Equivalence and Minimization for Model Checking Labeled Markov Chains

Peter Buchholz
Informatik IV, TU Dortmund
D-44221 Dortmund, Germany
peter.buchholz@cs.tu-dortmund.de

Jan Kriege
Informatik IV, TU Dortmund
D-44221 Dortmund, Germany
jan.kriege@cs.tu-dortmund.de

Dimitri Scheftelowitsch
Informatik IV, TU Dortmund
D-44221 Dortmund, Germany
dimitri.scheftelowitsch@cs.tu-dortmund.de

ABSTRACT

Model checking of Markov chains using logics like CSL or asCSL proves whether a logical formula holds for a state of the Markov chain. It has been developed in the last decade to a widely used approach to express performance and dependability quantities for models from a wide range of application areas. In this paper, model checking is extended to prove formulas for distributions rather than single states. This is a very natural way to express certain performance or dependability measures that depend on the state of the system rather than on a specific state in the state space of the Markov chain. It is shown that the mentioned logics can be easily extended from states to distributions and model checking algorithms can also be easily adopted. Furthermore, new equivalences will be introduced that are weaker than bisimulation but still characterize the extended logics.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Markov Processes
; D.4.8 [**Performance**]: Stochastic Analysis

General Terms

Performance

Keywords

Model Checking, Labeled Markov Chains, Equivalence

1. INTRODUCTION

Model checking is nowadays widely used in functional and also in quantitative system analysis. The general idea is that one defines a property that a system should observe or should not observe as a formula in a temporal logic and then proves for each state or some states of the system, whether the formula holds or does not hold.

The mentioned approach is natural in functional system analysis where model checking is applied to the states of a labeled transition system (LTS) or automaton and properties are defined in the temporal logic CTL [6]. Model checking can be applied in a fully automated way after the LTS and the formula have been specified. The corresponding algorithms are efficient in terms of the size of the LTS which, however, may grow exponentially in terms of the system specification. The outcome of model checking is a clear decision which states fulfill the required property and which do not fulfill the property.

The situation is different if model checking is extended to quantitative systems analysis as done with logics like CSL [1, 4, 3]. In this case, Markov chains with labeled states and possibly labeled transitions are considered. In contrast to functional properties two quantifications can be introduced. First, it can be required that a property has to hold within some time interval and it can be defined that a property holds at least or at most for a given probability. In this situation it is often not natural to prove whether the property or formula holds for a specific state since the state of a stochastic system is usually given as a probability distribution over the set of states. To give a simple abstract example, we consider some formula that should hold with a probability of at least p. If we know that the system is with probability q in state 1 where the formula will hold with probability $p_1 < p$ and it is with probability $(1-q)$ in state 2 where the formula holds with probability $p_2 > p$, then the formula holds if and only if $qp_1 + (1-q)p_2 \geq p$. This result cannot be achieved with the known approaches where the outcome is that the formula holds in state 2 and does not hold in state 1. The use of distributions rather than states is natural in stochastic systems. Two typical examples are:

- One wants to check whether a formula holds in a specific situation, for example in steady state or after a specific event like a component failure, which cannot be characterized by a single state but by a probability distribution.
- The states of the Markov chain do not correspond to system states since phase type distributions are integrated to model non-exponential timing. In this case the system state corresponds to a set of states in the Markov chain and if one knows the time since the phase type distributions has been initiated, a probability distribution over the set of phases can be computed and it can be decided whether a formula holds for the distribution which means that it holds for the corresponding abstract system state.

Surprisingly, model checking has, to the best of our knowledge, not been extended to distributions. Distributions contain the common viewpoint of states, because a distribution assigning probability 1 to a single state corresponds to the proof that the state fulfills the formula. However, it is in general not possible to prove formulas that are defined for distributions using state formulas.

In this paper we extend the established logics CSL and asCSL [3] to consider distributions rather than states. This can be done in a

VALUETOOLS 2015, December 14-16, Berlin, Germany

DOI 10.4108/eai.14-12-2015.2262622

very natural way and allows one to easily adopt model checking algorithms. A side effect is that new equivalence relations can be defined that preserve logical formulas. It is known that for CSL over states stochastic bisimulation is the equivalence that characterizes the logic. However, if one considers CSL over distributions, a weaker relation can be defined which relates distributions and includes the case that a state is related to a distribution over several states. The corresponding equivalence will be defined and it will be outlined that equivalence is still decidable and a minimal, possibly non-Markovian, representation can be computed for a Markov chain.

The outline of the paper is as follows. In the next section we briefly review related work. Then we introduce the basic model class, labeled Markov chains in continuous time. In Section 4 equivalence relations are defined and Section 5 introduces the extended logics, presents basic steps for model checking and shows that equivalent Markov models are indistinguishable under the logical formulas. All proofs are given in the appendix. Some smaller examples are used as running examples and Section 6 contains results for medium sized examples.

2. RELATED WORK

In the past labeled Markov processes have been defined in various forms. These definitions include stochastic process algebras [17] whose underlying stochastic model can be interpreted as a Markov process, stochastic automata [24] and interactive Markov chains [16] to mention only a few examples. For those Markov processes model checking approaches have been defined as extensions to the classical model checking of labeled transition systems or automata [6].

Usually model checking means to prove for each state whether a formula holds or does not hold. For performance or dependability analysis where time-dependent properties have to be analyzed the model is extended by stochastic timing information resulting in CTMCs. Model checking CTMCs based on the logic CSL has been introduced in [1]. CSL has been extended in several ways, one extension is asCSL [3] that supports transition labels, while CSL considers only state labels. Moreover, efficient algorithms have been developed to verify CSL formulas [4]. For more results in this area we refer the interested reader to the overview papers [5, 20].

To decrease the effort for the analysis of Markov chains in general and in particular for model checking often bisimulation is used to reduce the state space and to obtain a smaller but equivalent representation of the process. Bisimulation for untimed systems has been introduced in [23, 22] and was later extended to discrete time Markov chains in [21] and continuous time Markov chains in [8, 17]. The relation between CSL and bisimulation is derived in [2] and [18] presents several case studies that studied the effect of bisimulation on model checking. While the approaches for bisimulation mentioned so far all work at the state level, [13] extended bisimulation to the trace distribution of labeled Markov processes and [14] considers bisimulations if the initial state is given by a distribution rather than a single state. A model checking approach that extends CSL to models with phase type distributions is given in [9]. Finally, [12] presented transformations of Markovian and non-Markovian models that define general equivalence relations and allow for an efficient minimization of those models.

3. LABELED MARKOV PROCESSES

We consider Markov chains with labeled transitions and states in continuous time following similar models that have been proposed in the literature over more than two decades [8, 13, 15, 16, 17, 21, 24].

Definition 1 (Labeled Markov chains). *A continuous time labeled Markov chain (CTLMC) is defined by the tuple*

$$\mathcal{CM} = (S, \varphi, \mathcal{A}, \mathbf{G}_e(e \in \mathcal{A}), AP, L),$$

where

- $S = \{s_0, \ldots, s_{n-1}\}$ *is a finite set of states,*
- $\varphi \in [0,1]^{1,n}$ *defines a probability distribution over* S*,*
- $\mathcal{A}$ *is a finite alphabet of transition labels,*
- $\mathbf{G}_e \in \mathbb{R}^{n,n}_{\geq 0}$ *is for each* $e \in \mathcal{A}$ *an* $n \times n$ *transition rate matrix,*
- AP *is a set of atomic propositions, and*
- $L : S \to 2^{AP}$ *is the state labeling function.*

We often identify states by their number, i.e., i means s_i if the interpretation is clear from the context. Thus, $\mathbf{G}_e(i, j)$ is the rate of a transition from s_i to s_j labeled with e. We assume that in every state transitions with label e are either enabled such that $\sum_{j=1}^{n} \mathbf{G}_e(i, j) > 0$ or disabled such that $\sum_{j=1}^{n} \mathbf{G}_e(i, j) = 0$. The model includes CTMCs with only state labels (i.e., $\mathcal{A} = \emptyset$). In this case we write $\mathbf{G}$ rather than $\mathbf{G}_e$ for the transition matrix. Observe that in contrast to most other definitions of labeled Markov models, the initial distribution is part of the definition. To define a unique initial state s_i, vector $\varphi = \mathbf{e}_i$ is used where $\mathbf{e}_i$ is a row vector with 1 in position i and 0 elsewhere. CTLMCs can be completely described by vectors and matrices. Define for $a \in AP$ $\mathbf{r}_a \in \{0,1\}^{n,1}$ with $\mathbf{r}_a(i) = 1$ if $a \in L(s_i)$ and 0 if $a \notin L(s_i)$. Sometimes we use $\mathbf{R}_a = diag(\mathbf{r}_a)$ which is a $n \times n$ diagonal matrix with $\mathbf{r}_a(i)$ in position (i, i). $(\varphi, \mathbf{G}_e(e \in \mathcal{A}), \mathbf{r}_a(a \in AP))$ is a short hand notation for a CTLMC. Sometimes we skip the sets $\mathcal{A}$ and AP, if they are clear from the context or irrelevant.

