



**HAL**  
open science

# Introducing Divergence for Infinite Probabilistic Models

Alain Finkel, Serge Haddad, Lina Ye

► **To cite this version:**

Alain Finkel, Serge Haddad, Lina Ye. Introducing Divergence for Infinite Probabilistic Models. RP 2023 - 17th International Conference on Reachability Problems, Oct 2023, Nice, France. pp.127-140, 10.1007/978-3-031-45286-4\_10 . hal-04271224

**HAL Id: hal-04271224**

**<https://hal.science/hal-04271224v1>**

Submitted on 6 Nov 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



# Introducing Divergence for Infinite Probabilistic Models

Alain Finkel<sup>1,2</sup> , Serge Haddad<sup>1</sup> , and Lina Ye<sup>1,3</sup> 

<sup>1</sup> Université Paris-Saclay, CNRS, ENS Paris-Saclay, Laboratoire Méthodes Formelles, Gif-sur-Yvette, France

{alain.finkel, serge.haddad, lina.ye}@ens-paris-saclay.fr

<sup>2</sup> Institut Universitaire de France (IUF), Paris, France

<sup>3</sup> CentraleSupélec, Gif-sur-Yvette, France

**Abstract.** Computing the reachability probability in infinite state probabilistic models has been the topic of numerous works. Here we introduce a new property called *divergence* that when satisfied allows to compute reachability probabilities up to an arbitrary precision. One of the main interest of divergence is that our algorithm does not require the reachability problem to be decidable. Then we study the decidability of divergence for probabilistic versions of pushdown automata and Petri nets where the weights associated with transitions may also depend on the current state. This should be contrasted with most of the existing works that assume weights independent of the state. Such an extended framework is motivated by the modeling of real case studies. Moreover, we exhibit some divergent subclasses of channel systems and pushdown automata, particularly suited for specifying open distributed systems and networks prone to performance collapsing in order to compute the probabilities related to service requirements.

**Keywords:** Reachability probability · Infinite state probabilistic models · Divergence

## 1 Introduction

**Probabilistic Models.** In the 1980's, finite-state Markov chains have been considered for the modeling and analysis of probabilistic concurrent finite-state programs [19]. Since the 2000's, many works have been done to verify the infinite-state Markov chains obtained from probabilistic versions of automata extended with unbounded data (like stacks, channels, counters and clocks)<sup>1</sup>. The (qualitative and quantitative) model checking of *probabilistic pushdown automata* (pPDA) is studied in many papers, for example in [6, 10–12, 17] (see [5] for a

---

This work has been supported by ANR project BRAVAS (ANR-17-CE40-0028) with Alain Finkel and ANR project MAVeriQ (ANR-20-CE25-0012) with Serge Haddad.

<sup>1</sup> Surprisingly, in 1972, to the best of our knowledge, Santos gave the first definition of *probabilistic pushdown automata* [18] that did not open up a new field of research at the time.

survey). In 1997, Iyer and Narasimha [15] started the study of *probabilistic lossy channel systems* (pLCS) and later both some qualitative and quantitative properties were shown decidable for pLCS [1]. *Probabilistic counter machines* (pCM) have also been studied [7–9].

**Computing the Reachability Probability.** In finite Markov chains, there is a well-known algorithm for computing exactly the reachability probabilities in polynomial time [3]. Here we focus on the problem of *Computing the Reachability Probability up to an arbitrary precision* (CRP) in *infinite* Markov chains. There are (at least) two possible research directions:

The first one is to consider the Markov chains associated with a particular class of probabilistic models (like pPDA or probabilistic Petri nets (pPN)) and some specific target sets and to exploit the properties of these models to design a CRP-algorithm. For instance in [5], the authors exhibit a PSPACE algorithm for pPDA and PTIME algorithms for single-state pPDA and for one-counter automata.

The second one consists in exhibiting a property of Markov chains that yields a generic algorithm for solving the CRP problem and then looking for models that generate Markov chains that fulfill this property. *Decisiveness* of Markov chains is such a property. Intuitively, decisiveness w.r.t.  $s_0$  and  $A$  means that almost surely the random path  $\sigma$  starting from  $s_0$  will reach  $A$  or some state  $s'$  from which  $A$  is unreachable. It has been shown that pPDA are not (in general) decisive but both pLCS and probabilistic Petri nets (pPN) are decisive (for pPN: when the target set is upward-closed [2]).

**Two Limits of the Previous Approaches.** The generic approach based on the decisiveness property has numerous applications but suffers the restriction that the reachability problem must be decidable in the corresponding non deterministic model. To the best of our knowledge, all generic approaches rely on a *decidable reachability problem*.

In most of the works, the probabilistic models associate a *constant* weight for transitions and get transition probabilities by normalizing these weights among the enabled transitions in the current state. This *forbids to model phenomena* like congestion in networks (resp. performance collapsing in distributed systems) when the number of messages (resp. processes) exceeds some threshold leading to an increasing probability of message arrivals (resp. process creations) before message departures (resp. process terminations).

## Our Contributions

- In order to handle realistic phenomena (like congestion in networks), we consider *dynamic* weights i.e., weights depending on the current state.
- We introduce the new *divergence* property of Markov chains w.r.t.  $s_0$  and  $A$ : given some precision  $\theta$ , one can discard a set of states with either a small probability to be reached from  $s_0$  or a small probability to reach  $A$  such that the remaining subset of states is finite and thus allows for an approximate computation of the reachability probability up to  $\theta$ . For divergent Markov chains, we

provide a generic algorithm for the CRP-problem that *does not require* the decidability of the reachability problem. While decisiveness and divergence are not exclusive (both hold for finite Markov chains), they are complementary. In fact, divergence is somehow related to transience of Markov chains while decisiveness is somehow related to recurrence [13].

