Consistency for Parametric Interval Markov Chains

Benoît Delahaye

Université de Nantes / LINA
Nantes, France
benoit.delahaye@univ-nantes.fr

Abstract

Interval Markov Chains (IMCs) are the base of a classic probabilistic specification theory by Larsen and Jonsson in 1991. They are also a popular abstraction for probabilistic systems. In this paper we introduce and study an extension of Interval Markov Chains with parametric intervals. In particular, we investigate the consistency problem for such models and propose an efficient solution for the subclass of parametric IMCs with local parameters only. We also show that this problem is still decidable for parametric IMCs with global parameters, although more complex in this case.

1998 ACM Subject Classification D.2.4 Software/Program Verification, F.1.1 Models of Computation, G.3 Probability and Statistics

Keywords and phrases specification, parameters, Markov chains, consistency

Digital Object Identifier 10.4230/OASIcs.SynCoP.2015.17

1 Introduction

Interval Markov Chains (IMCs for short) extend Markov Chains, by allowing to specify intervals of possible probabilities on state transitions instead of precise probabilities. IMCs have been introduced by Larsen and Jonsson [16] as a *specification* formalism—a basis for a stepwise-refinement-like modeling method, where initial designs are very abstract and underspecified, and then they are made continuously more precise, until they are concrete. Unlike richer specification models such as Constraint Markov Chains [6] or Abstract Probabilistic Automata [9], IMCs are difficult to use for compositional specification due to lack of basic modeling operators. Nevertheless, IMCs have been intensively used in order to model real-life systems in domains such as systems biology, security or communication protocols [2, 12, 5, 19, 11].

The extension of Markov Chains into Interval Markov chains was motivated by the fact that, when modelling real-life systems, the actual exact value of transition probabilities may not be known precisely. Indeed, in most cases, these values are measured from observations or experiments which are subject to imprecision. In this case, using intervals of probabilities that take into account the precision of the measures makes more sense than using an arbitrary but precise value. We now take this reasoning a step further.

Complex systems are most often built by assembling multiple components. Assume that one of these components may fail with a given probability that depends on the quality of the materials involved in its fabrication. In practice, a prototype of the component is built and the failure probability of this component is measured by experiment with a certain

This work has been partially supported by project PACS ANR-14-CE28-0002-04.

imprecision. This failure probability and the subsequent imprecision are then taken into account in the model of the system by using an interval of probability. When one analyzes this model, every result will depend on the failure rate of this component, which itself depends on the choice of the quality of materials. If the conclusions of the analysis are that the failure probability is too high for the whole system to be viable, then a new prototype of the failing component is built using different materials and the modeling and analysis phase starts over. This process is repeated until a satisfactory failing component is identified.

Instead of using this “trial and error” methodology, we propose a new extension of Interval Markov Chains that allows using parameters in the definition of intervals of probability. In our example, the failure probability of the failing component is clearly a parameter of the system. The developer is interested in whether there exists a maximal value of this probability that will ensure that the whole system is satisfactory. When this value is identified, one can choose the materials of the failing component accordingly in order to produce a prototype with a lower maximal failing probability.

Therefore, we introduce in this paper the new formalism called parametric Interval Markov Chains (pIMCs), which extends IMCs by allowing the use of parameters as lower or upper endpoints of probability intervals. We also show that the problem of deciding whether a given pIMC is consistent (i.e. admits a valid implementation) is decidable and propose algorithms in order to solve this problem. In particular, we identify a subclass of pIMCs – *local pIMCs* – for which an efficient algorithm is proposed. In the rest of the paper, we limit ourselves to closed intervals. Nevertheless, all the results we propose can be extended with minor modifications to open/semi-open intervals whose lower/upper endpoints contain linear combinations of parameters and constants.

Related work. To the best of our knowledge, there is no existing work on parametric probabilistic specification theories as such, where parameters range over probability values. Still, classes of systems where parameters give some latitude on probability distributions, such as parametric Markov models [17] have been studied in the literature [18, 13]. The activity in this domain has yielded decidability results [15], parametric probabilistic model-checking algorithms [8] and even tools [14]. Continuous-time parametric and probabilistic models have also been considered in some very restricted settings [7]. Networks of probabilistic processes where the number of processes is a parameter have also been studied in [3, 4], and probabilistic timed automata with parameters in clock constraints and invariants have been studied in [1].

The paper proceeds as follows. In Section 2, we begin by introducing concepts and notations that will be used throughout the paper. Section 3 introduces the new formalism of parametric Interval Markov Chains, studies their relations to (Interval) Markov Chains and discusses what we call the *range* of parameters. In Section 4, we present original solutions to the consistency problem for IMCs and pIMCs. Finally, Section 5 concludes the paper and discusses future work.

2 Background

Throughout the paper, we use the notion of parameters. A parameter $p \in P$ is a variable ranging through the interval $[0, 1]$. A valuation for P is a function $\psi : P \to [0, 1]$ that associates values with each parameter in P. We write $\mathtt{Int}_{[0,1]}(P)$ for the set of all closed intervals of the form $[x, y]$ with $x, y \in [0, 1] \cup P$. When $P = \emptyset$, we write $\mathtt{Int}_{[0,1]} = \mathtt{Int}_{[0,1]}(\emptyset)$ to denote closed intervals with real-valued endpoints. Given an interval I of the form

$I = [a, b]$, $\texttt{Low}(I)$ and $\texttt{Up}(I)$ respectively denote the lower and upper endpoints of I, i.e. a and b. Given an interval $I = [a, b] \in \texttt{Int}_{[0,1]}$, we say that I is well-formed whenever $a \leq b$. In the following, we abuse notations and write $\emptyset$ for the empty interval, meaning any not well-formed interval. Given a finite set S, we write $\texttt{Dist}(S)$ for the set of distributions over S, i.e. the set of functions $\rho : S \to [0, 1]$ such that $\sum_{s \in S} \rho(s) = 1$. In the rest of the paper, we assume that all the states in our structures are equipped with labels taken from a fixed set of atomic propositions A. A state-labelling function over S is thus a function $V : S \to 2^A$ that assigns to each state a set of labels from A.

We recall the notion of Markov Chains (MCs), that will act as models for (parametric) IMCs. An example of a Markov Chain is given in Figure 1a.

▶ **Definition 1** (Markov Chain)**.** A Markov Chain is a tuple $\mathcal{M} = (S, s_0, M, A, V)$, where S is a finite set of states containing the initial state s_0, A is a set of atomic propositions, $V : S \to 2^A$ is a labeling function, and $M : S \times S \to [0, 1]$ is a probabilistic transition function such that $\forall s \in S, \sum_{t \in S} M(s, t) = 1$.

We now recall the notion of Interval Markov Chain (IMC), adapted from [10]. IMCs are a specification formalism that allows one to represent an infinite set of MCs. Roughly, IMCs extend MCs by replacing exact probability values on transitions with intervals of allowed probability values. An example of an IMC is given in Figure 1b.

▶ **Definition 2** (Interval Markov Chain [10])**.** An Interval Markov Chain (IMC) is a tuple $\mathcal{I} = (S, s_0, \varphi, A, V)$, where S, s_0, A and V are as for MCs, and $\varphi : S \times S \to \texttt{Int}_{[0,1]}$ is a transition constraint that associates with each potential transition an interval of probabilities.

The following definition recalls the notion of satisfaction introduced in [10]. Satisfaction (also called implementation in some cases) allows to characterize the set of MCs represented by a given IMC specification. Crucially, satisfaction abstracts from the syntactic structure of transitions in IMCs: a single transition in the implementation MC can contribute to satisfaction of more than one transition in the specification IMC, by distributing its probability mass against several transitions. Similarly many MC transitions can contribute to satisfaction of just one specification transition.