Transitions in continuous time Markov chains take place after an exponentially distributed duration. An infinite path of a CTLMC is defined as $\sigma = (s(0), e(0), t(0)), (s(1), e(1), t(1)), \ldots$ where $s(h) \in S$, $e(h) \in \mathcal{A}$ and $t(h) \in \mathbb{R}_{\geq 0}$ is the time between the hth and $h+1$th transition or the time before the first transition if $h = 0$. A finite path is given by

$$\sigma = \quad (s(0), e(0), t(0)), (s(1), e(1), t(1)), \ldots, \\ (s(|\sigma|-1), e(|\sigma|-1), t(|\sigma|-1)), s(|\sigma|)).$$

$\sigma(h) = (s(h), e(h), t(h))$ for $h < |\sigma|$, $s(h)$ for $h = |\sigma|$ and undefined otherwise. σ^i_h for $0 \leq h \leq i \leq |\sigma|$ is the subpath including the elements h through i.

Let $\bar{\mathbf{G}}_e = diag(\mathbf{G}_e \mathbb{1})$ and $\bar{\mathbf{G}} = \sum_{e \in \mathcal{A}} \bar{\mathbf{G}}_e$ be the diagonal matrix of transition rates, then

$$Dens(\sigma) = \prod_{h=0}^{|\sigma|-1} e^{-t(h)\bar{\mathbf{G}}(s(h),s(h))} \mathbf{G}_{e(h)}(s(h), s(h+1))$$

defines the value of the probability density for the path σ. Ω_i is the set of all paths of length i and Ω is the set of all finite paths.

For the definition of equivalent behavior, events and/or state propositions are observed and not detailed states. Thus, a sequence $\varsigma = \left((a_0, e_0, t_0), \ldots, (a_{|\varsigma|-1}, e_{|\varsigma|-1}, t_{|\varsigma|-1}), a_{|\varsigma|}\right)$ where $a_h \in AP$, $e_h \in \mathcal{A}$ and $t_h \in \mathbb{R}_{\geq 0}$ is a finite observable behavior. If for a CTLMC $\mathcal{A}$ is empty, we simply skip the second component and if states are not labeled we skip the first component. We define Υ as the set of

all finite observable behaviors. For a CTLMC and an observable behavior, the density is given by

$$Dens_{(\varphi,\mathbf{G}_e,\mathbf{r}_a)}(\varsigma) = \varphi\left(\prod_{h=0}^{|\varsigma|-1} \mathbf{R}_{a_h} e^{-t_h\tilde{\mathbf{G}}}\mathbf{G}_{e_h}\right)\mathbf{r}_{a_{|\varsigma|}}.$$

Running Examples

The *first example* is a CTLMC $(\varphi, \mathbf{G}, \mathbf{r}_a(a \in AP = \{a, b\}))$ without transition labels where

$$\mathbf{G} = \begin{pmatrix} 0 & 0 & 0 & \mu_0 \\ 0 & 0 & 0 & \mu_1 \\ 0 & 0 & 0 & \mu_2 \\ \lambda_0 & \lambda_1 & \lambda_2 & 0 \end{pmatrix},$$

states $s_0, \ldots, s_2$ are labeled with a and state s_3 is labeled with b.

4. EQUIVALENCE OF LABELED MARKOV CHAINS

We first define equivalence of CTLMCs as a natural adoption of trace equivalence for labeled Markov processes [13].

Definition 2. *Two CTLMCs* $(\varphi, \mathbf{G}_e(e \in \mathcal{A}), \mathbf{r}_a(a \in AP))$ *and* $(\upsilon, \mathbf{H}_e(e \in \mathcal{A}), \mathbf{s}_a(a \in AP))$ *are equivalent if and only if* $\forall\varsigma \in \Upsilon$*:* $Dens_{(\varphi,\mathbf{G}_e,\mathbf{r}_a)}(\varsigma) = Dens_{(\upsilon,\mathbf{H}_e,\mathbf{s}_a)}(\varsigma)$.

In the following we define different equivalences for CTLMCs based on the minimal non-Markovian representation of Markov models developed in [12] and extended to compositional models in [10]. These general equivalence relations are extensions to bisimulation on distributions as defined in [14].

Let $\mathcal{CM}_1 = (\varphi, \mathbf{G}_e(e \in \mathcal{A}), \mathbf{r}_a(a \in AP))$ and $\mathcal{CM}_2 = (\upsilon, \mathbf{H}_e(e \in \mathcal{A}), \mathbf{s}_a(a \in AP))$ be two CTLMCs with m and $n < m$ states, respectively. Then the following relations are defined:

1. $\mathcal{CM}_1 \sim \mathcal{CM}_2$ holds if and only if there exists a matrix $\mathbf{V} \in \mathbb{R}^{m,n}$ with $\mathbf{V}\mathbb{1}_n = \mathbb{1}_m$, such that $\varphi\mathbf{V} = \upsilon$, $\tilde{\mathbf{G}}\mathbf{V} = \mathbf{V}\tilde{\mathbf{H}}$, $\forall e \in \mathcal{A}$: $\mathbf{G}_e\mathbf{V} = \mathbf{V}\mathbf{H}_e$ and $\forall a \in AP : \mathbf{R}_a\mathbf{V} = \mathbf{V}\mathbf{S}_a$.
2. $\mathcal{CM}_1 \simeq \mathcal{CM}_2$ holds if and only if there exists a matrix $\mathbf{W} \in \mathbb{R}^{n,m}$ with $\mathbf{W}\mathbb{1}_m = \mathbb{1}_n$, such that $\varphi = \mathbf{W}\upsilon$, $\mathbf{W}\tilde{\mathbf{G}} = \tilde{\mathbf{H}}\mathbf{W}$, $\forall e \in \mathcal{A} : \mathbf{W}\mathbf{G}_e = \mathbf{H}_e\mathbf{W}$ and $\forall a \in AP : \mathbf{W}\mathbf{R}_a = \mathbf{S}_a\mathbf{W}$.

If $m = n$ a third relation is defined:

3. $\mathcal{CM}_1 \cong \mathcal{CM}_2$ holds if and only if there exists a matrix $\mathbf{U} \in \mathbb{R}^{n,n}$ with $\mathbf{U}\mathbb{1} = \mathbb{1}$, such that $\varphi\mathbf{U} = \upsilon$, $\tilde{\mathbf{G}}\mathbf{U} = \mathbf{U}\tilde{\mathbf{H}}$, $\forall e \in \mathcal{A}$: $\mathbf{G}_e\mathbf{U} = \mathbf{U}\mathbf{H}_e$ and $\forall a \in AP : \mathbf{R}_a\mathbf{U} = \mathbf{U}\mathbf{S}_a$.

The first relation is a natural extension of strong lumpability [19] and bisimulation for labeled Markov chains [8, 16, 17]. For strong lumpability or bisimulation matrix $\mathbf{V}$ contains only elements from $\{0, 1\}$ which implies that in each row of $\mathbf{V}$ exactly one element equals 1, the remaining elements are 0. Thus, strong lumpability and bisimulation describe a mapping where one state from the larger state space is mapped on exactly one element in the smaller state space. In the more general definition used here, a state in the larger state space is represented by a weighted sum of states from the smaller state space. In a similar way, the second equivalence can be related to weak lumpability of Markov chains [19]. For details we refer to [11].

The following theorem shows that any of the above equations assures equivalence, a proof is given in the appendix.

Theorem 1. *Two CTLMCs* $\mathcal{CM}_1$ *and* $\mathcal{CM}_2$ *that are in one of the relations* $\sim$, $\simeq$ *or* $\cong$ *are equivalent.*

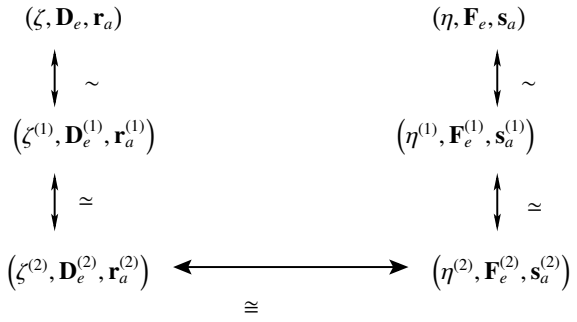

Figure 1: Relation $\approx$ for CTLMCs

From [12] it can be concluded that $\mathbf{U}$ is non-singular and that the matrices $\mathbf{V}$ and $\mathbf{W}$ have full rank, such that we can find a left- and right-inverse, respectively, i.e. $\mathbf{V}^{\#}\mathbf{V} = \mathbf{I}$ and $\mathbf{W}\mathbf{W}^{\#} = \mathbf{I}$.