- In order to check divergence, we provide several simpler sufficient conditions based on existing and new results of martingale theory.
- We study for different models the decidability of divergence. Our first undecidability result implies that whatever the infinite models, one must restrict the kind of dynamics weights. Here we limit to polynomial weights, i.e. where a weight is defined by a polynomial whose variables are characteristics of the current state (e.g. the marking of a place in a Petri net).
- We prove, by a case study analysis, that divergence is decidable for a subclass of polynomial pPDA (i.e. pPDA with *polynomial* weights). We show that divergence is undecidable for polynomial pPNs w.r.t. an upward closed set.
- We provide two classes of divergent polynomial models. The first one is a probabilistic version of channel systems particularly suited for the modeling of open queuing networks. The second one is the probabilistic version of pushdown automata restricted to some typical behaviors of dynamic systems.

**Organisation.** Section 2 recalls Markov chains, introduces divergent Markov chains, presents an algorithm for solving the CRP-problem. In Sect. 3, we study the decidability status of divergence for pPDA and pPN. Finally Sect. 4 presents two divergent subclasses of probabilistic channel systems and pPDA. All missing proofs and a second CRP-algorithm when reachability is decidable can be found in [14].

## 2 Divergence of Markov Chains

### 2.1 Markov Chains: Definitions and Properties

**Notations.** A set  $S$  is *countable* if there exists an injective function from  $S$  to the set of natural numbers: hence it could be finite or countably infinite. Let  $S$  be a countable set of elements called states. Then  $Dist(S) = \{\Delta : S \rightarrow \mathbb{R}_{\geq 0} \mid \sum_{s \in S} \Delta(s) = 1\}$  is the set of *distributions* over  $S$ . Let  $\Delta \in Dist(S)$ , then the *support* of  $\Delta$  is defined by  $Supp(\Delta) = \Delta^{-1}(\mathbb{R}_{>0})$ . Let  $T \subseteq S$ , then  $S \setminus T$  will also be denoted  $\bar{T}$ .

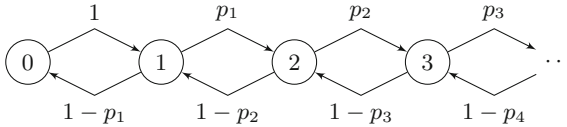
**Definition 1 (Effective Markov chain).** A Markov chain  $\mathcal{M} = (S, p)$  is a tuple where:

- $S$  is a countable set of states,
- $p$  is the transition function from  $S$  to  $Dist(S)$ ;

When for all  $s \in S$ ,  $Supp(p(s))$  is finite and computable and the function  $p$  is computable, one says that  $\mathcal{M}$  is effective.

**Notations.** The function  $p$  may be viewed as a  $S \times S$  matrix defined by  $p(s, s') = p(s)(s')$ . Let  $p^{(d)}$  denote the  $d^{th}$  power of the transition matrix  $p$ . When  $S$  is countably infinite, we say that  $\mathcal{M}$  is *infinite* and we sometimes identify  $S$  with  $\mathbb{N}$ . We also denote  $p(s, s') > 0$  by  $s \xrightarrow{p(s, s')} s'$ . A Markov chain is also viewed as a transition system whose transition relation  $\rightarrow$  is defined by  $s \rightarrow s'$  if  $p(s, s') > 0$ . Let  $A \subseteq S$ , one denotes  $Post_{\mathcal{M}}^*(A)$ , the set of states that can be reached from some state of  $A$  and  $Pre_{\mathcal{M}}^*(A)$ , the set of states that can reach  $A$ . As usual, we denote  $\rightarrow^*$ , the transitive closure of  $\rightarrow$  and we say that  $s'$  is *reachable from*  $s$  if  $s \rightarrow^* s'$ . We say that a subset  $A \subseteq S$  is *reachable* from  $s$  if some  $s' \in A$  is reachable from  $s$ . Note that every finite path of  $\mathcal{M}$  can be extended into (at least) one infinite path.

*Example 1.* Let  $\mathcal{M}_1$  be the Markov chain of Fig. 1. In any state  $i > 0$ , the probability for going to the “right”,  $p(i, i + 1)$ , is equal to  $0 < p_i < 1$  and for going to the “left”  $p(i, i - 1)$  is equal to  $1 - p_i$ . In state 0, one goes to 1 with probability 1.  $\mathcal{M}_1$  is effective if the function  $n \mapsto p_n$  is computable.



**Fig. 1.** A random walk  $\mathcal{M}_1$

Given an initial state  $s_0$ , the *sampling* of a Markov chain  $\mathcal{M}$  is an *infinite random sequence of states* (i.e., a path)  $\sigma = s_0 s_1 \dots$  such that for all  $i \geq 0$ ,  $s_i \rightarrow s_{i+1}$ . As usual, the corresponding  $\sigma$ -algebra whose items are called events is generated by the finite prefixes of infinite paths and the probability of an event  $Ev$  given an initial state  $s_0$  is denoted  $\mathbf{Pr}_{\mathcal{M}, s_0}(Ev)$ . In case of a finite path  $s_0 \dots s_n$ ,  $\mathbf{Pr}_{\mathcal{M}, s_0}(s_0 \dots s_n) = \prod_{0 \leq i < n} p(s_i, s_{i+1})$ .

**Notations.** From now on, **G** (resp. **F**, **X**) denotes the always (resp. eventual, next) operator of LTL.

Let  $A \subseteq S$ . We say that  $\sigma$  *reaches*  $A$  if  $\exists i \in \mathbb{N} s_i \in A$  and that  $\sigma$  *visits*  $A$  if  $\exists i > 0 s_i \in A$ . The probability that starting from  $s_0$ , the path  $\sigma$  reaches (resp. visits)  $A$  will be denoted by  $\mathbf{Pr}_{\mathcal{M}, s_0}(\mathbf{F}A)$  (resp.  $\mathbf{Pr}_{\mathcal{M}, s_0}(\mathbf{XFA})$ ).

We now state qualitative and quantitative properties of a Markov chain.

**Definition 2 (Irreducibility, recurrence, transience).** Let  $\mathcal{M} = (S, p)$  be a Markov chain and  $s \in S$ . Then  $\mathcal{M}$  is *irreducible* if for all  $s, s' \in S$ ,  $s \rightarrow^* s'$ .  $s$  is *recurrent* if  $\mathbf{Pr}_{\mathcal{M}, s}(\mathbf{X}\mathbf{F}\{s\}) = 1$  otherwise  $s$  is *transient*.

