REIHE INFORMATIK

3/99

Establishing Qualitative Properties for Probabilistic Lossy Channel Systems: an Algorithmic Approach

Christel Baier, Bettina Engelen, Markus Roggenbach
Universität Mannheim
Seminargebäude D 7, 27
D-68131 Mannheim

Establishing Qualitative Properties for Probabilistic Lossy Channel Systems: an Algorithmic Approach

Christel Baier, Bettina Engelen, Markus Roggenbach *

Universität Mannheim,
Fakultät für Mathematik & Informatik,
D7, 27, 68131 Mannheim, Germany
{baier,bengelen}@pi2.informatik.uni-mannheim.de
FAX: ++49/621/292-5364

Abstract. Lossy channel systems (LCSs for short) are models for communicating systems where the subprocesses are linked via unbounded FIFO channels which might lose messages. Link protocols, such as the Alternating Bit Protocol and HDLC can be modelled with these systems. The decidability of several verification problems of LCSs has been investigated by Abdulla & Jonsson [AJ93,AJ94], e.g. they have shown that the reachability problem for LCSs is decidable while LTL model checking is not. In this paper, we consider *probabilistic* LCSs (which are LCSs where the transitions are augmented with appropriate probabilities) as introduced by [IN97] and show that the question of whether or not a linear time property holds with probability 1 is decidable. More precisely, we show how $LTL_{\setminus X}$ model checking for (certain types of) probabilistic LCSs can be reduced to a reachability problem in a (non-probabilistic) LCS where the latter can be solved with the methods of [AJ93].[1]

1 Introduction

Traditional algorithmic verification methods for parallel systems are limited to finite state systems and fail for systems with an infinite state space, such as real-time programs with continuous clocks or programs that operate with unbounded data structures or protocols for processes that communicate via unbounded channels. Typically, such systems are modelled by a finite state machine that specifies the control part. The transitions between the control states are equipped with conditions (e.g. about the values of a counter or a clock or about the messages in a channel). The behaviour of such a system is then given by a (possibly infinite) transition system whose global states consist of a control state and an auxiliary component whose values range over an infinite domain (e.g. the interpretations for a counter or a clock or the contents of certain channels). Even a wide range of verification problems for such infinite systems is undecidable, various authors developed verification algorithms for special types of infinite systems.

* The second and the third author are partly sponsored by the DFG-Project MA 794/3-1.

[1] Here, $LTL_{\setminus X}$ denotes standard linear time logic without next step.

This paper is concerned with model checking algorithms for communication protocols where the (sub-)processes are linked via unbounded FIFO channels. Dealing with *perfect* channels, in which case one gets the same expressiveness as Turing Machines, most verification problems are undecidable [BZ83]. Several link protocols, like the Alternating Bit Protocol [BSW69] or HDLC [ISO79], are designed to work correctly even for unreliable channels. For such faulty systems, various verification problems can be solved automatically. Finkel [Fin94] considered *completely specified protocols* modelled by channel systems where the channels might lose their first message and showed that the termination problem is solvable. Abdulla & Jonsson [AJ93] present algorithms for a reachability analysis (see also [AKP97]) and the verification against (certain types of) safety and eventually properties for *lossy channel systems* (LCSs), i.e. channel systems that may lose arbitrary messages. Abdulla & Kindahl [AK95] have shown that also the task of establishing a branching time relation (simulation or bisimulation) between a LCS and a finite transition system can be automated. Decidability results for other types of unreliable FIFO systems have been developed e.g. by Cécé, Finkel & Iyer [CFI96] (where channel systems with insertion or duplication errors are considered) and Bouajjani & Mayr [BM98] (where lossy vector addition systems are investigated). Even if validating faulty channel systems is easier than reasoning about perfect channel systems, some verification problems are still undecidable for unreliable channel systems. Abdulla & Jonsson [AJ94] show the undecidability of model checking for LCSs against LTL or CTL specifications or establishing "eventually" properties under fairness assumptions about the channels.[2]

We follow here the approach of Iyer & Narasimha [IN97] and consider *probabilistic LCSs* (PLCSs for short). In PLCSs, one assumes that the failure rate of the channels is known and deals with a constant $\wp$ that stands for the probability that one of the channels loses a message. The other transitions are equipped with "weights" that yield the probabilities for the possible steps of the global states and turn the transition system for the underlying LCS into a (possibly infinite) Markov chain.

For probabilistic systems modelled by Markov chains, various (deductive and algorithmic) verification methods have been proposed in the literature, but only a minority of them is applicable for PLCSs. Most of the algorithmic methods are formulated for *finite* Markov chains and hence are not applicable for PLCSs, see e.g. [VW86,CY88,CC91,CC92,HT92,HJ94,CY95,IN96,BH97]. Even some of

[2] To overcome the limitations of algorithmic verification methods for LCSs due to undecidability results, [ABJ98] propose (possibly non-terminating) symbolic verification techniques based on a "on the fly" reachability analysis.

the axiomatic methods, see e.g. [HS86,JS90,LS92], fail for PLCSs since they are designed for *bounded* (or even finite) Markov chains.[3]

In this paper, we shrink our attention to temporal logical specifications; more precisely, to specifications given by formulas of propositional linear time temporal logic LTL. When interpreting a LTL formula f over the states of a Markov chain, the probability for f to hold in a state s, i.e. the probability measure of all paths starting in s and satisfying f, can be viewed as the "truth value" for f in state s. Thus, LTL can serve as specification formalism for both *qualitative* and *quantitative* temporal properties. In the former case, a LTL specification just consists of a LTL formula f; satisfaction of f in a state s means that f holds for *almost all* paths starting in s (i.e. with probability 1). Lehmann & Shelah [LS82] present sound and complete axiomatizations for (a logic that subsumes) LTL interpreted over Markov chains of arbitrary size; thus, the framework of [LS82] can serve as a proof-theoretic method for verifying qualitative properties for PLCSs. Quantitative properties can be expressed by a LTL formula f and a lower bound probability p; satisfaction in a state s means that the probability for f is beyond the given lower bound p.[4] In [IN97], an *approximative quantitative analysis* for PLCSs (i.e. an algorithm for approximating the probabilities for a $LTL_{\setminus X}$ formula f to hold in the initial state of a PLCS) is proposed. Here, $LTL_{\setminus X}$ means LTL without the next step operator X. This method yields a model checking procedure for verifying quantitative $LTL_{\setminus X}$ specifications with respect to a tolerence ϵ but it fails for qualitative properties (because of the tolerance).[5]

The main contribution of this paper is a verification algorithm for establishing qualitative properties specified by $LTL_{\setminus X}$ formulas for PLCSs. We use the ω-automaton approach à la Wolper, Vardi & Sistla [WVS83] and construct an ω-automaton $\mathcal{A}_f$ for the given formula f. Then, we define the product $\mathcal{PL} \times \mathcal{A}_f$ of the given PLCS $\mathcal{PL}$ and the ω-automaton $\mathcal{A}_f$ (yielding a new PLCS) and a formula f' of the form $f' = \bigvee \Diamond\Box(a_j \wedge \Diamond b_j)$ with atomic propositions a_j, b_j such that the probability for f to hold for $\mathcal{PL}$ equals the probability for f' to hold for $\mathcal{PL} \times \mathcal{A}_f$.

For finite Markov chains, it is well-known that whether or not a qualitative property can be established does not depend on the precise probability but just

[3] Boundedness of a Markov chain means that there is an upper bound $\epsilon > 0$ for the non-zero transition probabilities. In the Markov chain for a PLCS, the probability for the loss of a concrete message tends to 0 if the channel length tends to ∞; thus, they fail to be bounded.

[4] In the branching time framework (where one distinguishes between state and path formulas), the state formulas typically also assert that the probability for a certain event lies in a given interval; thus, the state formulas can be viewed as (special types of) quantitative LTL specifications. See e.g. [HJ94,ASBS95,BdA95].

[5] The tolerance ϵ specifies how precise the approximated value should be. I.e. the difference between the computed value q' and the precise probability q for the formula to hold in the initial state of the given PLCS is at most ϵ.

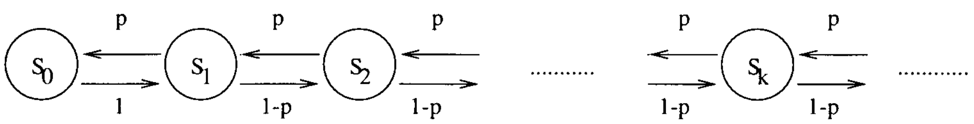

Fig. 1. An infinite (bounded) Markov chain

on the topology of the underlying directed graph [HSP83]. More precisely, qualitative properties of the type $f' = \bigvee \Diamond\Box(a_j \wedge \Diamond b_j)$ can be established by analyzing the bottom strongly connected components. This does not longer hold when we deal with infinite (bounded or unbounded) Markov chains. For an example, consider the system of Figure 1. The qualitative property stating that s_0 is visited infinitely many times cannot be established unless $p \geq \frac{1}{2}$.[6] To avoid a scenario as for the Markov chain in Figure 1 with $p < 1/2$ where a reachability analysis cannot help for establishing qualitative properties, we make an additional assumption about the underlying PLCS and require *probabilistic input enabledness*. This assumption allows us to reduce the question of whether a qualitative property specified by a formula f' as above is satisfied to a reachability problem in the underlying (non-probabilistic) LCS where the latter is solvable with conventional methods [AJ93,AKP97].

The reason why we do not deal with the next step operator will be explained in Section 4. Roughly speaking, the lack of next step ensures the invariance of the formulas with respect to losing a message. This is essential for characterizing the probability for f to hold for a PLCS $\mathcal{PL}$ by the probability for the above mentioned formula f' in the product system $\mathcal{PL} \times \mathcal{A}_f$. (See Lemma 2.)

Organization of the paper: In Section 2 we briefly explain our notations concerning Markov chains and linear time logic $LTL_{\setminus X}$ with its interpretation over Markov chains. The definitions of LCSs and PLCSs and related notations are given in Section 3. Our model checking algorithm is presented in Section 4. Section 5 concludes the paper.

Throughout the paper, we work with a finite non-empty set AP of atomic propositions which we use in the context of labelled Markov chains, $LTL_{\setminus X}$ formulas and LCSs. The reader should be familiar with basic notions of probability theory, see e.g. [Fel68,Bre68], further on with the main concepts of the temporal logic and model checking approach, see e.g. [CES86,Eme90,MP92], and also with the connection between temporal logic and ω-automaton, see e.g. [Tho90,Var96].

[6] This observation follows with standard arguments of Markov chain theory ("random walks"). For $p < \frac{1}{2}$, the probability to reach s_0 from s_k is $p^k/(1-p)^k < 1$.

2 Preliminaries: Markov chains and $LTL_{\setminus X}$

In the literature, a wide range of models for probabilistic processes is proposed. In this paper, we deal with (discrete time, labelled) Markov chains which is one of the basic models for specifying probabilistic systems. We briefly explain our notations concerning Markov chains and linear time logic $LTL_{\setminus X}$ with its interpretation over Markov chains.

Markov chains: A *Markov chain* over AP is a tuple $\mathsf{M} = (\mathsf{S}, \mathsf{P}, \mathsf{L})$ where S is a set of *states*, $\mathsf{L} : \mathsf{S} \longrightarrow 2^{AP}$ a *labelling function* which assigns to each state $\mathsf{s} \in \mathsf{S}$ a set of atomic propositions and $\mathsf{P} : \mathsf{S} \times \mathsf{S} \longrightarrow [0, 1]$ a *transition probability function* such that for all $\mathsf{s} \in \mathsf{S}$: $\mathsf{P}(\mathsf{s}, \mathsf{t}) > 0$ for at most countably many states $\mathsf{t} \in \mathsf{S}$ and $\sum_{\mathsf{t} \in \mathsf{S}} \mathsf{P}(\mathsf{s}, \mathsf{t}) = 1$.