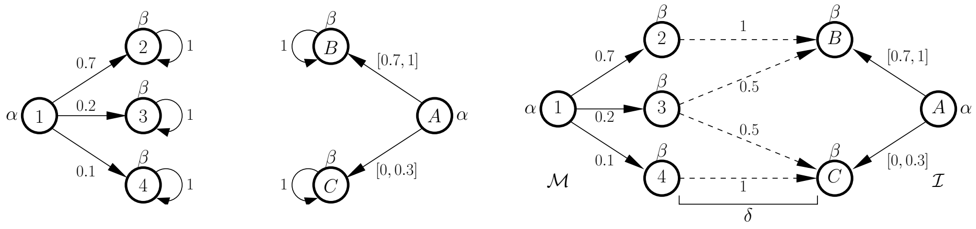
▶ **Definition 3** (Satisfaction Relation [10])**.** Let $\mathcal{I} = (S, s_0, \varphi, A, V^I)$ be an IMC and $\mathcal{M} = (T, t_0, M, A, V^M)$ be a MC. A relation $\mathcal{R} \subseteq T \times S$ is a *satisfaction relation* if whenever $t\mathcal{R}s$,

1. the valuations of s and t agree: $V^M(t) = V^I(s)$,
2. there exists a function $\delta : T \to (S \to [0, 1])$ such that
   - **a.** for all $t' \in T$ such that $M(t, t') > 0$, $\delta(t')$ is a distribution on S,
   - **b.** for all $s' \in S$, we have $(\sum_{t' \in T} M(t, t') \cdot \delta(t')(s')) \in \varphi(s, s')$, and
   - **c.** for all $t' \in T$ and $s' \in S$, if $\delta(t')(s') > 0$, then $(t', s') \in \mathcal{R}$.

   We say that $\mathcal{M}$ satisfies $\mathcal{I}$ (written $\mathcal{M} \models \mathcal{I}$) iff there exists a satisfaction relation containing (t_0, s_0).

The set of MCs satisfying a given IMC $\mathcal{I}$ is written $[\![\mathcal{I}]\!]$. Formally, $[\![\mathcal{I}]\!] = \{\mathcal{M} \mid \mathcal{M} \models \mathcal{I}\}$. In the rest of the paper, we write $\perp$ for the *empty* IMC, i.e.$\perp = (\emptyset, \emptyset, \emptyset, A, \emptyset)$. By construction, we have $[\![\perp]\!] = \emptyset$.

The notion of satisfaction between the MC $\mathcal{M}$ from Figure 1a and the IMC $\mathcal{I}$ from Figure 1b is illustrated in Figure 1c.



(a) A Markov Chain $\mathcal{M}$. **(b)** An IMC $\mathcal{I}$.

(c) An example of satisfaction relation.

Figure 1 Markov Chain, Interval Markov Chain and satisfaction relation [10].

3 Parametric Interval Markov Chains

In this section, we propose a new formalism, called parametric Interval Markov Chains (pIMC) that extends IMCs by allowing parameters as the lower/upper endpoints of the transition intervals. We start by giving the main definitions of pIMCs and their relations with IMCs and MCs, and then distinguish two subclasses of interest of pIMCs: *local* and *global* pIMCs.

3.1 pIMCs and their relations to IMCs/MCs

We now propose an extension of IMCs that allows using parameters in the definition of intervals.

▶ **Definition 4** (Parametric Interval Markov Chain). A parametric Interval Markov Chain (pIMC) is a tuple $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$, where S, s_0, A and V or as for IMCs, P is a set of variables (parameters) ranging over $[0, 1]$ and $\varphi_P : S \times S \to \texttt{Int}_{[0,1]}(P)$ associates to each potential transition a (parametric) interval.

In the following, we abuse notations and also write $\perp$ for the *empty* pIMC, i.e. $\perp = (\emptyset, \emptyset, \emptyset, A, \emptyset, \emptyset)$.

Roughly, an instance of a pIMC $\mathcal{I}^P$ is a pair $(\mathcal{I}, \psi)$, where $\mathcal{I}$ is an IMC that respects the structure and labels of $\mathcal{I}^P$ and such that its transition constraint is the instantiation of φ_P according to the valuation for the parameters ψ.

▶ **Definition 5** (Instance of a pIMC). An instance of pIMC $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$ is a pair $(\mathcal{I}, \psi)$ (written $(\mathcal{I}, \psi) \vdash \mathcal{I}^P$), where $\mathcal{I} = (S, s_0, \varphi, A, V)$ is an IMC respecting the structure and labels of $\mathcal{I}^P$, $\psi : P \to [0, 1]$ is a valuation for the parameters, and $\varphi \equiv \varphi_P[p \leftarrow \psi(p)]$.

We sometimes write $\mathcal{I} \vdash_\psi \mathcal{I}^P$ instead of $(\mathcal{I}, \psi) \vdash \mathcal{I}^P$ and say that $\mathcal{I}$ is an instance of $\mathcal{I}^P$ through ψ. We say that $\mathcal{I}$ is an instance of $\mathcal{I}^P$, written $\mathcal{I} \vdash \mathcal{I}^P$, whenever there exists a valuation ψ such that $\mathcal{I} \vdash_\psi \mathcal{I}^P$.

A MC $\mathcal{M} = (T, t_0, M, A, V^M)$ implements pIMC $\mathcal{I}^P$, written $\mathcal{M} \models \mathcal{I}^P$, iff there exists an instance $\mathcal{I}$ of $\mathcal{I}^P$ such that $\mathcal{M} \models \mathcal{I}$. We write $[\![\mathcal{I}^P]\!]$ for the set of MCs implementing $\mathcal{I}^P$.

▶ **Example 6.** Consider pIMC $\mathcal{I}^P$ given in the left of Figure 2. $\mathcal{I}^P$ represents a family of dispensable probabilistic beverage machines (dpbm) that have a probability greater or equal to 0.5 of delivering tea and a probability lower or equal to 0.5 of delivering coffee. In addition, we use parameter p to model the fact that the machine can fail to deliver anything with probability at most p. The value of p depends on the quality of a potentially failing

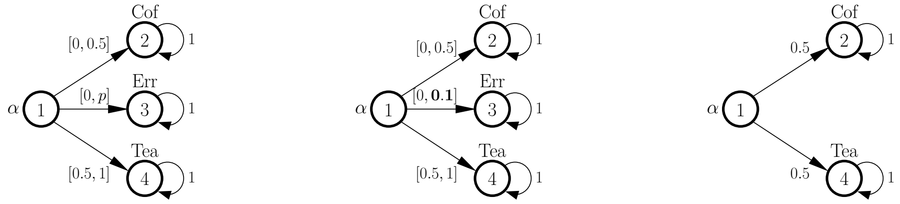


Figure 2 pIMC $\mathcal{I}^P$ (left) with one of its instances $\mathcal{I}$ (middle) and an implementation $\mathcal{M}$ (right).

component. The IMC $\mathcal{I}$ given in the middle of Figure 2 depicts the family of dpbm for which the potentially failing component has a maximal failure probability of 0.1. Finally, MC $\mathcal{M}$ given in the right of Figure 2 depicts a given dpbm of this family, where the actual probabilities of delivering tea and coffee are fixed to 0.5 and 0.5 respectively, and where the potentially failing component does not fail.

As for IMCs, one question of interest for pIMCs is to decide whether they admit at least one implementation – the so-called consistency problem. Given the definition of implementation, deciding whether a pIMC is consistent amounts to verifying whether it admits at least one instance that is itself consistent. Nevertheless, we will see in Section 4, that in the case of local pIMCs, consistency can be decided using a polynomial algorithm on the pIMC itself without having to go through any of its instances.

3.2 Local VS Global Parameters

We now propose two subclasses that distinguish different trends in the use of parameters throughout a given structure. Parameters can be used at two different levels in a given pIMC: either in a local fashion – reflecting small tuning artifacts in a model; or in a global fashion – reflecting potential design choices. In the following, we formally define these subclasses.