In an irreducible Markov chain, all states are in the same category, either recurrent or transient [16]. Thus an irreducible Markov chain will be said transient or recurrent depending on the category of its states. In the remainder of this section, we will relate this category with techniques for computing reachability probabilities.

*Example 2.* Clearly  $\mathcal{M}_1$  is irreducible. Moreover (see [14]),  $\mathcal{M}_1$  is recurrent if and only if  $\sum_{n \in \mathbb{N}} \prod_{1 \leq m < n} \rho_m = \infty$  with  $\rho_m = \frac{1-p_m}{p_m}$ , and when transient, the probability that starting from  $i$  the random path visits 0 is equal to  $\frac{\sum_{i \leq n} \prod_{1 \leq m < n} \rho_m}{\sum_{n \in \mathbb{N}} \prod_{1 \leq m < n} \rho_m}$ .

One of our goals is to approximately compute reachability probabilities in infinite Markov chains. Let us formalize it. Given a finite representation of a subset  $A \subseteq S$ , one says that this representation is *effective* if one can decide the membership problem for  $A$ . With a slight abuse of language, we identify  $A$  with any effective representation of  $A$ . The Computing of Reachability Probability (CRP) problem is defined by:

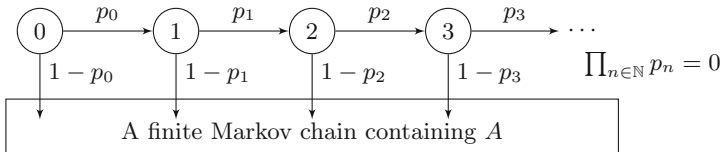
- Input: an effective Markov chain  $\mathcal{M}$ , an (initial) state  $s_0$ , an effective subset of states  $A$ , and a rational number  $\theta > 0$ .
- Output: an interval  $[low, up]$  such that  $up - low \leq \theta$  and  $\mathbf{Pr}_{\mathcal{M}, s_0}(\mathbf{F}A) \in [low, up]$ .

## 2.2 Divergent Markov Chains

Let us first discuss two examples before introducing the notion of *divergent* Markov chains.

*Example 3.* Consider again the Markov chain  $\mathcal{M}_1$  of Fig. 1 with for all  $n > 0$ ,  $p_n = p > \frac{1}{2}$ . In this case, for  $m \geq 0$ ,  $\mathbf{Pr}_{\mathcal{M}_1, m}(\mathbf{F}\{0\}) = \rho^m$  with  $\rho = \frac{1-p}{p}$ . Thus here the key point is that not only this reachability probability is less than 1 but it goes to 0 when  $m$  goes to  $\infty$ . This means that given some precision  $\theta$ , one could “prune” states  $n \geq n_0$  and compute the reachability probabilities of  $A$  in a finite Markov chain.

Consider the Markov chain of Fig. 2, where  $\mathbf{Pr}_{\mathcal{M}, 0}(\mathbf{F}\{m, m+1, \dots\}) = \prod_{n < m} p_n$  goes to 0 when  $m$  goes to  $\infty$ . As in the precedent example, computing the reachability probabilities of  $A$  can be also done in a finite Markov chain after pruning states  $n \geq n_0$ , given some precision  $\theta$ .



**Fig. 2.** An infinite (divergent) Markov chain

Intuitively, a divergent Markov chain w.r.t.  $s_0$  and  $A$  generalizes these examples: given some precision  $\theta$ , one can discard a set of states with either a small probability to be reached from  $s_0$  ( $f_0^{-1}([0, \theta])$  in the next definition) or a small probability to reach  $A$  (from any state of the set  $f_1^{-1}([0, \theta])$  in the next definition), such that the remaining subset of states is finite and thus allows for an approximate computation of the reachability probability up to  $\theta$ .

**Definition 3 (divergent Markov chain).** Let  $\mathcal{M}$  be a Markov chain,  $s_0 \in S$  and  $A \subseteq S$ . We say that  $\mathcal{M}$  is divergent w.r.t.  $s_0$  and  $A$  if there exist two computable functions  $f_0$  and  $f_1$  from  $S$  to  $\mathbb{R}_{\geq 0}$  such that:

- For all  $0 < \theta < 1$ ,  $\Pr_{\mathcal{M}, s_0}(\mathbf{F}f_0^{-1}([0, \theta])) \leq \theta$ ;
- For all  $s \in S$ ,  $\Pr_{\mathcal{M}, s}(\mathbf{F}A) \leq f_1(s)$ ;
- For all  $0 < \theta < 1$ ,  $\{s \mid f_0(s) \geq \theta \wedge f_1(s) \geq \theta\} \cap \text{Post}_{\mathcal{M}}^*(\{s_0\})$  is finite.

**Observation and Illustration.** Let us remark that there may not exist, for general Markov chains, an algorithm to decide the existence of such functions  $f_0, f_1$  and if there exist, to find them. Indeed as for decisiveness, divergence is a semantical property. But there exist some simpler sufficient conditions for divergence.

A finite Markov chain is divergent (letting  $f_0 = f_1 = 1$ ) w.r.t. any  $s_0$  and any  $A$ . In the first Markov chain of Example 3,  $f_0 = 1$  and  $f_1(m) = \rho^m$  and in the second Markov chain,  $f_1 = 1$ ,  $f_0(m) = \prod_{0 \leq n < m} p_n$  and  $f_0(s) = 1$  for all  $s$  in the finite Markov chain containing  $A$ . Generalizing these two examples, the next proposition introduces a sufficient condition for divergence. Its proof is immediate by choosing ( $f = f_0$  and  $f_1 = 1$ ) or ( $f = f_1$  and  $f_0 = 1$ ).

**Proposition 1.** Let  $\mathcal{M}$  be a Markov chain,  $s_0 \in S$ ,  $A \subseteq S$ , and a computable function  $f$  from  $S$  to  $\mathbb{R}_{\geq 0}$  such that:

- For all  $0 < \theta < 1$ ,  $\Pr_{\mathcal{M}, s_0}(\mathbf{F}f^{-1}([0, \theta])) \leq \theta$   
or for all  $s \in S$ ,  $\Pr_{\mathcal{M}, s}(\mathbf{F}A) \leq f(s)$ ;
- For all  $0 < \theta < 1$ ,  $\{s \mid f(s) \geq \theta\} \cap \text{Post}_{\mathcal{M}}^*(\{s_0\})$  is finite.

