CSL Model Checking for Generalized Stochastic Petri Nets*

Davide Cerotti, Susanna Donatelli, András Horváth, and Jeremy Sproston
Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy
{cerotti,susi,horvath,sproston}@di.unito.it

Abstract

*This paper presents a Continuous Stochastic Logic (*CSL*) model-checking algorithm for Generalized Stochastic Petri Nets (GSPNs).* CSL *is a temporal logic defined over Continuous Time Markov Chains (CTMCs). GSPNs are a class of Stochastic Petri Nets in which sojourn times in states are either exponentially distributed (tangible states) or deterministically zero (vanishing states). Although vanishing states have zero probabilities, they can be relevant for the definition of system properties expressed as* CSL *formulae: the semantics of* CSL *is therefore modified accordingly. The paper then shows how the set of GSPN states which satisfy a* CSL *formula can be computed through the solution of CTMCs produced from a series of embedded Discrete Time Markov Chains modified according to the formula being checked.*

1. Introduction

Generalized Stochastic Petri Nets (GSPNs) [1] are a class of Stochastic Petri Nets in which sojourn times in states are either exponentially distributed (tangible states) or deterministically zero (vanishing states). Transient and steady state probabilities of GSPN states can be computed from a CTMC defined over the tangible states, with the corresponding probabilities of vanishing states being zero. GSPNs have been used widely for the computation of performance evaluation metrics. As has been recognized in, for example, the pioneering works [9] and [21], it may be important to express metrics that relate to execution paths. More precisely, in [21] a class of path-based reward variables is introduced, where an automaton is used for the specification of the set of paths of interest, while in [9] the reward language is based on a simple modal logic.

Continuous Stochastic Logic (CSL) [3, 6] is an extension of the temporal logic CTL [10] for the specification of guarantees on system performance, including path-based measures. The logic CSL includes a probabilistic path operator to reason about the probability with which a certain (possibly timed) property is satisfied: for example, "does the system reach an error state, without passing through a recovery state, within 5 minutes with probability greater than 0.01". This property requires the accumulation of probability along the subset of paths representing executions that do not pass through recovery states, and can be computed using transient analysis in which the recovery states and error states are made absorbing [6]. The logic CSL also includes a steady-state operator which can refer to the probability of the system being in certain states in equilibrium.

CSL model-checking algorithms for CTMCs are implemented in the tools ETMCC [14], PRISM [15], MRMC [18], and the APNN toolbox [8]. CSL model checking of GSPNs is possible in tools like GreatSPN [11] (via an export to MRMC and to PRISM) and the APNN toolbox, but in all cases only a model-dependent subclass of CSL formulae, which are invariant under the elimination of vanishing states, can be verified [8]. However, vanishing states can play a role in the definition of formulae, as we show in the next section. Even when it is possible to rephrase the property in terms of tangible states only, this may not be convenient or intuitive.

In this paper we extend CSL to deal with GSPNs. To achieve this goal, in Section 2 we first describe the stochastic process over vanishing and tangible states corresponding to a GSPN, which we will call a Vanishing Continuous-Time Markov Chain (VCTMC), and then, in Section 3, we define the semantics of CSL over VCTMCs. In Section 4, we show how to modify the CSL model-checking algorithm for CTMCs proposed by Baier et al. [6] to the case of VCTMCs. The paper concludes with an example of model checking a cyclic polling system modelled as a GSPN.

Related work. A recent paper developed concurrently to ours has the objective to define a model-checking algorithm for IM-SPDL on stochastic systems with vanishing states (ESLTS) [20]. The logic IM-SPDL takes into account labels of states and *actions*, and allows the specification of complex properties on execution paths using program-like expressions. We note that IM-SPDL is more powerful than

* Supported in part by Miur project Firb-Perf and EEC project Crutial.

CSL, and as a consequence the model-checking algorithm is more complicated and requires the construction of a product of a formula automaton with the ESLTS to produce a modified ESLTS, which is then reduced to a CTMC by eliminating vanishing states; this CTMC is subsequently model-checked against a probabilistic timed reachability property of CSL. The reduced CTMCs we define in this paper are similar or identical to those generated by the construction of [20] (the paper [20] does not enter into the details of open and closed intervals at 0, and point intervals) .

A CSL model-checking algorithm for semi-Markov chains is presented in [17]. It assumed that, in all states, some time elapses with positive probability; this assumption means that considered model cannot represent faithfully VCTMCs. A variant of CSL, called eCSL, has been defined on semi-Markov stochastic Petri nets (SM-SPNs), which are Petri nets in which the sojourn time in states follows a general distribution [7]. GSPNs can be considered as a subclass of SM-SPNs in which sojourn times are either exponential or deterministically null. From a SM-SPN it is possible to generate a semi-Markov process; however, the semi-Markov process used in the semantics of eCSL is defined over tangible states only. A model-checking algorithm for stochastic transition systems has been developed in [12]; we note that the subclass of stochastic transition systems in which nondeterminism is not permitted corresponds to VCTMCs. However, the temporal logic considered in [12] does not permit CSL-like time-bounds in path formulae.

2. GSPNs and motivating examples

2.1. Introduction to GSPNs

Let $\mathbb{R}_{\geq 0}$ be the set of non-negative reals, and $\mathbb{R}_{>0}$ be the set of positive reals.

Definition 2.1 *A* generalized stochastic Petri net *[1] (GSPN) is a tuple $\mathcal{G} = (P, T, \mathsf{pri}, \mathrm{I}, \mathrm{O}, \mathrm{H}, W, \mathsf{m0})$, where P is the set of places, T is the set of transitions,* $\mathsf{pri} : T \to \mathbb{N}$ *is a priority function,* $\mathrm{I}, \mathrm{O}, \mathrm{H} : T \to P \to \mathbb{N}$ *define the input, output, and inhibitor arcs with associated multiplicity, respectively, $W : T \to \mathbb{R}_{\geq 0}$ assigns a non-negative real to each transition, and* $\mathsf{m0} : P \to \mathbb{N}$ *describes the initial marking.*

Transitions $\mathsf{t} \in T$ with $\mathsf{pri}(\mathsf{t}) \geq 1$ are called *immediate* and fire in zero time, while transitions with $\mathsf{pri}(\mathsf{t}) = 0$ are called *timed* and have an exponentially-distributed delay. The firing rule of classical Petri nets is changed accordingly to take priorities into account. This induces a partition of T into the set T_{imm} of immediate transitions and the set T_{tim} of timed transitions. For a transition $\mathsf{t} \in T$: if $\mathsf{t} \in T_{\mathsf{imm}}$ then $W(\mathsf{t})$ is the "weight" of the transition in comparison with other immediate transitions, and is used to compute the probability that t fires in a given marking; if $\mathsf{t} \in T_{\mathsf{tim}}$, then $W(\mathsf{t})$ is the rate of the exponential distribution associated with t.

We let RS denote the reachability set of a GSPN (the set of all markings reachable from m0). As usual we use interchangeably the terms "state" and "marking". Due to the presence of immediate transitions we distinguish *vanishing* states, in which there is at least one outgoing immediate transition (therefore the sojourn time is deterministically zero), from *tangible* states, in which time elapses. We indicate with VRS and TRS the set of vanishing and tangible states respectively. If s and s' are states, we indicate with $s[\mathsf{t}\rangle s'$ the fact that the firing of transition t in s produces the state s'. We let RG denote the reachability graph defined over RS: nodes are the states of RS, and there is an arc labelled with transition t from node s to node s' if and only if $s[\mathsf{t}\rangle s'$. We let TRG denote the tangible reachability graph, the nodes of which are in one-to-one correspondence with tangible states. There is an arc of TRG from s to s', where s, s' are tangible states, if and only if $s[\mathsf{t}\rangle s'$, where t is a timed transition, (in which case the arc is labelled with t), or through the firing of a timed transition followed by a sequence of immediate transitions leading to s' (in which case the arc is labelled with t followed by a sequence of immediate transitions).

