project MIUR-FIRB Perf.

REFERENCES

- [1] M.A. Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modeling with Generalized Stochastic Petri Nets*. John Wiley and Sons, 1995.
- [2] A. Aziz, V. Singhal, F. Balarin, R.K. Brayton, and A.L. Sangiovanni-Vincentelli, "It Usually Works: The Temporal Logic of Stochastic Systems," *Proc. Seventh Int'l Conf. Computer Aided Verification (CAV '95)*, 1995.

- [3] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton, "Model-Checking Continuous Time Markov Chains," *ACM Trans. Computational Logic*, vol. 1, no. 1, pp. 162-170, 2000.
- [4] P. Ballarini, "Towards Compositional CSL Model Checking," PhD thesis, Dipartimento di Informatica, Univ. di Torino, 2004.
- [5] M.C. Browne, E.M. Clarke, and O. Grumberg, "Characterizing Finite Kripke Structures in Propositional Temporal Logic," *Theoretical Computer Science*, vol. 59, pp. 115-131, 1988.
- [6] M. Bernardo and R. Gorrieri, "A Tutorial on EMPA: A Theory of Concurrent Processes with Nondeterminism, Priorities, Probabilities and Time," *Theoretical Computer Science*, vol. 202, pp. 1-54, 1998.
- [7] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "On the Logical Characterisation of Performability Properties," *Proc. 12th Int'l Colloquium on Automata, Languages and Programming (ICALP '00)*, 2000.
- [8] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Automated Performance and Dependability Evaluation Using Model Checking," *Performance Evaluation of Complex Systems: Techniques and Tools*, 2002.
- [9] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Model-Checking Algorithms for Continuous-Time Markov Chains," *IEEE Trans. Software Eng.*, vol. 29, no. 6, pp. 524-541, June 2003.
- [10] C. Baier, H. Hermanns, J.-P. Katoen, and V. Wolf, "Comparative Branching-Time Semantics for Markov Chains," *Information and Computation*, vol. 200, no. 2, pp. 149-214, 2005.
- [11] J.T. Bradley, N.J. Dingle, P.G. Harrison, and W.J. Knottenbelt, "Performance Queries on Semi-Markov Stochastic Petri Nets with an Extended Continuous Stochastic Logic," *Proc. 10th Int'l Workshop Petri Nets and Performance Models (PNPM '03)*, pp. 62-71, 2003.
- [12] P. Buchholz, "Exact and Ordinary Lumpability in Finite Markov Chains," *J. Applied Probability*, vol. 31, pp. 59-74, 1994.
- [13] P. Buchholz, "On a Markovian Process Algebra," Technical Report 500, Fachbereich Informatik, Univ. of Dortmund, 1994.
- [14] P. Buchholz, "Exact Performance Equivalence: An Equivalence Relation for Stochastic Automata," *Theoretical Computer Science*, vol. 215, nos. 1-2, pp. 263-287, 1999.
- [15] G. Chiola, C. Duthillet, G. Franceschinis, and S. Haddad, "Stochastic Well-Formed Coloured Nets for Symmetric Modeling Applications," *IEEE Trans. Computers*, vol. 42, no. 11, pp. 1343-1360, Nov. 1993.
- [16] L. Capra, C. Duthillet, G. Franceschinis, and J.-M. Ilić, "Exploiting Partial Symmetries for Markov Chain Aggregation," *Proc. First Int'l Workshop Models for Time-Critical Systems (MTCS 2000)*, 2000.
- [17] A. Cimatti, E.M. Clarke, F. Giunchiglia, and M. Roveri, "NUMMV: A New Symbolic Model Checker," *Software Tools for Technology Transfer*, vol. 2, no. 4, pp. 410-425, 2000.
- [18] E.M. Clarke, E.A. Emerson, and A.P. Sistla, "Automatic Verification of Finite-State Concurrent Systems Using Temporal Logic Specifications," *ACM Trans. Programming Languages and Systems*, vol. 8, no. 2, pp. 244-263, 1986.
- [19] E.M. Clarke, O. Grumberg, and D.A. Peled, *Model Checking*. MIT Press, 1999.
- [20] T.H. Cormen, C.E. Leiserson, and R.L. Rivest, *Introduction to Algorithms*. MIT Press, 1990.
- [21] C. Courcoubetis and M. Yannakakis, "The Complexity of Probabilistic Verification," *J. ACM*, vol. 42, no. 4, pp. 857-907, 1995.
- [22] S. Derisavi, H. Hermanns, and W. Sanders, "Optimal State-Space Lumping in Markov Chains," *Information Processing Letters*, vol. 87, no. 6, pp. 309-315, 2003.
- [23] J. Desharnais and P. Panangaden, "Continuous Stochastic Logic Characterizes Bisimulation of Continuous-Time Markov Processes," *J. Logic and Algebraic Programming*, vol. 56, nos. 1-2, pp. 99-115, 2003.
- [24] H. Hermanns, U. Herzog, and V. Mertsiotakis, "Stochastic Process Algebras: Between LOTOS and Markov Chains," *Computer Networks and ISDN Systems*, vol. 30, nos. 9-10, pp. 901-924, 1998.
- [25] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle, "A Tool for Model-Checking Markov Chains," *Software Tools for Technology Transfer*, vol. 4, no. 2, pp. 153-172, 2003.
- [26] J. Hillston, *A Compositional Approach to Performance Modeling*. Cambridge Univ. Press, 1996.
- [27] H. Hansson and B. Jonsson, "A Logic for Reasoning about Time and Reliability," *Formal Aspects of Computing*, vol. 6, no. 5, pp. 512-535, 1994.
- [28] T.A. Henzinger, O. Kupferman, and S. Qadeer, "From Prehistoric to Postmodern Symbolic Model Checking," *Formal Methods in System Design*, vol. 23, pp. 303-327, 2003.
- [29] G.J. Holzmann, "The Model Checker SPIN," *IEEE Trans. Software Eng.*, vol. 23, no. 5, pp. 279-295, May 1997.
- [30] R.A. Howard, *Dynamic Probabilistic Systems*, vols. I, II. John Wiley and Sons, 1971.
- [31] M. Kwiatkowska, G. Norman, and D. Parker, "PRISM 2.0: A Tool for Probabilistic Model Checking," *Proc. First Int'l Conf. Quantitative Evaluation of Systems (QEST '04)*, pp. 322-323, 2004.
- [32] P. Kanellakis and S. Smolka, "CCS Expressions, Finite State Processes, and Three Problems of Equivalence," *Information and Computation*, vol. 86, pp. 43-68, 1990.
- [33] J.G. Kemeny and J.L. Snell, *Finite Markov Chains*. Van Nostrand, 1960.
- [34] N. Lynch and F.W. Vaandrager, "Forward and Backward Simulations Part I: Untimed Systems," *Information and Computation*, vol. 121, no. 2, pp. 214-233, 1995.
- [35] A. Pnueli, "The Temporal Logic of Programs," *Proc. 18th Ann. Symp. Foundations of Computer Science (FOCS '77)*, pp. 46-57, 1977.
- [36] P.J. Schweitzer, "Aggregation Methods for Large Markov Chains," *Math. Computer Performance and Reliability*, pp. 275-302, 1984.
- [37] J. Sproston and S. Donatelli, "Backward Stochastic Bisimulation in CSL Model Checking," *Proc. First Int'l Conf. Quantitative Evaluation of Systems (QEST '04)*, pp. 220-229, 2004.
- [38] M. Vardi, "Automatic Verification of Probabilistic Concurrent Finite-State Programs," *Proc. 16th Ann. Symp. Foundations of Computer Science (FOCS '85)*, pp. 327-338, 1985.