Then  $\mathcal{M}$  is divergent w.r.t.  $s_0$  and  $A$ .

### 2.3 An Algorithm for Divergent Markov Chains

We now design an algorithm for accurately framing the reachability probability for a divergent (effective) Markov chain w.r.t.  $s_0$  and an effective  $A$ .

Let us describe this algorithm. It performs an exploration of reachable states from  $s_0$  maintaining  $S'$ , the set of visited states, and stopping an exploration when the current state  $s$  fulfills: either (1) for some  $i \in \{0, 1\}$ ,  $f_i(s) \leq \frac{\theta}{2}$  in which case  $s$  is inserted in the *AlmostLoose<sub>i</sub>* set (initially empty), or (2)  $s \in A$  in which case  $s$  is inserted in  $A'$  (initially empty). When the exploration is ended, if  $A'$  is empty, the algorithm returns the interval  $[0, \theta]$ . Otherwise it builds  $\mathcal{M}' = (S', p')$  a finite Markov chain over  $S'$  whose transition probabilities are the ones of  $\mathcal{M}$  except for the states of *AlmostLoose<sub>0</sub>*  $\cup$  *AlmostLoose<sub>1</sub>*  $\cup$   $A'$ , which are made absorbing. Finally it computes the vector of reachability probabilities starting from  $s_0$  in  $\mathcal{M}'$  (function `CompFinProb`) and returns the interval  $[\text{preach}(A'), \text{preach}(A') + \text{preach}(\text{AlmostLoose}_0) + \frac{\theta}{2} \cdot \text{preach}(\text{AlmostLoose}_1)]$ . The next proposition establishes the correctness of the algorithm.

**Proposition 2.** Let  $\mathcal{M}$  be a divergent Markov chain with  $s_0 \in S$ ,  $A \subseteq S$  and  $\theta > 0$ . Then Algorithm 1 solves the CRP problem.

---

**Algorithm 1:** Framing the reachability probability
 

---

```

CompProb( $\mathcal{M}, s_0, A, \theta$ )
 $AlmostLoose_0 \leftarrow \emptyset$ ;  $AlmostLoose_1 \leftarrow \emptyset$ ;  $S' \leftarrow \emptyset$ 
 $A' \leftarrow \emptyset$ ;  $Front \leftarrow \emptyset$ ; Insert( $Front, s_0$ )
while  $Front \neq \emptyset$  do
     $s \leftarrow \text{Extract}(Front)$ ;  $S' \leftarrow S' \cup \{s\}$ 
    if  $f_0(s) \leq \frac{\theta}{2}$  then  $AlmostLoose_0 \leftarrow AlmostLoose_0 \cup \{s\}$ 
    else if  $f_1(s) \leq \frac{\theta}{2}$  then  $AlmostLoose_1 \leftarrow AlmostLoose_1 \cup \{s\}$ 
    else if  $s \in A$  then  $A' \leftarrow A' \cup \{s\}$ 
    else for  $s \rightarrow s' \wedge s' \notin S'$  do Insert( $Front, s'$ )
end
if  $A' = \emptyset$  then return  $(0, \theta)$ 
 $Abs \leftarrow AlmostLoose_0 \cup AlmostLoose_1 \cup A'$ 
for  $s \in Abs$  do  $p'(s, s) \leftarrow 1$ 
for  $s \in S' \setminus Abs \wedge s' \in S'$  do  $p'(s, s') \leftarrow p(s, s')$ 
 $preach \leftarrow \text{CompFinProb}(\mathcal{M}', s_0)$  //  $\mathcal{M}' = (S', p')$ : a finite Markov chain
return  $(preach(A'), preach(A') + preach(AlmostLoose_0) + \frac{\theta}{2} \cdot preach(AlmostLoose_1))$ 
    
```

---

We also provide an algorithm for models with a decidable reachability problem that returns  $[0, 0]$  when  $A$  is unreachable and  $[\ell, u]$  with  $\ell > 0$  otherwise. This algorithm and the proof of its correctness are both presented in [14].

### 3 (Un)Decidability Results

We now study probabilistic versions of well-known models like Pushdown Automaton (PDA) and Petri nets (PN), for which we analyse the decidability of the divergence property.

#### 3.1 Probabilistic Pushdown Automata

Let  $\Gamma$  be a finite alphabet.  $\Gamma^{\leq k}$  is the set of words over  $\Gamma$  with length at most  $k$ . Let  $w \in \Gamma^*$ , then  $|w|$  denotes its length.  $\varepsilon$  denotes the empty word.

**Definition 4 (pPDA).** A (dynamic-)probabilistic pushdown automaton (*pPDA*) is a tuple  $\mathcal{A} = (Q, \Gamma, \Delta, W)$  where:

- $Q$  is a finite set of control states;
- $\Gamma$  is a finite stack alphabet with  $Q \cap \Gamma = \emptyset$ ;
- $\Delta$  is a subset of  $Q \times \Gamma^{\leq 1} \times Q \times \Gamma^{\leq 2}$  such that for all  $(q, \varepsilon, q', w) \in \Delta$ ,  $|w| \leq 1$ ;
- $W$  is a computable function from  $\Delta \times \Sigma^*$  to  $\mathbb{Q}_{>0}$ .

In the version of pPDA presented in [12], the weight function  $W$  goes from  $\Delta$  to  $\mathbb{Q}_{>0}$ . In order to emphasize this restriction here and later we say that, in this case, the weight function is *static* and the corresponding models will be called static pPDA. In what follows, pPDA denotes the dynamic version.



An item  $(q, a, q', w)$  of  $\Delta$  is also denoted  $q \xrightarrow{?a!w} q'$  and  $?a!\varepsilon$  is also simply denoted by  $?a$ . A *configuration* of  $\mathcal{A}$  is a pair  $(q, w) \in Q \times \Gamma^*$ . We use the letters  $a, b, c, x, y$  for elements in  $\Gamma$  and  $w$  for a word in  $\Gamma^*$ .