2.2. Underlying stochastic process of a GSPN

2.2.1. Markov chains. We now define the notation for discrete- and continuous-time Markov chains. We consider Markov chains extended with a function which labels each state with a set of atomic propositions interpreted as being valid in that state. Let AP be a set of atomic propositions. A *probability distribution* on a set of elements S is a function $\alpha : S \to [0,1]$ such that $\sum_{s \in S} \alpha(s) = 1$.

Definition 2.2 *A* labelled discrete-time Markov chain (DTMC) D *is denoted by a quadruple $(S, \mathbf{P}, \textsc{init}, L)$ comprising a finite set S of states, a* probability matrix $\mathbf{P} : S \times S \to [0,1]$ *of conditional probabilities (also called a probability transition matrix, and which is such that $\sum_{s' \in S} \mathbf{P}(s, s') = 1$ for each state $s \in S$), an* initial probability distribution $\textsc{init}$ *on S, and a* labelling function $L : S \to 2^{AP}$.

As usually when dealing with CSL, we use the rate transition matrix $\mathbf{R}$ instead of the infinitesimal generator matrix in the description of continuous-time Markov chains.

Definition 2.3 *A* labelled continuous-time Markov chain (CTMC) C *is denoted by a quadruple $(S, \mathbf{R}, \textsc{init}, L)$, comprising a finite set S of states, a* rate matrix $\mathbf{R} : S \times S \to \mathbb{R}_{\geq 0}$ *of rates (also called a rate transition matrix), an* initial probability distribution $\textsc{init}$ *on* S*, and a* labelling function $L : S \rightarrow 2^{AP}$.

The interpretation of the rate transition matrix is that $\mathbf{R}(s, s') > 0$ if and only if there exists a CTMC transition from state s to state s', and that the probability that this transition is triggered within δ time units is $1 - e^{-\mathbf{R}(s,s')\cdot\delta}$. Let the *exit rate* $\mathrm{E}(s)$ for the state $s \in S$ be defined by $\mathrm{E}(s) = \sum_{s' \in S} \mathbf{R}(s, s')$. A state s is called *absorbing* if and only if $\mathrm{E}(s) = 0$. We denote the usual notion of transient probability of being in state s at time $\delta \in \mathbb{R}_{\geq 0}$, given that the CTMC starts with the probability distribution α, by $\pi^{\mathsf{C}}(\alpha, s, \delta)$. Similarly, the steady-state probability of being in state set $S' \subseteq S$, given that the CTMC starts with the distribution α, is denoted by $\pi^{\mathsf{C}}(\alpha, S')$.

2.2.2. Vanishing CTMCs. In most of the papers on GSPNs, the stochastic process of a GSPN is classified as a semi-Markov process in which sojourn time in states is exponentially distributed or deterministically zero. However, when looking carefully at the stochastic process of a GSPN, we can observe that defining such a process as a semi-Markov process may not be an appropriate choice. The same basic definition of a continuous stochastic process does not apply when the sample space is the whole reachability set RS. Consider that, in a stochastic process $X(\delta)$, the state at time $\delta \in \mathbb{R}_{\geq 0}$ is defined over the set of states, which in our case is RS; however, if δ is the time of firing of a timed transition followed by a sequence of immediate transitions, then $X(\delta)$ can evaluate to a *subset* of states. We could consider a definition in which $X(\delta)$ is equal to the last tangible state, but then $X(\delta)$ is defined over TRS only, in which case CTMCs is enough. Alternatively, $X(\delta)$ could be equal to a sequence of states of RS.

Even if the semi-Markov process is not totally adequate to describe the semantics of a GSPN, we can observe that the embedded DTMC defined as "$X(n) =$ state of the system after the n-th change of state" is well-defined over RS. We can compute $P_{ss'}$, the probability of reaching state s' from s in one step, using the rates of the timed transitions enabled in s, if s is a tangible state, and the weights of the immediate transitions enabled in s, if s is vanishing, as described in [2]. The only missing information is the sojourn time in states, although again this is easily recoverable from the rates of timed transitions enabled in a state.

In the next paragraph we define the stochastic process associated to a GSPN through the embedded DTMC defined over RS plus the sojourn times, which we call a *CTMC with vanishing states* (VCTMC). Our approach to the definition of VCTMCs is inspired by the definition of CTMCs given by Kulkarni [19], where a stochastic process is a CTMC if it has an embedded DTMC that describes the change of state and if sojourn times are exponentially distributed.

Definition 2.4 *A labelled CTMC with vanishing states (VCTMC) is a quadruple* $\mathsf{V} = (\mathsf{D}, S_T, S_V, \Lambda)$ *where* $\mathsf{D} = (S, \mathbf{P}, \textsc{init}, L)$ *defines a labelled DTMC with a finite set* S *of states, such that* $S_T \cup S_V = S$ *and* $S_V \cap S_T = \emptyset$*, and* $\Lambda = \{\Lambda_s | s \in S\}$ *is a set of random variables which describe the sojourn time in states of* D*, with the constraint that* Λ_s *is deterministically zero if* $s \in S_V$ *(vanishing state), and is exponentially distributed otherwise (tangible state).*

If X_n *is the random variable that describes the state of* D *after the* n*-th state change and if* Y_n *is the random variable that describes the sojourn time in state* X_n *then, given states* $s, s' \in S$ *and duration* $\delta \in \mathbb{R}_{\geq 0}$*, we can define the following joint CDF of* V*:*

$$P\{X_{n+1} = s', Y_n \geq \delta | X_n = s, X_{n-1}, \cdots, Y_1, X_0\}$$
$$= \begin{cases} \mathbf{P}(s, s') \cdot e^{-\lambda(s)\cdot\delta} & \textit{if } s \in S_T \\ \mathbf{P}(s, s') & \textit{if } s \in S_V \wedge \delta = 0 \\ 0 & \textit{if } s \in S_V \wedge \delta > 0 \end{cases}$$

where $\lambda(s)$ *is the rate of the exponential distribution associated with the random variable* Λ_s *of a tangible state* $s \in S_T$*.*

For a VCTMC $\mathsf{V} = (\mathsf{D}, S_T, S_V, \Lambda)$, we consider the probability matrix $\mathbf{P}$ of D partitioned in the following way:

$$\mathbf{P} = \begin{vmatrix} \mathbf{P}_{VV} & \mathbf{P}_{VT} \\ \mathbf{P}_{TV} & \mathbf{P}_{TT} \end{vmatrix}$$

where $\mathbf{P}_{VV}$ contains transition probabilities from a vanishing state to a vanishing state, $\mathbf{P}_{VT}$ from a vanishing state to a tangible state, etc. Throughout this paper, to avoid behavior in which time does not exceed some bound, we assume that $\lim_{n\rightarrow\infty} \mathbf{P}^n_{VV} = 0$; that is, the probability of remaining within the set of vanishing states without ever reaching a tangible state converges to 0 in the limit. This condition can be checked by computing the strongly-connected components of the directed graph of D restricted to states in S_V; such a strongly-connected component exists if and only if $\lim_{n\rightarrow\infty} \mathbf{P}^n_{VV} \neq 0$.

VCTMC of a GSPN. We now define the VCTMC of a GSPN. Let $s[\mathsf{t}\rangle$ denote the fact that t is enabled in s.