Local parameters. Parameters are said to be *local* if they only appear in transition probabilities outgoing from a unique state. In this sense they reflect small tuning artifacts because of their small impact on the structure of the pIMC. The pIMC $\mathcal{I}^P$ in Figure 2 illustrates this notion: in $\mathcal{I}^P$, parameter p is local as it only appears in transitions outgoing from a single state (State 1). In essence, p models the failure probability of a single component, only used once in pIMC $\mathcal{I}^P$.

We write $\texttt{Range}(p)$ for the range of a given parameter p, i.e. the set of states s such that p is either the lower or the upper endpoint of the probability interval associated with an outgoing transition of s. Formally, given pIMC $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$, $\texttt{Range}_{\mathcal{I}^P}(p) = \{s \in S \mid \exists s' \in S \text{ s.t. } p \in \texttt{Low}(\varphi_P(s, s')) \cup \texttt{Up}(\varphi_P(s, s'))\}$. When clear from the context, we write $\texttt{Range}(p)$ instead of $\texttt{Range}_{\mathcal{I}^P}(p)$. We say that a parameter $p \in P$ is *local* in $\mathcal{I}^P$ iff $|\texttt{Range}(p)| \leq 1$. A pIMC $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$ is *local* iff all its parameters are local.

Since all parameters are local in local pIMCs, it is very easy to check whether the outgoing transitions of a given state are *consistent* in the sense that it is possible to find out easily whether there exist values of the parameters such that the outgoing intervals of a given state are not empty.

Global parameters. Parameters are *global* if they are not local, i.e. if they appear in the outgoing probability intervals of at least two states.

Formally, we say that parameter $p \in P$ is *global* in $\mathcal{I}^P$ iff $|\texttt{Range}_{\mathcal{I}^P}(p)| > 1$. We say that pIMC $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$ is *global* iff at least one of its parameters is global.

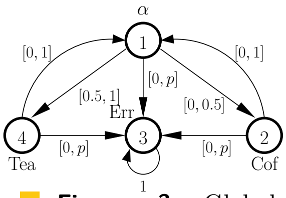


Figure 3 Global pIMC $\mathcal{I}_2^P$ with global parameter p.

▶ **Example 7.** pIMC $\mathcal{I}_2^P$ from Figure 3 depicts a family of beverage machines in which all modules use the same potentially failing component. The maximal probability of failure of this component is modeled using a parameter p. This parameter is *global* in $\mathcal{I}_2^P$ as it appears in the outgoing transitions of several states (States 1, 2, 4). In $\mathcal{I}_2^P$, the choice of a value for p has more impact on the potential behaviors of the global pIMC than in the case of pIMC $\mathcal{I}^P$ from Figure 2, where parameter p was only local.

In the case of global pIMCs, checking whether the outgoing transitions of a given state are consistent becomes more tricky, since the potential values of the parameters may be subject to constraints coming from other states.

4 Consistency

As said in Section 3, one question of interest given a pIMC $\mathcal{I}^P$ is to decide whether it admits at least one instance that itself admits at least one implementation. This is what we call the consistency problem. In this section, we start by recalling the consistency problem in the case of IMCs and solutions to this problem that have been proposed in the literature. We propose an alternative solution to the consistency problem for IMCs and then extend it to the case of local pIMCs. Finally, we show that the problem is more complex in the case of pIMCs with global parameters.

4.1 Consistency of IMCs

The consistency problem for IMCs has already been studied in the literature [10] and it has been proven that it is decidable and can be solved in polynomial time. We first recall one of the existing algorithms and then propose an alternative, more direct solution.

In [10], the consistency problem for IMCs has been considered as a special case of the *common implementation* problem, which consists in deciding, given a finite number of IMCs, whether there exists at least one implementation satisfying them all. One can solve the consistency problem for a given IMC $\mathcal{I}$ by deciding whether $\mathcal{I}$ admits a common implementation with itself. The proposed solution to the consistency problem is based on the notion of consistency relation, also introduced in [10]. It is shown that an IMC $\mathcal{I}$ is consistent iff there exists a consistency relation between $\mathcal{I}$ and itself, which can be decided in polynomial time.

As explained in [10], a consistency relation allows one state of a given IMC to contribute to the consistency of other states. Although this was justified by the fact that satisfaction abstracts from the structure of transitions in IMCs, we show in the following theorem that whenever an IMC is consistent, it admits one implementation with the same structure. As a consequence, one transition in this implementation only contributes to satisfying the exact same transition in the specification IMC, which will allow us to avoid the use of consistency relations in the rest of the paper.

▶ **Theorem 8.** *An IMC $\mathcal{I} = (S, s_0, \varphi, A, V)$ is consistent iff it admits an implementation of the form $\mathcal{M} = (S, s_0, M, A, V)$ where, for all reachable state s in $\mathcal{M}$, it holds that $M(s, s') \in \varphi(s, s')$ for all s'.*

Proof. Let $\mathcal{I} = (S, s_0, \varphi, A, V)$ be an IMC. One direction of this theorem is trivial: if $\mathcal{I}$ admits an implementation of the form $\mathcal{M} = (S, s_0, M, A, V)$ where, for all reachable state s in $\mathcal{M}$, it holds that $M(s, s') \in \varphi(s, s')$ for all s', then $\mathcal{I}$ is consistent. We now prove the other direction.

Assume that $\mathcal{I}$ is consistent and let $\mathcal{M}' = (T, t_0, M', A, V')$ be a MC such that $\mathcal{M}' \models \mathcal{I}$ with satisfaction relation $\mathcal{R}$. From $\mathcal{M}'$, we build a new implementation of $\mathcal{I}$ of the desired form. Let $f : S \to T$ be a function that associates to each state of $\mathcal{I}$ one of the states in $\mathcal{M}'$ contributing to its satisfaction, if any. Formally, f is such that for all $s \in S$, if $f(s)$ is defined, then $(f(s), s) \in \mathcal{R}$. Let $\delta_{(f(s),s)}$ be the function given by $\mathcal{R}$ (item 2 of Definition 3). We now define the desired implementation $\mathcal{M} = (S, s_0, M, A, V)$. Let $S' = \{s \in S \mid \exists t \in T, (t, s) \in \mathcal{R}\}$ and $M(s, s') = \sum_{t \in T} \delta_{f(s),s}(t)(s') \cdot M'(f(s), t)$ if $s \in S'$ and 0 otherwise.

We observe that, by definition of $\mathcal{R}$ we have $M(s, s') \in \varphi(s, s')$ for all $(s, s') \in S' \times S$. Moreover, whenever $M(s, s') > 0$, there exists at least one state $t \in T$ such that $\delta_{f(s),s}(t)(s') > 0$ and $M'(f(s), t) > 0$. Therefore, by definition of δ, we have $(t, s') \in \mathcal{R}$ and thus $s' \in S'$. It thus follows that only states from S' can be reachable in $\mathcal{M}$.

Consider the identity relation $\mathcal{R}'$ over S' and let $(s, s) \in \mathcal{R}'$. Let $\delta' : S \to (S \to [0, 1])$ be such that $\delta'(s')(s'') = 1$ whenever $s' \in S'$ and $s'' = s'$, and 0 otherwise.

- Let $s' \in S$ be such that $M(s, s') > 0$. By construction, we have $s' \in S'$ and thus $\delta'(s')$ is a distribution on S.
- Let $s' \in S$ and consider $\sum_{s'' \in S} M(s, s'') \cdot \delta(s'')(s')$.
  - If $s' \notin S'$, then $\sum_{s'' \in S} M(s, s'') \cdot \delta'(s'')(s') = 0$ and we know by $\mathcal{R}$ that $0 \in \varphi(s, s')$ (because there is no $t \in T$ such that $\delta(t)(s') > 0$).
  - Otherwise, we have $\sum_{s'' \in S} M(s, s'') \cdot \delta'(s'')(s') = M(s, s') \in \varphi(s, s')$
- For all $s', s'' \in S$ such that $\delta'(s')(s'') > 0$, we have $s' = s''$ and $s' \in S'$, therefore $(s', s'') \in \mathcal{R}'$.