**Definition 5.** Let  $\mathcal{A}$  be a pPDA. Then the Markov chain  $\mathcal{M}_{\mathcal{A}} = (S_{\mathcal{A}}, p_{\mathcal{A}})$  is defined by:

- $S_{\mathcal{A}} = Q \times \Gamma^*$  is the set of configurations;
- For all  $(q, \varepsilon) \in S_{\mathcal{A}}$  s.t.  $\{t = q \xrightarrow{?\varepsilon!w_t} q'\}_{t \in \Delta} = \emptyset$ ,  $p_{\mathcal{A}}((q, \varepsilon), (q, \varepsilon)) = 1$ ;
- For all  $(q, \varepsilon) \in S_{\mathcal{A}}$  s.t.  $\{t = q \xrightarrow{?\varepsilon!w_t} q'\}_{t \in \Delta} \neq \emptyset$ , let  $W(q, \varepsilon) = \sum_{t=q \xrightarrow{?\varepsilon!w_t} q'} W(t, \varepsilon)$ .  
Then: for all  $t = q \xrightarrow{?\varepsilon!w_t} q' \in \Delta$ ,  $p_{\mathcal{A}}((q, \varepsilon), (q', w_t)) = \frac{W(t, \varepsilon)}{W(q, \varepsilon)}$
- For all  $(q, wa) \in S_{\mathcal{A}}$  s.t.  $\{t = q \xrightarrow{?a!w_t} q'\}_{t \in \Delta} = \emptyset$ ,  $p_{\mathcal{A}}((q, wa), (q, wa)) = 1$ ;
- For all  $(q, wa) \in S_{\mathcal{A}}$  s.t.  $\{t = q \xrightarrow{?a!w_t} q'\}_{t \in \Delta} \neq \emptyset$ ,  
let  $W(q, wa) = \sum_{t=q \xrightarrow{?a!w_t} q'} W(t, wa)$ . Then:  
for all  $t = q \xrightarrow{?a!w_t} q' \in \Delta$ ,  $p_{\mathcal{A}}((q, wa), (q', ww_t)) = \frac{W(t, wa)}{W(q, wa)}$

We now show that even for pPDA with a single state and with a stack alphabet reduced to a singleton, divergence is undecidable.

**Theorem 1.** The divergence problem for pPDA is undecidable even with a single state and stack alphabet  $\{a\}$ .

Due to this negative result on such a basic model, it is clear that one must restrict the possible weight functions. A pPDA  $\mathcal{A}$  is said *polynomial* if for all  $t \in \Delta$ ,  $W(t, w)$  is a positive integer polynomial (i.e. whose coefficients are non negative and the constant one is positive) whose single variable is  $|w|$ .

**Theorem 2.** The divergence problem w.r.t.  $s_0$  and finite  $A$  for polynomial pPDA with a single state and stack alphabet  $\{a\}$  is decidable (in linear time).

### 3.2 Probabilistic Petri Nets

A probabilistic Petri net (resp. a probabilistic VASS) is a Petri net (resp. a VASS) with a computable weight function  $W$ . In previous works [2, 4], the weight function  $W$  is a *static* one: i.e., a function from  $T$ , the finite set of transitions of the Petri net, to  $\mathbb{N}_{>0}$ . As above, we call these models *static* probabilistic Petri nets. We introduce here a more powerful function where the weight of a transition depends on the current marking.

**Definition 6.** A (dynamic-)probabilistic Petri net (pPN)  $\mathcal{N} = (P, T, \mathbf{Pre}, \mathbf{Post}, W, \mathbf{m}_0)$  is defined by:

- $P$ , a finite set of places;
- $T$ , a finite set of transitions;
- $\mathbf{Pre}, \mathbf{Post} \in \mathbb{N}^{P \times T}$ , resp. the pre and post condition matrices;
- $W$ , a computable function from  $T \times \mathbb{N}^P$  to  $\mathbb{Q}_{>0}$  the weight function;

–  $\mathbf{m}_0 \in \mathbb{N}^P$ , the initial marking.

When for all  $t \in T$ ,  $W(t, -)$  is a positive polynomial whose variables are the place markings, we say that  $\mathcal{N}$  is a *polynomial* pPN.

A marking  $\mathbf{m}$  is an item of  $\mathbb{N}^P$ . Let  $t$  be a transition. Then  $t$  is *enabled* in  $\mathbf{m}$  if for all  $p \in P$ ,  $\mathbf{m}(p) \geq \mathbf{Pre}(p, t)$ . When enabled, the *firing* of  $t$  leads to marking  $\mathbf{m}'$  defined for all  $p \in P$  by  $\mathbf{m}'(p) = \mathbf{m}(p) + \mathbf{Post}(p, t) - \mathbf{Pre}(p, t)$  which is denoted by  $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ . Let  $\sigma = t_1 \dots t_n$  be a sequence of transitions. We define the enabling and the firing of  $\sigma$  by induction. The empty sequence is always enabled in  $\mathbf{m}$  and its firing leads to  $\mathbf{m}$ . When  $n > 0$ ,  $\sigma$  is enabled if  $\mathbf{m} \xrightarrow{t_1} \mathbf{m}_1$  and  $t_2 \dots t_n$  is enabled in  $\mathbf{m}_1$ . The firing of  $\sigma$  leads to the marking reached by  $t_2 \dots t_n$  from  $\mathbf{m}_1$ . A marking  $\mathbf{m}$  is *reachable* from  $\mathbf{m}_0$  if there is a firing sequence  $\sigma$  that reaches  $\mathbf{m}$  from  $\mathbf{m}_0$ .