Definition 2.5 *The VCTMC of a GSPN* $\mathcal{G}$ *is denoted by* $\mathsf{V}(\mathcal{G}) = ((S, \mathbf{P}, \textsc{init}, L), S_T, S_V, \Lambda)$*, where* $S_T = \mathrm{TRS}$ *and* $S_V = \mathrm{VRS}$*, where the probability transition matrix* $\mathbf{P}$ *is defined, for all states* $s, s' \in S$*, by:*

$$\mathbf{P}(s, s') = \begin{cases} \frac{\sum_{\mathsf{t}:s[\mathsf{t}\rangle s'} W(\mathsf{t})}{\sum_{\mathsf{t}:s[\mathsf{t}\rangle} W(\mathsf{t})} & \textit{if } \exists \mathsf{t} \textit{ such that } s[\mathsf{t}\rangle \\ 1 & \textit{if } \nexists \mathsf{t} \textit{ such that } s[\mathsf{t}\rangle \textit{ and } s = s' \\ 0 & \textit{otherwise,} \end{cases}$$

where $\textsc{init}$ *is such that* $\textsc{init}(\mathsf{m0}) = 1$ *for the initial marking* $\mathsf{m0}$ *of* $\mathcal{G}$*, and* $\textsc{init}(s) = 0$ *for all other markings* $s \in S \setminus \{\mathsf{m0}\}$*, where* $L(s) = at_s$*, and where* Λ *is defined by* $\lambda(s) = \sum_{\mathsf{t}:s[\mathsf{t}\rangle} W(\mathsf{t})$ *for each tangible state* $s \in S_T$*.*

The set $L(s)$ can be extended to include arbitrary expressions on markings and enabling of transitions as in [11].

2.2.3. Solution process. There are two different ways of computing the transient and steady-state probabilities of a GSPN. If no cycle of immediate transitions is present then the easiest approach (followed by most GSPN tools) is to build the RG, reduce the RG to the TRG, and then to use the information on TRG arcs to compute the CTMC. If a cycle of immediate transitions exists the CTMC is computed passing through the embedded DTMC defined over RS, as described in [2]. We revisit this construction in our context in the following definition.

Definition 2.6 *Let* $\mathsf{V} = (\mathsf{D}, S_T, S_V, \Lambda)$ *be a VCTMC. The* reduced CTMC $\mathsf{R}[\mathsf{V}] = (S, \mathbf{R}, \textsc{init}, L)$ *of the VCTMC* V *is defined as follows: let* $S = S_T$*, let* L *be the labelling function of* D *projected over* S_T*, and let* $\mathbf{R}$ *and* $\textsc{init}$ *be defined as follows.*

First we consider the probability matrix $\mathbf{P}$ *of* D *partitioned into the matrices* $\mathbf{P}_{VV}$, $\mathbf{P}_{VT}$, $\mathbf{P}_{TV}$ *and* $\mathbf{P}_{TT}$*. The probability matrix* $\mathbf{P}'$ *among tangible states can be computed as* $\mathbf{P}' = \mathbf{P}_{TT} + \mathbf{P}_{TV}(I - \mathbf{P}_{VV})^{-1}\mathbf{P}_{VT}$ *where* I *is the identity matrix and* $(I - \mathbf{P}_{VV})^{-1}$ *takes into account the possible paths among vanishing states. Then the entries of the rate matrix* $\mathbf{R}$ *of the underlying CTMC are given as* $\mathbf{R}(s, s') = \lambda(s)\mathbf{P}'(s, s')$.

Now consider the initial distribution of D *as a vector partitioned into a vector* $\textsc{init}_V$ *over* S_V *and a vector* $\textsc{init}_T$ *over* S_T*. Then the initial distribution of* $\mathsf{R}[\mathsf{V}]$ *is defined by* $\textsc{init} = \textsc{init}_T + \textsc{init}_V(I - \mathbf{P}_{VV})^{-1}\mathbf{P}_{VT}$.

2.3. Motivating examples

To argue for the relevance of considering vanishing states in CSL, we consider two simple GSPNs, depicted in Figure 1 and Figure 2. Both model the execution of a job during which errors can occur: errors might be coverered or not by the error recovery strategy. The two models behave differently upon a successful recovery. In the first model, the error stops the execution and the resulting state of the system either corresponds to a token in place failed or, in the case the error is covered, in place completed. In the second model an uncovered error stops the execution and results in failure, while a covered error does not alter the job execution. The RG and TRG of the first and second models are shown in Figure 3 and Figure 4, respectively.

Assume that we are interested in the probability of a path along which no errors occur and in which the system arrives in a marking in which place completed is marked. The CSL formula that seems most adequate is $\Phi ::= \mathcal{P}_{\geq\rho}[(\mathsf{work} \wedge \mathsf{safe})\ U^{[0,\infty)}\ \mathsf{completed}]$ for some $\rho \in [0, 1]$ (where the atomic propositions used refer to marking of places in the obvious manner). For the GSPN of Figure 1, if Φ is checked on the TRG, the probability is accumulated over the paths $m_0 \rightarrow m_2$ and $m_0 \rightarrow m_3$, while on the RG only $m_0 \rightarrow m_2$ is considered, because the path $m_0 \rightarrow m_1 \rightarrow m_3$ passes through an unsafe state. For the GSPN of Figure 2 the probability is accumulated along the path $m_0 \rightarrow m_2$ on the RG and on paths of the form $(m_0 \rightarrow)^+ m_2$ in the TRG.

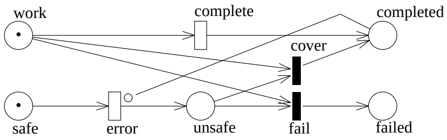

Figure 1. First simple GSPN

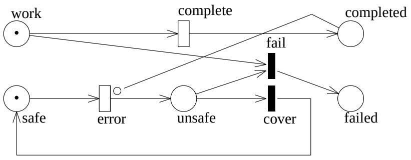

Figure 2. Second simple GSPN

Observe that an equivalent formula exists that is correctly computed on both the RG and the TRG for the first model: $\mathcal{P}_{\geq\rho}[(\mathsf{work} \wedge \mathsf{safe})\ U^{[0,\infty)}\ (\mathsf{completed} \wedge \mathsf{safe})]$. However, this equivalent formula does not help for the second model: the states involved in the paths over the RG and the TRG involve the same states, and since atomic propositions can be associated to states only, it is not possible to find a formula that is equivalent on the RG and on the TRG.

3. Continuous Stochastic Logic

3.1. Paths and probability measures

Paths of VCTMCs. Let $\mathsf{V} = (\mathsf{D}, S_T, S_V, \Lambda)$ be a VCTMC. If a tangible state $s \in S_T$ is such that $\lambda(s) = 0$, then we say that s is *absorbing*. If, for tangible state $s \in S_T$ and arbitrary state $s' \in S$, we have both $\mathbf{P}(s, s') > 0$ and $\lambda(s) > 0$, then, for any $\delta \in \mathbb{R}_{>0}$, we say that there exists a VCTMC transition of duration δ from state s to state s', denoted by $s \xrightarrow{\delta} s'$. If, for a vanishing state $s \in S_V$ and $s' \in S$, we have $\mathbf{P}(s, s') > 0$, then we say that there exists a VCTMC transition of duration 0 from state s to state s', denoted by $s \xrightarrow{0} s'$. An infinite path is a sequence $s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \cdots$ of VCTMC transitions. A finite path is a sequence $s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \cdots s_{n-1} \xrightarrow{\delta_{n-1}} s_n$ of VCTMC transitions such that s_n is absorbing. Let $Path^{\mathsf{V}}$ be the set of paths of V.