Jeremy Sproston received the bachelor's degree in mathematical economics in 1995, the master's degree in computer science in 1996, and the PhD degree in computer science in 2001 from the University of Birmingham, United Kingdom. He has held postdoctoral research positions at the University of Birmingham and the University of Turin, Italy. Since 2002, he has held a researcher position at the University of Turin, during which time he has been an honorary research fellow at the University of Birmingham, and a visiting researcher at the Laboratoire Spécification et Vérification, École Normale Supérieure de Cachan, France. His research interests include the use of formal methods to verify the correctness and reliability of computer systems, with emphasis on model-checking techniques for probabilistic and timed systems.



Susanna Donatelli received the master's degree in electrical and computer engineering from the University of Massachusetts at Amherst in 1987, and the PhD degree in computer science from the University of Turin, Italy, in 1989. She is currently a full professor of computer science at the University of Turin. Her research has focused on performance evaluation and probabilistic verification of discrete event systems based on queuing networks, stochastic Petri nets, high-level Petri nets, and stochastic process algebras. Her current research interests concentrate mainly on the automatic generation of dependability models from UML specifications, with special attention to the field of large critical infrastructures and on the definition and realization of a multiformalism modeling and validation framework. She serves as a member of the steering committees of the Petri nets community and of the International Conference on Quantitative Evaluation of Systems. She is a member of the ACM and the IEEE Computer Society.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.