Execution sequences arise by resolving the probabilistic choices. Formally, an *execution sequence* in M is a nonempty (finite or infinite) sequence $\pi = \mathsf{s}_0, \mathsf{s}_1, \mathsf{s}_2, \ldots$ where s_i are states and $\mathsf{P}(\mathsf{s}_{i-1}, \mathsf{s}_i) > 0$, $i = 1, 2, \ldots$. An infinite execution sequence π is also called a *path*. We denote by $word(\pi)$ the to π associated sequence of atomic propositions, i.e. $word(\pi) = \mathsf{L}(\mathsf{s}_0), \mathsf{L}(\mathsf{s}_1), \mathsf{L}(\mathsf{s}_2), \ldots$. The first state of π is denoted by $first(\pi)$. $\pi(k)$ denotes the $(k+1)$-th state of π, i.e. if $\pi = \mathsf{s}_0, \mathsf{s}_1, \mathsf{s}_2, \ldots$ then $\pi(k) = \mathsf{s}_k$. $Reach_{\mathsf{M}}(\mathsf{s})$ denotes the set of states that are reachable from s, i.e. $Reach_{\mathsf{M}}(\mathsf{s})$ is the set of states $\pi(k)$ where π is a path with $first(\pi) = \mathsf{s}$. $Path_{\mathsf{M}}(\mathsf{s})$ denotes the set of paths π with $first(\pi) = \mathsf{s}$ and $Path_{fin,\mathsf{M}}(\mathsf{s})$ denotes the set of finite paths starting in s. For $\mathsf{s} \in \mathsf{S}$, let $\Sigma_{\mathsf{M}}(\mathsf{s})$ be the smallest σ-algebra on $Path_{\mathsf{M}}(\mathsf{s})$ which contains the basic cylinders $\{\pi \in Path_{\mathsf{M}}(\mathsf{s}) : \rho \text{ is a prefix of } \pi\}$ where ρ ranges over all finite execution sequences starting in s. The probability measure $Prob_{\mathsf{M}}$ on $\Sigma_{\mathsf{M}}(\mathsf{s})$ is the unique measure with

$$Prob_{\mathsf{M}}\{\ \pi \in Path_{\mathsf{M}}(\mathsf{s})\ : \rho \text{ is a prefix of } \pi\ \} = \mathsf{P}(\rho)$$

where $\mathsf{P}(\mathsf{s}_0, \mathsf{s}_1, \ldots, \mathsf{s}_k) = \mathsf{P}(\mathsf{s}_0, \mathsf{s}_1) \cdot \mathsf{P}(\mathsf{s}_1, \mathsf{s}_2) \cdot \ldots \cdot \mathsf{P}(\mathsf{s}_{k-1}, \mathsf{s}_k)$. If it is clear from the context, we omit the subscript M and briefly write $Path(\mathsf{s})$, $Reach(\mathsf{s})$, etc..

Linear Time Logic $LTL_{\setminus X}$:

$$f \ ::= \ tt \ \Big| \ a \ \Big| \ f_1 \wedge f_2 \ \Big| \ \neg f \ \Big| \ f_1 \mathcal{U} f_2$$

$LTL_{\setminus X}$ formulas are build from the above grammar where a is an atomic proposition ($a \in AP$) and $\mathcal{U}$ the temporal operator "until". As usual, operators for modelling "eventually" or "always" can be derived by $\Diamond f = tt\ \mathcal{U}\ f$ and $\Box f = \neg \Diamond \neg f$. The interpretation of $LTL_{\setminus X}$ formulas over the paths and states of a Markov chain is as follows. Let $\mathsf{M} = (\mathsf{S}, \mathsf{P}, \mathsf{L})$ be a Markov chain over AP. The satisfaction relation (denoted $\models_{\mathsf{M}}$ or briefly $\models$) for path formulas is as in the non-probabilistic case, i.e. it is given by: $\pi \models a$ iff $\pi(0) \models a$, $\pi \models f_1 \wedge f_2$ iff

$\pi \models f_i$, $i = 1, 2$, $\pi \models \neg f$ iff $\pi \not\models f$ and $\pi \models f_1 \mathcal{U} f_2$ iff there exists $k \geq 0$ with $\pi \uparrow i \models f_1$, $i = 0, 1, \ldots, k-1$ and $\pi \uparrow k \models f_2$.[7]

For $\mathsf{s} \in \mathsf{S}$, we define the "truth value" $p_{\mathsf{s}}^{\mathsf{M}}(f)$ (or briefly $p_{\mathsf{s}}(f)$) as the measure of all paths that start in s and satisfy f, i.e. $p_{\mathsf{s}}(f) = Prob\{\pi \in Path(\mathsf{s}) : \pi \models f\}$. The satisfaction relation for the states (also denoted $\models_{\mathsf{M}}$ or $\models$) is given by $\mathsf{s} \models f$ iff $p_{\mathsf{s}}(f) = 1$.

3 Probabilistic Lossy Channel Systems

We recall the definitions of (non-probabilistic and probabilistic) LCSs as introduced by [AJ93] and [IN97]. A LCS models the behaviour of a number of processes which communicate over certain unreliable channels. The control part of a LCS is specified by a finite state machine with (conditional) action-labelled transitions. The transitions can either be labelled by τ (which stands for an autonomous (internal) move for one of the processes) or by a communication action $c?m$ (where a process receives message m from channel c) or $c!m$ (where a process sends message m via channel c). The global behaviour depends on the current control state s and the contents of the channels. While the enabledness of the internal actions τ and the output actions $c!m$ just depends on the control state, enabledness of an input action $c?m$ requires that m is the first message of c and that the current control state s has an outgoing transition labelled by $c?m$.

The effect of an input action $c?m$ is that the first message m is removed from c while the output action $c!m$ inserts m at the end of c. The internal action τ does not change the channel contents. Moreover, in each global state, any messages in a channel can be lost in which case the control state does not change.

Definition 1. (cf. [AJ93]) *A* Lossy Channel System (LCS) *is a tuple* $\mathcal{L} = (S_{control}, s_0, L, Ch, Mess, \hookrightarrow)$ *where*

- $S_{control}$ *is a finite set of* control states,
- $s_0 \in S_{control}$ *is an* initial *control state,*
- L *is a* labelling function, *i.e.* $L : S_{control} \longrightarrow 2^{AP}$,
- Ch *is a finite set of* channels,
- $Mess$ *is a finite set of* messages,
- $\hookrightarrow \subseteq S_{control} \times Act \times S_{control}$

where $SendAct = \{c!m : c \in Ch,\ m \in Mess\}$, $RecAct = \{c?m : c \in Ch,\ m \in Mess\}$ *and* $Act = SendAct \cup RecAct \cup \{\tau\}$.[8]

[7] Here, $\pi \uparrow k$ denotes the k-th suffix of π, i.e. the path $\pi(k), \pi(k+1), \pi(k+2), \ldots$.

[8] The finite representation of a LCS in the sense of Definition 1 just specifies the control part. Since the loss of messages does not affect the control state, transitions obtained by losing a message are not specified by the transition relation $\hookrightarrow$.

The (global) behaviour of a LCS can be formalized by an action-labelled transition system (which might have infinitely many states). We use the action set $\mathsf{Act}_\ell = Act \cup \{\ell_{c,i} : c \in Ch, i = 0,1,2,\ldots\}$ where the auxiliary labels $\ell_{c,i}$ denote that the i-th message of channel c is lost. The global states are pairs $\mathsf{s} = \langle s, w\rangle$ consisting of a control state s and an additional component w that gives rise about the channel contents. Formally, w is a function $Ch \longrightarrow Mess^*$ which assigns to each channel c a finite string $w.c$ of messages. We use the symbol $\emptyset$ to denote both the empty string and the function that assigns to any channel c the empty string. For $c \in Mess^*$, $c \neq \emptyset$, $first(c)$ is the first message in c. $|c|$ denotes the length of c; i.e. $|\emptyset| = 0$ and $|m_1 \ldots m_k| = k$. $w[c := x]$ denotes the unique function $w' : Ch \longrightarrow Mess^*$ with $w'.c = x$ and $w'.d = w.d$ for $d \neq c$. The total channel length $|w|$ is defined as the sum over the lengths of the contents of the vector w; i.e. $|w| = \sum_{c \in Ch} |w.c|$. Further on, $|\mathsf{s}| = |w|$ and $\mathsf{s}.c = w.c$ for the global state $\mathsf{s} = \langle s, w\rangle$. The transition system associated with $\mathcal{L}$ is

$$\mathsf{TS}(\mathcal{L}) \;=\; (\mathsf{S}_{global}, \longrightarrow, \mathsf{L}, \mathsf{s}_0)$$

where $\mathsf{S}_{global} = S_{control} \times (Ch \longrightarrow Mess^*)$, $\mathsf{s}_0 = \langle s_0, \emptyset\rangle$ is the *initial global state* and $\mathsf{L}(\langle s, w\rangle) = L(s)$ for all $\langle s, w\rangle \in \mathsf{S}_{global}$. Furthermore the transition relation $\longrightarrow \subseteq \mathsf{S}_{global} \times \mathsf{Act}_\ell \times \mathsf{S}_{global}$ is the smallest set such that, for $w = m_1 m_2 \ldots m_k$:

- If $s \overset{c!m}{\hookrightarrow} t$ then $\langle s, w\rangle \xrightarrow{c!m} \langle t, w[c := m_1 \ldots m_k m]\rangle$.
- If $s \overset{c?m}{\hookrightarrow} t$ and $k \geq 1$ then $\langle s, w[c := m m_1 \ldots m_k]\rangle \xrightarrow{c?m} \langle t, w\rangle$.
- If $k \geq 1$ and $i \in \{1, \ldots, k\}$ then $\langle s, w\rangle \xrightarrow{\ell_{c,i}} \langle s, w[c := m_1 \ldots m_{i-1} m_{i+1} \ldots m_k]\rangle$.
- If $s \overset{\tau}{\hookrightarrow} t$ then $\langle s, w\rangle \xrightarrow{\tau} \langle t, w\rangle$.

We write $\mathsf{s} \xrightarrow{\ell} \mathsf{t}$ iff $\mathsf{s} \xrightarrow{\ell_{c,i}} \mathsf{t}$ for some c and i and $\mathsf{s} \xrightarrow{\alpha}$ iff $\mathsf{s} \xrightarrow{\alpha} \mathsf{t}$ for some global state t. We define $act(\mathsf{s})$ to be the set of actions $\alpha \in Act$ that are *enabled* in the global state s. Formally, $act(\mathsf{s}) = \{\alpha \in Act : \mathsf{s} \xrightarrow{\alpha}\}$. In what follows, we require that in all global states at least one action is enabled. This is guaranteed by the requirement that, for any control state s, there is some action $\alpha \in SendAct \cup \{\tau\}$ and control state t with $s \overset{\alpha}{\hookrightarrow} t$.[9]

Definition 2. (cf. [IN97]) *A* PLCS *is a tuple $\mathcal{PL} = (\mathcal{L}, P_{control}, \wp)$ where $\mathcal{L}$ is a LCS, $\wp \in]0,1[$ the* failure probability *and*

$$P_{control} : S_{control} \times Act \times S_{control} \longrightarrow [0,1]$$

a function with $P_{control}(s, \alpha, t) > 0$ iff $s \overset{\alpha}{\hookrightarrow} t$.

The Markov chain associated with a PLCS $\mathcal{PL} = (\mathcal{L}, P_{control}, \wp)$ arises by augmenting the transitions of the transition system $\mathsf{TS}(\mathcal{L})$ with probabilities.