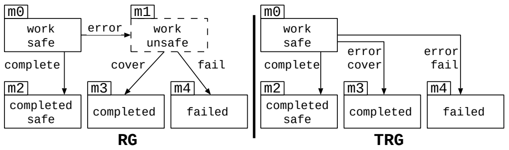

Figure 3. RG **and** TRG **of the GSPN of Fig. 1**

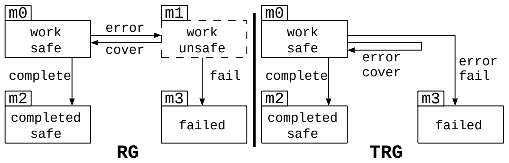

Figure 4. RG **and** TRG **of the GSPN of Fig. 2**

For any infinite path $\sigma = s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \cdots$ of a VCTMC and any $i \in \mathbb{N}$, let $\sigma(i) = s_i$, the $(i+1)$st state of σ, and let $\Delta(\sigma, i) = \delta_i$. Similarly, for any finite path $s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \cdots s_{n-1} \xrightarrow{\delta_{n-1}} s_n$ of a VCTMC, $\sigma(i)$ and $\Delta(\sigma, i)$ are defined only for $i \leq n$; we define $\sigma(i)$ as in the case of infinite paths, whereas $\Delta(\sigma, i)$ is defined as for infinite paths if $i < n$, and $\Delta(\sigma, n) = \infty$.

We now introduce notation to refer to the states occupied along a path at an exact time point. Consider the path in Figure 5, where s_i are states and t denotes time. At time $t = 9$ the system moves from tangible state s_1 to a vanishing state s_2, from where it evolves into the vanishing states s_3 and then s_4, to finally reach the tangible state s_5. It is then clear that, in contrast to the case of CTMCs, a path may occupy a *sequence* of states at a particular point in time; in our example, we say that the path occupies the sequence $\langle s_1 s_2 s_3 s_4 \rangle$ of states at time $t = 9$ (we assume that the VCTMC is left-continuous).

Definition 3.1 *Let* $\sigma = s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \cdots$ *be an* infinite *path such that* $\sum_{i \geq 0} \delta_i$ *does not converge. For* $\delta \in \mathbb{R}_{\geq 0}$, *let* $i \in \mathbb{N}$ *be the smallest index such that* $\delta \leq \sum_{j=0}^{i} \delta_j$. *If* $\delta < \sum_{j=0}^{i} \delta_j$, *then let* $\sigma @ \delta = \langle s_i \rangle$. *If* $\delta = \sum_{j=0}^{i} \delta_j$, *for the largest index* $k > i$ *such that* $\sum_{j=i+1}^{k} \delta_j = 0$, *let* $\sigma @ \delta = \langle s_i ... s_k \rangle$. *Let* $s_0 \xrightarrow{\delta_0} s_1 \xrightarrow{\delta_1} \cdots s_{n-1} \xrightarrow{\delta_{n-1}} s_n$ *be a* finite *path. For* $\delta > \sum_{j=0}^{n-1} \delta_j$, *let* $\sigma @ \delta = \langle s_n \rangle$; *otherwise,* $\sigma @ \delta$ *is defined as for infinite paths.*

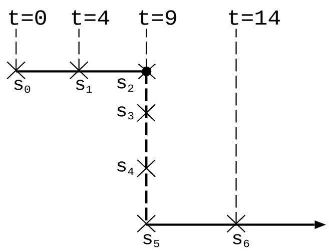

Figure 5. A sample path of a VCTMC

We often interpret a (possibly empty) sequence $\langle s_0 ... s_k \rangle$ as the set of its constituent states, so that we can write, for example, $s \in \langle s_0 ... s_k \rangle$, which is true if $s \in \{s_0, ..., s_k\}$. If $s \in \langle s_0 ... s_k \rangle$, then for some $0 \leq i \leq k$ we have $s = s_i$; then we let $\mathit{Pref}(\langle s_0 ... s_k \rangle, s) = \langle s_0 ... s_{i-1} \rangle$ if $i > 0$, and let $\mathit{Pref}(\langle s_0 ... s_k \rangle, s)$ be the empty sequence otherwise.

Probability measure of VCTMC paths. Given a state $s \in S$ of the VCTMC V, we now define the probability measure $\mathrm{Pr}_s^{\mathsf{V}}$ following the precedent of the analogous definition for CTMCs [6]. Consider the sequence $\langle s_0 ... s_k \rangle$ of states with $\mathbf{P}(s_i, s_{i+1}) > 0$ for $0 \leq i < k$, and the sequence $\langle I_0 ... I_{k-1} \rangle$ of non-empty intervals in $\mathbb{R}_{\geq 0}$. We use $C(s_0, I_0, ..., I_{k-1}, s_k)$ to denote the cylinder set consisting of paths $\sigma \in \mathit{Path}^{\mathsf{V}}$ such that $\sigma(i) = s_i$ for $0 \leq i \leq k$, and $\Delta(\sigma, i) \in I_i$ for $0 \leq i < k$. Let $\mathcal{F}_s(\mathit{Path}^{\mathsf{V}})$ be the smallest sigma-algebra on $\mathit{Path}^{\mathsf{V}}$ containing the sets $C(s_0, I_0, ..., I_{k-1}, s_k)$, where $s_0, I_0, ..., I_{k-1}, s_k$ ranges over the set of sequences of the form described above, and where $s_0 = s$. The probability measure $\mathrm{Pr}_s^{\mathsf{V}}$ on $\mathcal{F}_s(\mathit{Path}^{\mathsf{V}})$ is defined by induction on k by $\mathrm{Pr}_s^{\mathsf{V}}(C(s)) = 1$, and for $k \geq 0$:

$$\begin{aligned} & \mathrm{Pr}_s^{\mathsf{V}}(C(s_0, I_0, ..., s_k, I', s')) \\ = \;\; & \mathrm{Pr}_s^{\mathsf{V}}(C(s_0, I_0, ..., s_k)) \cdot \mathbf{P}(s_k, s') \cdot \rho \,, \end{aligned}$$

where if $s_k \in S_T$, then $\rho = (e^{-\lambda(s_k) \cdot \inf I'} - e^{-\lambda(s_k) \cdot \sup I'})$ (we let $e^{-\mu \cdot \infty} = 0$), if $s_k \in S_V$ and $0 \in I'$, then $\rho = 1$, otherwise $\rho = 0$.

The following proposition states that the probability of visiting a certain *tangible* state at a certain time point in a VCTMC is the same as the transient probability associated to the state and time point in the reduced Markov chain of the VCTMC. As there exist methods for the computation of transient probabilities on CTMCs, this provides us with methods for the computation of performance indices of VCTMCs (and on GSPNs) defined with regard to tangible states. We use α_s^1 to denote the probability distribution with probability 1 in the single element s. The subsequent proposition follows from the definition of $\mathrm{Pr}_s^{\mathsf{V}}$ and $\mathsf{R}[\mathsf{V}]$.

Proposition 3.2 *Let* V *be a VCTMC. Then for an arbitrary state* $s \in S$ *and a tangible state* $s' \in S_T$ *of* V, *and a duration* $\delta \in \mathbb{R}_{\geq 0}$, *we have* $\mathrm{Pr}_s^{\mathsf{V}}\{\sigma \in \mathit{Path}^{\mathsf{V}} \mid s' \in \sigma @ \delta\} = \pi^{\mathsf{R}[\mathsf{V}]}(\alpha_s, s', \delta)$, *where* α_s *is the distribution over tangible states defined by* $\alpha_s = \alpha_s^1 (I - \mathbf{P}_{VV})^{-1} \mathbf{P}_{VT}$ *if* $s \in S_V$, *and* $\alpha_s = \alpha_s^1$ *if* $s \in S_T$.

3.2. CSL syntax and semantics

We now recall the syntax of CSL [3, 6], and extend its semantics to the case of VCTMCs.

Definition 3.3 *The syntax of* CSL *is defined as* $\Phi ::= a \mid \Phi \wedge \Phi \mid \neg\Phi \mid \mathcal{P}_{\bowtie\rho}(X^I\Phi) \mid \mathcal{P}_{\bowtie\rho}(\Phi U^I \Phi) \mid \mathcal{S}_{\bowtie\rho}(\Phi)$, *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* $\rho \in [0,1]$ *is a probability.*

Informally, the interpretation of the *path formulae* $X^I\Phi$ and $\Phi_1 U^I \Phi_2$ is as follows: the *next* formula $X^I\Phi$ 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 *until* 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.