Now, assume that two CTLMCs, $(\zeta, \mathbf{D}_e, \mathbf{r})$ with m states and $(\eta, \mathbf{F}_e, \mathbf{s}_a)$ with n states are given. Then the two processes are in relation $\approx$ if the diagram in Figure 1 commutes. For each step in the diagram, an efficient algorithm exists that computes a minimal equivalent representation according to the required relation. Computation of the matrices $\mathbf{V}$ and $\mathbf{W}$ and the corresponding minimal representations can be done with the staircase algorithm from [12] with an effort in $O(|\mathcal{A}| \cdot m^4)$ for CTLMCs of order m. Matrix $\mathbf{U}$ can be computed with an effort in $O(|\mathcal{A}| \cdot m^3)$ for processes of order m using algorithms for the solution of Sylvester equations [7].

The following corollary follows from Theorem 1.

Corollary 1. *If two CTLMCs are in relation* $\approx$*, then they are equivalent.*

Running Examples

For the *first example* assume that $\mu_0 \neq \mu_1$ and let $p = (\mu_2 - \mu_1)/(\mu_0 - \mu_1)$. Then define matrices

$$\mathbf{V} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ p & 1-p & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{H} = \begin{pmatrix} 0 & 0 & \mu_0 \\ 0 & 0 & \mu_1 \\ \lambda_0 + p\lambda_2 & \lambda_1 + (1-p)\lambda_2 & 0 \end{pmatrix}$$

and the CTLMC $(\upsilon = \varphi\mathbf{V}, \mathbf{H}, \mathbf{s}_a(a \in \{a, b\}))$. Assume that the last state is labeled with b and the first two states are labeled with a. Then $(\upsilon = \varphi\mathbf{V}, \mathbf{H}, \mathbf{s}_a(a \in \{a, b\}))$ is in relation $\sim$ to $(\varphi, \mathbf{G}, \mathbf{r}_a)$. Observe that the new process is a CTLMC if and only if $p \in [0, 1]$.

5. MODEL CHECKING LABELED MARKOV CHAINS

Different logics have been defined for model checking Markov chains with state and transition labels. Usually model checking means to prove for each state whether a formula holds or does not hold. This viewpoint has been transferred from qualitative system analysis to quantitative system analysis using transition probabilities or rates. However, if the state is defined in terms of a distribution rather than a unique state, as it is the case here and in [13], then a formula should hold with respect to a distribution which does not necessarily mean that it has to hold for all states with non-zero probabilities as shown below.

We consider here the logics CSL [4] for CTLMCs with state labels and asCSL [3] for CTLMCs with state and transition labels. In

all cases we extend the logics to check a formula for a CTLMC with a given initial distribution. By defining an initial distribution equal to $\mathbf{e}_i$, this includes the common case, where a formula is checked for state s_i. We extend the two logics CSL and asCSL in the mentioned way resulting in logics dCSL and dasCSL, where the letter d stands for distribution.

We begin with *dCSL* which is defined over CTLMCs without transition labels such that $C\mathcal{M} = (\mu, \mathbf{G}, \mathbf{r}_a(a \in AP))$.

Definition 3 (dCSL). *A dCSL formula is defined as*

$$\Gamma ::= \mathcal{DS}_{\bowtie p}(\Phi) \mid \mathcal{DP}_{\bowtie p}(\Psi)$$

where $\bowtie \in \{\leq, \geq\}$, $p \in [0,1]$, $I \subset \mathbb{R}_{\geq 0}$ *is some non-empty interval,* Φ *is a state formula*

$$\Phi ::= tt \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathcal{S}_{\bowtie p}(\Phi) \mid \mathcal{P}_{\bowtie p}(\Psi)$$

and Ψ *is a path formula*

$$\Psi ::= \Phi \mathcal{U}^I \Phi \mid X_I \Phi.$$

The syntax and semantics of the state and path formulas is as in CSL [1, 4].

For a state distribution formula Φ and state $s \in S$, $s \models \Phi$ or $s \not\models \Phi$ holds. Let $\mathbf{r}_\Phi$ be a vector with $\mathbf{r}_\Phi(i) = 1$ if $s_i \models \Phi$ and 0 otherwise. As before $\mathbf{R}_\Phi = diag(\mathbf{r}_\phi)$. Furthermore we define for some transition matrix of a CTLMC $\mathbf{G}$ and a state formula Φ a matrix $\mathbf{G}[\Phi]$ as the matrix that results from $\mathbf{G}$ when all states where Φ does not hold are made absorbing. This means that $\mathbf{G}[\phi](i\bullet) = (0, \ldots, 0)$ if $s_i \not\models \Phi$ and $\mathbf{G}[\phi](i\bullet) = \mathbf{G}(i\bullet)$ if $s_i \models \Phi$[1].

Then for some distribution μ,

$$\mu \models \mathcal{DS}_{\bowtie p}(\Phi) \;\Leftrightarrow\; \mu \mathbf{r}_\Phi \bowtie p. \tag{1}$$

For $\mu = \mathbf{e}_i$, $\bowtie = \geq$ and $p = 1$, $\mu \models \mathcal{DS}_{\bowtie p}(\Phi)$ is equivalent to $s_i \models \Phi$ which shows that for any CSL formula an equivalent dCSL formula exists.

To handle path formulas we define in accordance to [4] $\gamma(\sigma, h) = t(h)$, the time spent between the $h-1$th and hth transition at path σ and $a@t = s(h)$ where $\sum_{j=1}^{h-1} t(j) \leq t$ and $\sum_{j=1}^{h} t(j) > t$. The latter sum is, of course, only defined for paths of a length $\geq h$. The following two sets are defined

$$\begin{aligned}
\Omega_i(\Phi_1 \mathcal{U}^I \Phi_2) &= \{\sigma | s(0) = s_i \wedge \exists t \in I : a@t \models \Phi_2 \wedge \\
&\qquad \forall t' < t : a@t' \models \Phi_1\} \\
\Omega_i(X_I \Phi) &= \{\sigma | s(1) \models \Phi \wedge \gamma(\sigma, 0) \in I\}
\end{aligned}$$

Under appropriate measurability conditions the probability of an arbitrary path from $\Omega_i(\cdot)$ is well defined (see [4]) and can be computed from the following equations

$$\begin{aligned}
\mathbf{q}_{\Phi_1 \mathcal{U}^{[t_1,t_2]} \Phi_2} &= e^{t_1(\mathbf{G}[\Phi_1] - \bar{\mathbf{G}}[\Phi_1])} \mathbf{R}_{\Phi_1} e^{(t_2 - t_1)(\mathbf{G}[\Phi_1 \wedge \neg\Phi_2] - \bar{\mathbf{G}}[\Phi_1 \wedge \neg\Phi_2])} \mathbf{r}_{\Phi_2} \quad (2) \\
&\qquad \text{for } t_1 > 0 \\
\mathbf{q}_{\Phi_1 \mathcal{U}^{[0,t_2]} \Phi_2} &= e^{t_2(\mathbf{G}[\Phi_1 \wedge \neg\Phi_2] - \bar{\mathbf{G}}[\Phi_1 \wedge \neg\Phi_2])} \mathbf{r}_{\Phi_2} \qquad \text{for } t_1 = 0 \quad (3) \\
\mathbf{q}_{X_{[t_1,t_2]}\Phi} &= e^{-t_1 \bar{\mathbf{G}}} \int_{t=t_1}^{t_2} e^{-t\bar{\mathbf{G}}} \mathbf{G} \mathbf{r}_\Phi dt
\end{aligned}$$

where $I = [t_1, t_2]$ with $0 \leq t_1 \leq t_2$. Observe that for dCSL transitions are not labeled such that we can write $\mathbf{G}$ rather than $\mathbf{G}_e$. The vectors $\mathbf{q}_{\Phi_1 \mathcal{U}^{[t_1,t_2]} \Phi_2}$ and $\mathbf{q}_{X_{t_1}\Phi}$ can be computed efficiently using uniformization as shown in [4].

[1] Observe that the definition of $\mathbf{G}[\Phi]$ corresponds to $\mathbf{G}[\neg\Phi]$ in [4] such that the equations syntactically differ but have the same semantics.