**Definition 7.** Let  $\mathcal{N}$  be a pPN. Then the Markov chain  $\mathcal{M}_{\mathcal{N}} = (S_{\mathcal{N}}, p_{\mathcal{N}})$  associated with  $\mathcal{N}$  is defined by:

- $S_{\mathcal{N}}$  is the set of reachable markings from  $\mathbf{m}_0$ ;
- Let  $\mathbf{m} \in S_{\mathcal{N}}$  and  $T_{\mathbf{m}}$  be the set of transitions enabled in  $\mathbf{m}$ . If  $T_{\mathbf{m}} = \emptyset$  then  $p_{\mathcal{N}}(\mathbf{m}, \mathbf{m}) = 1$ . Otherwise let  $W(\mathbf{m}) = \sum_{\mathbf{m} \xrightarrow{t} \mathbf{m}_t} W(t, \mathbf{m})$ . Then for all  $\mathbf{m} \xrightarrow{t} \mathbf{m}_t$ ,  $p_{\mathcal{N}}(\mathbf{m}, \mathbf{m}_t) = \frac{W(t, \mathbf{m})}{W(\mathbf{m})}$ .

Contrary to the previous result, restricting the weight functions to be polynomials does not yield decidability for pPNs.

**Theorem 3.** The divergence problem of polynomial pPNs w.r.t. an upward closed set is undecidable.

## 4 Illustration of Divergence

Due to the undecidability results, we propose syntactical restrictions for standard models like pushdown automata and channel systems that ensure divergence. Observing that function  $f_1$  of Definition 3 is somewhat related to transience of Markov chains, we first establish a sufficient condition of transience from which we derive a sufficient condition of divergence for infinite Markov chains used for our two illustrations.

**Theorem 4.** Let  $\mathcal{M}$  be a Markov chain and  $f$  be a function from  $S$  to  $\mathbb{R}$  with  $B = \{s \mid f(s) \leq 0\}$  fulfilling  $\emptyset \subsetneq B \subsetneq S$ ,  $\varepsilon, K \in \mathbb{R}_{>0}$  and  $d \in \mathbb{N}^*$  such that:

$$\text{for all } s \in S \setminus B \quad \sum_{s' \in S} p^{(d)}(s, s') f(s') \geq f(s) + \varepsilon \text{ and } \sum_{|f(s') - f(s)| \leq K} p(s, s') = 1 \quad (1)$$

Then for all  $s \in S$  such that  $f(s) > dK$ ,

$$\Pr_{\mathcal{M}, s}(\mathbf{FB}) \leq c_1 e^{-c_2(f(s) - dK)}$$

where  $c_1 = \sum_{n \geq 1} e^{-\frac{\varepsilon^2 n}{2(\varepsilon + K)^2}}$  and  $c_2 = \frac{\varepsilon}{(\varepsilon + K)^2}$ ,

which implies transience of  $\mathcal{M}$  when it is irreducible.

**Proposition 3.** *Let  $\mathcal{M}$  be a Markov chain and  $f$  be a computable function from  $S$  to  $\mathbb{R}$  with  $B = \{s \mid f(s) \leq 0\}$  fulfilling  $\emptyset \subsetneq B \subsetneq S$ , and for some  $\varepsilon, K \in \mathbb{R}_{>0}$  and  $d \in \mathbb{N}^*$ , Equation (1). Assume in addition that for all  $n \in \mathbb{N}$ ,  $\{s \mid f(s) \leq n\}$  is finite. Then  $\mathcal{M}$  is divergent w.r.t. any  $s_0$  and any finite  $A$ .*

#### 4.1 Probabilistic Channel Systems

Now we introduce a probabilistic variant of channel systems particularly appropriate for the modelling of open queuing networks. Here a special input channel  $c_{in}$  (that works as a counter) only receives the arrivals of anonymous clients all denoted by  $\$$  (item 1 of the next definition). Then the service of a client corresponds to a message circulating between the other channels with possibly change of message identity until the message disappears (items 2 and 3).

**Definition 8.** *A probabilistic open channel system (pOCS)  $\mathcal{S} = (Q, Ch, \Sigma, \Delta, W)$  is defined by:*

- a finite set  $Q$  of states;
- a finite set  $Ch$  of channels, including  $c_{in}$ ;
- a finite alphabet  $\Sigma$  including  $\$$ ;
- a transition relation  $\Delta \subseteq Q \times Ch \times \Sigma_\varepsilon \times Ch \times \Sigma_\varepsilon \times Q$  that fulfills:
  1. For all  $q \in Q$ ,  $(q, c_{in}, \varepsilon, c_{in}, \$, q) \in \Delta$ ;
  2. For all  $(q, c, a, c', a', q') \in \Delta$ ,  $a = \varepsilon \Rightarrow a' = \$ \wedge c = c' = c_{in}$ ;
  3. For all  $(q, c, a, c', a', q') \in \Delta$ ,  $c \neq c_{in} \Rightarrow c' \neq c_{in}$ ;
- $W$  is a function from  $\Delta \times (\Sigma^*)^{Ch}$  to  $\mathbb{Q}_{>0}$ .

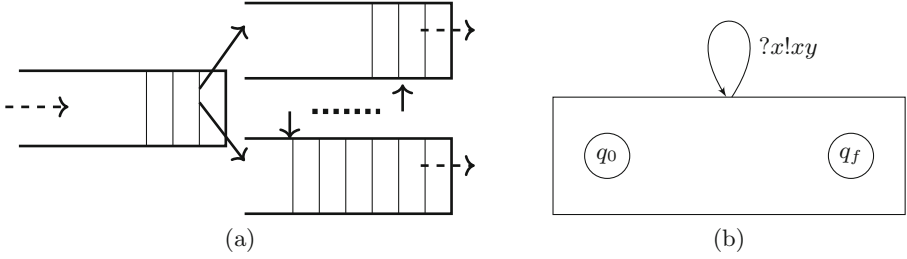
**Illustration.** We consider three types of transitions between configurations: sending messages to the input channel (i.e.,  $(q, c_{in}, \varepsilon, c_{in}, \$, q)$ ) representing client arrivals; transferring messages between different channels (i.e.,  $(q, c, a, c', a', q')$  with  $\varepsilon \notin \{a, a'\}$  and  $c' \neq c_{in}$ ) describing client services; and terminating message processing (i.e.,  $(q, c, a, c', \varepsilon, q')$  with  $a \neq \varepsilon$ ) meaning client departures. All messages entering  $c_{in}$  are anonymous (i.e., denoted by  $\$$ ). The left part of Fig. 3 is a schematic view of such systems. The left channel is  $c_{in}$ . All dashed lines represent message arrivals (to  $c_{in}$ ) or departures. The solid lines model message transferrings.