We conclude that $\mathcal{R}'$ is a satisfaction relation between $\mathcal{M}$ and $\mathcal{I}$. Moreover, we know by construction that $(t_0, s_0) \in \mathcal{R}$, thus $s_0 \in S'$. ◀

The fact that a consistent IMC necessarily admits an implementation with the same structure implies that using a cross-product such as introduced in the notion of consistency relation in order to prove consistency is not necessary. Therefore, one does not need to search for local inconsistencies in $S \times S$, as is done in [10], but only needs to check and avoid local inconsistencies on S.

We thus propose an alternative solution to the consistency problem for IMCs. Our solution is based on the notion of pruning. The aim of pruning is to detect and remove from a given structure all the states that cannot contribute to any of its implementations. Such states are called *inconsistent*. The algorithm we propose will follow the same principle: it will detect and propagate local inconsistencies (i.e. states whose transition intervals cannot be satisfied) through the state-space of the IMC until either the initial state is locally inconsistent – the IMC is thus inconsistent – or only consistent states are reachable, implying that the IMC is consistent. Because of Theorem 8, an implementation of the original IMC $\mathcal{I}$ can be directly derived from its pruned version.

A pruning algorithm was also proposed in [10], but it was based on the notion of consistency relation, therefore using the cross-products we are trying to avoid. In [10], a quadratic number of iterations is needed in order to build the consistency relation, each iteration being itself quadratic in the number of states. A linear number of iterations is then needed in order to prune the consistency relation. In contrast, the algorithm we propose in the following only needs a linear number of iterations, each iteration being linear itself.

The pruning operator we propose is based on the notion of local state-consistency.

▶ **Definition 9.** Given an IMC $\mathcal{I} = (S, s_0, \varphi, A, V)$, a state $s \in S$ is *locally consistent* if there exists a distribution $\rho \in \mathtt{Dist}(S)$ such that for all $s' \in S$, $\rho(s') \in \varphi(s, s')$.

Being able to check whether a given state in an IMC is locally consistent is thus of paramount importance. Fortunately, this can be done quite easily: checking whether a state is locally consistent amounts to solving a set of linear inequations. Indeed, assuming that $S = \{s_0, s_1, \ldots s_n\}$, checking whether $s_i \in S$ is consistent amounts to deciding whether the following system of inequations admits a solution.

$$\exists x_0, \ldots x_n,\ x_0 + \ldots + x_n = 1 \wedge\ x_0 \in \varphi(s_i, s_0) \wedge \ldots \wedge\ x_n \in \varphi(s_i, s_n)$$

In fact, one does not need to solve the system in order to decide whether it admits a solution. If φ contains intervals that are not well-formed, then s_i is trivially inconsistent. Otherwise, assuming all the intervals in φ are well-formed, then one only needs to check whether the sum of all lower endpoints is below 1 and whether the sum of all upper endpoints is above 1.

▶ **Proposition 10.** *Given an IMC $\mathcal{I} = (S, s_0, \varphi, A, V^I)$, a state $s \in S$ is locally consistent iff $\varphi(s, s')$ is well-formed for all s', and $\sum_{s' \in S} \mathit{Low}(\varphi(s, s')) \leq 1 \leq \sum_{s' \in S} \mathit{Up}(\varphi(s, s'))$.*

Checking whether a state is locally consistent can thus be done in linear time. Once locally inconsistent states have been identified, they will be made unreachable in $\mathcal{I}$ by iterating the following pruning operator β. In the following, we say that a state s of IMC $\mathcal{I} = (S, s_0, \varphi, A, V^I)$ is *inconsistent* iff there is no implementation of $\mathcal{I}$ in which s is satisfied. In practice, s is inconsistent iff it is locally inconsistent or there are transitions with non-zero probability leading from s to another inconsistent state s', i.e. such that $0 \notin \varphi(s, s')$. In order to keep track of inconsistent states that have already been processed, we equip IMCs with a marking function $\lambda : S \to \{0, 1\}$. States s such that $\lambda(s) = 1$ are inconsistent states that have already been identified and made unreachable in a previous iteration of β. The notion of satisfaction is not impacted by this marking function.

▶ **Definition 11** (Pruning operator β for IMCs)**.** Let $\mathcal{I} = (S, s_0, \varphi, A, V, \lambda)$ be an IMC. The pruning operator β is defined as follows. Let $\lambda_0(S) = \{s \in S \mid \lambda(s) = 0\}$.

1. If $\lambda_0(S)$ does not contain any locally inconsistent state or if $\mathcal{I} = \bot$, then $\beta(\mathcal{I}) = \mathcal{I}$.
2. Else, if s_0 is locally inconsistent, then $\beta(\mathcal{I}) = \bot$.
3. Otherwise, let $s_i \in \lambda_0(S)$ be a new locally inconsistent state in $\mathcal{I}$. We then define $\beta(\mathcal{I}) = (S, s_0, \varphi', A, V, \lambda')$, with $\lambda'(s_i) = 1$ and $\lambda'(s) = \lambda(s)$ for all $s \neq s_i$, $\varphi'(s, s') = \varphi(s, s')$ if $s' \neq s_i$, and
$$\varphi'(s, s_i) = \begin{cases} \varphi(s, s_i) & \text{if } \lambda(s) = 1 \\ [0, 0] & \text{if } \lambda(s) = 0 \text{ and } 0 \in \varphi(s, s_i) \\ \emptyset & \text{otherwise} \end{cases}$$

As seen in the above definition, the pruning operator does not remove inconsistent states but makes them unreachable. When 0 is an allowed probability for incoming transitions, β enforces this choice by modifying the subsequent intervals to $[0, 0]$. When 0 is not allowed, then the only possibility is to modify the interval probabilities to $\emptyset$, which propagates local inconsistency to predecessors. The first application of β should always be done with an empty marking function, i.e. assigning 0 to all states.

Since pruning potentially propagates local inconsistencies to predecessor states, the pruning operator β has to be applied iteratively until it converges to a fixpoint. The IMC obtained in this fixpoint is either $\perp$ or an IMC with no reachable locally inconsistent states (item 1 of Definition 11 above). Since at least one inconsistent state is detected and made unreachable at each iteration (items 2 and 3 of Definition 11), the number of iterations needed in order to converge is bounded by $|S|$. The complexity of applying pruning to $\mathcal{I}$ until it converges is thus polynomial. The result of this iteration on IMC $\mathcal{I}$ is written $\beta^*(\mathcal{I})$ in the rest of the document.

▶ **Theorem 12.** *For all IMC $\mathcal{I} = (S, s_0, \varphi, A, V)$ and marking function λ such that $\lambda(s) = 0$ for all $s \in S$, it holds that $[\![\beta^*((S, s_0, \varphi, A, V, \lambda))]\!] = [\![\mathcal{I}]\!]$.*

Proof. Let $\mathcal{I} = (S, s_0, \varphi, A, V)$ be an IMC and let λ be a Marking function such that $\lambda(s) = 0$ for all $s \in S$. Let $\mathcal{I}' = (S, s_0, \varphi', A, V, \lambda') = \beta^n((S, s_0, \varphi, A, V, \lambda))$ for some $n \in \mathbb{N}$. We show that for all MC $\mathcal{M}$, we have $\mathcal{M} \models \mathcal{I}' \iff \mathcal{M} \models \beta(\mathcal{I}')$.

If $\{s \in S \mid \lambda'(s) = 0\}$ does not contain any inconsistent state or if s_0 is inconsistent, then the result is trivial. We thus assume that an inconsistent state $s_i \in \{s \in S \mid \lambda'(s) = 0\}$ is found and made unreachable by β.