[9] Note that for any control state s where the system has terminated we may assume that there is a τ-loop, i.e. $s \overset{\tau}{\hookrightarrow} s$.

First, we define the probabilities for the action-labelled transitions. Then, we abstract from the action-labels and deal with the probabilities $\mathsf{P}_{global}(\mathsf{s},\mathsf{t})$ to move from s to t via any action. In any global state s where $|\mathsf{s}| \neq 0$, the probability for losing one of the messages is $\wp$ where all transitions $\mathsf{s} \xrightarrow{\ell_{c,i}} \mathsf{t}$ have equal probability. The other transition probabilities (for the transitions labelled by actions $\alpha \in Act$) are derived from $P_{control}$ (that assigns "weights" to the transitions) with the help of the *normalization function* $\nu : \mathsf{S}_{global} \longrightarrow I\!\!R_{>0}$ which is defined by:

$$\nu(\langle s, w\rangle) \;=\; \sum_{\alpha \in act(\langle s,w\rangle)} P_{control}(s,\alpha)$$

where $P_{control}(s,\alpha) \;=\; \sum_t \; P_{control}(s,\alpha,t)$.[10] The conditional probability (under the assumption that no message will be lost in the next step) for an α-labelled transition $\langle s, w\rangle \xrightarrow{\alpha} \langle s', w'\rangle$ is given by the "weight" $P_{control}(s,\alpha,s')$ divided by $\nu(\langle s,w\rangle)$. We define the action-labelled transition probability function P_{global} : $\mathsf{S}_{global} \times \mathsf{Act}_\ell \times \mathsf{S}_{global} \longrightarrow [0,1]$ as follows. If $\alpha \in Act$, $\langle s,w\rangle \xrightarrow{\alpha} \langle s',w'\rangle$, $|w| \neq 0$ then

$$\mathsf{P}_{global}(\langle s,w\rangle, \alpha, \langle s',w'\rangle) \;=\; \frac{1-\wp}{\nu(\langle s,w\rangle)} \cdot P_{control}(s,\alpha,s').$$

For the loss of a message, corresponding to the transition $\mathsf{s} \xrightarrow{\ell_{c,i}} \mathsf{t}$[11], we define

$$\mathsf{P}_{global}(\mathsf{s}, \ell_{c,i}, \mathsf{t}) \;=\; \frac{\wp}{|\mathsf{s}|}.$$

For the global states with empty channels we put $\mathsf{P}_{global}(\langle s,\emptyset\rangle, \alpha, \langle s',w'\rangle) = P_{control}(s,\alpha,s')/\nu(\langle s,\emptyset\rangle)$. In all remaining cases, we define $\mathsf{P}_{global}(\mathsf{s},\alpha,\mathsf{t}) = 0$. We define

$$\mathsf{P}_{global}(\mathsf{s},\alpha) \;=\; \sum_{\mathsf{t}\in \mathsf{S}_{global}} \mathsf{P}_{global}(\mathsf{s},\alpha,\mathsf{t}), \quad \mathsf{P}_{global}(\mathsf{s},\mathsf{t}) \;=\; \sum_{\alpha \in Act_\ell} \mathsf{P}_{global}(\mathsf{s},\alpha,\mathsf{t}).$$

The Markov chain[12] associated with $\mathcal{PL}$ is $\mathsf{MC}(\mathcal{PL}) \;=\; (\mathsf{S}_{global}, \mathsf{P}_{global}, \mathsf{L}, \mathsf{s}_0)$ where P_{global} is viewed as a function $\mathsf{S}_{global} \times \mathsf{S}_{global} \longrightarrow [0,1]$. Dealing with $LTL_{\setminus X}$ as formalism for specifying qualitative properties for PLCSs, we deal with the satisfaction relation $\mathcal{PL} \models f$ iff $\mathsf{s}_0 \models_{\mathsf{MC}(\mathcal{PL})} f$ where $\mathsf{s}_0 = \langle s_0, \emptyset\rangle$ is the initial global state of $\mathsf{MC}(\mathcal{PL})$.

[10] Since we assume that any control state s has at least one transition $s \overset{\alpha}{\hookrightarrow} t$ for some $\alpha \in SendAct \cup \{\tau\}$, the normalization factor $\nu(\langle s,w\rangle)$ is always > 0.

[11] Note that $|\mathsf{s}| \neq 0$ because we cannot lose a message from the empty channel.

[12] To be precisely, we deal with a *pointed* Markov chain by which we mean a Markov chain that is endowed with an initial state. For simplicity, we briefly refer to "pointed Markov chains" as "Markov chains".

4 Model checking

In this section, we describe a $LTL_{\setminus X}$ model checking procedure for PLCSs. More precisely, the input of our algorithm is a PLCS $\mathcal{PL}$ and a $LTL_{\setminus X}$ formula f; the output is "yes" or "no" depending on whether or not $\mathcal{PL} \models f$. The basic idea of our method is the reduction of the $LTL_{\setminus X}$ model checking problem to a reachability problem in a (non-probabilistic) LCS where the latter can be solved with the methods proposed in [AJ93] or [AKP97].

Before we explain how our algorithm works we briefly sketch the algorithmic methods that have been developed for verifying finite probabilistic systems against LTL formulas.

Courcoubetis & Yannakakis [CY88] deal with finite Markov chains and present an algorithm that is based on a recursive procedure that successively removes the temporal modalities from the formula (i.e. replaces each subformula g whose outermost operator is a temporal operator, e.g. $\mathcal{U}$, by a new atomic proposition a_g) where at the same time each state s of the underlying Markov chain M is splitted into the two states $\langle \mathsf{s}, a_g \rangle$ and $\langle \mathsf{s}, \neg a_g \rangle$. The transition probabilities in the new Markov chain M_g are computed with the help of the probabilities $p_{\mathsf{s}}(g)$ for the path formula g. This method is very tricky and elegant for finite Markov chains but it seems to be not adequate for infinite systems (like PLCSs) since it would require the computation of infinitely many transition probabilities.

An alternative method is based on the ω-automaton approach proposed by Vardi & Wolper [Var85,VW86]. This approach has been used later by several other authors, see e.g. [CY95,IN96,dA97,BK98a]. The basic idea behind the ω-automata theoretic approach can be sketched as follows. The starting point is a probabilistic system $\mathcal{S}$, e.g. described by a Markov chain or Markov decision process, and a linear time formula f. Using well-known methods, one constructs an ω-automaton $\mathcal{A}_f$ for the formula f and defines a new probabilistic system $\mathcal{S} \times \mathcal{A}_f$ by taking the "product" $\mathcal{S} \times \mathcal{A}_f$ of $\mathcal{S}$ and $\mathcal{A}_f$. From the acceptance condition of $\mathcal{A}_f$, a set V' of states in $\mathcal{S} \times \mathcal{A}_f$ can be derived such that the probability that f holds in a state s agrees with the probability for a certain state s' in $\mathcal{S} \times \mathcal{A}_f$ to reach a state in V'.

Similar ideas are used in the tableau-based method of Pnueli & Zuck [PZ93] where the "product" of the probabilistic system and the "tableau" for f (obtained from the Fischer-Ladner closure of f) is analyzed.

In this paper, we follow the approachs of [dA97,BK98a] and use a deterministic Rabin automaton to get an alternative characterization of the probability that a $LTL_{\setminus X}$ formula f holds in a global state.[13]

[13] [dA97,BK98a] deal with finite probabilistic systems with non-determinism, i.e. Markov Decision Processes rather than Markov chains. It is still open whether or not a *non-deterministic* ω-automaton would still be sufficient for our purposes as it is the case for finite Markov chains [CY95,IN96].

We recall the basic definitions and explain our notations. A *deterministic Rabin automaton* $\mathcal{A}$ is a tuple $(Q, q_0, Alph, \delta, AccCond)$ where

- Q is a non-empty finite set of states,
- $q_0 \in Q$ is the initial state,
- $Alph$ is a finite alphabet,
- $\delta : Q \times Alph \longrightarrow Q$ is the transition function,
- $AccCond$ is the acceptance condition, i.e. $AccCond \subseteq 2^Q \times 2^Q$.

An infinite sequence $\mathbf{p} = p_0, p_1, p_2, \ldots \in Q^\omega$ is said to satisfy the acceptance condition of the automaton $\mathcal{A}$ (denoted $\mathbf{p} \models AccCond$) iff there exists $(A, B) \in AccCond$ such that $inf(\mathbf{p}) \subseteq A$ and $inf(\mathbf{p}) \cap B \neq \emptyset$. Here, $inf(\mathbf{p})$ denotes the set of automaton states that occur infinitely often in $\mathbf{p}$.

A *run* $\mathbf{r}$ of $\mathcal{A}$ over an infinite word $a_0, a_1, a_2, \ldots \in Alph^\omega$ is a sequence $\mathbf{r} = q_0, q_1, q_2, \ldots \in Q^\omega$ (starting in the initial state q_0 of $\mathcal{A}$) with $q_{i+1} = \delta(q_i, a_i)$ for all $i \geq 0$. A run $\mathbf{r}$ of $\mathcal{A}$ is called *accepting* iff $\mathbf{r} \models AccCond$. A word $\mathbf{a} = a_0, a_1, a_2 \ldots \in Alph^\omega$ is called *accepted* iff there is an accepting run $\mathbf{r}$ over $\mathbf{a}$. Let $AccWords(\mathcal{A})$ denote the set of accepting words.

It is well-known [WVS83,Saf88,VW94] that, for any LTL formula f (in particular, for any $LTL_{\setminus X}$ formula) with atomic propositions in AP, a deterministic Rabin automaton $\mathcal{A}_f$ with the alphabet $Alph = 2^{AP}$ can be constructed such that $AccWords(\mathcal{A}_f)$ is exactly the set of infinite words $\mathbf{a} = a_0, a_1, \ldots$ over 2^{AP} where f is true.[14] The product $\mathsf{M} \times \mathcal{A}_f$ of a Markov chain $\mathsf{M} = (\mathsf{S}, \mathsf{P}, \mathsf{L})$ and the automaton $\mathcal{A}_f$ is defined as follows.

$$\mathsf{M} \times \mathcal{A}_f \;=\; (\mathsf{S} \times Q, \mathsf{P}', \mathsf{L}')$$

where $\mathsf{L}'(\langle \mathsf{s}, q\rangle) = \mathsf{L}(\mathsf{s})$ and

$$\mathsf{P}'(\langle \mathsf{s}, q\rangle, \langle \mathsf{t}, p\rangle) \;=\; \begin{cases} \mathsf{P}(\mathsf{s},\mathsf{t}) & \text{if } p = \delta(q, \mathsf{L}(\mathsf{t})) \\ 0 & \text{otherwise.} \end{cases}$$

Let $AccCond = \{(A_j, B_j) : j = 1, \ldots, k\}$ be the acceptance condition of $\mathcal{A}_f$. Hence we define $A'_j = \mathsf{S} \times A_j$, $B'_j = \mathsf{S} \times B_j$. Let V'_j be the smallest set such that $V'_j \subseteq A'_j$ and $Reach_{\mathsf{M}\times\mathcal{A}_f}(v') \subseteq V'_j$, $Reach_{\mathsf{M}\times\mathcal{A}_f}(v') \cap B'_j \neq \emptyset$ for all $v' \in V'_j$.[15] Let $V' = V'_1 \cup \ldots \cup V'_k$. As in [dA97,BK98a] it can be shown that

(*) $Prob_{\mathsf{M}}\{\pi \in Path_{\mathsf{M}}(\mathsf{s}) : \pi \models f\} = Prob_{\mathsf{M}\times\mathcal{A}_f}\{\pi \in Path_{\mathsf{M}\times\mathcal{A}_f}(\mathsf{s}') : \pi \models \Diamond V'\}$.

for all states $\mathsf{s} \in \mathsf{S}$. Here, s' denotes the state $\langle \mathsf{s}, \delta(q_0, \mathsf{L}(\mathsf{s}))\rangle$ and $\pi \models \Diamond V'$ is an abbreviation of "π will eventually reach a state of V'". Thus, the test whether $p_{\mathsf{s}}(f) = 1$ can be done by first computing $\mathcal{A}_f$ and then performing a probabilistic reachability analysis in the product $\mathsf{M} \times \mathcal{A}_f$ to check whether

[14] Here, satisfaction of LTL formulas interpreted over infinite words over 2^{AP} is defined in the obvious way.