Distribution μ observes the distributional path formulas, if the following equations hold.

$$\begin{aligned}
\mu \mathbf{q}_{\Phi_1 \mathcal{U}^{[t_1,t_2]} \Phi_2} \bowtie p &\;\Leftrightarrow\; \mu \models \mathcal{DP}_{\bowtie p}\left(\Phi_1 \mathcal{U}^{[t_1,t_2]} \Phi_2\right) \\
\mu \mathbf{q}_{X_{[t_1,t_2]}\Phi} \bowtie p &\;\Leftrightarrow\; \mu \models \mathcal{DP}_{\bowtie p}\left(X_{t_1} \Phi\right)
\end{aligned}$$

Observe that the major effort to verify $\mu \models \mathcal{DP}_{\bowtie p}(\ldots)$ is required to compute the vectors $\mathbf{q}_{\cdot}$ in (2,3). After the vectors are available at most one inner product has to be computed. This means that the asymptotic effort to verify dCSL formulas is the same than the effort to verify CSL formulas.

The following theorem shows the relation between $\approx$ and dCSL formulas, the proof can be found in the appendix.

Theorem 2. *Let* $C\mathcal{M}_1 = (\varphi, \mathbf{G}, \mathbf{r}_a(a \in \mathcal{A}))$ *and* $C\mathcal{M}_2 = (\upsilon, \mathbf{H}, \mathbf{s}_a(a \in \mathcal{A}))$ *be two CTLMCs with* $C\mathcal{M}_1 \approx C\mathcal{M}_2$*, then for any dCSL formula* Γ*:*

$$\varphi \models \Gamma \;\Leftrightarrow\; \upsilon \models \Gamma.$$

Finally, we consider the extension of asCSL to *dasCSL* which follows the same ideas used for the definition of dCSL.

Definition 4 (dasCSL). *A dasCSL formula is defined as*

$$\Gamma ::= \mathcal{DS}_{\bowtie p}(\Phi) \mid \mathcal{DP}_{\bowtie p}(\Psi)$$

where $\bowtie \in \{\leq, \geq\}$, $p \in [0,1]$, $I \subset \mathbb{R}_{\geq 0}$ *is some non-empty interval,* Φ *is a state formula*

$$\Phi ::= tt \mid a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathcal{S}_{\bowtie p}(\Phi) \mid \mathcal{P}_{\bowtie p}(\Psi)$$

and Ψ *is a path formula*

$$\Psi ::= \alpha^I$$

which is formally defined below.

For the definition of path formulas we follow [3]. α is a program that specifies properties which have to hold for finite paths. Programs are specified by the following grammar

$$\alpha ::= \varepsilon \mid (\Phi, e) \mid \alpha; \alpha \mid \alpha \cup \alpha \mid \alpha^*$$

where Φ is a dasCSL state formula and $e \in \mathcal{A} \cup \{\surd\}$, $\surd$ is a symbol that does not belong to $\mathcal{A}$. We define $\alpha_0 = \varepsilon$ and $\alpha_i = \alpha; \alpha_{i-1}$ for $i \geq 1$. Symbol ; denotes the concatenation of two programs and α^* the Kleene star, the n-fold sequential composition for arbitrary $n > 0$.

The set of paths $\Omega(\alpha)$ that fulfill the program α can be defined inductively from the following sets of finite paths

$$\begin{aligned}
\Omega(\varepsilon) &= \{\sigma \mid |\sigma| = 0\} \\
\Omega(\Phi, e) &= \{\sigma \mid s(0) \models \Phi \wedge e(0) = e\} \\
\Omega(\Phi, \surd) &= \{\sigma \mid s(0) \models \Phi \wedge |\sigma| = 0\} \\
\Omega(\alpha_1; \alpha_2) &= \{\sigma \mid \exists h \in \{0, 1, \ldots, |\sigma| - 1\} : \\
&\qquad \sigma_0^h \in \Omega(\alpha_1) \wedge \sigma_h^{|\sigma|} \in \Omega(\alpha_2)\Big\} \\
\Omega(\alpha_1 \cup \alpha_2) &= \{\sigma \mid \sigma \in \Omega(\alpha_1) \cup \Omega(\alpha_2)\} \\
\Omega(\alpha^*) &= \{\sigma \mid \exists h \geq 0 : \sigma \in \Omega(\alpha_h)\}
\end{aligned}$$

A path σ belongs to the set $\Omega(\alpha^I)$, if $\sigma \in \Omega(\alpha)$ and $\sum_{h=0}^{|\sigma|-1} t(h) \in I$.

Since programs are regular expressions they can be equivalently characterized as the language of a finite acceptor. This acceptor can be transformed into a deterministic acceptor which means that for each path σ, there exists a unique run of the acceptor and the path is accepted if it ends in a final state. Details of the construction of the acceptor for a program α and a CTLMC $C\mathcal{M}$ can be found in [3].

For some program α and CTLMC $C\mathcal{M}$ let $\mathcal{DA} = (Z, \mathcal{B}, \delta, z_0, \mathcal{F})$ where

- Z is a finite set of states,
- $\mathcal{B}$ is an alphabet of transition labels of the form (Φ, e) where Φ is a state formula for $\mathcal{CM}$ and $e \in \mathcal{A}$,
- $\delta : Z \times \mathcal{B} \to Z$ is the (deterministic) transition function,
- $z_0 \in Z$ is the unique initial state,
- $\mathcal{F} \subset Z$ is the set of final states,

be a deterministic automaton that accepts exactly the paths of $\mathcal{CM}$ that result in a successful run of program α. Observe that $\surd$ does not appear as transition inscription here. The generation of the deterministic automaton including the elimination of $\surd$ is described in [3].

To check $\mu \models \mathcal{DP}_{\bowtie p}\left(\alpha^I\right)$, we build the composed automaton $\mathcal{DA} \times \mathcal{CM}\left(\mathcal{CM} = (S, \varphi, \mathcal{A}, \mathbf{G}_a, AP, L)\right)$ without transition labels which can be interpreted as a CTLMC without transition labels. We build the automaton on the complete state space $Z \times S$ which might contain unreachable states and is not necessary for model checking but helps to prove the preservation of formulas by the previously proposed equivalence relations.

The composed automaton is

$$\mathcal{CM}' = \mathcal{DA} \times \mathcal{CM} = (Z \times S, \upsilon, \emptyset, \mathbf{F}, AP', L')$$

where

- $\upsilon = (\varphi, \mathbf{0}, \ldots, \mathbf{0})$,
- $\mathbf{F}$ is a matrix of order $|Z| \cdot |S| \times |Z| \cdot |S|$ which is built by $|Z| \times |Z|$ submatrices $\mathbf{F}_{z,z'}$ of order $|S| \times |S|$ where $\mathbf{F}_{z,z'} = \sum_{(\Phi,e):\delta(\Phi,e)=z'} \mathbf{R}_\Phi \mathbf{G}_e$,
- $AP' = AP \cup \{fin\}$ with $fin \notin AP$,
- $L'(z,s) = \begin{cases} L(s) & \text{if } z \notin \mathcal{F}, \\ L(s) \cup \{fin\} & \text{if } z \in \mathcal{F}. \end{cases}$

Then model checking can be performed using standard methods for the new CTLMC. Let $\mathbf{r}_{fin}$ be a vector with 1 in position i if $s_i = (s, z)$, $z \in \mathcal{F}$ and 0 otherwise. If $I = [0, t]$, then vector

$$\mathbf{q}_{\alpha^I} = e^{t(\mathbf{F}[\neg fin] - \bar{\mathbf{F}}[\neg fin])} \mathbf{r}_{fin} \tag{4}$$

is computed by a transient analysis and

$$\varphi \models \mathcal{DP}_{\bowtie p}\left(\alpha^I\right) \Leftrightarrow \upsilon \mathbf{q}_{\alpha^I} \bowtie p. \tag{5}$$

For $I = [t_1, t_2]$ with $0 < t_1 \leq t_2$, vector $\mathbf{q}_{\alpha^I}$ is computed in two steps which are explained in [3]. We first compute the vector

$$\hat{\mathbf{q}}_{\alpha^I} = e^{(t_2 - t_1)(\mathbf{F}[\neg fin] - \bar{\mathbf{F}}[\neg fin])} \mathbf{r}_{fin} \tag{6}$$

by a transient analysis. This corresponds to the second step described in [3]. Then

$$\mathbf{q}_{\alpha^I} = e^{t_1(\mathbf{F} - \bar{\mathbf{F}})} \hat{\mathbf{q}}_{\alpha^I} \tag{7}$$

is computed and used in (5) to check the formula. Again the asymptotic effort to check a dasCSL formula for distribution μ is the same than the effort to check the same formula in asCSL for a single state because the evaluation of (6) and (7) requires much more time than the computation of the final result by an inner product (dasCSL) or by selecting a single element (asCSL).