We start by observing that, by construction, all states $s \in S$ such that $\lambda'(s) = 1$ are such that for all $s' \in S$ with $\lambda(s') = 0$, we have either $\varphi'(s', s) = [0, 0]$ or $\varphi'(s', s) = \emptyset$.

$\boxed{\Rightarrow}$ Let $\mathcal{M} = (T, t_0, M, A, V^M)$ be a MC such that $\mathcal{M} \models \mathcal{I}'$. Let $\mathcal{R}$ be the associated satisfaction relation. We show that $\mathcal{R}$ is still a satisfaction relation between $\mathcal{M}$ and $\beta(\mathcal{I}') = (S, s_0, \varphi'', A, V, \lambda'')$. Let $(t, s) \in \mathcal{R}$.

1. Since β has no effect on valuations, we still have $V^M(t) = V(s)$.
2. Let $\delta : T \to (S \to [0, 1])$ be the function given by $\mathcal{R}$. By construction, it holds that
   a. for all $t' \in T$ such that $M(t, t') > 0$, $\delta(t')$ is a distribution on S,
   b. for all $s' \in S$, we have $(\sum_{t' \in T} M(t, t') \cdot \delta(t')(s')) \in \varphi'(s, s')$, and
   c. for all $t' \in T$ and $s' \in S$, if $\delta(t')(s') > 0$, then $(t', s') \in \mathcal{R}$.

Items 2.*a*. and 2.*c*. are not impacted by β. We now show that Item 2.*b*. still holds.

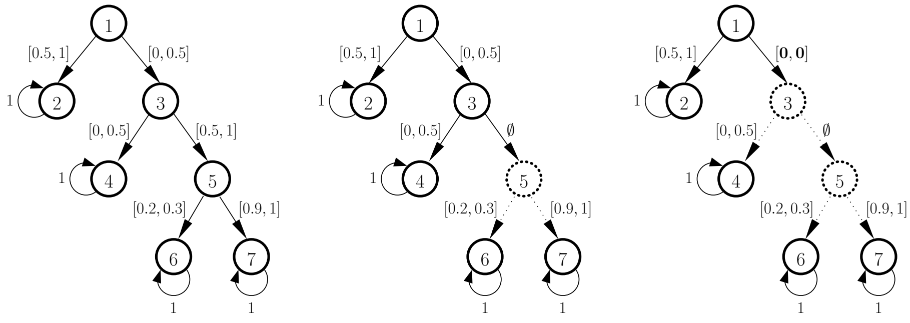
For all $s' \in S$ such that $s' \neq s_i$, we have $\varphi''(s, s') = \varphi'(s, s')$ and Item 2.*b*. still holds. Furthermore, since s_i is inconsistent in $\mathcal{I}'$, we necessarily have that for all $t' \in T$, $(t', s_i) \notin \mathcal{R}$, and thus $\delta(t')(s_i) = 0$. Therefore, we have $(\sum_{t' \in T} M(t, t') \cdot \delta(t')(s_i)) = 0$.

- If s is such that $0 \in \varphi'(s, s_i)$, then we still have $0 \in \varphi''(s, s_i)$ since $\varphi''(s, s_i)$ is either $\varphi'(s, s_i)$ or $[0, 0]$.
- Otherwise, if $0 \notin \varphi'(s, s_i)$, then we have $(\sum_{t' \in T} M(t, t') \cdot \delta(t')(s')) \notin \varphi'(s, s')$, which is a contradiction w.r.t. the definition of $\mathcal{R}$. As a consequence, there exists no $t \in T$ such that $(t, s) \in \mathcal{R}$ and the modification of $\varphi'(s, s_i)$ into $\varphi''(s, s_i) = \emptyset$ has no consequence on Item 2.*b*.

Finally, $\mathcal{R}$ is still a satisfaction relation between $\mathcal{M}$ and $\beta(\mathcal{I}')$ and therefore $\mathcal{M} \models \beta(\mathcal{I}')$.

$\boxed{\Leftarrow}$ Let $\mathcal{M} = (T, t_0, M, A, V^M)$ be a MC such that $\mathcal{M} \models \beta(\mathcal{I}') = (S, s_0, \varphi'', A, V, \lambda'')$. Let $\mathcal{R}$ be the associated satisfaction relation. We show that $\mathcal{R}$ is also a satisfaction relation between $\mathcal{M}$ and $\mathcal{I}'$. Let $(t, s) \in \mathcal{R}$ and let $\delta : T \to (S \to [0, 1])$ be the function given by $\mathcal{R}$. As above, β has no effect on valuations and on Items 2.*a*. and 2.*c*. of the definition of a satisfaction relation. We show that Item 2.*b*. also holds between $\mathcal{M}$ and $\mathcal{I}'$.

For all $s' \in S$ such that $s' \neq s_i$, we have $\varphi''(s, s') = \varphi'(s, s')$ and Item 2.*b*. trivially holds. Furthermore, since s_i is inconsistent in $\mathcal{I}$, it is also inconsistent in $\mathcal{I}'$. As a consequence, we necessarily have that for all $t' \in T$, $(t', s_i) \notin \mathcal{R}$, and thus $\delta(t')(s_i) = 0$. Therefore, we have $(\sum_{t' \in T} M(t, t') \cdot \delta(t')(s_i)) = 0$.



(a) An IMC $\mathcal{I}$. **(b)** $\beta(\mathcal{I})$. **(c)** $\beta^2(\mathcal{I}) = \beta^*(\mathcal{I})$.

Figure 4 Iterative application of the pruning operator β to IMC $\mathcal{I}$ until convergence.

- If s is such that $0 \in \varphi'(s, s_i)$, then $(\sum_{t' \in T} M(t,t') \cdot \delta(t')(s_i)) \in \varphi'(s, s_i)$ and Item 2.*b*. holds.
- Otherwise, if $0 \notin \varphi'(s, s_i)$, then we have $\varphi''(s, s_i) = \emptyset$. As a consequence $(\sum_{t' \in T} M(t,t') \cdot \delta(t')(s_i)) \notin \varphi''(s, s_i)$, which is a contradiction. Therefore, there exists no $t \in T$ such that $(t, s) \in \mathcal{R}$ and $0 \notin \varphi'(s, s_i)$.

Finally, $\mathcal{R}$ is also a satisfaction relation between $\mathcal{M}$ and $\mathcal{I}'$ and therefore $\mathcal{M} \models \mathcal{I}'$.

◀

▶ **Example 13.** Figure 4 illustrates the iteration of pruning operator β on an IMC. Consider IMC $\mathcal{I}$ from Figure 4a. Applying β on $\mathcal{I}$ consists in two steps: (1) searching for a locally inconsistent state in $\mathcal{I}$, and (2) modifying $\mathcal{I}$ in order to make the selected locally inconsistent state unreachable. At first, the only locally inconsistent state in $\mathcal{I}$ is State 5. As a consequence, applying β will either reduce all incoming interval transition probabilities to $[0, 0]$ when 0 is already allowed or to $\emptyset$ when this is not the case. The only incoming transition for State 5 is equipped with interval $[0.5, 1]$, therefore it is replaced with $\emptyset$. $\beta(\mathcal{I})$ is then depicted in Figure 4b. In the second iteration of β, State 3 is identified as locally inconsistent because it has an outgoing transition equipped with $\emptyset$. State 3 only has one incoming transition, which is equipped with interval $[0, 0.5]$. Since this interval contains 0, it is replaced with $[0, 0]$. $\beta^2(\mathcal{I})$ is represented in Figure 4c. Since $\beta^2(\mathcal{I})$ does not contain any reachable locally inconsistent state, the fixed point is reached and $\beta^*(\mathcal{I}) = \beta^2(\mathcal{I})$.

4.2 Consistency of pIMCs

We now move to the setting of pIMCs. Recall that a pIMC $\mathcal{I}^P$ is consistent iff it admits at least one consistent instance, i.e. $\exists \mathcal{I}, \exists \mathcal{M} \mid \mathcal{M} \models \mathcal{I}$ and $\mathcal{I} \vdash \mathcal{I}^P$.