[15] The existence of such a set V'_j can be shown with the help of Tarski's fixed point theorem for monotonic set-valued operators.

(**) $Prob_{\mathsf{M}\times\mathcal{A}_f}\{\pi \in Path_{\mathsf{M}\times\mathcal{A}_f}(\mathsf{s}') : \pi \models \Diamond V'\} = 1$.
For finite Markov chains, the latter (the test of (**)) can be done with non-probabilistic (graph theoretical) methods.[16] In our case, where we deal with infinite Markov chains obtained by a PLCS (i.e. Markov chains of the form $\mathsf{M} = \mathsf{MC}(\mathcal{PL})$), condition (*) still holds but it is not clear (at least not for the authors) how to test condition (**). The problem is that the reachability algorithm of [AJ93] (or [AKP97]) cannot be applied since the underlying transition system of the so obtained Markov chain $\mathsf{MC}(\mathcal{PL}) \times \mathcal{A}_f$ might not be the transition system of a LCS (see Remark 1). For this reason, we do not deal with the product $\mathsf{MC}(\mathcal{PL}) \times \mathcal{A}_f$ but switch to the product of the PLCS $\mathcal{PL}$ and the automaton $\mathcal{A}_f$ (which yields a new PLCS $\mathcal{PL} \times \mathcal{A}_f$) and then show how to apply conventional methods for a reachability analysis in the LCS $\mathcal{L} \times \mathcal{A}_f$ to reason about the probabilities in $\mathsf{MC}(\mathcal{PL} \times \mathcal{A}_f)$.

4.1 The product of a PLCS and an ω-automaton

In the sequel, let $\mathcal{PL}$ be a PLCS and $\mathcal{A}$ a deterministic Rabin automaton with the alphabet 2^{AP} where the components of $\mathcal{PL}$ and $\mathcal{A}$ are as before; i.e. $\mathcal{PL} = (\mathcal{L}, P_{control}, \wp)$ and $\mathcal{A} = (Q, q_0, 2^{AP}, \delta, AccCond)$ where $\mathcal{L}$ is as in Definition 1 and $AccCond = \{(A_j, B_j) : j = 1, \ldots, k\}$.

Definition 3. *$\mathcal{PL} \times \mathcal{A}$ denotes the PLCS $(\mathcal{L} \times \mathcal{A}, P_{\mathcal{A}}, \wp)$ where*

$$\mathcal{L} \times \mathcal{A} = (S_{control} \times Q, \langle s_0, p_0\rangle, L_{\mathcal{A}}, Ch, Mess, \hookrightarrow_{\mathcal{A}})$$

with $p_0 = \delta(q_0, L(s_0))$, $L_{\mathcal{A}}(\langle s, q\rangle) = L(s)$ and

$$\langle s, q\rangle \overset{\alpha}{\hookrightarrow}_{\mathcal{A}} \langle t, p\rangle \quad \textit{iff} \quad s \overset{\alpha}{\hookrightarrow} t \textit{ and } p = \delta(q, L(t))$$

and, if $\langle s, q\rangle \overset{\alpha}{\hookrightarrow}_{\mathcal{A}} \langle t, p\rangle$ then $P_{\mathcal{A}}(\langle s, q\rangle, \alpha, \langle t, p\rangle) = P_{control}(s, \alpha, t)$.

We use the notation $\langle s, w, q\rangle \in S_{control} \times (Ch \longrightarrow Mess^*) \times Q$ rather than $\langle\langle s, q\rangle, w\rangle$ for the global states in $\mathsf{MC}(\mathcal{PL} \times \mathcal{A})$.

Remark 1. The Markov chain $\mathsf{MC}(\mathcal{PL} \times \mathcal{A})$ induced by $\mathcal{PL} \times \mathcal{A}$ differs from the product $\mathsf{MC}(\mathcal{PL}) \times \mathcal{A}$. We assume that $q \neq q'$. We regard the loss of messages in both constructions. Let $q' = \delta(q, L(s))$ and $w : Ch \longrightarrow Mess^*$ such that $w.c = m_1 \ldots m_{i-1} m_i m_{i+1} \ldots m_k$ and $w' = w[c := m_1 \ldots m_{i-1} m_{i+1} \ldots m_k]$. In $\mathsf{MC}(\mathcal{PL}) \times \mathcal{A}$, the state $\langle s, w, q\rangle$ can move to $\langle s, w', q'\rangle$ (via the action $\ell_{c,i}$), but possibly not to the state $\langle s, w', q\rangle$. In $\mathsf{MC}(\mathcal{PL} \times \mathcal{A})$, we have

$$\langle s, w, q\rangle \xrightarrow{\ell_{c,i}}_{\mathcal{A}} \langle s, w', q\rangle \; .$$

[16] One just has to check whether all states reachable from the state s' via an execution sequence that does not pass V' can reach a V'-state.

Thus, $\mathsf{P}'(\langle s,w,q\rangle,\langle s,w',q\rangle) = 0 \ < \ \mathsf{P}_{global}(\langle s,w,q\rangle,\langle s,w',q\rangle)$ is possible.[17] This signifies that it is possible that the underlying graph of $\mathsf{MC}(\mathcal{PL}) \times \mathcal{A}$ cannot be obtained by the transition system of a LCS. ■

We now assume that $\mathcal{A} = \mathcal{A}_f$ is a deterministic automaton for a $LTL_{\setminus X}$ formula f. Recall that $p_{\mathsf{s}}^{\mathsf{M}}(f)$ denotes $Prob_{\mathsf{M}}\{\pi \in Path_{\mathsf{M}}(\mathsf{s}) : \pi \models_{\mathsf{M}} f\}$.

Lemma 1. *Let* s *be a global state in* $\mathcal{PL}$ *and* $\mathsf{s}' = \langle \mathsf{s}, \delta(q_0, \mathsf{L}(\mathsf{s}))\rangle$. *Then,*

$$p_{\mathsf{s}}^{\mathsf{MC}(\mathcal{PL})}(f) \ = \ p_{\mathsf{s}'}^{\mathsf{MC}(\mathcal{PL}\times\mathcal{A}_f)}(f).$$

Proof. For every path $\pi = \langle s_0, w_0\rangle\langle s_1, w_1\rangle\langle s_2, w_2\rangle \ldots$ in $\mathsf{MC}(\mathcal{PL})$ and for every $i \in \mathbb{N}_0$ the state $\langle s_{i+1}, w_{i+1}\rangle$ can be reached from $\langle s_i, w_i\rangle$ via different actions; this means that we can have transitions $\langle s_i, w_i, q_i\rangle \longrightarrow \langle s_{i+1}, w_{i+1}, q_{i+1}\rangle$ with $q_{i+1} = q_i$ in the case of a loss of a message or $q_{i+1} = \delta(q_i, L(s_{i+1}))$ in all other cases. Thus, a path in $\mathsf{MC}(\mathcal{PL})$ induces a set of according paths in $\mathsf{MC}(\mathcal{PL}\times\mathcal{A})$.[18] If we mark the transitions with labels and construct an action-labelled Markov chain from $\mathsf{MC}(\mathcal{PL})$ we can show that there is a one-one relation between the paths in this action-labelled Marcov chain and the paths in the Markov chain $\mathsf{MC}(\mathcal{PL} \times \mathcal{A})$.

After that, we will show that the probability of a measurable subset of paths in $\mathsf{MC}(\mathcal{PL})$ equals the probability of an "associated" measurable subset of paths in $\mathsf{MC}(\mathcal{PL} \times \mathcal{A})$. We will explain later what we mean by "associated" set.

The to the Markov chain $\mathsf{MC}(\mathcal{PL})$ associated action-labelled Markov chain $\mathsf{MC}_{action}(\mathcal{PL})$ is defined by:

$$\mathsf{MC}_{action}(\mathcal{PL}) = (\mathsf{S}_{global}, Act_{\ell}, \mathsf{P}_{global}, \mathsf{L}, \mathsf{s}_0)$$

where $\mathsf{S}_{global}, \mathsf{L}, \mathsf{s}_0, Act_{\ell}$ are as before and P_{global} is regarded (as in the original definition, see Page 8) as action-labelled transition probability function. In $\mathsf{MC}_{action}(\mathcal{PL})$ we deal with action labelled paths of the form

$$\langle s_0, w_0\rangle \xrightarrow{\alpha_1} \langle s_1, w_1\rangle \xrightarrow{\alpha_2} \langle s_2, w_2\rangle \xrightarrow{\alpha_3} \ldots$$

In the sequel we need the projection of a path π' in $\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})$ to a path π in $\mathsf{MC}_{action}(\mathcal{PL})$. We denote this projection by pr, i.e. if $\pi' = \langle s_0, w_0, q_0\rangle \xrightarrow{\alpha_1} \langle s_1, w_1, q_1\rangle \xrightarrow{\alpha_2} \ldots$ then

$$\mathsf{pr}(\pi') = \langle s_0, w_0\rangle \xrightarrow{\alpha_1} \langle s_1, w_1\rangle \xrightarrow{\alpha_2} \ldots$$

The definition of pr for a path $\pi \in \mathsf{MC}(\mathcal{PL} \times \mathcal{A})$ (this path has no action labels) is obvious.

---

17 Note that the control state (which consists in $\mathcal{L} \times \mathcal{A}$ of a control state in $\mathcal{L}$ and an automaton state) does not change if a message is lost.

18 The reader should notice that we do not make a difference between finite or infinite paths here, i.e. every (finite or infinite) path induces such a set.

Claim 1: Let $\pi \in Path_{\mathsf{MC}_{action}(\mathcal{PL})}$. Then there exists exactly one path $\pi' \in Path_{\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})}$ with $\mathsf{pr}(\pi') = \pi$.

Proof:

For every path $\pi = \langle s_0, w_0\rangle \xrightarrow{\alpha_1} \langle s_1, w_1\rangle \xrightarrow{\alpha_2} \langle s_2, w_2\rangle \ldots$ in $\mathsf{MC}_{action}(\mathcal{PL})$ we can construct a path $\pi' = \langle s_0, w_0, \delta(q_0, L(s_0))\rangle \xrightarrow{\alpha_1} \langle s_1, w_1, q_1\rangle \xrightarrow{\alpha_2} \langle s_2, w_2, q_2\rangle \ldots$ where q_i for $i \in \mathbb{N}_0$ is uniquely defined because of the action labels. Obviously we have $\mathsf{pr}(\pi') = \pi$.

As we already mentioned, a path in $\mathsf{MC}(\mathcal{PL})$ induces a set of according paths in $\mathsf{MC}(\mathcal{PL}\times\mathcal{A})$. Let π be a path in $\mathsf{MC}(\mathcal{PL})$, then we will denote the set of according paths in $\mathsf{MC}(\mathcal{PL}\times\mathcal{A})$ by $\Pi(\pi)$ and we have $\Pi(\pi) = \{\pi' \in Path_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})} : \mathsf{pr}(\pi') = \pi\}$. Clearly, we have $word(\pi') = word(\pi)$ for every path $\pi' \in \Pi(\pi)$.