THEOREM 3. *Let $\mathcal{CM}_1 = (\varphi, \mathbf{G}, \mathbf{r}_a (a \in \mathcal{A}))$ and $\mathcal{CM}_2 = (\upsilon, \mathbf{H}, \mathbf{s}_a (a \in \mathcal{A}))$ be two CTLMCs with $\mathcal{CM}_1 \approx \mathcal{CM}_2$, then, for any dasCSL formula Γ:*

$$\varphi \models \Gamma \Leftrightarrow \upsilon \models \Gamma.$$

The proof can be found in the appendix.

6. EXAMPLES

6.1 An M/M/1 Queue

We consider a simple M/M/1 queue with capacity 5 as a running example to present dasCSL model checking. Arrivals occur with rate $\lambda = 1$, the service rate is $\mu = 2$. The CTLMC corresponding to the system is shown in Fig. 2. We use two transition labels a and b. a is used for normal transitions where either a customer arrives or is served, b is used for customers that are lost due to a full queue. The atomic proposition *empty* is associated with state s_0. The other states are labeled with *busy*. We analyze the system for two different scenarios regarding the initial distribution of the queue length. The distribution $\mu_1 = [0.4, 0.3, 0.2, 0.1, 0, 0]$ models a normal load, while distribution $\mu_2 = [0.1, 0.15, 0.2, 0.3, 0.15, 0.1]$ models a high load.

Then, dasCSL is used to investigate the probability that at most 2 customers are lost before the *empty* state is reached again within the next t time units. The deterministic automaton describing the

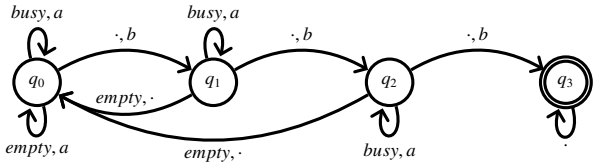

Figure 3: Deterministic Automaton

mentioned property is shown in Fig. 3. It accepts inputs that violate the property mentioned above, i.e. sequences with more than two blocked customers before the *empty* state is reached. The composed automaton has a 24×24 matrix $\mathbf{F}$. The submatrices are given by $\mathbf{F}_{q_0,q_0} = \mathbf{G}_a$, $\mathbf{F}_{q_0,q_1} = \mathbf{F}_{q_1,q_2} = \mathbf{F}_{q_2,q_3} = \mathbf{G}_b$, $\mathbf{F}_{q_1,q_0} = \mathbf{F}_{q_2,q_0} = \mathbf{R}_{empty}\mathbf{G}_a$, $\mathbf{F}_{q_1,q_1} = \mathbf{F}_{q_2,q_2} = \mathbf{R}_{busy}\mathbf{G}_a$ and $\mathbf{F}_{q_3,q_3} = \mathbf{G}_a + \mathbf{G}_b$. All other submatrices are zero. With these matrices we evaluated Eq. 4 for different values for t between 0 to 600. The probabilities p for that $\mu_i \models \mathcal{DP}_{\leq p}\left(\alpha^{[0,t]}\right)$ is fulfilled are shown in Fig. 4.

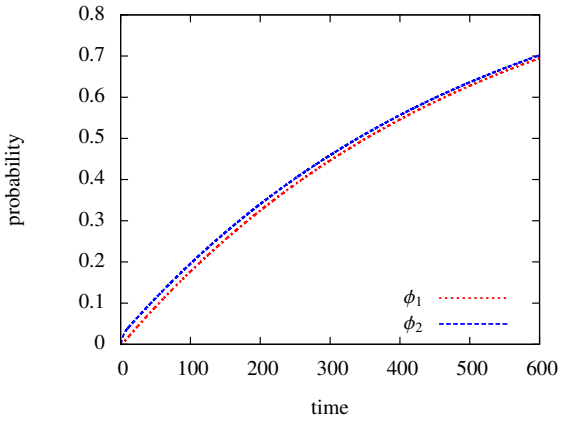

Figure 4: Results for the Queueing Example

6.2 A PH/PH/1 Queue

As second example we consider a finite capacity queue with 2-phase hyperexponential arrival and service time distributions. The inter-arrival time distribution function has an initial state vector $(0.95227, 0.047733)$ and phase rates $(1.5236, 0.076373)$ (i.e., a 2-phase hyperexponential distribution with mean 1.25 and squared

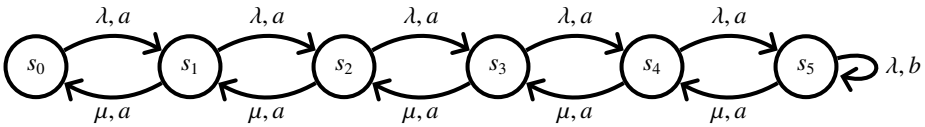

Figure 2: CTLMC of Queueing example

coefficient of variation 10). For the service time distribution we choose a 2-phase hyperexponential distribution with initial state vector (0.97559, 0.024405) and phase rates (1.99512, 0.04881) (i.e., a distribution with mean 1 and squared coefficient of variation of 20). The queue has a maximal capacity of 10 and we add an additional absorbing state that is entered, if a customer arrives to a fully occupied queue. States can be described by a triple (i, j, k) where $i \in \{-1, 0, \ldots, 10\}$. $i = -1$ defines the absorbing state, and $i = 0, \ldots, 10$ describes the number of customers in the queue. j indicates the phase of the hyperexponential distribution for the arrivals and becomes 0 for $i = -1$. Similarly, k indicates the phase of the hyperexponential distribution for the service and becomes 0 for $i = -1$ and $i = 0$. Thus, the state space consists of 43 states, 2 states where the queue is empty, 40 states with 1 through 10 customers in the queue and one absorbing state indicating an overflow. We assume that the two states describing the empty queue observe an atomic proposition *empty* and the absorbing state observes atomic proposition *full*. Transitions are not labeled.

t	(1, 1, 1)	(1, 1, 2)	(1, 2, 1)	(1, 2, 2)	initial dist.
5	3.349e-03	2.157e-01	3.876e-15	1.036e-11	8.124e-03
10	1.166e-02	8.904e-01	8.013e-15	6.147e-09	3.152e-02
20	1.820e-02	9.679e-01	8.110e-15	2.209e-06	3.941e-02
50	1.976e-02	9.680e-01	8.110e-15	1.276e-03	4.086e-02
100	1.977e-02	9.680e-01	8.110e-15	3.112e-02	4.090e-02
500	1.977e-02	9.680e-01	8.110e-15	3.449e-01	4.127e-02

Table 1: Value of p for which $\mathcal{DP}_{>p}\left(\neg\, empty\, \mathcal{U}^{[0,t]}\, full\right)$ holds for the PH/PH/1 queue.

The goal is now to compute $\mathcal{DP}_{>p}\left(\neg\, empty\, \mathcal{U}^{[0,t]}\, full\right)$ which equals the probability that a customer is lost before the system becomes empty. We assume that the analysis begins immediately after the first customer arrived to an empty system. Thus, the system can be in one of four states (1, 1, 1), (1, 1, 2), (1, 2, 1) and (1, 2, 2). This implies that the initial distribution can be easily computed from the two initial vectors of the hyperexponential distributions. Naturally, this defines a distribution and not a single state. Table 1 shows the probabilities p for which the formula holds for different values of t, for the initial distribution and for the different states in which the system can be in for population 1. Obviously, p depends heavily on the state and model checking of the isolated states does not answer the question whether the formula holds for the system or does not hold. Whereas it holds with some probability between the extreme values for the states when the distribution is considered. Standard CSL cannot be used to express the required property.

6.3 The Workstation Cluster

The workstation cluster is a widely used benchmark in stochastic model checking [18]. The model describes two clusters of workstations connected via a backbone net and each cluster itself is realized by a star topology with a central switch and N workstations. The system provides *premium* service as long as at least N connected workstations are available and *minimum* service if at least $\lceil N/2 \rceil$ connected workstations are available. We assume that states are labeled with *min* if they provide *minimum* but not *premium* service, they are labeled with *prem* if they provide *premium* service. As shown in [18] stochastic bisimulation can be applied to reduce the state space of the model by a factor of approximately two, a further reduction with the equivalence relation presented here is not possible.

A typical formula for the analysis of the workstation cluster is $\mathcal{DS}_{\geq p}\left(min\, \mathcal{U}^{[0,t]}\, prem\right)$ which states that the system delivers *minimum* service and recovers within t units of time to a state where *premium* service is provided without reaching a state where the service level drops below minimum. Typically one would like to analyze the formula starting from the point where the service drops from *premium* to *minimum*.