We now define formally the satisfaction relation $\models$, where $s \models \Phi$ indicates that the CSL formula Φ is satisfied in state s, for the case of VCTMCs. Given the satisfaction relation $\models$, let *Sat*$(\Phi) = \{s \in S \mid s \models \Phi\}$ be the set of states of V which satisfy Φ. We use $Prob^{\mathsf{V}}(s, \varphi)$ to denote the probability measure of paths from s satisfying φ; formally, we have $Prob^{\mathsf{V}}(s,\varphi) = \Pr_s^{\mathsf{V}}\{\sigma \in Path^{\mathsf{V}} \mid \sigma \models \varphi\}$.

Definition 3.4 *For* $\mathsf{V} = ((S, \mathbf{P}, \textsc{init}, L), S_T, S_V, \Lambda)$ *and state* $s \in S$*, the satisfaction relation* $\models$ *is defined as:*

$$\begin{array}{lcll}
s & \models & a & \textit{iff} \quad a \in L(s) \\
s & \models & \Phi_1 \wedge \Phi_2 & \textit{iff} \quad s \models \Phi_1 \textit{ and } s \models \Phi_2 \\
s & \models & \neg\Phi & \textit{iff} \quad s \not\models \Phi \\
s & \models & \mathcal{S}_{\bowtie\rho}(\Phi) & \textit{iff} \quad \pi^{\mathsf{R}[\mathsf{V}]}(\alpha_s, \textit{Sat}(\Phi) \cap S_T) \bowtie \rho \\
s & \models & \mathcal{P}_{\bowtie\rho}(\varphi) & \textit{iff} \quad Prob^{\mathsf{V}}(s,\varphi) \bowtie \rho \\
\sigma & \models & X^I\Phi & \textit{iff} \quad \sigma(1) \textit{ is defined, and} \\
 & & & \qquad \sigma(1) \models \Phi \wedge \Delta(\sigma, 0) \in I \\
\sigma & \models & \Phi_1 U^I \Phi_2 & \textit{iff} \quad \exists \delta \in I. \exists s \in \sigma@\delta . s \models \Phi_2, \\
 & & & \qquad \wedge \forall s' \in \textit{Pref}(\sigma@\delta, s). s' \models \Phi_1 \\
 & & & \qquad \wedge \forall \delta' \in [0,\delta). \forall s'' \in \sigma@\delta'. s'' \models \Phi_1 \ .
\end{array}$$

Given the assumption $\lim_{n\to\infty} \mathbf{P}_{VV}^n = 0$, we can use the notation $\sigma@\delta$ in the description of measurable paths in Definition 3.4. For the path in Figure 5, consider the case in which states s_i, for $0 \leq i \leq 5$, satisfy Φ_1, and s_6 satisfies Φ_2; the path satisfies the until formula $\Phi_1 U^{[13,15]} \Phi_2$, because Φ_2 is satisfied at a time point in the interval $[13, 15]$, and Φ_1 is satisfied until this point. On the other hand, the path does not satisfy the formula $\Phi_1 U^{[12,13]} \Phi_2$ (because Φ_2 is satisfied after 13 time units have elapsed). However, if s_5 satisfies both Φ_1 and Φ_2, the formula $\Phi_1 U^{[12,13]} \Phi_2$ is satisfied. Now consider the case in which s_i, for $0 \leq i \leq 3$, satisfies Φ_1, and s_4 satisfies Φ_2 (the other states are immaterial). Then this path satisfies $\Phi_1 U^{[8,10]} \Phi_2$: the formula Φ_2 is true in a vanishing state witnessed after 9 time units have elapsed, and Φ_1 holds in all preceding vanishing states and time points.

4. Model-checking algorithms

In this section, we present CSL model-checking algorithms for VCTMCs, and, by extension, for GSPNs. Let $\mathsf{V} = ((S, \mathbf{P}, \textsc{init}, L), S_T, S_V, \Lambda)$ be a VCTMC, which we assume is fixed throughout this section. The overall algorithm proceeds in the usual manner as for branching-time temporal logics such as CTL [10] or CSL [6]: the set *Sat*(Φ) of states satisfying the CSL formula Φ to be verified is obtained by computing recursively the set of states satisfying the subformulae of Φ, and then using these state sets to compute *Sat*(Φ). As in [6], we assume that for each state $s \in S$, we have the atomic proposition at_s in AP for which $at_s \in L(s)$ and $at_s \notin L(s')$ for all $s' \in S \setminus \{s\}$. For a set S' of states, we let $at_{S'} = \bigvee_{s \in S'} at_s$.

Let $\mathcal{B}_{\mathsf{C}}$ be the set of bottom strongly-connected components (BSCCs) of the directed graph of C, and, for a BSCC B of C, we denote by π^B the steady-state distribution corresponding to states of B. The following proposition is a minor adjustment to the analogous proposition in the case of CTMCs [6] and allows us to verify steady-state formulae of CSL on both tangible and vanishing states.

Proposition 4.1 *Let* V *be a VCTMC with set* S *of states, and let* $s \in S$ *and* $S' \subseteq S_T$*. Then we have that* $\pi^{\mathsf{R}[\mathsf{V}]}(\alpha_s, S')$ *equals:*

$$\sum_{s' \in S_T} (\alpha_s(s') \cdot \sum_{B \in \mathcal{B}_{\mathsf{R}[\mathsf{V}]}} (Prob^{\mathsf{R}[\mathsf{V}]}(s', at_B) \cdot \sum_{s'' \in B \cap S'} \pi^B(s''))).$$

Proposition 4.1, together with standard algorithms for computing $Prob^{\mathsf{R}[\mathsf{V}]}(s', at_B)$ (see [6]) and $\pi^{\mathsf{R}[\mathsf{V}]}(s'')$, offers a method for verifying properties of the form $\mathcal{S}_{\bowtie\rho}(\Phi)$ on V.

The case in which we wish to identify the set of states satisfying a next path formula is also a minor extension to the CTMC case [6]. Recall that, for a state $s \in S$ of V, we have $s \models \mathcal{P}_{\bowtie\rho}(\varphi)$ if and only if $Prob^{\mathsf{V}}(s,\varphi) \bowtie \rho$.

Proposition 4.2 *Let* V *be a VCTMC with set* S *of states, let* $s \in S$*, let* $I \subseteq \mathbb{R}_{\geq 0}$ *be an interval, and let* Φ *be a* CSL *state formula. Then we have:*

$$Prob^{\mathsf{V}}(s, X^I\Phi) = \begin{cases} (e^{-\mathrm{E}(s)\cdot \inf I} - e^{-\mathrm{E}(s)\cdot \sup I}) \cdot \sum_{s' \in \textit{Sat}(\Phi)} \mathbf{P}(s,s') & \\ & \textit{if } s \in S_T \\ \sum_{s' \in \textit{Sat}(\Phi)} \mathbf{P}(s,s') & \textit{if } s \in S_V \textit{ and } 0 \in I \\ 0 & \textit{if } s \in S_V \textit{ and } 0 \notin I \ . \end{cases}$$

The proof follows from the definition of probability measure in Section 3.1. We note that the case in which $I = [0, \infty)$ requires analysis of the DTMC component of V only, and can be subject to algorithms for DTMCs [13].

We now consider model checking of CSL formulae $\mathcal{P}_{\bowtie\rho}(\Phi U^I \Psi)$ containing an until path formula. As in the case of the next operator, the case in which $I = [0, \infty)$

can be subject to analysis of the DTMC of V [13]. For the general case, and following similar results obtained previously in the context of CSL model checking for CTMCs [6], we now describe how we can reduce the model-checking problem for the until operator to a transient analysis of (adjusted versions of) $\mathsf{R}[\mathsf{V}]$.