Claim 2:

Let Δ be a measurable subset of $Path_{\mathsf{MC}(\mathcal{PL})}(\mathsf{s})$. Then,

a) the set $\Delta' = \bigcup_{\pi\in\Delta} \Pi(\pi)$ is a measurable subset of $Path_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}')$, and

b) $Prob_{\mathsf{MC}(\mathcal{PL})}(\Delta) = Prob_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}(\Delta')$.

Proof:

a) Naturally, a path in $\mathsf{MC}(\cdot)$ induces a set of paths in $\mathsf{MC}_{action}(\cdot)$ and we denote this set by $\Pi_{action}(\pi)$ for a path π. It is also clear that for a measurable subset Δ in $Path_{\mathsf{MC}}$ the set $\bigcup_{\pi\in\Delta} \Pi_{action}(\pi)$ is measurable and that $Prob_{\mathsf{MC}}(\Delta) = Prob_{\mathsf{MC}_{action}}(\bigcup_{\pi\in\Delta} \Pi_{action}(\pi))$.
Now, let Δ be a measurable subset of $Path_{\mathsf{MC}(\mathcal{PL})}(\mathsf{s})$. Since we have a one-one relation[19] between the paths in $\mathsf{MC}_{action}(\mathcal{PL})$ and the paths in $\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})$ we get that $\tilde{\Delta} = \bigcup_{\pi\in\Delta}\{\pi' \in Path_{\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}') : \mathsf{pr}(\pi') = \tilde{\pi}, \tilde{\pi} \in \Pi_{action}(\pi)\}$ is a measurable subset of $Path_{\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}')$. Thus, we get $\Delta' = \bigcup_{\pi\in\Delta}\{\pi' \in Path_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}') : \mathsf{pr}(\pi') = \pi\}$ is a measurable subset of $Path_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}')$.

b) Let B be a basic cylinder with prefix ρ. Then,

$$Prob_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}\Big(\bigcup_{\sigma'\in\Pi(\rho)} \{\pi' \in Path_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}') : \sigma' \text{ is a prefix of } \pi'\}\Big) =$$

$$\sum_{\sigma'\in\Pi(\rho)} Prob_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}\{\pi' \in Path_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}') : \sigma' \text{ is a prefix of } \pi'\} =$$

$$\sum_{\sigma'\in\Pi(\rho)} Prob_{\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})}\Big(\bigcup_{\tilde{\sigma}'\in\Pi_{action}(\sigma')} \{\tilde{\pi}' \in Path_{\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}') : \tilde{\sigma}' \text{ is a prefix of } \tilde{\pi}'\}\Big) =$$

[19] In Claim 1 we only have shown the direction from $\mathsf{MC}_{action}(\mathcal{PL})$ to $\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})$. We have not shown the other direction since it is easy to see.

$$\sum_{\sigma' \in \Pi(\rho)} \sum_{\tilde{\sigma}' \in \Pi_{action}(\sigma')} Prob_{\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})}\{\tilde{\pi}' \in Path_{\mathsf{MC}_{action}(\mathcal{PL}\times\mathcal{A})}(\mathsf{s}') : \tilde{\sigma}' \text{ is a prefix of } \tilde{\pi}'\} =$$

$$\sum_{\tilde{\sigma} \in \Pi_{action}(\rho)} Prob_{\mathsf{MC}_{action}(\mathcal{PL})}\{\tilde{\pi} \in Path_{\mathsf{MC}_{action}(\mathcal{PL})}(\mathsf{s}) : \tilde{\sigma} \text{ is a prefix of } \tilde{\pi}\}$$

where $\tilde{\pi}$ and $\tilde{\sigma}$ are the uniquely defined paths in $\mathsf{MC}_{action}(\mathcal{PL})$ to $\tilde{\pi}'$ and $\tilde{\sigma}'$ (see Claim 1). The last equation holds because of the definition of $P_{\mathcal{A}}$, Claim 1 and the fact that a path in $\mathsf{MC}(\mathcal{PL})$ induces a set of paths in $\mathsf{MC}(\mathcal{PL} \times \mathcal{A})$.

$$\sum_{\tilde{\sigma} \in \Pi_{action}(\rho)} Prob_{\mathsf{MC}_{action}(\mathcal{PL})}\{\tilde{\pi} \in Path_{\mathsf{MC}_{action}(\mathcal{PL})}(\mathsf{s}) : \tilde{\sigma} \text{ is a prefix of } \tilde{\pi}\} = Prob_{\mathsf{MC}(\mathcal{PL})}(\{\pi \in Path_{\mathsf{MC}(\mathcal{PL})}(\mathsf{s}) : \rho \text{ is a prefix of } \pi\})$$

Thus we get for every measurable set Δ:

$$Prob_{\mathsf{MC}(\mathcal{PL})}(\Delta) = Prob_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A})}(\Delta') .$$

With Claim 1 and Claim 2 we get

$$\begin{aligned}
p_{\mathsf{s}}^{\mathsf{MC}(\mathcal{PL})}(f) &= Prob_{\mathsf{MC}(\mathcal{PL})}\left\{\pi \in Path_{\mathsf{MC}(\mathcal{PL})}(\mathsf{s}) : \pi \models f\right\} \\
&= Prob_{\mathsf{MC}(\mathcal{PL})}\left\{\pi \in Path_{\mathsf{MC}(\mathcal{PL})}(\mathsf{s}) : word(\pi) \in AccWords(\mathcal{A}_f)\right\} \\
&= Prob_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A}_f)}\left\{\pi' \in Path_{\mathsf{MC}(\mathcal{PL}\times\mathcal{A}_f)}(\mathsf{s}') : word(\pi') \in AccWords(\mathcal{A}_f)\right\} \\
&= p_{\mathsf{s}'}^{\mathsf{MC}(\mathcal{PL}\times\mathcal{A}_f)}(f)
\end{aligned}$$

■

For the construction $\mathsf{M} \times \mathcal{A}_f$, the projection of a path π in $\mathsf{M} \times \mathcal{A}_f$ to the automaton states yields a run in $\mathcal{A}_f$ which is accepting iff $\pi \models f$. Unfortunately, the projection of the paths in $\mathsf{MC}(\mathcal{PL} \times \mathcal{A}_f)$ to the automaton states does not yield a run in $\mathcal{A}_f$ since the loss of a message (more precisely, a step of the form $\langle s, w, q\rangle \xrightarrow{\ell} \langle s, w', q\rangle$ where $\delta(q, L(s)) \neq q$) does not correspond to a transition in $\mathcal{A}_f$. However, the loss of a message does not affect the control and automaton state and hence can be viewed as a *stutter step*. Since we do not deal with the next step operator and since the atomic propositions only depend on the control components (but not on the channel contents), the formula f is insensitive with respect to such stutter steps [BCG88]. Thus, $\pi \models f$ iff $\mathbf{r}$ is accepting where $\mathbf{r}$ is the run induced by the sequence of automaton states that results from π by removing all stutter steps.

Let $A_j' = S_{control} \times A_j$, $B_j' = S_{control} \times B_j$. In the sequel, we treat A_j', B_j' as atomic propositions with the obvious meaning; e.g. $A_j' \in L_{\mathcal{A}}(\langle s, q\rangle)$ if $\langle s, q\rangle \in A_j'$.

Lemma 2. *For any path π in $\mathsf{MC}(\mathcal{PL} \times \mathcal{A}_f)$:*

$$\pi \models f \quad \textit{iff} \quad \pi \models \bigvee_{1 \le j \le k} \Diamond\Box(A_j' \wedge \Diamond B_j').$$

Proof. We denote by $\equiv_{st}$ the stuttering equivalence relation for infinite sequences $\mathbf{x} = x_0, x_1, x_2, \ldots$ over an arbitrary set X; i.e. $\equiv_{st}$ is the smallest equivalence relation on X^ω which identifies all sequences $\mathbf{x} = x_0, x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots$ and $\mathbf{x}' = x_0, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots$ where $x_i = x_{i+1}$. Let $\mathbf{a} = a_0, a_1, \ldots$ and $\mathbf{a}' = a_0', a_1', \ldots$ be infinite sequences over 2^{AP}. It is well known that in absence of the next step operator X stuttering equivalent sequences over a set of atomic propositions satisfy the same linear time formulas [BCG88]. Hence,

(1) If $\mathbf{a} \equiv_{st} \mathbf{a}'$ then $\mathbf{a} \models f$ iff $\mathbf{a}' \models f$.

(2) Let $\mathbf{p}$, $\mathbf{p}' \in Q^\infty$. If $\mathbf{p} \equiv_{st} \mathbf{p}'$ then $\mathbf{p} \models AccCond$ iff $\mathbf{p}' \models AccCond$.

Let π be a path in $\mathsf{MC}(\mathcal{PL} \times \mathcal{A}_f)$ and $\pi(i) = \langle \vec{s}_i, w_i, p_i \rangle$. For any i, we choose some $\alpha_i \in Act \cup \{\ell\}$[20] such that $\pi(i) \xrightarrow{\alpha_i} \pi(i+1)$. Note that there are infinitely many indices i with $\alpha_i \neq \ell$ as we only deal with finite channel contents which means that we only can have a finite number of ℓ's consecutively. Let $\langle \vec{s}_0{}', w_0', p_0' \rangle, \langle \vec{s}_1{}', w_1', p_1' \rangle, \ldots$ be the sequence that results from π by removing the i-th tuple $\pi(i) = \langle \vec{s}_i, w_i, p_i \rangle$ if $\alpha_i = \ell$. Let $\mathbf{a} = L(\vec{s}_0), L(\vec{s}_1), \ldots$, $\mathbf{a}' = L(\vec{s}_0{}'), L(\vec{s}_1{}'), \ldots$, $\mathbf{p} = p_0, p_1, \ldots$ and $\mathbf{p}' = p_0', p_1', \ldots$. We have $\langle \vec{s}_i, p_i \rangle = \langle \vec{s}_{i+1}, p_{i+1} \rangle$ for all indices i with $\alpha_i = \ell$. Thus, $\mathbf{a} \equiv_{st} \mathbf{a}'$ and $\mathbf{p} \equiv_{st} \mathbf{p}'$. By definition of $\mathcal{PL} \times \mathcal{A}_f$, we have $p_{i+1}' = \delta(p_i', L(\vec{s}_{i+1}{}'))$, $i = 0, 1, 2, \ldots$. Thus, $\mathbf{p}'$ is a run over $\mathbf{a}'$. Hence, $\mathbf{a}' \models f$ iff $\mathbf{a}' \in AccWords(\mathcal{A}_f)$ iff $\mathbf{p}' \models AccCond$. By (1) and (2) $\pi \models f$ iff $\mathbf{a} \models f$ iff $\mathbf{a}' \models f$ iff $\mathbf{p}' \models AccCond$ iff $\mathbf{p} \models AccCond$. Clearly, $\mathbf{p} \models AccCond$ is an equivalent formulation for $\pi \models \bigvee_j \Diamond\Box(A_j' \wedge \Diamond B_j')$. ■

4.2 Probabilistic input enabledness

Because of Lemma 1 and Lemma 2 we can shrink our attention to formulas of the form $\bigvee \Diamond\Box(a_j \wedge \Diamond b_j)$ where a_j, b_j are atomic propositions. We aim at a condition that allows to establish qualitative properties specified by formulas of this type by analyzing the graph of the underlying LCS. For this, we need a condition that allows us to abstract from the concrete transition probabilities. In contrast to the finite-state case, for infinite Markov chains, the precise transition probabilities might be essential for establishing qualitative properties.