The next definitions formalize the semantics of pOCS.

**Definition 9.** *Let  $\mathcal{S}$  be a pOCS,  $(q, \nu) \in Q \times (\Sigma^*)^{Ch}$  be a configuration and  $t = (q, c, a, c', a', q') \in \Delta$ . Then  $t$  is enabled in  $(q, \nu)$  if  $\nu(c) = aw$  for some  $w$ . The firing of  $t$  in  $(q, \nu)$  leads to  $(q', \nu')$  defined by:*

- if  $c = c'$  then  $\nu'(c) = wa'$  and for all  $c'' \neq c$ ,  $\nu'(c'') = \nu(c'')$ ;
- if  $c \neq c'$  then  $\nu'(c) = w$ ,  $\nu'(c') = \nu(c')a'$   
and for all  $c'' \notin \{c, c'\}$ ,  $\nu'(c'') = \nu(c'')$ .

As usual one denotes the firing by  $(q, \nu) \xrightarrow{t} (q', \nu')$ . Observe that from any configuration at least one transition (a client arrival) is enabled.



**Fig. 3.** A schematic view of pOCS (left) and a pPDA (right)

**Definition 10.** Let  $\mathcal{S}$  be a pOCS. Then the Markov chain  $\mathcal{M}_{\mathcal{S}} = (S_{\mathcal{S}}, p_{\mathcal{S}})$  is defined by:

- $S_{\mathcal{S}} = Q \times (\Sigma^*)^{C_h}$  is the set of configurations;
- For all  $(q, \nu) \in S_{\mathcal{S}}$  let  $W(q, \nu) = \sum_{(q', \nu') \xrightarrow{t} (q, \nu)} W(t, \nu)$ . Then:  
for all  $(q, \nu) \xrightarrow{t} (q', \nu')$ ,  $p_{\mathcal{S}}((q, \nu), (q', \nu')) = \frac{W(t, \nu)}{W(q, \nu)}$ .

The restrictions on pOCS w.r.t. standard CS do not change the status of the reachability problem.

**Proposition 4.** The reachability problem of pOCS is undecidable.

As discussed in the introduction, when the number of clients exceeds some threshold, the performances of the system drastically decrease and thus the ratio of arrivals w.r.t. the achievement of a task increase. We formalize it by introducing *uncontrolled* pOCS where the weights of transitions are constant except the ones of client arrivals which are specified by positive non constant polynomials. Let  $\nu \in (\Sigma^*)^{C_h}$ . Then  $|\nu|$  denotes  $\sum_{c \in C_h} |\nu(c)|$ .

**Definition 11.** Let  $\mathcal{S}$  be a pOCS. Then  $\mathcal{S}$  is uncontrolled if:

- For all  $t = (q, c, a, c', a', q') \in \Delta$  with  $a \neq \varepsilon$ ,  $W(t, \nu)$  only depends on  $t$  and will be denoted  $W(t)$ ;
- For all  $t = (q, c_{in}, \varepsilon, c_{in}, \$, q)$ ,  $W(t, \nu)$  is a positive non constant polynomial, whose single variable is  $|\nu|$ , and will be denoted  $W_{in}(q, |\nu|)$ .

The next proposition establishes that an uncontrolled pOCS generates a divergent Markov chain. This model illustrates the interest of divergence: while reachability of a pOCS is undecidable, we can apply Algorithm 1.

**Proposition 5.** Let  $\mathcal{S}$  be a uncontrolled pOCS. Then  $\mathcal{M}_{\mathcal{S}}$  is divergent.

## 4.2 Probabilistic Pushdown Automata

**Increasing pPDA.** We introduce the subset of *increasing pairs*, denoted as  $Inc(\mathcal{A})$ , which is a subset of  $Q \times \Gamma$  that contains pairs  $(q, a)$  such that from

state  $(q, wa)$ , the height of the stack can increase without decreasing before. When some conditions on  $\text{Inc}(\mathcal{A})$  are satisfied, we obtain a syntactic *sufficient* condition for  $\mathcal{M}_{\mathcal{A}}$  to be divergent. This set  $\text{Inc}(\mathcal{A})$  can be easily computed in polynomial time by a saturation algorithm.

**Definition 12.** Let  $(q, a), (q', a') \in Q \times \Gamma$ . Then  $(q', a')$  is reachable from  $(q, a)$  if either  $(q, a) = (q', a')$  or there is a sequence of transitions of  $\Delta$ ,  $(t_i)_{0 \leq i < d}$  such that:  $t_i = q_i \xrightarrow{?a_i!a_{i+1}} q_{i+1}$ ,  $(q_0, a_0) = (q, a)$ ,  $(q_d, a_d) = (q', a')$  and for all  $i$ ,  $a_i \neq \varepsilon$ . The set of increasing pairs  $\text{Inc}(\mathcal{A}) \subseteq Q \times \Gamma$  is the set of pairs  $(q, a)$  that can reach a pair  $(q', a')$  with some  $q' \xrightarrow{?a'!bc} q'' \in \Delta$ .

**Definition 13.** A pPDA  $\mathcal{A}$  is increasing if:

- $\text{Inc}(\mathcal{A}) = Q \times \Gamma$ ;
- for all  $t = q \xrightarrow{?a!w} q' \in \Delta$  such that  $|w| \leq 1$ ,  $W(t, -)$  is an integer constant denoted  $W_t$ ;
- for all  $t = q \xrightarrow{?a!bc} q' \in \Delta$ ,  $W(t, -)$  is a non constant integer polynomial where its single variable is the height of the stack denoted  $W_t$ ;
- for all  $q \xrightarrow{?a} q' \in \Delta$ , there exists  $q \xrightarrow{?a!bc} q'' \in \Delta$ .