The formula can be checked for $\mathcal{CM} = (S, \varphi, \emptyset, \mathbf{G}, \{min,\ prem,\ none\}, L)$ where $\mathbf{G} - \bar{\mathbf{G}}$ is the generator matrix of the CTMC described by the workstation cluster. Let $\bar{\varphi}(\mathbf{G} - \bar{\mathbf{G}}) = \mathbf{0}$ be the stationary vector of the CTMC which is for the example unique since the CTMC is ergodic. Then the initial vector for the mentioned situation is computed as

$$\varphi(i) = \begin{cases} 0 & \text{if } min \notin L(s_i) \\ \frac{\sum_{j:\, prem\, \in L(s_j)} \bar{\varphi}(j)\mathbf{G}(j,i)}{\sum_{k:\, prem\, \in L(s_k)} \sum_{l:\, min \in L(s_l)} \bar{\varphi}(k)\mathbf{G}(k,l)} & \text{if } min \in L(s_i) \end{cases}$$

Then the until formula can be evaluated as described above.

7. CONCLUSIONS

We presented an extended approach for model checking Markov models with labeled transitions and states. In contrast to known approaches, distributions rather than single states are considered. It is shown that common logics can be easily extended to adopt this viewpoint and that the approach allows one to extend equivalence relations that preserve logical formulas beyond bisimulation. The use of distributions can be applied to prove properties of a system that hold after specific events that have been observed but do not necessarily imply that the system is in a specific state. Examples are arrivals or departures of customers, failure of components or even the steady state distribution conditioned on some subset of the state space. It is quite natural to ask whether a system fulfills a formula in such a situation which is possible with the extended logics proposed here but cannot be analyzed with standard approaches for model checking. Standard model checking algorithms can be easily adopted for the extended logics whereas the introduction of the new equivalence relations is a real extension of bisimulation at state level since bisimulation no longer characterizes the smallest system that is indistinguishable under a formula. The latter aspect will be investigated in the future.

It is, of course, possible to extend the approach to other logics that have been proposed in a Markovian setting and also to discrete time models. Furthermore, compositionality of the approach should be considered in the future.

8. REFERENCES

[1] A. Aziz, K. Sanwal, V. Singhal, and R. Brayton. Model checking continuous time Markov chains. *ACM Trans. on Computational Logic*, 1(1):162–170, 2000.

[2] A. Aziz, V. Singhal, and F. Balarin. It usually works: The temporal logic of stochastic systems. In P. Wolper, editor, *CAV*, volume 939 of *Lecture Notes in Computer Science*, pages 155–165. Springer, 1995.

[3] C. Baier, L. Cloth, B. Haverkort, M. Kuntz, and M. Siegle. Model checking Markov chains with actions and state labels. *IEEE Transactions on Software Engineering*, 33(4):209–224, 2007.

[4] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking algorithms for continuous time Markov chains. *IEEE Transactions on Software Engineering*, 29(7):524–541, 2003.

[5] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Performance evaluation and model checking join forces. *Comm. ACM*, 53(9):76–85, 2010.

[6] C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.

[7] R. H. Bartels and G. W. Stewart. Solution of the matrix equation AX+XB=C. *Comm. ACM*, 15:820–826, 1972.

[8] P. Buchholz. Markovian process algebra: composition and equivalence. In U. Herzog and M. Rettelbach, editors, *Proc. of the 2nd Work. on Process Algebras and Performance Modelling*, pages 11–30. Arbeitsberichte des IMMD, University of Erlangen, no. 27, 1994.

[9] P. Buchholz, J. Kriege, and D. Scheftelowitsch. Model checking stochastic automata for dependability and performance measures. In *DSN*, 2014.

[10] P. Buchholz and M. Telek. Composition and equivalence of Markovian and non-Markovian models. In *QEST*, pages 213–222. IEEE Computer Society, 2011.

[11] P. Buchholz and M. Telek. Rational processes related to communicating Markov processes. *J. Appl. Probab.*, 49(1):40–59, 2012.

[12] P. Buchholz and M. Telek. On minimal representations of rational arrival processes. *Annals OR*, 202(1):35–58, 2013.

[13] L. Doyen, T. A. Henzinger, and J.-F. Raskin. Equivalence of labeled Markov chains. *Int. J. Found. Comput. Sci.*, 19(3):549–563, 2008.

[14] C. Eisentraut, H. Hermanns, J. Krämer, A. Turrini, and L. Zhang. Deciding bisimilarities on distributions. In *Quantitative Evaluation of Systems - 10th International Conference, QEST*, volume 8054 of *Lecture Notes in Computer Science*, pages 72–88. Springer, 2013.

[15] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[16] H. Hermanns. *Interactive Markov Chains: The Quest for Quantified Quality*, volume 2428 of *Lecture Notes in Computer Science*. Springer, 2002.

[17] J. Hillston. A compositional approach for performance modelling. Phd thesis, University of Edinburgh, Dep. of Comp. Sc., 1994.

[18] J.-P. Katoen, T. Kemna, I. S. Zapreev, and D. N. Jansen. Bisimulation minimisation mostly speeds up probabilistic model checking. In O. Grumberg and M. Huth, editors, *TACAS*, volume 4424 of *Lecture Notes in Computer Science*, pages 87–101. Springer, 2007.

[19] J. G. Kemeny and J. L. Snell. *Finite Markov chains*. University series in undergraduate mathematics. VanNostrand, New York, repr edition, 1969.

[20] M. Z. Kwiatkowska, G. Norman, and D. Parker. Stochastic model checking. In M. Bernardo and J. Hillston, editors, *SFM*, volume 4486 of *Lecture Notes in Computer Science*, pages 220–270, 2007.

[21] K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

[22] R. Milner. *Communication and concurrency*. Prentice Hall, 1989.

[23] D. Park. Concurrency and automata on infinite sequences. In *Proc. 5th GI Conference on Theoretical Computer Science*, pages 167–183. Springer, 1981.

[24] B. Plateau. On the stochastic structure of parallelism and synchronisation models for distributed algorithms. *ACM Performance Evaluation Review*, 13:142–154, 1985.

APPENDIX

Proof of Theorem 1

We show the proof for relation $\sim$. Let $CM_1 = (\mu, \mathbf{G}_e(e \in \mathcal{A}), \mathbf{r}_a(a \in AP))$ of order m and $CM_2 = (\psi, \mathbf{H}_e(e \in \mathcal{A}), \mathbf{s}_a(a \in AP))$ of order n be in relation $CM_1 \sim CM_2$. Then

$$\mathbf{R}_a\mathbf{G}_e\mathbf{V} = \mathbf{R}_a\mathbf{V}\mathbf{H}_e = \mathbf{V}\mathbf{S}_a\mathbf{H}_e$$

for all $a \in AP$, all $e \in \mathcal{A}$. Observe that $\mathbf{R}_a\mathbf{V} = \mathbf{V}\mathbf{S}_a$ implies $\mathbf{r}_a = \mathbf{V}\mathbf{s}_a$ since

$$\mathbf{R}_a\mathbf{V}\mathbb{1} = \mathbf{R}_a\mathbb{1} = \mathbf{r}_a \text{ and } \mathbf{R}_a\mathbf{V}\mathbb{1} = \mathbf{V}\mathbf{S}_a\mathbb{1} = \mathbf{V}\mathbf{s}_a.$$

Observe that

$$\begin{aligned} e^{t\tilde{\mathbf{G}}}\mathbf{V} &= \sum_{h=0}^{\infty} \frac{t\tilde{\mathbf{G}}^h}{h!}\mathbf{V} &&= \sum_{h=0}^{\infty} \frac{t\tilde{\mathbf{G}}^h\mathbf{V}}{h!} \\ &= \mathbf{V}\left(t\mathbf{I} + t\left(\tilde{\mathbf{H}}\right)\right) + \sum_{h=2}^{\infty} \frac{t\tilde{\mathbf{G}}^h\mathbf{V}}{h!} &&= \mathbf{V}\sum_{h=0}^{\infty} \frac{t\tilde{\mathbf{H}}^h}{h!} \qquad (8) \\ &= \mathbf{V}e^{t\tilde{\mathbf{H}}_e}. \end{aligned}$$