Example 1. The Markov chain of Figure 1 can be viewed as the Markov chain associated with a PLCS consisting of a single control state s, one channel c, one message m, the transition $s \overset{c!m}{\hookrightarrow} s$ and the failure probability $\wp = p$. Then, the state s_k of Figure 1 represents the global state $\langle s, m^k \rangle$ in which the total channel

[20] The reader should be aware of the notation of ℓ; $\{\ell\}$ stands for the set $\{\ell_{c,i},\ c \in Ch,\ i \in \{1, 2 \ldots\}\}$.

length is k. The qualitative property stating that the initial global state s_0 is visited infinitely often holds for $p \geq 1/2$ but not for $p < 1/2$. ■

The problem in the above example is that, for $p < 1/2$, with non-zero probability, the channels grow in an "uncontrolled" way. To prevent such situations, we shrink our attention to *probabilistic input enabled* PLCSs. Probabilistic input enabledness is a condition which ensures that with probability at least $1/2$ any global state s moves within one step to a global state t where $|\mathsf{t}| = |\mathsf{s}| - 1$ and which guarantees that almost all executions visit infinitely many global states where all channels are empty (see Lemma 3). In particular, it ensures that with probability 1 any message m received in a certain channel c will either be lost or will be consumed by a process (via the action $c?m$).

The formal definition of probabilistic input enabledness can be viewed as a probabilistic "variant" of the standard notion of input enabledness for I/O-automata, see [LT87,Lyn95]. In fact we work with a slightly different meaning of input enabledness. For I/O-automata, communication works synchronously and input enabledness guarantees that the output of messages cannot be blocked. This effect is already obtained for systems where the communication works asynchronously (as for LCSs). Our notion of input enabledness can be viewed as a condition that asserts some kind of "channel fairness" as it rules out the pathological case where a certain message m (produced and send by a process via the action $c!m$) is totally ignored (i.e. never lost nor consumed via the action $c?m$). We adapt the notion of input enabledness for I/O-automata (which asserts that in any (global) state all input actions are enabled) for PLCSs in such a way that, for any global state s where $|\mathsf{s}| \geq 1$, the probability for any input action $c?m$ is "sufficiently" large.

Definition 4. *A PLCS $\mathcal{PL}$ is called* probabilistic input enabled *iff for all $s \in S_{control}$ and all $c \in Ch$, $m \in Mess$:*

$$P_{control}(s, c?m) \geq \ (1 - 2\wp) \cdot \left(\sum_{\alpha \in SendAct \cup \{\tau\}} P_{control}(s, \alpha) \right).$$

It should be noticed that any PLCS with failure probability $\wp \geq 1/2$ is probabilistic input enabled. Clearly, with $\mathcal{PL}$, also the product $\mathcal{PL} \times \mathcal{A}$ is probabilistic input enabled. In the sequel, we assume that $\mathcal{PL} = (\mathcal{L}, P_{control}, \wp)$ is a probabilistic input enabled PLCS where $\mathcal{L}$ is as in Definition 1.

Let $\mathsf{S}_\emptyset = \{\mathsf{s} \in \mathsf{S}_{global} : |\mathsf{s}| = 0\}$ be the set of all global states where all channels are empty. We write $\pi \models \Box\Diamond \mathsf{S}_\emptyset$ to denote that π passes infinitely many global states in $\mathsf{S}_\emptyset$, i.e. $|\pi(i)| = 0$ for infinitely many indices i.

Lemma 3. *For all global states* s*:*

$$\sum_{\substack{\mathsf{t} \\ |\mathsf{t}|=|\mathsf{s}|-1}} \mathsf{P}_{global}(\mathsf{s},\mathsf{t}) \;\geq\; \tfrac{1}{2}.$$

and $Prob\{\pi \in Path(\mathsf{s}) \;:\; \pi \models \Box\Diamond \mathsf{S}_\emptyset \;\} \;=\; 1.$

Proof. For the proof of the first part we define $\ell_{Set}(\mathsf{s}) = \{\ell_{c,i} \;:\; c \in Ch, i \in \{1,\ldots,|\mathsf{s}|\}\;\}$ and $\nu'(\mathsf{s}) = \sum_{\alpha \in SendAct \cup \{\tau\}} P_{control}(\mathsf{s},\alpha)$.

$$\begin{aligned}
\sum_{\substack{\mathsf{t} \\ |\mathsf{t}|=|\mathsf{s}|-1}} \mathsf{P}_{global}(\mathsf{s},\mathsf{t}) &\stackrel{Def}{=} \sum_{\substack{\mathsf{t} \\ |\mathsf{t}|=|\mathsf{s}|-1}} \sum_{\alpha \in \mathsf{Act}_\ell} \mathsf{P}_{global}(\mathsf{s},\alpha,\mathsf{t}) \\
&= \sum_{\substack{\mathsf{t} \\ |\mathsf{t}|=|\mathsf{s}|-1}} \left(\sum_{\alpha \in \ell_{Set}(\mathsf{s})} \mathsf{P}_{global}(\mathsf{s},\alpha,\mathsf{t}) + \sum_{\alpha \in RecAct} \mathsf{P}_{global}(\mathsf{s},\alpha,\mathsf{t}) \right) \\
&\stackrel{Def}{=} \sum_{\alpha \in \ell_{Set}(\mathsf{s})} \frac{\wp}{|\mathsf{s}|} + \sum_{\substack{\mathsf{t} \\ |\mathsf{t}|=|\mathsf{s}|-1}} \sum_{\alpha \in RecAct} \frac{1-\wp}{\nu(\mathsf{s})} \cdot P_{control}(\mathsf{s},\alpha,\mathsf{t}) \\
&\stackrel{Def}{=} \wp + \sum_{\alpha \in RecAct} \frac{1-\wp}{\nu(\mathsf{s})} \cdot P_{control}(\mathsf{s},\alpha) \\
&= \wp + \frac{1-\wp}{\nu(\mathsf{s})} \cdot \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha) \\
&\stackrel{Def}{=} \wp + \frac{1-\wp}{\nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha)} \cdot \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha) \\
&= \frac{\wp \cdot \nu'(\mathsf{s}) + \wp \cdot \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha)}{\nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha)} \\
&\quad - \frac{\wp \cdot \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha)}{\nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha)} \\
&= \frac{\wp \cdot \nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha)}{\nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s},\alpha)}
\end{aligned}$$

$$= \frac{\wp \cdot \nu'(\mathsf{s}) + \nu'(\mathsf{s}) - \nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s}, \alpha)}{\nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s}, \alpha)}$$

$$= 1 - \nu'(\mathsf{s}) \cdot \frac{1 - \wp}{\nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s}, \alpha)}$$

Let $\tilde{\alpha} = c?m$ be an arbitrary element of *RecAct*. Then,

$$\begin{aligned}
1 - \nu'(\mathsf{s}) \cdot \frac{1 - \wp}{\nu'(\mathsf{s}) + \sum_{\alpha \in RecAct} P_{control}(\mathsf{s}, \alpha)} &\geq 1 - \nu'(\mathsf{s}) \cdot \frac{1 - \wp}{\nu'(\mathsf{s}) + P_{control}(\mathsf{s}, \tilde{\alpha})} \\
&\geq 1 - \nu'(\mathsf{s}) \cdot \frac{1 - \wp}{\nu'(\mathsf{s}) \cdot (1 + 1 - 2\wp)} \\
&= \frac{1}{2}
\end{aligned}$$

For the second part it suffices to show that $\mathsf{p}(\mathsf{s}) = 1$ for all global states s where $\mathsf{p}(\mathsf{s}) = Prob\{\pi \in Path(\mathsf{s}) : \pi \models \Diamond \mathsf{S}_\emptyset\}$. We put $\mathsf{p}(k) = \min\{\, \mathsf{p}(\mathsf{s}) : \mathsf{s} \in \mathsf{S}_{global}, |\mathsf{s}| \leq k\}$ for $k \in \mathbb{N}_0$. Then, $1 = \mathsf{p}(0) \geq \mathsf{p}(1) \geq \ldots$. Let $\mathsf{s} \in \mathsf{S}_{global}$, $|\mathsf{s}| = k$ where $k \geq 1$. Then,

$$\begin{aligned}
\mathsf{p}(\mathsf{s}) = & \sum_{\substack{\mathsf{t} \\ |\mathsf{t}| \in \{k, k+1\}}} \mathsf{P}_{global}(\mathsf{s}, \mathsf{t}) \cdot \mathsf{p}(\mathsf{t}) + \sum_{\substack{\mathsf{t} \\ |\mathsf{t}| = k-1}} \mathsf{P}_{global}(\mathsf{s}, \mathsf{t}) \cdot \mathsf{p}(\mathsf{t}) \\
\geq & \sum_{\substack{\mathsf{t} \\ |\mathsf{t}| \in \{k, k+1\}}} \mathsf{P}_{global}(\mathsf{s}, \mathsf{t}) \cdot \mathsf{p}(k+1) + \sum_{\substack{\mathsf{t} \\ |\mathsf{t}| = k-1}} \mathsf{P}_{global}(\mathsf{s}, \mathsf{t}) \cdot \mathsf{p}(k-1) \\
\geq & \, (1 - Q(k, k-1)) \cdot \mathsf{p}(k+1) + Q(k, k-1) \cdot \mathsf{p}(k-1),
\end{aligned}$$

where $Q(k, k-1) = \min\{\mathsf{P}_{global}(\mathsf{s}, \mathsf{t}) : \mathsf{s}, \mathsf{t} \in \mathsf{S}_{global}, |\mathsf{s}| = |\mathsf{t}| + 1 \leq k\}$. By Lemma 3 we get $Q(k, k-1) \geq \frac{1}{2}$. Let $p = \inf_{k \geq 1} Q(k, k-1)$. Then,

$$\mathsf{p}(k) \geq (1 - p) \cdot \mathsf{p}(k+1) + p \cdot \mathsf{p}(k-1).$$

We can now define an operator $\mathsf{F} : (\mathbb{N}_0 \to [0,1]) \longrightarrow (\mathbb{N}_0 \to [0,1])$ such that $\mathsf{F}(f)(k) = (1-p) \cdot f(k+1) + p \cdot f(k-1)$. Obviously we get $\mathsf{F}(\mathsf{p}) \leq \mathsf{p}$. For the least fixpoint of F, denoted by $\mathsf{lfp}(\mathsf{F})$, we get the situation as in Figure 1, which is known in the literature as random walk. There $\mathsf{lfp}(\mathsf{F})(k)$ can be viewed as the probability to reach s_0 from s_k, and (since $p \geq \frac{1}{2}$) we get $\mathsf{lfp}(\mathsf{F})(k) = 1$ for all $k \geq 1$. Thus, we get $\mathsf{lfp}(\mathsf{F}) \equiv 1$ and since $\mathsf{p} \geq \mathsf{lfp}(\mathsf{F})$ we finally get $\mathsf{p}(k) = 1$ for all $k \geq 1$. ■

We now show how, for probabilistic input enabled PLCSs, qualitative properties specified by a formula $f' = \bigvee \Diamond\Box(a_j \wedge \Diamond b_j)$ can be established by proving a qualitative eventually property $\Diamond U$ where U is a finite set of control states. For showing that $p_s(\Diamond U) = 1$, we use a reachability analysis in the underlying (non-probabilistic) LCS.[21] More precisely, the set U is defined by means of the *bottom strongly connected components* (BSCCs for short) of the directed graph $G_\emptyset(\mathcal{L})$ whose nodes represent the global states $\langle s, \emptyset\rangle$ and whose edges represent the reachability relation between them. The condition $p_\mathsf{s}(\Diamond U) = 1$ can shown to be equivalent to $p_\mathsf{s}(\Diamond \overline{U}) = 0$ where $\overline{U}$ characterizes all global states $\langle s, \emptyset\rangle$ that belong to a BSCC of $G_\emptyset(\mathcal{L})$ and that are not contained in U. To check whether $p_\mathsf{s}(\Diamond \overline{U}) = 0$, it suffices to show that the global state s cannot reach a global state $\langle \overline{u}, \emptyset\rangle$ where $\overline{u} \in \overline{U}$.