As we will see later, the difficulty of deciding whether a given pIMC $\mathcal{I}^P$ is consistent highly depends on the nature of the parameters in $\mathcal{I}^P$. This is due to the fact that the notion of local state-consistency only makes sense for states whose transition probability intervals only contain local parameters. Indeed, in the case of global parameters, the local consistency of one state might be incompatible with the local consistency of another due to the incompatible choice of parameter valuations. In the following, we propose an intuitive and efficient solution for deciding whether a local pIMC is consistent. We then show that

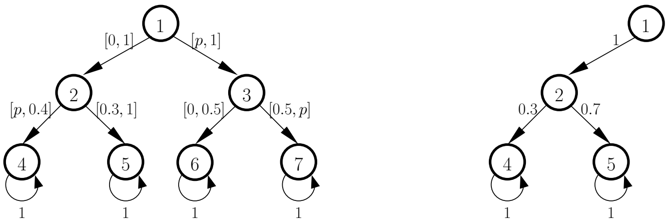


Figure 5 Global pIMC $\mathcal{I}^P$ with global parameter p (left) and one of its implementations (right).

consistency is also decidable in the case of global parameters although the algorithm we propose is more complex.

▶ **Example 14.** Consider pIMC $\mathcal{I}^P$ given on the left of Figure 5. Parameter p in $\mathcal{I}^P$ is global as it affects outgoing transitions of States 1, 2 and 3. If one is checking local state-consistency, it looks like all states in $\mathcal{I}^P$ are locally consistent. Indeed, outgoing transitions of State 1 can be satisfied with $p = 0$; outgoing transitions of State 2 can be satisfied with $p = 0$; and outgoing transitions of State 3 can be satisfied with $p = 1$. From a purely local point of view, it thus seems that $\mathcal{I}^P$ is consistent. However, it also appears that State 2 requires that $p \leq 0.4$ while State 3 requires that $p \geq 0.5$. One could therefore conclude that $\mathcal{I}^P$ is inconsistent. Despite of this fact, we claim that $\mathcal{I}^P$ is consistent: if p is set to 0, then we can reduce the transition interval from State 1 to State 3 to $[0, 0]$, which makes State 3 unreachable. Therefore, one no longer needs to have $p \geq 0.5$ and a correct implementation of $\mathcal{I}^P$ can be found. Such an implementation is Given on the right of Figure 5.

Consistency of local pIMCs. In order to check consistency of local pIMCs, a similar algorithm to the one used for checking consistency of IMCs can be used. In fact, due to Theorem 8, one does not need to consider particular instances of $\mathcal{I}^P$ in order to find out whether $\mathcal{I}^P$ is consistent: since all instances of $\mathcal{I}^P$ share the same structure, $\mathcal{I}^P$ will be consistent iff there exists an implementation that shares this structure. Since the notion of local state-consistency makes sense in the case of local pIMCs, we adapt the pruning algorithm presented in Section 4.1 to local pIMCs.

Let $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$ be a local pIMC and let $s \in S$. We write $\texttt{param}(s) = \{p \in P \mid \texttt{Range}(p) = \{s\}\}$ for the set of parameters appearing in the outgoing transition intervals of s. We then say that s is *locally consistent* iff there exists a valuation ψ over $\texttt{param}(s)$ and a distribution $\rho \in \texttt{Dist}(S)$ such that for all $s' \in S$, $\rho(s') \in \varphi_P(s, s')[p \leftarrow \psi(p)]$. Recall that the only parameters potentially appearing in $\varphi_P(s, s')$ are necessarily from $\texttt{param}(s)$.

In a similar fashion to the case of IMCs, local consistency of state $s_i \in S$ can be reduced to checking whether a system of inequations admits a solution. In order to facilitate presentation, we assume that $S = \{s_1, \ldots, s_n\}$ and we use the parameters in $\texttt{param}(s_i)$ as variables taking values in $[0, 1]$. The system is then as follows:

$$\textstyle\sum_{j=1}^{n} \texttt{Low}(\varphi_P(s_i, s_j)) \leq 1 \wedge \sum_{j=1}^{n} \texttt{Up}(\varphi_P(s_i, s_j)) \geq 1 \wedge \texttt{Low}(\varphi_P(s_i, s_1)) \leq \texttt{Up}(\varphi_P(s_i, s_2)) \\ \wedge \ldots \wedge \texttt{Low}(\varphi_P(s_i, s_n)) \leq \texttt{Up}(\varphi_P(s_i, s_n))$$

In this system, the first two inequations reflect the definition of local state-consistency while the other inequations ensure that all the intervals expressed using parameters are well-formed. In the case of IMCs, we were able to remove this check by assuming beforehand that our IMCs were well-formed. In the case of pIMCs, we cannot assume the same as well-formedness will depend on the actual value given to parameters. Nevertheless, solving such a system of inequations can be done in polynomial time w.r.t. $|S|$ and $|P|$.

We now propose a pruning algorithm for local pIMCs based on the notion of local state-consistency. The outline of this algorithm is similar to the algorithm for IMCs, and only the modification of interval probabilities following the discovery of a new locally inconsistent state s_i is slightly modified: We start by identifying the set of parameters appearing as the lower bound of a transition interval leading to s_i and then enforce the value of these parameters to be 0 in order to be able to make s_i unreachable. Formally, we write $\texttt{enforce}(s_i) = \{p \in P \mid \exists s \in S, p = \texttt{Low}(\varphi_P(s, s_i))\}$ for this set of parameters. As for IMCs, we use a marking function $\lambda : S \to \{0, 1\}$ in order to keep track of locally inconsistent states that have already been processed. The notions of instantiation and satisfaction are not impacted by this marking function.

▶ **Definition 15** (Pruning operator β for pIMCs). Let $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P, \lambda)$ be a pIMC. The pruning operator β for pIMCs is defined as follows. Let $\lambda_0(S) = \{s \in S \mid \lambda(s) = 0\}$.

1. If $\lambda_0(S)$ does not contain any locally inconsistent state or if $\mathcal{I}^P = \perp$ then $\beta(\mathcal{I}^P) = (\mathcal{I}^P)$.
2. Else, if s_0 is locally inconsistent, then $\beta(\mathcal{I}^P) = \perp$.
3. Otherwise, let $s_i \in \lambda_0(S)$ be a new locally inconsistent state in $\mathcal{I}^P$. We then define $\beta(\mathcal{I}^P) = (S, s_0, \varphi'_P, A, V, P, \lambda')$, with $\lambda'(s_i) = 1$ and $\lambda'(s) = \lambda(s)$ for all $s \neq s_i$, $\varphi'_P(s, s') = \varphi_P(s, s')[\texttt{enforce}(s_i) \leftarrow 0]$ if $s' \neq s_i$, and

$$\varphi'_P(s, s_i) = \begin{cases} \varphi_P(s, s_i)[\texttt{enforce}(s_i) \leftarrow 0] \text{ if } \lambda(s) = 1 \\ [0, 0] \text{ if } \lambda(s) = 0 \text{ and } \varphi_P(s, s_i)[\texttt{enforce}(s_i) \leftarrow 0] = [0, .] \\ \emptyset \text{ otherwise} \end{cases}$$

As for IMCs, the pruning operator β for pIMCs propagates local inconsistencies to predecessor states. Therefore, β has to be applied iteratively until a fixpoint if reached. The pIMC obtained in this fixpoint is either $\perp$ or a pIMC with no reachable locally inconsistent state (item 1 of Definition 15). Since at least one inconsistent state is identified and made unreachable at each iteration (items 2 and 3 of Definition 15), the number of iterations needed in order to converge is bounded by $|S|$. Therefore, the complexity of applying pruning to a given local pIMC until convergence is polynomial in $|S|$ and $|P|$. The result of this iteration on pIMC $\mathcal{I}^P$ is written $\beta^*(\mathcal{I}^P)$.