**Illustration.** The right part of Fig. 3 is an abstract view of a pPDA modelling of a server simultaneously handling multiple requests. The requests may occur at any time and are stored in the stack. The loop labelled by  $?x!xy$  is a symbolic representation of several loops: one per triple  $(q, x, y)$  with  $q \in Q$ ,  $x \in \Gamma$  and  $y \in \Gamma$ . Due to the symbolic loop, the set of increasing pairs of the  $pPDA_{\text{server}}$  is equal to  $Q \times \Gamma$  and there is always a transition increasing the height of the stack outgoing from any  $(q, a)$ . Assume now that for any other transition, its weight does not depend on the size of the stack and that a transition  $t = q \xrightarrow{?a!ab} q$  has weight  $W_t(n) = c_t \times n$ . Then  $\mathcal{A}$  is increasing. The dependance on  $n$  means that due to congestion, the time to execute tasks of the server increases with the number of requests in the system and thus increase the probability of a new request that occurs at a constant rate. One is interested in computing the probability to reach  $(q_f, \varepsilon)$  from  $(q_0, \varepsilon)$  representing the probability that the server reaches an idle state having served all the incoming requests.

We establish that an increasing pPDA generates a divergent Markov chain.

**Proposition 6.** Let  $\mathcal{A}$  be an increasing pPDA. Then the Markov chain  $\mathcal{M}_{\mathcal{A}}$  is divergent w.r.t. any  $s_0$  and finite  $A$ .

## 5 Conclusion and Perspectives

We have introduced the divergence property of Markov chains and designed two generic CRP-algorithms depending on the status of the reachability problem. Then we have studied the decidability of divergence for pPDA and for pPN for

different kinds of weights and target sets. Finally, we have provided two useful classes of divergent models within pCS and pPDA.

In the future, we plan to study the model checking of polynomial pPDA (as a possible extension of [12]) and some heuristics to find functions  $f_0$  and  $f_1$ .

## References

1. Abdulla, P.A., Bertrand, N., Rabinovich, A.M., Schnoebelen, P.: Verification of probabilistic systems with faulty communication. *Inf. Comput.* **202**(2), 141–165 (2005)
2. Abdulla, P.A., Henda, N.B., Mayr, R.: Decisive Markov chains. *Log. Methods Comput. Sci.* **3**(4) (2007)
3. Baier, C., Katoen, J.: *Principles of Model Checking*. MIT Press, Heidelberg (2008)
4. Brázdil, T., Chatterjee, K., Kucera, A., Novotný, P., Velan, D., Zuleger, F.: Efficient algorithms for asymptotic bounds on termination time in VASS. In: Dawar, A., Grädel, E. (eds.) *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018*, pp. 185–194. ACM (2018)
5. Brázdil, T., Esparza, J., Kiefer, S., Kucera, A.: Analyzing probabilistic pushdown automata. *Formal Methods Syst. Des.* **43**(2), 124–163 (2013)
6. Brázdil, T., Esparza, J., Kucera, A.: Analysis and prediction of the long-run behavior of probabilistic sequential programs with recursion (extended abstract). In: *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 521–530. IEEE Computer Society (2005)
7. Brázdil, T., Kiefer, S., Kučera, A.: Efficient analysis of probabilistic programs with an unbounded counter. In: Gopalakrishnan, G., Qadeer, S. (eds.) *CAV 2011*. LNCS, vol. 6806, pp. 208–224. Springer, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-22110-1\\_18](https://doi.org/10.1007/978-3-642-22110-1_18)
8. Brázdil, T., Kiefer, S., Kucera, A.: Efficient analysis of probabilistic programs with an unbounded counter. *J. ACM* **61**(6), 41:1–41:35 (2014)
9. Brázdil, T., Kiefer, S., Kucera, A., Novotný, P., Katoen, J.: Zero-reachability in probabilistic multi-counter automata. In: *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS 2014*, pp. 22:1–22:10. ACM (2014)
10. Esparza, J., Kucera, A., Mayr, R.: Model checking probabilistic pushdown automata. In: *Proceedings of 19th IEEE Symposium on Logic in Computer Science (LICS)*, pp. 12–21. IEEE Computer Society (2004)
11. Esparza, J., Kucera, A., Mayr, R.: Quantitative analysis of probabilistic pushdown automata: expectations and variances. In: *Proceedings of the 20th IEEE Symposium on Logic in Computer Science (LICS)*, pp. 117–126. IEEE Computer Society (2005)
12. Esparza, J., Kucera, A., Mayr, R.: Model checking probabilistic pushdown automata. *Logical Methods Comput. Sci.* **2**(1) (2006)
13. Finkel, A., Haddad, S., Ye, L.: About decisiveness of dynamic probabilistic models. *CoRR* abs/2305.19564 (2023). <https://doi.org/10.48550/arXiv.2305.19564>, to appear in CONCUR'23
14. Finkel, A., Haddad, S., Ye, L.: Introducing divergence for infinite probabilistic models. *CoRR* abs/2308.08842 (2023). <https://doi.org/10.48550/arXiv.2308.08842>

15. Iyer, P., Narasimha, M.: Probabilistic lossy channel systems. In: Bidoit, M., Dauchet, M. (eds.) CAAP 1997. LNCS, vol. 1214, pp. 667–681. Springer, Heidelberg (1997). <https://doi.org/10.1007/BFb0030633>
16. Kemeny, J., Snell, J., Knapp, A.: Denumerable Markov Chains, 2nd edn. Springer, Heidelberg (1976). <https://doi.org/10.1007/978-1-4684-9455-6>
17. Kucera, A., Esparza, J., Mayr, R.: Model checking probabilistic pushdown automata. *Log. Methods Comput. Sci.* **2**(1) (2006). [https://doi.org/10.2168/LMCS-2\(1:2\)2006](https://doi.org/10.2168/LMCS-2(1:2)2006)
18. Santos, E.S.: Probabilistic grammars and automata. *Inf. Control.* **21**(1), 27–47 (1972). [https://doi.org/10.1016/S0019-9958\(72\)90026-5](https://doi.org/10.1016/S0019-9958(72)90026-5)
19. Vardi, M.Y.: Automatic verification of probabilistic concurrent finite-state programs. In: Proceedings of the 26th Annual Symposium on Foundations of Computer Science, pp. 327–338. IEEE Computer Society (1985)