Using Eq. 8 we have for an arbitrary observable behavior $\varsigma \in \Upsilon$ that

$$\begin{aligned} Dens_{(\psi,\mathbf{H}_e,\mathbf{s}_a)}(\varsigma) &= \psi\left(\prod_{h=0}^{|\varsigma|-1} \mathbf{S}_{a_h} e^{-t_h\tilde{\mathbf{H}}}\mathbf{H}_{e_h}\right)\mathbf{s}_{a_{|\varsigma|}} \\ &= \mu\mathbf{V}\left(\prod_{h=0}^{|\varsigma|-1} \mathbf{S}_{a_h} e^{-t_h\tilde{\mathbf{H}}}\mathbf{H}_{e_h}\right)\mathbf{s}_{a_{|\varsigma|}} \\ &= \mu\mathbf{R}_{a_h} e^{-t_h\tilde{\mathbf{G}}}\mathbf{G}_{e_h}\mathbf{V}\left(\prod_{h=1}^{|\varsigma|-1} \mathbf{S}_{a_h} e^{-t_h\tilde{\mathbf{H}}}\mathbf{H}_{e_h}\right)\mathbf{s}_{a_{|\varsigma|}} \\ &= \mu\left(\prod_{h=1}^{|\varsigma|-1} \mathbf{R}_{a_h} e^{-t_h\tilde{\mathbf{G}}}\mathbf{G}_{e_h}\right)\mathbf{V}\mathbf{s}_{a_{|\varsigma|}} \\ &= \mu\left(\prod_{h=1}^{|\varsigma|-1} \mathbf{R}_{a_h} e^{-t_h\tilde{\mathbf{G}}}\mathbf{G}_{e_h}\right)\mathbf{r}_{a_{|\varsigma|}} \end{aligned}$$

For the proofs of $\simeq$ and $\cong$ the relations

$$\mathbf{W}e^{t\tilde{\mathbf{G}}} = e^{t\tilde{\mathbf{H}}}\mathbf{W} \text{ and } e^{t\tilde{\mathbf{G}}}\mathbf{U} = \mathbf{U}e^{t\tilde{\mathbf{H}}}$$

can be shown similarly to Eq. 8. Then the proofs follow immediately by observing

$$\mathbf{W}\mathbf{R}_a\mathbf{G}_e = \mathbf{S}_a\mathbf{W}\mathbf{G}_e = \mathbf{S}_a\mathbf{H}_e\mathbf{W} \text{ and } \mathbf{R}_a\mathbf{G}_e\mathbf{U} = \mathbf{R}_a\mathbf{U}\mathbf{H}_e = \mathbf{U}\mathbf{S}_a\mathbf{H}_e.$$

Proof of Theorem 2

We have to prove the theorem for the 3 equivalences used in Fig. 1. If each equivalence preserves the result of dCSL formulas, then the same holds for relation $\approx$. The detailed proof for relation $\sim$ will be presented, the proofs for $\simeq$ and $\cong$ are very similar.

For the following proofs we consider $C\mathcal{M}_1 = (\varphi, \mathbf{G}, \mathbf{r}_a)$ of order m and $C\mathcal{M}_2 = (\upsilon, \mathbf{H}, \mathbf{s}_a)$ of order n ($< m$) that are in relation $\sim$ which implies $\varphi\mathbf{V} = \upsilon$, $\mathbf{GV} = \mathbf{VH}$ and $\mathbf{R}_a\mathbf{V} = \mathbf{VS}_a$. We first prove the theorem for state formulas Φ that do not contain an until or next operator.

If $\Phi = tt$, then $\mathbf{R}_\Phi = \mathbf{R}_{tt} = \mathbf{I}_m$ and $\mathbf{I}_m\mathbf{V} = \mathbf{VI}_n$. Observe that the first identity matrix is of order m and the second of order n. If $\Phi = a \in AP$, then $\mathbf{R}_a\mathbf{V} = \mathbf{VS}_a = \mathbf{VI}_n[a]$ by assumption.

Now assume that $\mathbf{R}_{\Phi_1}\mathbf{V} = \mathbf{VR}_{\Phi_1}$ and $\mathbf{R}_{\Phi_2}\mathbf{V} = \mathbf{VR}_{\Phi_2}$, then $\mathbf{R}_{\neg\Phi_1} = \mathbf{I} - \mathbf{R}_{\Phi_1}$ and $\mathbf{R}_{\Phi_1\wedge\Phi_2} = \mathbf{R}_{\Phi_1} \cdot \mathbf{R}_{\Phi_2}$ such that

$$\begin{aligned}\mathbf{R}_{\neg\Phi_1}\mathbf{V} &= \mathbf{I}_m\mathbf{V} - \mathbf{R}_{\Phi_1}\mathbf{V} = \mathbf{VI}_n - \mathbf{VR}_{\Phi_1} = \mathbf{VR}_{\neg\Phi_1},\\ \mathbf{R}_{\Phi_1\wedge\Phi_2}\mathbf{V} &= \mathbf{R}_{\Phi_1}\mathbf{R}_{\Phi_2}\mathbf{V} = \mathbf{R}_\Phi\mathbf{VS}_{\Phi_2} = \mathbf{VS}_{\Phi_1}\mathbf{S}_{\Phi_2} = \mathbf{VS}_{\Phi_1\wedge\Phi_2}.\end{aligned}$$

Consequently, the relations between the matrices holds for all formulas built from the logical composition of atomic propositions.

We continue with the analysis of $\mathcal{S}_{\bowtie p}(\Phi)$ and assume $\mathbf{R}_\Phi\mathbf{V} = \mathbf{VS}_\Phi$ which implies $\mathbf{r}_\Phi = \mathbf{Vs}_\Phi$. Then let $\bar\varphi = \lim_{t\to\infty}\varphi e^{t(\mathbf{G}-\bar{\mathbf{G}})}$ and $\bar\upsilon = \lim_{t\to\infty}\upsilon e^{t(\mathbf{H}-\bar{\mathbf{H}})}$. Since $\mathbf{GV} = \mathbf{VH}$ and $\bar{\mathbf{G}}\mathbf{V} = \mathbf{V}\bar{\mathbf{H}}$ the relation

$$\lim_{t\to\infty}\upsilon e^{t(\mathbf{H}-\bar{\mathbf{H}})} = \lim_{t\to\infty}\varphi e^{t(\mathbf{G}-\bar{\mathbf{G}})}\mathbf{V}$$

can be easily shown in a similar way as Eq. 8. Therefore, also $\bar\varphi\mathbf{V} = \bar\upsilon$ holds. This implies

$$\bar\varphi\mathbf{r}_\Phi = \bar\varphi\mathbf{Vs}_\Phi = \bar\upsilon\mathbf{s}_\Phi \text{ and } \varphi \models \mathcal{S}_{\bowtie p}(\Phi) \;\Leftrightarrow\; \upsilon \models \mathcal{S}_{\bowtie p}(\Phi).$$

Now we consider the path formula $\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2$. According to (2) we have

$$\begin{aligned}\mathbf{q}^1_{\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2} &= e^{t_1(\mathbf{G}[\Phi_1]-\bar{\mathbf{G}}[\Phi_1])}\mathbf{R}_{\Phi_1}e^{t(\mathbf{G}[\Phi_1\wedge\neg\Phi_2]-\bar{\mathbf{G}}[\Phi_1\wedge\neg\Phi_2])}\mathbf{r}_{\Phi_2},\\ \mathbf{q}^2_{\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2} &= e^{t_1(\mathbf{H}[\Phi_1]-\bar{\mathbf{H}}[\Phi_1])}\mathbf{S}_{\Phi_1}e^{t(\mathbf{H}[\Phi_1\wedge\neg\Phi_2]-\bar{\mathbf{G}}[\Phi_1\wedge\neg\Phi_2])}\mathbf{s}_{\Phi_2}.\end{aligned}$$

We assume that $\mathbf{R}_{\Phi_i}\mathbf{V} = \mathbf{VS}_{\Phi_i}$ holds for $i = 1, 2$, i.e. formulas Φ_1, Φ_2 have already been evaluated and equivalence has been proved. Since $(\mathbf{G} - \bar{\mathbf{G}})\mathbf{V} = \mathbf{V}(\mathbf{H} - \bar{\mathbf{H}})$ we also have

$$(\mathbf{G}[\Phi_i]-\bar{\mathbf{G}}[\Phi_i])\mathbf{V} = \mathbf{R}_{\Phi_i}(\mathbf{G}-\bar{\mathbf{G}})\mathbf{V} = \mathbf{VS}_{\Phi_i}(\mathbf{H}-\bar{\mathbf{H}}) = \mathbf{V}(\mathbf{H}[\Phi_i]-\bar{\mathbf{H}}[\Phi_i])$$

and

$$\left(\mathbf{G}[\Phi_1\wedge\neg\Phi_2] - \bar{\mathbf{G}}[\Phi_1\wedge\neg\Phi_2]\right)\mathbf{V} = \mathbf{V}\left(\mathbf{H}[\Phi_1\wedge\neg\Phi_2] - \bar{\mathbf{G}}[\Phi_1\wedge\neg\Phi_2]\right).$$