First we require the following notation: given a CSL formula Φ, we define $\mathsf{V}[\Phi]$ as the VCTMC obtained by modifying V so that all states that satisfy Φ become absorbing (and, by consequence, tangible).

Definition 4.3 *For a VCTMC* $\mathsf{V} = (\mathsf{D}, S_T, S_V, \Lambda)$, *with* $\mathsf{D} = (S, \mathbf{P}, \text{INIT}, L)$, *and a* CSL *state formula* Φ, *let* $\mathsf{V}[\Phi] = (\mathsf{D}', S'_T, S'_V, \Lambda')$ *be defined as follows. Let* $\mathsf{D}' = (S, \mathbf{P}', \text{INIT}, L)$, *where, for all states* $s, s' \in S$, *we have:*

$$\mathbf{P}'(s,s') = \begin{cases} 1 & \text{if } s \models \Phi \text{ and } s' = s \\ \mathbf{P}(s,s') & \text{if } s \not\models \Phi \\ 0 & \text{otherwise.} \end{cases}$$

We let $S'_T = S_T \cup Sat(\Phi)$ *and* $S'_V = S_V \setminus Sat(\Phi)$. *Finally, for* $s \in Sat(\Phi)$, *we let* Λ'_s *be an exponentially distributed variable with rate 0, and for* $s \notin Sat(\Phi)$, *we let* $\Lambda'_s = \Lambda_s$.

To compute $Prob^{\mathsf{V}}(s, \Phi U^I \Psi)$ for each state $s \in S$ we consider a number of cases, depending on the form of I. By the semantics of CSL, the computation of $Prob^{\mathsf{V}}(s, \Phi U^I \Psi)$ suffices to determine whether s satisfies $\mathcal{P}_{\bowtie\rho}(\Phi U^I \Psi)$. In the following, we write $Sat_T(\Phi)$ to abbreviate $S_T \cap Sat(\Phi)$ (and similarly for Ψ and V to obtain $Sat_V(\Phi)$, $Sat_T(\Psi)$ and $Sat_V(\Psi)$).

Case 1: time-bounded until. We first consider the case in which $I = [0,t]$ for $t > 0$. The following proposition specifies that, to compute $Prob^{\mathsf{V}}(s, \Phi U^{[0,t]} \Psi)$ for state $s \in S$, it suffices to compute transient probabilities on the reduced Markov chain of $\mathsf{V}[\neg\Phi \vee \Psi]$ (that is, of V with states satisfying $\neg\Phi \vee \Psi$ made absorbing).

Proposition 4.4 *Let* V *be a VCTMC and* $\Phi U^{[0,t]} \Psi$ *be a* CSL *until formula with* $t > 0$. *Then, for any state* $s \in S$ *of* V, *we have:*

$$Prob^{\mathsf{V}}(s, \Phi U^{[0,t]} \Psi) \quad = \quad \sum_{s' \in Sat_T(\Psi)} \pi^{\mathsf{R}[\mathsf{V}[\neg\Phi\vee\Psi]]}(\alpha_s, s', t)$$

The proposition follows from the fact that behavior of V after a state satisfying $\neg\Phi$ or Ψ is reached has no impact on the required probability; hence such states can be made absorbing.

Case 2: non-immediate point-interval until. The second proposition considers $I = [t,t]$ for $t > 0$, and specifies that, to compute $Prob^{\mathsf{V}}(s, \Phi U^{[t,t]} \Psi)$ for state $s \in S$, it suffices to compute transient probabilities on the reduced Markov chain of $\mathsf{V}[\neg\Phi]$.

Proposition 4.5 *Let* V *be a VCTMC and* $\Phi U^{[t,t]} \Psi$ *be a* CSL *until formula such that* $t > 0$. *Then, for any state* $s \in S$ *of* V, *we have:*

$$Prob^{\mathsf{V}}(s, \Phi U^{[t,t]} \Psi) = \sum_{s' \in Sat_T(\Phi\wedge\Psi)} \pi^{\mathsf{R}[\mathsf{V}[\neg\Phi]]}(\alpha_s, s', t) \; .$$

Similarly to Case 1, the proposition relies on the fact that behavior after reaching states satisfying $\neg\Phi$ has no impact on the required probability. Another property used is that, given state s and time t, the probability of the set of paths from s for which a sequence featuring vanishing states is observed at time t is zero; formally, we have $\Pr^{\mathsf{V}}_s\{\sigma \in Path^{\mathsf{V}} \mid |\sigma@t| > 1\} = 0$. This follows from the fact that the probability of the set of paths in which a transition is fired from a tangible state to a vanishing state at exactly time t is equal to zero: firing such a transition is the only way to exhibit a path featuring at least one vanishing state at exactly time t. In this way, the probability in Proposition 4.5 can be obtained by summing over tangible states only: reaching vanishing states at time t has probability zero, and therefore such vanishing states do not need to be considered in the summation. We also note that the summation is over states satisfying $\Phi \wedge \Psi$ for the same reason as in Corollary 2 of [6].

Case 3: immediate point-interval until. The third case considers $I = [0,0]$. It can be observed that a formula $\Phi U^{[0,0]} \Psi$ is satisfied by paths σ of one of the following forms:

1. the first state $\sigma(0)$ along the path satisfies Ψ, or
2. an initial prefix of σ follows a sequence of vanishing Φ-states before terminating in a state satisfying Ψ (formally, for some $i \geq 1$, we have $\sigma(i) \in Sat_V(\Psi)$, and $\sigma(j) \in Sat_V(\Phi)$ for each $j < i$).

Hence, the only non-trivial probability computation that is required concerns transitions between vanishing states only. We can express the paths of interest using an until formula with a trivial time-bound interval $[0,\infty)$, thus allowing computation of the required probability on the DTMC of V.

Proposition 4.6 *Let* V *be a VCTMC and* $\Phi U^{[0,0]} \Psi$ *be a* CSL *until formula. Then, for any state* $s \in S$ *of* V, *we have:*

$$Prob^{\mathsf{V}}(s, \Phi U^{[0,0]} \Psi) = \begin{cases} 1 & \text{if } s \in Sat_T(\Psi) \\ Prob^{\mathsf{V}}(s, at_{Sat_V(\Phi)} U^{[0,\infty)} at_{Sat_V(\Psi)}) & \text{if } s \in S_V \\ 0 & \text{otherwise.} \end{cases}$$

Case 4: interval until. We now consider the case in which $I = [t,t']$ for $0 < t < t'$. As in the case of CTMCs [6], the computation of the required probability is split into two parts; more precisely, we can show that

$\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{[t,t']}\Psi)$ equals:

$$\sum_{s' \in \mathit{Sat}_T(\Phi)} \mathit{Prob}^{\mathsf{V}}(s, \Phi U^{[t,t]} at_{s'}) \cdot \mathit{Prob}^{\mathsf{V}}(s', \Phi U^{[0,t'-t]}\Psi) \; .$$

It suffices to consider only tangible states within the above sum, because, as in Case 2, the probability of the set of paths exhibiting a sequence containing vanishing states at time t is zero, and hence such paths do not contribute to the overall probability. Applying Proposition 4.5 and Proposition 4.4 then yields the following result.