Definition 5. *Let $\mathcal{L}$ be a LCS as in Definition 1. We define*

$$G_\emptyset(\mathcal{L}) = (S_{control}, \rightsquigarrow_\mathcal{L})$$

where the relation $\rightsquigarrow_\mathcal{L} \subseteq S_{control} \times S_{control}$ is given by $s \rightsquigarrow_\mathcal{L} t$ iff the global state $\langle t, \emptyset\rangle$ is reachable from the global state $\langle s, \emptyset\rangle$ in $\mathsf{TS}(\mathcal{L})$.

If $U \subseteq S_{control}$ then we write $s \rightsquigarrow_\mathcal{L} U$ iff $s \rightsquigarrow_\mathcal{L} u$ for some $u \in U$. $s \not\rightsquigarrow_\mathcal{L} U$ denotes that there is no $u \in U$ with $s \rightsquigarrow_\mathcal{L} u$.

Notation 1. *For any global state s, we define $\Pi_{BSCC}(\mathsf{s})$ to be the set of paths $\pi \in Path(\mathsf{s})$ such that, for some BSCC C of $G_\emptyset(\mathcal{L})$, all global states $\langle \vec{t}, \emptyset\rangle$, $\vec{t} \in C$, are visited infinitely often.*

Lemma 4. *For all global states s:*

$$Prob(\Pi_{BSCC}(\mathsf{s})) = 1\,.$$

Proof. We define a set Γ by

$$\Gamma = \bigcup_{\substack{\mathsf{t},\mathsf{u} \\ \mathsf{P}_{global}(\mathsf{t},\mathsf{u})>0}} \Gamma_{\mathsf{t},\mathsf{u}}$$

where $\Gamma_{\mathsf{t},\mathsf{u}} = \{\pi \in Path \;:\; t \in \inf(\pi), \mathsf{u} \notin \inf(\pi)\}$

First we show that $Prob(\Gamma_{\mathsf{t},\mathsf{u}}(\mathsf{s})) = 0$ (see the following Claim). We will use this fact to show that the probability of the paths which are not in Γ and which satisfy $\Box\Diamond \mathsf{S}_\emptyset$ is 1. With the help of this fact and two observations we finally can show that $Prob(\Pi_{BSCC}(\mathsf{s})) = 1$.

[21] We write $p_\mathsf{s}(\Diamond U)$ to denote the probability for the global state s to reach a global state of the form $\langle u, w\rangle$ for some $u \in U$.

Claim: Let t, u be global states such that $\mathsf{P}_{global}(\mathsf{t}, \mathsf{u}) > 0$. Let $\Gamma_{\mathsf{t},\mathsf{u}}$ be defined as above, i.e. $\Gamma_{\mathsf{t},\mathsf{u}} = \{\pi \in Path \ : \ t \in \inf(\pi), \mathsf{u} \notin \inf(\pi)\}$. Then, for all global states s:

$$Prob(\Gamma_{\mathsf{t},\mathsf{u}}(\mathsf{s})) = 0 \,.$$

Proof: We define a set

$$\Gamma'_{\mathsf{t},\mathsf{u}} = \{\pi \in Path \ : \ t \in \inf(\pi), \pi(i) \neq \mathsf{u} \quad \forall i \in \mathbb{N}\} \,.$$

In a very similar manner as in [BK98b] we define a set

$$\Omega_{\mathsf{s}} = \{\rho \in Path_{fin}(\mathsf{s}) \ : \rho(i) \neq \mathsf{t}, \rho(i) \neq \mathsf{u}, \ i = 1, \ldots, |\rho| - 1, last(\rho) = \mathsf{t}\}$$

where $|\rho|$ denotes the length of the finite path ρ and $last(\rho)$ denotes the last element in the finite path ρ. It is obvious that

$$\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{s}) = \bigcup_{\rho \in \Omega_{\mathsf{s}}} \rho \Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t}) \,.$$

Now we have to show that

$$Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) = 0 \,.$$

Since $\mathsf{P}_{global}(\mathsf{t}, \mathsf{u}) > 0$ we get

$$\sum_{\rho \in \Omega_{\mathsf{t}}} \mathsf{P}(\rho) \leq \sum_{\mathsf{r} \neq \mathsf{u}} \mathsf{P}_{global}(\mathsf{t}, \mathsf{r}) = 1 - \mathsf{P}_{global}(\mathsf{t}, \mathsf{u}) \,.$$

Further on we have

$$Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) = \sum_{\rho \in \Omega_{\mathsf{t}}} \mathsf{P}(\rho) \cdot Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) \,.$$

Thus we get

$$Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) \leq (1 - \mathsf{P}_{global}(\mathsf{t}, \mathsf{u})) \cdot Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) \,.$$

It follows that

$$\mathsf{P}_{global}(\mathsf{t}, \mathsf{u}) \cdot Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) \leq 0 \,.$$

and since $\mathsf{P}_{global}(\mathsf{t}, \mathsf{u}) > 0$ we have

$$Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) = 0$$

We now show that $Prob(\Gamma_{\mathsf{t},\mathsf{u}}(\mathsf{s})) = 0$. It is easy to see that

$$\Gamma_{\mathsf{t},\mathsf{u}} \subseteq \bigcup_{\rho} \rho \Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})$$

where ρ ranges over all finite paths such that $last(\rho) = \mathsf{t}$ and there is some $j < |\rho|$ with $\rho(j) = \mathsf{u}$ and $\rho(l) \neq \mathsf{u}$ for $l = j+1, \ldots, |\rho| - 1$ and $\rho(j-1) = \mathsf{t}$. It follows that $Prob(\Gamma_{\mathsf{t},\mathsf{u}}(\mathsf{s})) \leq \sum_\rho \mathsf{P}(\rho) \cdot Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t}))$ where ρ ranges over the paths described above. Since $Prob(\Gamma'_{\mathsf{t},\mathsf{u}}(\mathsf{t})) = 0$ we get $Prob(\Gamma_{\mathsf{t},\mathsf{u}}(\mathsf{s})) = 0$ for all global states s. $\square$

We know now that $Prob(\Gamma_{\mathsf{t},\mathsf{u}}(\mathsf{s})) = 0$ for all global states s. In the sequel we will investigate the paths that are not in Γ. For a path π which is not in Γ it follows that, for all $\mathsf{t} \in \inf(\pi)$ and for all u with $\mathsf{P}_{global}(\mathsf{t},\mathsf{u}) > 0$ deduces that u is an element which appears infinitely often on the path π, e.g. $\mathsf{u} \in \inf(\pi)$.

In particular, this means that for $\pi \notin \Gamma$ and $\mathsf{t} \in \inf(\pi)$ it follows that $Reach(\mathsf{t}) \subseteq \inf(\pi)$. Let Π denote the set of all paths which are not in Γ and where infinitely often a state in $\mathsf{S}_\emptyset$ is reached, e.g. $\Pi = \{\pi \in Path \setminus \Gamma \,:\, \pi \models \Box\Diamond \mathsf{S}_\emptyset\}$.

Thus,

$$Prob(\Pi(\mathsf{s})) = 1$$

for all global states s by Lemma 3 and the above Claim.

Let $\pi \in \Pi$. Then,

(1) $\inf(\pi) \cap \mathsf{S}_\emptyset \neq \emptyset$
(2) $\forall \mathsf{s} \in \inf(\pi) \cap \mathsf{S}_\emptyset \,:\, Reach(\mathsf{s}) \subseteq \inf(\pi)$

Since $Prob(\Pi(\mathsf{s})) = 1$ and by (1) and (2) we get $Prob(\Pi_{BSCC}(\mathsf{s})) = 1$. ■

Notation 2. *Let a_j, $b_j \in AP$ and $A_j = \{s \in S_{control} : a_j \in L(s)\}$, $B_j = \{s \in S_{control} : b_j \in L(s)\}$. Let U_j be the union of all BSCCs C of $G_\emptyset(\mathcal{L})$ such that $C \subseteq A_j$ and $C \cap B_j \neq \emptyset$, $j = 1, \ldots, k$, and $U = U_1 \cup \ldots \cup U_k$; consequently $\overline{U}$ is the union of all BSCCs C of $G_\emptyset(\mathcal{L})$ such that, for all $j \in \{1, \ldots, k\}$, either $C \not\subseteq A_j$ or $C \cap B_j = \emptyset$.*

Lemma 5. *For all global states* s*:*

$$p_{\mathsf{s}}(\Diamond U) \;+\; p_{\mathsf{s}}(\Diamond \overline{U}) \;=\; 1$$

Proof. With the definition of $p_{\mathsf{s}}(\Diamond U)$ and Lemma 4 we get

$$\begin{aligned} p_{\mathsf{s}}(\Diamond U) \;+\; p_{\mathsf{s}}(\Diamond \overline{U}) &= Prob\{\pi \in Path(\mathsf{s}) \,:\, (\pi \models \Diamond U) \vee (\pi \models \Diamond \overline{U})\} \\ &= Prob\{\pi \in \Pi_{BSCC}(\mathsf{s}) \,:\, (\pi \models \Diamond U) \vee (\pi \models \Diamond \overline{U})\} \\ &= 1 \end{aligned}$$

Lemma 6. *For all control states $\vec{s}$:*

$$Prob \left\{ \pi \in Path(\langle \vec{s}, \emptyset \rangle) : \pi \models \bigvee_{1 \leq j \leq k} \Diamond\Box(a_j \wedge \Diamond b_j) \right\} \;=\; 1 \quad \mathit{iff} \quad \vec{s} \not\rightsquigarrow_{\mathcal{L}} \overline{U}.$$

Proof. It is easy to see that $\pi \models \Diamond\Box(a_j \wedge \Diamond b_j)$ iff $\pi \models \Diamond U_j$ for any path $\pi \in \Pi_{BSCC}(\mathsf{s})$. By Lemma 4 and Lemma 5, we get:

$$p_{\mathsf{s}}\left(\bigvee_{1\le j\le k} \Diamond\Box(a_j \wedge \Diamond b_j)\right) = p_{\mathsf{s}}(\Diamond U) = 1 - p_{\mathsf{s}}(\Diamond \overline{U}).$$

Hence, $p_{\mathsf{s}}\left(\bigvee_{1\le j\le k} \Diamond\Box(a_j \wedge \Diamond b_j)\right) = 1$ iff $p_{\mathsf{s}}(\Diamond \overline{U}) = 0$. Since any global state $\langle \overline{u}, w\rangle$ can reach the state $\langle \overline{u}, \emptyset\rangle$ (via losing all messages), we have $p_{\mathsf{s}}(\Diamond \overline{U}) = 0$ iff s cannot reach a global state of the form $\langle \overline{u}, \emptyset\rangle$ where $\overline{u} \in \overline{U}$. ■

4.3 The model checking algorithm

Combining Lemma 1, 2 and 6 we get the following theorem which builds the basis of our model checking algorithm.

Theorem 1. *Let $\mathcal{PL} = (\mathcal{L}, P_{control}, \wp)$ be a probabilistic input enabled PLCS where $\mathcal{L}$ is as in Definition 1, f a $LTL_{\setminus X}$ formula and $\mathcal{A}_f$ a deterministic Rabin automaton for f. Let $\overline{U}'$ be the union of all BSCCs C' of the directed graph $G_\emptyset(\mathcal{L} \times \mathcal{A}_f)$ such that, for all $j \in \{1, \dots, k\}$, either $C' \not\subseteq A'_j$ or $C' \cap B'_j = \emptyset$. Then,*

$$\mathcal{PL} \models f \ \text{ iff } \ s'_0 \not\rightsquigarrow_{\mathcal{L}\times\mathcal{A}_f} \overline{U}' .$$

Here, $s'_0 = \langle s_0, \delta(q_0, L(s_0))\rangle$ denotes the initial control state of $\mathcal{L} \times \mathcal{A}_f$ and A'_j, B'_j are as in Lemma 2.