▶ **Example 16.** Figure 6 illustrates the pruning operator for local pIMCs. Consider local pIMC $\mathcal{I}^P$ given in the left of Figure 6. We start by searching for locally inconsistent states in $\mathcal{I}^P$: State 3 is chosen. The first application of pruning operator β will therefore try to make State 3 unreachable by forcing all incoming transition intervals to $[0, 0]$. This can only be done if either 0 is already the lower bound of the incoming interval or if a parameter p is the lower bound of the incoming interval and p can be forced to 0 throughout the whole pIMC. In $\mathcal{I}^P$, State 3 only has one incoming transition, which is equipped with interval $[p, 1]$. Parameter p is thus forced to 0 in all other transitions and the incoming interval to State 3 is reduced to $[0, 0]$. The result $\beta(\mathcal{I}^P)$ is given in the right of Figure 6. Since there are no more locally inconsistent states in $\beta(\mathcal{I}^P)$, we have $\beta^*(\mathcal{I}^P) = \beta(\mathcal{I}^P)$.

We now show that the result of iterating β on a given pIMC $\mathcal{I}^P$ is a pIMC with the same set of implementations as $\mathcal{I}^P$.

▶ **Theorem 17.** *For all local pIMC $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$ and marking function λ such that $\lambda(s) = 0$ for all $s \in S$, it holds that for all MC $\mathcal{M}$, $\mathcal{M} \models \mathcal{I}^P$ iff $\mathcal{M} \models \beta^*((S, s_0, \varphi_P, A, V, P, \lambda))$.*

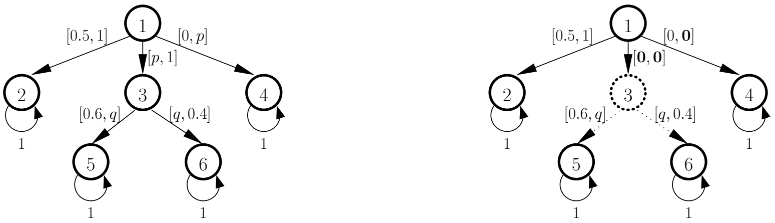


Figure 6 Iterative application of the pruning operator to pIMC $\mathcal{I}^P$ (left) until convergence (right).

Proof. Let $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$ be a local pIMC and let λ be a marking function such that $\lambda(s) = 0$ for all $s \in S$. Let $\mathcal{I}^{P'} = (S, s_0, \varphi'_P, A, V, P, \lambda') = \beta^n((S, s_0, \varphi_P, A, V, P, \lambda))$ for some $n \in \mathbb{N}$. We show that for all MC $\mathcal{M}$, we have $\mathcal{M} \models \mathcal{I}^{P'} \iff \mathcal{M} \models \beta(\mathcal{I}^{P'})$.

If $\{s \in S \mid \lambda'(s) = 0\}$ does not contain any locally inconsistent state or if s_0 is locally inconsistent, then the result is trivial. We thus assume that a locally inconsistent state $s_i \in \{s \in S \mid \lambda'(s) = 0\}$ is found and made unreachable by β.

We start by observing that, by construction, all states $s \in S$ such that $\lambda'(s) = 1$ are such that for all $s' \in S$ with $\lambda(s') = 0$, we have either $\varphi'_P(s', s) = [0, 0]$ or $\varphi'_P(s', s) = \emptyset$.

$\boxed{\Rightarrow}$ Let $\mathcal{I} = (S, s_0, \varphi, A, V, P)$ be an IMC and let $\psi : P \to [0, 1]$ be a valuation for the parameters such that $\mathcal{I} \vdash_\psi \mathcal{I}^{P'}$. Let $\mathcal{M} = (T, t_0, M, A, V^M)$ be a MC such that $M \models \mathcal{I}$ with satisfaction relation $\mathcal{R} \subseteq T \times S$. We show that there exists an IMC $\mathcal{I}'$ such that $M \models \mathcal{I}'$ and $\mathcal{I}' \vdash \beta(\mathcal{I}^{P'})$.

The proof proceeds in two steps: we first build the IMC $\mathcal{I}'$ and show that $\mathcal{I}' \vdash \beta(\mathcal{I}^{P'})$ and then show that $\mathcal{M} \models \mathcal{I}'$.

Let $\psi' : P \to [0, 1]$ be a new valuation for the parameters such that $\psi'(p) = 0$ if $p \in \texttt{enforce}(s_i)$ and $\psi'(p) = \psi(p)$ otherwise. Let $\mathcal{I}' = (S, s_0, \varphi', A, V, P)$ be such that $\varphi'(s, s') = \varphi_P(s, s')[p \leftarrow \psi'(p)]$ if $s' \neq s_i$ or if $\lambda(s) = 1$, $\varphi'(s, s_i) = [0, 0]$ if $\lambda(s) = 0$ and $\varphi_P(s, s_i)[\texttt{enforce}(s_i) \leftarrow 0] = [0, .]$, and $\varphi'(s, s_i) = \emptyset$ otherwise. By construction, it follows that $\mathcal{I}' \vdash_{\psi'} \beta(\mathcal{I}^{P'})$.

We now show that $\mathcal{R}$ is a satisfaction relation between $\mathcal{M}$ and $\mathcal{I}'$. Let $(t, s) \in \mathcal{R}$.

1. Since β has no effect on valuations, we have $V^M(t) = V(s)$.
2. Let δ be the function given in item 2 of Definition 3. By construction, it holds that
   a. for all $t' \in T$ such that $M(t, t') > 0$, $\delta(t')$ is a distribution on S,
   b. for all $s' \in S$, we have $(\sum_{t' \in T} M(t, t') \cdot \delta(t')(s')) \in \varphi(s, s')$, and
   c. for all $t' \in T$ and $s' \in S$, if $\delta(t')(s') > 0$, then $(t', s') \in \mathcal{R}$.

Items 2.a. and 2.c. are not impacted by β. We now show that item 2.b. still holds when considering φ' instead of φ.

Remark that since s_i is locally inconsistent in $\mathcal{I}'$, we necessarily have that for all $t' \in T$, $(t', s_i) \notin \mathcal{R}$ and therefore $\delta(t')(s_i) = 0$. Let $s' \in S$ and consider $\varphi'(s, s')$.

- If $s' \neq s_i$, we have $\varphi'(s, s') = \varphi_P(s, s')[p \leftarrow \psi'(p)]$. If $\varphi_P(s, s') = [x, y]$ with $x \in [0, 1] \cup P$ and $y \in [0, 1] \cup (P \setminus \texttt{enforce}(s_i))$, then either $\varphi'(s, s') = \varphi(s, s')$ or $\varphi(s, s') \subseteq \varphi'(s, s')$ and therefore $(\sum_{t' \in T} M(t, t') \cdot \delta(t')(s')) \in \varphi'(s, s')$. The only difficulty appears when $y = p \in \texttt{enforce}(s_i)$. In this case, there must exist $s'' \in S$ such that $\varphi_P(s'', s_i) = [p, .]$. Moreover, since $\mathcal{I}^P$ is *local*, we must have $s = s''$. By $\mathcal{R}$, we know that $(\sum_{t' \in T} M(t, t') \cdot$

$\delta(t')(s_i)) \in \varphi(s, s_i)$. Since $\delta(t')(s_i) = 0$ for all $t' \in T$, we have that $0 \in \varphi(s, s_i) = [\psi(p), .]$, thus $\psi(p) = \psi'(p) = 0$ and $\varphi'(s, s') = \varphi(s, s')$. As a consequence, item 2.b. still holds.