Proposition 4.7 *Let* V *be a VCTMC and* $\Phi U^{[t,t']}\Psi$ *be a* CSL *until formula with* $0 < t < t'$*. Then, for any state* $s \in S$ *of* V*, we have:*

$$\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{[t,t']}\Psi) = \sum_{s' \in \mathit{Sat}_T(\Phi)} \sum_{s'' \in \mathit{Sat}_T(\Psi)} \pi^{\mathsf{R}[\mathsf{V}[\neg\Phi]]}(\alpha_s, s', t) \cdot \pi^{\mathsf{R}[\mathsf{V}[\neg\Phi\vee\Psi]]}(\alpha^1_{s'}, s'', t'-t) \; .$$

Case 5: unbounded until. The final case that we consider concerns $I = [t, \infty)$ for $t > 0$. Following the same reasoning as in Case 4, we obtain the following proposition, which specifies that the computation of the required probability reduces to an application of Case 2 and the computation of probabilities relating to a formula with time bound $[0, \infty)$ on the DTMC of V.

Proposition 4.8 *Let* V *be a VCTMC and* $\Phi U^{[t,\infty)}\Psi$ *be a* CSL *until formula with* $0 < t < t'$*. Then, for any state* $s \in S$ *of* V*, we have:*

$$\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{[t,\infty)}\Psi) = \sum_{s' \in \mathit{Sat}_T(\Phi)} \mathit{Prob}^{\mathsf{V}}(s, \Phi U^{[t,t]} at_{s'}) \cdot \mathit{Prob}^{\mathsf{V}}(s', \Phi U^{[0,\infty)}\Psi) \; .$$

Other intervals. In the above cases, the time-bound intervals are either closed (of the form $[t, t']$) or left-closed and unbounded (of the form $[t, \infty)$). As for CTMCs, probabilities of formulae with half-open intervals, or open intervals, can be computed using the model-checking algorithms corresponding to their respective closed intervals, as presented in the five cases above; however, the left-open intervals are exceptions to this rule, and require a different technique. Consider the case in which we aim to compute $\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{(0,t]}\Psi)$ for some state $s \in S$. If $s \in S_T$ (s is a tangible state), then $\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{(0,t]}\Psi)$ can be computed as for Case 1. However, if $s \in S_V$ (s is a vanishing state), then we have to account for the fact that, to satisfy $\Phi U^{(0,t]}\Psi$, a path of the VCTMC must pass from vanishing states to a tangible state before satisfying Ψ.

Proposition 4.9 *Let* V *be a VCTMC and* $\Phi U^{(0,t]}\Psi$ *be a* CSL *until formula. Then, for any tangible state* $s \in S_T$ *of* V*, we have that* $\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{(0,t]}\Psi)$ *equals* $\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{[0,t]}\Psi)$*. For any vanishing state* $s \in S_V$ *of* V*, we have that* $\mathit{Prob}^{\mathsf{V}}(s, \Phi U^{(0,t]}\Psi)$ *equals:*

$$\sum_{s' \in \mathit{Sat}_T(\Phi)} \sum_{s'' \in \mathit{Sat}_T(\Psi)} \mathit{Prob}^{\mathsf{V}}(s, at_{\mathit{Sat}_V(\Phi)} U^{[0,\infty)} at_{s'}) \cdot \pi^{\mathsf{R}[\mathsf{V}[\neg\Phi\vee\Psi]]}(\alpha^1_{s'}, s'', t) \; .$$

This result can also be applied to the case of interval $(0, t)$. The case of interval $(0, \infty)$ can be obtained using a similar equation (we replace $\pi^{\mathsf{R}[\mathsf{V}[\neg\Phi\vee\Psi]]}(\alpha^1_{s'}, s'', t)$ by $\mathit{Prob}^{\mathsf{V}}(s', \Phi U^{[0,\infty)} at_{s''})$).

Complexity. The worst-case time complexity of the above algorithms is $O(|\Phi| \cdot (M \cdot q \cdot t_{max} + N^3))$, where $|\Phi|$ is the length of the overall CSL formula, M is the number of non-zero entries in the matrix $\mathbf{P}$ of the VCTMC, N is the number of states of the VCTMC, q is the uniformization rate used for computing the transient solutions, and t_{max} is the maximum time bound occurring in Φ. This coincides with the complexity for model checking CSL on CTMCs [6], because computing the reduced Markov chain of a VCTMC takes $O(N^3)$ time.

5. Numerical examples

To illustrate the procedure we consider the model of a cyclic polling system as shown in Figure 6, in which servers visit cyclically 4 stations, and which has been used widely as an example of a GSPN [16, 2]. When a server arrives at station i (a token in place Ppi, $0 \leq i \leq 3$), it either finds a client in queue i (token in place Pqi) and serves it (token in place Psi) or starts immediately to walk toward station $i + 1$ (token in place Pwi). In all of the examples below, we let the maximal number of clients in each queue be $K = 3$. In the CSL formulae, we use the name of a place to denote the atomic proposition which labels all states in which the place has at least one token.

We consider the single server case (therefore $L = 1$) and study how the server moves from the first station to the next one. The corresponding path formula is $\varphi_1 ::= \neg\mathsf{Pw1}\ U^{[y,z]}\ \mathsf{Pp1}$ ($\neg$Pw1 is necessary in order to rule out full cycles). We are interested in computing the probability of the formula for all states $s \in S$, in order to identify the states which satisfy $\mathcal{P}_{\bowtie\rho}(\varphi_1)$ for some $\bowtie$ and ρ. Observe that this formula involves a vanishing state (place Pp1 is marked only in vanishing states). Figure 7 plots, for various values of y and z, the probability for all states in which Pp0 is marked and Pq0 is not (and therefore the server will not provide service), while Figure 8 plots the probability for all states in which Pp0 and Pq0 are both marked (and therefore the server will provide service).

In the case $y > 0$, Proposition 4.7 applies and two VCTMCs are used to calculate the results. Up to time y the calculations are carried out over V[Pw1], the VCTMC

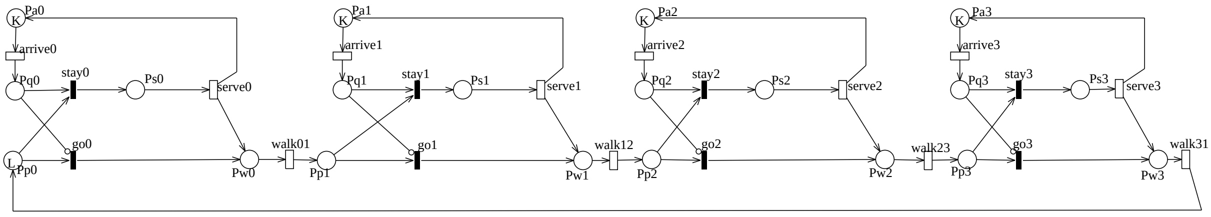

Figure 6. Polling model

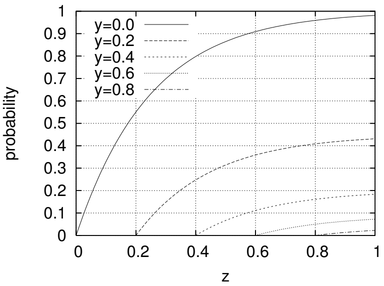

Figure 7. Probability of a path satisfying φ_1 (states in which Pp0 is marked and Pq0 is not)

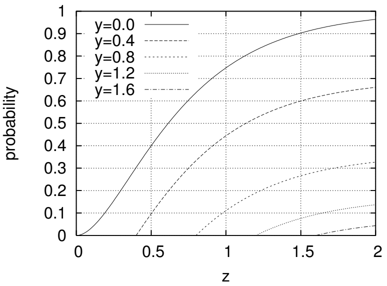

Figure 8. Probability of a path satisfying φ_1 (states in which Pp0 and Pq0 are marked)

in which states that satisfy Pw1 are made absorbing. From time y to time z, the VCTMC $\mathsf{V}[\mathsf{Pw1} \vee \mathsf{Pp1}]$ is applied. The VCTMC $\mathsf{V}[\mathsf{Pw1} \vee \mathsf{Pp1}]$ has the initial probability vector that corresponds to the transient probabilities of $\mathsf{V}[\mathsf{Pw1}]$ at time y. Observe that this is an implicit way of computing the double sum of Proposition 4.7. Note that $\mathsf{V}[\mathsf{Pw1}]$ is started from a vanishing state (corresponding to a token in Pp0). Accordingly, the CTMC that is used to calculate the transient probabilities of $\mathsf{V}[\mathsf{Pw1}]$ is started from the tangible state reached immediately after this initial vanishing state. The VCTMC $\mathsf{V}[\mathsf{Pw1} \vee \mathsf{Pp1}]$ has instead some absorbing states that were vanishing (corresponding to a token in Pp1). In the case $y = 0$, Proposition 4.4 applies and only the second VCTMC is necessary.