With all the above preliminaries, we are now able to formulate our model checking algorithm. (see Figure 2). The input is a probabilistic input enabled PLCS $\mathcal{PL}$ and a $LTL_{\setminus X}$ formula f. First, we construct a deterministic Rabin automaton $\mathcal{A}_f$ for f and the LCS $\mathcal{L} \times \mathcal{A}_f$. Then, we compute the reachability relation $\rightsquigarrow_{\mathcal{L}\times\mathcal{A}_f}$ for the LCS $\mathcal{L} \times \mathcal{A}_f$ which yields the graph $G_\emptyset(\mathcal{L} \times \mathcal{A}_f)$. For this, we may apply the methods of [AJ93] (or [AKP97]).

Using standard methods of graph theory, we calculate the BSCCs of the graph $G_\emptyset(\mathcal{L} \times \mathcal{A}_f)$ and obtain the set $\overline{U}'$ (defined as in Theorem 1). Finally, we check whether the initial control state s'_0 of $\mathcal{L} \times \mathcal{A}_f$ can reach a node of $\overline{U}'$ with respect to the edge relation $\rightsquigarrow_{\mathcal{L}\times\mathcal{A}_f}$.

5 Conclusion and future work

We have shown that, for probabilistic input enabled PLCSs, model checking against qualitative $LTL_{\setminus X}$ specifications is decidable. This should be contrasted with the undecidability of LTL model checking for (non-probabilistic) LCSs [AJ94].[22] Thus, adding appropriate transition probabilities to a LCS, can be

[22] Note that in the probabilistic setting, a linear time formula f is viewed to hold in a state s iff f holds on *almost all* paths starting in s (but f might be wrong on some paths) while, in the non-probabilistic case, f is viewed to be correct for a state s iff f holds on *all* paths starting in s.

Input: a probabilistic input enabled PLCS $\mathcal{PL} = (\mathcal{L}, P_{control}, \wp)$ and a $LTL_{\setminus X}$ formula f

Output: if $\mathcal{PL} \models f$ then **yes** else **no**

Method:

1. Compute the deterministic Rabin automaton $\mathcal{A}_f$ for the formula f.
2. Compute the LCS $\mathcal{L} \times \mathcal{A}_f$.
3. Compute the reachability relation $\leadsto_{\mathcal{L} \times \mathcal{A}_f}$ (which yields the graph $G_\emptyset(\mathcal{L} \times \mathcal{A}_f)$).
4. Compute the set $\overline{U}'$ (defined as in Theorem 1) by means of the BSCCs in $G_\emptyset(\mathcal{L} \times A_f)$.
5. If $s_0' \not\leadsto_{\mathcal{L} \times \mathcal{A}_f} \overline{U}'$ then return **yes** else return **no**.

Fig. 2. The $LTL_{\setminus X}$ model checking algorithm

viewed as a technique to overcome the limitations of algorithmic verification that are due to undecidability results.

Whether or not the probabilistic input enabledness is a necessary condition is still open. The correctness of our method is based on the observation that, with probability 1, a BSCC C of the graph $G_\emptyset(\mathcal{L})$ is reached and that all states of C are visited infinitely often. This property holds for probabilistic input enabled systems (see Lemma 6) but is wrong for general PLCSs as we have seen in Example 1.

In this paper, we used the interpretation of a PLCS by a (sequential) Markov chain as proposed in [IN97]. This model is adequate e.g. if the underlying parallel composition for the processes that communicate via the channels is a probabilistic shuffle operator in the style of [BBS92]. This kind of parallel composition assumes a scheduler that decides randomly (according to the "weights" specified by the function $P_{control}$) which of the processes performs the next step. Alternatively, the global behaviour of a PLCS could be described by a model for probabilistic systems with non-determinism (such as concurrent Markov chains [Var85] or the more general models of [BdA95,Seg95,BK98a]), where the non-determinism can be used to describe the interleaving behaviour of the communicating processes.

Unfortunately, we cannot report on experimental results. The implementation of our algorithm (combined with the methods of [AJ93] or [AKP97]), case studies and a complexity analysis will be future topics. Moreover, we intend to investigate how our algorithm can be modified for probabilistic systems with non-determinism and an interpretation of $LTL_{\setminus X}$ formulas over PLCSs that involve (process) fairness, i.e. an interpretation in the style $\mathcal{PL} \models f$ iff f holds with probability 1 for any fair scheduler. Another future direction is to study a CTL^*-like temporal logic that combines $LTL_{\setminus X}$ and the branching time logic of [HS86] where state formulas of the form $\forall f$ (asserting that f holds with probability 1) are considered.

References

[ABJ98] P. Abdulla, A. Bouajjani, and B. Jonsson. On-the-fly analysis of systems with unbounded, lossy FIFO channels. *LNCS*, 1427:305–318, 1998.

[AJ93] P. Abdulla and B. Jonsson. Verifying programs with unreliable channels. *Proc. LICS'93*, pages 160–170, 1993. The full version with the same title has been published in *Information and Computation*, 127:91–101, 1996.

[AJ94] P. Abdulla and B. Jonsson. Undecidable verification problems for programs with unreliable channels. *LNCS*, 820:316–327, 1994. The full version with the same title has been published in *Information and Computation*, 130:71–90, 1996.

[AK95] P. Abdulla and M. Kindahl. Decidability of simulation and bisimulation between lossy channel systems and finite state systems. *LNCS*, 962:333–347, 1995.

[AKP97] P. Abdulla, M. Kindahl, and D. Peled. An improved search strategy for lossy channel systems. In *PSTV/FORTE*. Chapman-Hall, 1997.

[ASBS95] A. Aziz, V. Singhal, R. Brayton, and A. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. *Proc. CAV'95*, 939:155–165, 1995.

[BBS92] J. Baeten, J. Bergstra, and S. Smolka. Axiomatizing probabilistic processes: ACP with generative probabilities (extended abstract). *CONCUR'92*, 630:472–485, 1992. The full version with the same title has been published in *Information and Computation*, 122:234–255, 1995.

[BCG88] M. Browne, E. Clarke, and O. Grumberg. Characterizing finite Kripke structures in propositional temporal logic. *Theoretical Computer Science*, 59:115–131, 1988.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. *LNCS*, 1026:499–513, 1995.

[BH97] C. Baier and H. Hermanns. Weak bisimulation for fully probabilistic processes. *LNCS*, 1254:119–130, 1997.

[BK98a] C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11:125–155, 1998.

[BK98b] C. Baier and M. Kwiatkowska. On the verification of qualitative properties of probabilistic processes under fairness constraints. *Information Processing Letters*, 66(2):71–79, 1998.

[BM98] A. Bouajjani and R. Mayr. Model checking lossy vector addition systems. 1998. To appear in *Proc. STACS'99, LNCS.*

[Bre68] L. Breiman. *Probability*. Addison-Wesley Publishing Company, 1968.

[BSW69] K. Bartlett, R. Scantlebury, and P. Wilkinson. A note on reliable full-duplex transmission over half-duplex links. *Communications of the ACM*, 12(5):260–261, 1969.

[BZ83] D. Brand and P. Zafiropulo. On communicating finite-state machines. *Journal of the ACM*, 30(2):323–342, 1983.

[CC91] L. Christoff and I. Christoff. Efficient algorithms for verification of equivalences for probabilistic processes. *Proc. CAV'91, LNCS*, 575:310–321, 1991.

[CC92] L. Christoff and I. Christoff. Reasoning about safety and liveness properties for probabilistic processes. *Proc. 12th Conference on Foundations of Software Technology and Theoretical Computer Science, LNCS*, 652:342–355, 1992.

[CES86] E. Clarke, E. Emerson, and A. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2):244–263, 1986.

[CFI96] G. Cécé, A. Finkel, and S. Iyer. Unreliable channels are easier to verify than perfect channels. *Information and Computation*, 124(1):20–31, 1996.

[CY88] C. Courcoubetis and M. Yannakakis. Verifying temporal properties of finite-state probabilistic programs. *Proc. FOCS'88*, pages 338–345, 1988.

[CY95] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the ACM*, 42(4):857–907, 1995.

[dA97] L. de Alfaro. Temporal logics for the specification of performance and reliability. *Proc. STACS'97, LNCS*, 1200:165–176, 1997.

[Eme90] E. Emerson. Temporal and modal logic. *Handbook of Theoretical Computer Science*, B:995–1072, 1990.

[Fel68] W. Feller. *An Introduction to Probability Theory and its Application.* John Wiley and Sons, New York, 1968.

[Fin94] A. Finkel. Decidability of the termination problem for completely specified protocols. *Distributed Computing*, 7(3):129–135, 1994.

[HJ94] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[HS86] S. Hart and M. Sharir. Probabilistic propositional temporal logics. *Information and Control*, 70(2/3):97–155, 1986. This is the extended version of "Probabilistic Temporal Logics for Finite and Bounded Models". In *Proc. STOCS'84*, 1–13, 1984.

[HSP83] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent program. *ACM Transactions on Programming Languages and Systems*, 5(3):356–380, 1983.

[HT92] T. Huynh and L. Tian. On some equivalence relations for probabilistic processes. *Fundamenta Informaticae*, 17:211–234, 1992.

[IN96] P. Iyer and M. Narasimha. "Almost always" and "sometime definitely" are not enough: Probabilistic quantifiers and probabilistic model-checking. Technical Report TR-96-16, Department of Computer Science, North Carolina State University, 1996.

[IN97] P. Iyer and M. Narasimha. Probabilistic lossy channel systems. *Proc. TAPSOFT'97, LNCS*, 1214:667–681, 1997.

[ISO79] ISO. Data communications - HDLC procedures - elements of procedures. Technical Report TR-ISO-4335, International Standards Organization, Geneva, 1979.

[JS90] C. Jou and S. Smolka. Equivalences, congruences, and complete axiomatizations for probabilistic processes. In *Proc. CONCUR'90, LNCS*, 458:367–383, 1990.

[LS82] D. Lehmann and S. Shelah. Reasoning about time and chance. *Information and Control*, 53(3):165–198, 1982.

[LS92] K. Larsen and A. Skou. Compositional verification of probabilistic processes. In *CONCUR'92, LNCS*, 630:456–471, 1992.

[LT87] N. Lynch and M. Tuttle. Hierarchical Correctness Proofs For Distributed Algorithms. *PODC'87*, pages 137–151, 1987.

[Lyn95] N. Lynch. *Distributed Algorithms.* Morgan Kaufmann Series in Data Management Systems. Morgan Kaufmann Publishers, 1995.

[MP92] Z. Manna and A. Pnueli. *The Temporal Logic of Reactive and Concurrent Systems-Specification.* Springer-Verlag, 1992.

[PZ93] A. Pnueli and L. Zuck. Probabilistic Verification. *Information and Computation*, 103(1):1–29, 1993. This is the extended version of "Probabilistic Verification by Tableaux". In *Proc. LICS'86*, 322–331, 1986.

[Saf88] S. Safra. On the complexity of ω-automata. *FOCS'88*, pages 319–327, 1988.

[Seg95] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, Massachusetts Institute of Technology, 1995.

[Tho90] W. Thomas. Automata on infinite objects. *Handbook of Theoretical Computer Science*, B:133–191, 1990.

[Var85] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. *FOCS'85*, pages 327–338, 1985.

[Var96] M. Vardi. An automata-theoretic approach to linear temporal logic. *LNCS*, 1043:238–266, 1996.

[VW86] M. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. *LICS '86*, pages 332–345, 1986.

[VW94] M. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.

[WVS83] P. Wolper, M. Vardi, and A. Sistla. Reasoning about infinite computation paths (extended abstract). *FOCS'83*, pages 185–194, 1983.