- If $s' = s_i$, then since s_i is inconsistent, we have $\sum_{t' \in T} M(t, t') \cdot \delta(t')(s_i) = 0 \in \varphi(s, s_i)$. As a consequence, we necessarily have $\varphi(s, s_i) = [0, .]$ and thus $\varphi_P(s, s_i)[\texttt{enforce}(s_i) \leftarrow 0] = [0, .]$. Therefore, by construction, we still have $0 \in \varphi'(s, s_i)$ and item 2.b. holds.

Finally, $\mathcal{R}$ is still a satisfaction relation between $\mathcal{M}$ and $\mathcal{I}'$ and therefore $\mathcal{M} \models \mathcal{I}'$.

$\boxed{\Leftarrow}$ The proof of $\Leftarrow$ is straightforward with symmetric arguments. ◀

Consistency of global pIMCs. Unfortunately, the pruning algorithm we propose above cannot be ported to the setting of global pIMCs. Indeed, as illustrated in Example 14, the notion of local state consistency does not make sense in this setting, as restrictions on the values of parameters given by the local consistency of a given state can impact the local consistency of another. Nevertheless, consistency of global pIMCs is decidable: one can derive another, more complex, pruning algorithm from the one proposed in Definition 15. Since this algorithm is not optimal and only serves to prove decidability, we only present the outline of the algorithm without going into too much details.

Since fixing the value of given parameters may impact several states, we propose to group states that share given parameters and check inter-consistency of this group of states instead of local consistency of all states taken separately. We thus define groups of states that share parameters and propose a system of inequations that will decide whether this group of states is *inter-consistent*. Formally, given global pIMC $\mathcal{I}^P = (S, s_0, \varphi_P, A, V, P)$ and states $s_1, s_2 \in S$, we say that s_1 and s_2 are inter-dependent, written $s_1 \leftrightarrow s_2$ iff either $\texttt{param}(s_1) \cap \texttt{param}(s_2) \neq \emptyset$ or there exists s_3 such that $s_1 \leftrightarrow s_3$ and $s_3 \leftrightarrow s_2$. The groups of states we consider for the new notion of *inter-consistency* will thus be equivalence classes under $\leftrightarrow$.

Given such an equivalence class $\overline{s}$, we say that $\overline{s}$ is inter-consistent iff the system of inequations consisting of all inequations for local consistency of all states in $\overline{s}$ admits a solution. When $\overline{s}$ is **not** inter-consistent, the pruning algorithm will nondeterministically choose one of the states in $\overline{s}$, try to make it unreachable as in Definition 15 and mark it. From this point, if pruning goes on until $\mathcal{I}^P$ is proven consistent, then we can conclude positively. However, if the initial state is ultimately proven inconsistent, then we cannot conclude and the algorithm will backtrack and try making another state from $\overline{s}$ unreachable instead until all possible combinations of states in $\overline{s}$ have been considered. Only then can we conclude that $\mathcal{I}^P$ is inconsistent. Since there are only finitely many combinations of states in S, the algorithm will ultimately converge and allow deciding whether global pIMC $\mathcal{I}^P$ is consistent.

5 Concluding remarks

In this paper, we have introduced the new formalism of parametric Interval Markov Chains, that extends Interval Markov Chains by allowing the use of parameters as lower or upper bounds to the interval probabilities of transitions. We have also shown that the consistency problem is decidable for pIMCs and proposed an efficient algorithm for checking consistency of pIMCs with local parameters only. While we limit ourselves to intervals where parameters can only appear as lower or upper bound, our work can be directly extended to intervals with linear expressions over parameters and constants. In fact, this change does not impact any of the proposed solutions for local or global pIMCs : the systems of inequations we propose for deciding local or inter-consistency and the subsequent pruning algorithms remain unchanged.

The first direction for future work is to design better-suited algorithms for solving the consistency problem in the case of global pIMCs. Our second direction for future work is to consider other problems of interest for pIMCs, e.g. parameter synthesis with respect to some optimality criterion such as reachability. Finally, as has been argued in the literature, IMCs are quite limited as a specification theory as they are not closed under compositional operators such as parallel composition or conjunction. Therefore, we plan to extend our reasoning to more expressive specification theories such as Constraint Markov Chains [6] or Abstract Probabilistic Automata [9].

References

1 É. André, L. Fribourg, and J. Sproston. An extension of the inverse method to probabilistic timed automata. *Formal Methods in System Design*, (2):119–145, 2013.

2 R. Barbuti, F. Levi, P. Milazzo, and G. Scatena. Probabilistic model checking of biological systems with uncertain kinetic rates. *Theor. Comput. Sci.*, 419(0):2 – 16, 2012.

3 N. Bertrand and P. Fournier. Parameterized verification of many identical probabilistic timed processes. In *FSTTCS*, volume 24 of *LIPIcs*, pages 501–513, 2013.

4 N. Bertrand, P. Fournier, and A. Sangnier. Playing with probabilities in reconfigurable broadcast networks. In *FoSSaCS*, volume 8412 of *LNCS*, pages 134–148. Springer, 2014.

5 F. Biondi, A. Legay, B.F. Nielsen, and A. Wasowski. Maximizing entropy over markov processes. In *LATA*, volume 7810 of *LNCS*, pages 128–140. Springer, 2013.

6 B. Caillaud, B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, and A. Wasowski. Constraint markov chains. *Theor. Comput. Sci.*, 412(34):4373–4404, 2011.

7 N. Chamseddine, M. Duflot, L. Fribourg, C. Picaronny, and J. Sproston. Computing expected absorption times for parametric determinate probabilistic timed automata. In *QEST*, pages 254–263. IEEE Computer Society, 2008.

8 C. Daws. Symbolic and parametric model checking of discrete-time Markov chains. In *ICTAC*, volume 3407 of *LNCS*, pages 280–294. Springer, 2004.

9 B. Delahaye, J-P. Katoen, K.G. Larsen, A. Legay, M.L. Pedersen, F. Sher, and A. Wasowski. Abstract probabilistic automata. *Inf. Comput.*, 232:66–116, 2013.

10 B. Delahaye, K.G. Larsen, A. Legay, M.L. Pedersen, and A. Wasowski. Consistency and refinement for interval markov chains. *J. Log. Algebr. Program.*, 81(3):209–226, 2012.

11 L.M. Ferrer Fioriti, E.M. Hahn, H. Hermanns, and B. Wachter. Variable probabilistic abstraction refinement. In *ATVA*, volume 7561 of *LNCS*, pages 300–316. Springer, 2012.

12 R. Gori and F. Levi. An analysis for proving probabilistic termination of biological systems. *Theor. Comput. Sci.*, 471(0):27 – 73, 2013.

13 E.M. Hahn, T. Han, and L. Zhang. Synthesis for PCTL in parametric Markov decision processes. In *NSFM*, volume 6617 of *LNCS*, pages 146–161. Springer, 2011.

14 E.M. Hahn, H. Hermanns, B. Wachter, and L. Zhang. PARAM: A model checker for parametric Markov models. In *CAV*, volume 6174 of *LNCS*, pages 660–664. Springer, 2010.

15 E.M. Hahn, H. Hermanns, and L. Zhang. Probabilistic reachability for parametric Markov models. *Software Tools for Technology Transfer*, 13(1):3–19, 2011.

16 B. Jonsson and K.G. Larsen. Specification and refinement of probabilistic processes. In *LICS*, pages 266–277. IEEE Computer, 1991.

17 R. Lanotte, A. Maggiolo-Schettini, and A. Troina. Decidability results for parametric probabilistic transition systems with an application to security. In *SEFM*, pages 114–121. IEEE Computer Society, 2004.

18 R. Lanotte, A. Maggiolo-Schettini, and A. Troina. Parametric probabilistic transition systems for system design and analysis. *Formal Aspects of Computing*, 19(1):93–109, 2007.

19 K. Sen, M. Viswanathan, and G. Agha. Model-checking markov chains in the presence of uncertainties. In *TACAS*, volume 3920 of *LNCS*, pages 394–410. Springer, 2006.