Next we study the characteristics of a full cycle with single server, expressed by the formula $\varphi_2 ::= \neg\mathsf{Pp0}\ U^{(y,z]}\ \mathsf{Pp0}$, where $\neg\mathsf{Pp0}$ is used to rule out multiple cycles. The first graph of Figure 9 depicts the probability of a path satisfying φ_2 for the state s in which Pw0 is marked and all the queues are empty. If we want to investigate different types of cycle we can refine φ_2. In order to study a cycle in which the server does not work we calculate the probability for the formula $\varphi_3 ::= (\neg\mathsf{Pp0} \wedge \neg\mathsf{Ps1} \wedge \neg\mathsf{Ps2} \wedge \neg\mathsf{Ps3})\ U^{(y,z]}\ \mathsf{Pp0}$. The probability for state s is shown in the second graph Figure 9. The third graph of Figure 9 depicts instead the probability of a path in which the server has to work at every queue: $\varphi_4 ::= (\neg\mathsf{Pp0} \wedge \neg((\mathsf{Pp1} \wedge \mathsf{Pq1}) \vee (\mathsf{Pp2} \wedge \mathsf{Pq2}) \vee (\mathsf{Pp3} \wedge \mathsf{Pq3})))\ U^{(y,z]}\ \mathsf{Pp0}$, from the state in which Ps0 is marked and all the queues are empty. Observe that the curves for the cases $y = 0$ and $y = 1$ cannot be distinguished because the probability of completing the cycle in 1 time unit is negligible.

6. Conclusions

This paper has presented a model-checking algorithm that can be applied to GSPNs. As discussed using simple motivating examples, vanishing states cannot simply be eliminated before model checking. We therefore define the underlying stochastic process of a GSPN as an extended form of CTMCs that consider vanishing states and that we call vanishing CTMCs. The semantics of CSL has then been extended to VCTMCs, and the algorithms based on transient analysis defined for CSL on CTMCs [6] have been extended to the case of VCTMCs. The usefulness of the approach has been shown on a classical GSPN example.

Despite the development of logics such as CSRL [5], eCSL and asCSL [4], which offer the possibility of using rewards, initial distributions and action labels, the use of stochastic model-checking approaches in performance evaluation is not yet widespread. In future work we intend to investigate whether the techniques from stochastic model checking may be combined with the path-based measures of Obal and Sanders [21], and to extend CSRL to the case of vanishing states.

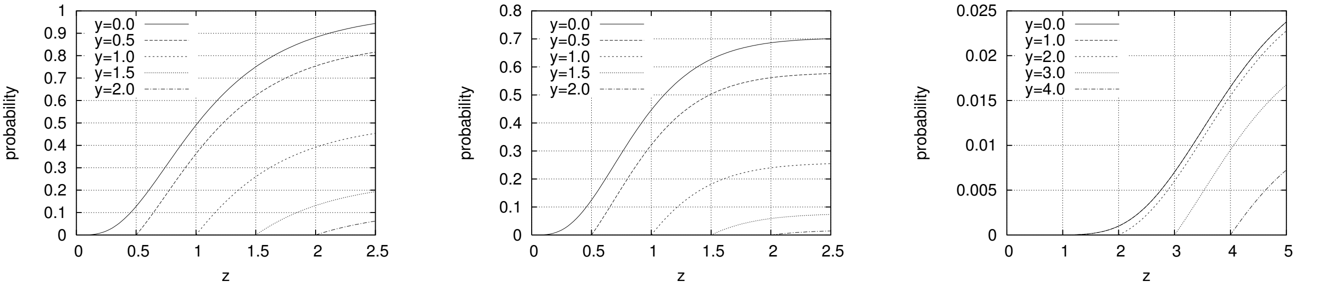

Figure 9. Probability of a path satisfying φ_2, φ_3 and φ_4, respectively

References

[1] M. Ajmone Marsan, G. Balbo, and G. Conte. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Transactions on Computer Systems*, 2:93–122, 1984.

[2] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. J. Wiley, 1995.

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

[4] C. Baier, L. Cloth, B. Haverkort, M. Kuntz, and M. Siegle. Model checking action- and state-labelled Markov chains. In *Proceedings of the International Conference on Dependable Systems and Networks (DSN'04)*, pages 701–710. IEEE Computer Society, 2004.

[5] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. On the logical characterisation of performability properties. In *Proceedings of the 12th International Colloquium on Automata, Languages and Programming (ICALP'00)*, volume 1853 of *LNCS*, pages 780–792. Springer, 2000.

[6] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Transactions on Software Engineering*, 29(6):524–541, 2003.

[7] 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. In *Proceedings of the 10th International Workshop on Petri Nets and Performance Models (PNPM'03)*, pages 62–71. IEEE Computer Society, 2003.

[8] P. Buchholz, J.-P. Katoen, P. Kemper, and C. Tepper. Model-checking large structured Markov chains. *Journal of Logic and Algebraic Programming*, 56:69–96, 2003.

[9] G. Clark and J. Hillston. Towards automatic derivation of performance measures from PEPA models. In *Proceedings of the UK Performance Engineering Workshop*, 1996.

[10] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[11] D. D'Aprile, S. Donatelli, and J. Sproston. CSL model checking for the GreatSPN tool. In *Proceedings of the 19th International Symposium on Computer and Information Sciences (ISCIS'04)*, volume 3280 of *LNCS*, pages 543–552. Springer, 2004.

[12] L. de Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.

[13] H. A. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[14] H. Hermanns, J.-P. Katoen, J. Meyer-Kayser, and M. Siegle. A tool for model-checking Markov chains. *International Journal on Software Tools for Technology Transfer*, 4(2):153–172, 2003.

[15] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. PRISM: A tool for automatic verification of probabilistic systems. In *Proceedings of the 12th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'06)*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

[16] O. C. Ibe and K. Trivedi. Stochastic Petri net models of polling systems. *IEEE Journal on Selected Areas of Communication*, 8(9):1649–1657, 1990.

[17] G. Infante López, H. Hermanns, and J.-P. Katoen. Beyond memoryless distributions: Model checking semi-Markov chains. In *Proceedings of the Joint International Workshop on Process Algebra and Probabilistic Methods (PAPM-PROBMIV'01)*, volume 2165 of *LNCS*, pages 57–70. Springer, 2001.

[18] J.-P. Katoen, M. Khattri, and I. S. Zapreev. A Markov reward model checker. In *Proceedings of the 2nd International Conference on the Quantitative Evaluation of Systems (QEST'05)*, pages 243–244. IEEE Computer Society, 2005.

[19] V. G. Kulkarni. *Modeling and Analysis of Stochastic Systems*. Chapman Hall, 1995.

[20] M. Kuntz and M. Siegle. Symbolic model checking of stochastic systems: Theory and implementation. In *Proceedings of the 13th International SPIN Workshop on Model Checking Software*, volume 3925 of *LNCS*, pages 89–107. Springer, 2006.

[21] W. D. Obal II and W. H. Sanders. State-space support for path-based reward variables. *Performance Evaluation*, 35(3-4):233–251, 1999.