Then

$$\begin{aligned}\mathbf{q}^1_{\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2} &= e^{t_1(\mathbf{G}[\Phi_1]-\bar{\mathbf{G}}[\Phi_1])}\mathbf{R}_{\Phi_1}e^{t(\mathbf{G}[\Phi_1\wedge\neg\Phi_2]-\bar{\mathbf{G}}[\Phi_1\wedge\neg\Phi_2])}\mathbf{r}_{\Phi_2}\\ &= e^{t_1(\mathbf{G}[\Phi_1]-\bar{\mathbf{G}}[\Phi_1])}\mathbf{R}_{\Phi_1}e^{t(\mathbf{G}[\Phi_1\wedge\neg\Phi_2]-\bar{\mathbf{G}}[\Phi_1\wedge\neg\Phi_2])}\mathbf{Vs}_{\Phi_2}\\ &= e^{t_1(\mathbf{G}[\Phi_1]-\bar{\mathbf{G}}[\Phi_1])}\mathbf{R}_{\Phi_1}\mathbf{V}e^{t(\mathbf{H}[\Phi_1\wedge\neg\Phi_2]-\bar{\mathbf{H}}[\Phi_1\wedge\neg\Phi_2])}\mathbf{s}_{\Phi_2}\\ &= \mathbf{V}e^{t_1(\mathbf{H}[\Phi_1]-\bar{\mathbf{H}}[\Phi_1])}\mathbf{S}_{\Phi_1}e^{t(\mathbf{H}[\Phi_1\wedge\neg\Phi_2]-\bar{\mathbf{H}}[\Phi_1\wedge\neg\Phi_2])}\mathbf{s}_{\Phi_2}\\ &= \mathbf{V}\mathbf{q}^2_{\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2}\end{aligned}$$

which implies

$$\varphi\mathbf{q}^1_{\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2} = \varphi\mathbf{V}\mathbf{q}^2_{\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2} = \upsilon\mathbf{q}^2_{\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2}$$

and

$$\varphi \models \mathcal{DP}_{\bowtie p}\left(\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2\right) \;\Leftrightarrow\; \upsilon \models \mathcal{DP}_{\bowtie p}\left(\Phi_1\mathcal{U}^{[t_1,t_2]}\Phi_2\right).$$

Finally, we analyze $X_I(\Phi)$ and define

$$\mathbf{q}^1_{X_{[t_1,t_2]}\Phi} = e^{-t_1\bar{\mathbf{G}}}\int_{t=t_1}^{t_2} e^{-t\bar{\mathbf{G}}}\mathbf{G}\mathbf{r}_\Phi\,\mathrm{d}t \text{ and } \mathbf{q}^2_{X_{[t_1,t_2]}\Phi} = e^{-t_1\bar{\mathbf{H}}}\int_{t=t_1}^{t_2} e^{-t\bar{\mathbf{H}}}\mathbf{H}\mathbf{s}_\Phi\,\mathrm{d}t.$$

Then

$$\begin{aligned}\mathbf{q}^1_{X_{[t_1,t_2]}\Phi} &= e^{-t_1\bar{\mathbf{G}}}\int_{t=t_1}^{t_2} e^{-t\bar{\mathbf{G}}}\mathbf{G}\mathbf{r}_\Phi\,\mathrm{d}t &&= e^{-t_1\bar{\mathbf{G}}}\int_{t=t_1}^{t_2} e^{-t\bar{\mathbf{G}}}\mathbf{G}\mathbf{V}\mathbf{s}_\Phi\,\mathrm{d}t\\ &= e^{-t_1\bar{\mathbf{G}}}\int_{t=t_1}^{t_2} \mathbf{V}e^{-t\bar{\mathbf{H}}}\mathbf{H}\mathbf{s}_\Phi\,\mathrm{d}t &&= e^{-t_1\bar{\mathbf{G}}}\mathbf{V}\int_{t=t_1}^{t_2} e^{-t\bar{\mathbf{H}}}\mathbf{H}\mathbf{s}_\Phi\,\mathrm{d}t\\ &= \mathbf{V}e^{-t_1\bar{\mathbf{H}}}\mathbf{V}\int_{t=t_1}^{t_2} e^{-t\bar{\mathbf{H}}}\mathbf{H}\mathbf{s}_\Phi\,\mathrm{d}t &&= \mathbf{V}\mathbf{q}^2_{X_{[t_1,t_2]}\Phi}\end{aligned}$$

which implies

$$\varphi\mathbf{q}^1_{X_{[t_1,t_2]}(\Phi)} = \varphi\mathbf{V}\mathbf{q}^2_{X_{[t_1,t_2]}(\Phi)} = \upsilon\mathbf{q}^2_{X_{[t_1,t_2]}(\Phi)}$$

and

$$\varphi \models \mathcal{DP}_{\bowtie p}\left(X_{[t_1,t_2]}(\Phi)\right) \;\Leftrightarrow\; \upsilon \models \mathcal{DP}_{\bowtie p}\left(X_{[t_1,t_2]}(\Phi)\right)$$

A recursive application of the transformations implies that the identities hold for all state and path formulas. The proofs for the relations $\simeq$ and $\cong$ are again similar. Since the equivalence between dCSL formulas holds for all transformations, it also holds along the diagram in Fig. 1.

Proof of Theorem 3

Again we show the proof for $\sim$, the proofs for the remaining relations are similar. We consider $C\mathcal{M}_1 = (\varphi, \mathbf{G}, \mathbf{r}_a)$ of order m and $C\mathcal{M}_2 = (\upsilon, \mathbf{H}, \mathbf{s}_a)$ of order n ($\leq m$) with $C\mathcal{M}_1 \sim C\mathcal{M}_2$ which implies the existence of a matrix $\mathbf{V}$ such that $\varphi\mathbf{V} = \upsilon$, $\mathbf{GV} = \mathbf{VH}$ and $\mathbf{R}_a\mathbf{V} = \mathbf{VS}_a$. It is sufficient to show equivalent behavior on path formulas, as the rest follows from Theorem 2. We consider a dasCSL path formula α^I. For the path formula first a non-deterministic program automata is generated which depends only on α. Then from the non-deterministic automaton and the CTLMC a deterministic automaton is generated. The construction depends only on the accepting paths of the CTLMC (see [3, Sect. 6.1]) which means that for automata in relation $\approx$ the same deterministic automaton is generated since the paths are identical due to the equivalence of the processes. Thus, we obtain a finite acceptor $\mathcal{DA} = (Z, \mathcal{B}, \delta, z_0, \mathcal{F})$. The satisfiability properties of α^I in $C\mathcal{M}_1$ and $C\mathcal{M}_2$ depend on the composed automata $C\mathcal{M}'_1 = \mathcal{DA} \times C\mathcal{M}_1 = (Z \times S_1, \upsilon_1, \emptyset, \mathbf{F}_1, AP', L')$ and $C\mathcal{M}'_2 = \mathcal{DA} \times C\mathcal{M}_2 = (Z \times S_2, \upsilon_2, \emptyset, \mathbf{F}_2, AP', L')$. Let $\varphi \models \mathcal{DP}_{\bowtie p}\left(\alpha^I\right)$. We want to show that it is equivalent to $\upsilon \models \mathcal{DP}_{\bowtie p}\left(\alpha^I\right)$. For this, we consider the matrix

$$\mathbf{V}' = \begin{pmatrix}\mathbf{V} & 0 & \cdots & 0\\ 0 & \mathbf{V} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & \cdots & 0 & \mathbf{V}\end{pmatrix}.$$

It is easy to see that $\mathbf{F}_1\mathbf{V}'$ will be a matrix consisting of submatrices which have the form $\sum_{(\Phi,a):\delta(\Phi,a)=z'}\mathbf{R}_\Phi\mathbf{G}_a\mathbf{V}$ for $z, z' \in Z$. By assumption, we know that this is equal to $\sum_{(\Phi,a):\delta(\Phi,a)=z'}\mathbf{R}_\Phi\mathbf{V}\mathbf{H}_a = \mathbf{V}\sum_{(\Phi,a):\delta(\Phi,a)=z'}\mathbf{S}_\Phi\mathbf{H}_a$, so we get $\mathbf{F}_1\mathbf{V}' = \mathbf{V}'\mathbf{F}_2$. Also by assumption we get $\upsilon_1\mathbf{V}' = \mathbf{V}'\upsilon_2$. For label vectors $\mathbf{t}^1_a$ in $C\mathcal{M}_1$ and $\mathbf{t}^2_a$ in $C\mathcal{M}_2$, it is also easy to see that they are products of Z and $\mathbf{r}_a$ and respectively, Z and $\mathbf{s}_a$; thus, we also have $\mathbf{T}^1_a\mathbf{V}' = \mathbf{V}'\mathbf{T}^2_a$. Thus, it follows that the product automata are bisimilar, and thus, $\varphi \models \mathcal{DP}_{\bowtie p}\left(\alpha^I\right) \iff \upsilon \models \mathcal{DP}_{\bowtie p}\left(\alpha^I\right)$.