Version of Record: https://www.sciencedirect.com/science/article/pii/S2352220817301372
Manuscript_eafa3f59f4bf0d6bc8135b56c5eff1a0

Consistency in Parametric Interval Probabilistic Timed Automata[☆]

Étienne André[*]

Université Paris 13, LIPN, CNRS, UMR 7030, F-93430, Villetaneuse, France

Benoît Delahaye, Paulin Fournier

Université de Nantes / LS2N UMR CNRS 6004, Nantes, France

Abstract

We propose a new abstract formalism for probabilistic timed systems, Parametric Interval Probabilistic Timed Automata, based on an extension of Parametric Timed Automata and Interval Markov Chains. In this context, we consider the consistency problem that amounts to deciding whether a given specification admits at least one implementation. In the context of Interval Probabilistic Timed Automata (with no timing parameters), we show that this problem is decidable and propose a constructive algorithm for its resolution. We show that the existence of timing parameter valuations ensuring consistency is undecidable in the general context, but still exhibit a syntactic condition on parameters to ensure decidability. We also propose procedures that resolve both the consistency and the consistent reachability problems when the parametric probabilistic zone graph is finite.

Keywords: parametric verification, timed probabilistic systems, parametric probabilistic timed automata

[☆]This work is partially supported by the ANR national research program PACS (ANR-14-CE28-0002).

[*]This work was partially done during Étienne André's *délégation CNRS* at École Centrale de Nantes, IRCCyN, CNRS, UMR 6597, France (2015–2016)

Email addresses: `first.last@lipn.univ-paris13.fr` (Étienne André), `first.last@univ-nantes.fr` (Benoît Delahaye, Paulin Fournier)

URL: `https://lipn.univ-paris13.fr/~andre/` (Étienne André), `http://pagesperso.lina.univ-nantes.fr/~delahaye-b/` (Benoît Delahaye, Paulin Fournier)

1. Introduction

Motivation. Nowadays, automata-based modeling and verification methods are mainly used in two different ways: for designing digital systems based on (mostly informal) specifications expressed by the end-users of these systems or from the knowledge designers have of their environment; and in order to abstract existing (not necessarily software) systems that are too complex to comprehend in their entirety. In both cases the complexity of the systems being designed calls for increasingly expressive abstraction artifacts such as time and probabilities. Timed automata, introduced in Alur and Dill (1994), are a widely recognized modeling formalism for reasoning about real-time systems. This modeling formalism, based on finite control automata equipped with clocks, which are real-valued variables which increase uniformly at the same rate, has been extended to the probabilistic framework in Gregersen and Jensen (1995); Kwiatkowska et al. (2002). In this context, discrete actions are replaced with probabilistic discrete distributions over discrete actions, allowing to model uncertainties in the system's behavior. This formalism has been applied to a number of case studies, e. g., in Kwiatkowska et al. (2006).

Unfortunately, building a system model based either on imprecise specifications or on imprecise observations often requires to fix arbitrarily a number of constants in the model, which are then calibrated by a fastidious comparison of the model behavior and the expected behavior. This is the case for instance for timing constants or transition probability values. In order to incorporate these uncertainties in the model and to develop automatic calibration, more abstract formalisms have been introduced separately in the timed setting and in the probabilistic setting.

In the timed setting, *parametric timed automata* (PTAs) introduced by Alur et al. (1993) allow using parameter variables in the guards of timed transitions in order to account for the uncertainty on their values. The reachability emptiness problem, i. e., the emptiness of the set of valuations for which a given discrete state is reachable, is undecidable for parametric timed automata as shown in Alur et al. (1993), even for bounded parameters as shown by Miller (2000), for a single integer-valued parameter as shown by Beneš et al. (2015), or only when strict inequalities are used as shown by Doyen (2007). Decidable subclasses were exhibited (e. g., Hune et al. (2002); Bozzelli and La Torre (2009); Jovanović et al. (2015); André et al. (2016)).

Parametric probabilistic timed automata were proposed in André et al. (2013) to answer the following question: given a timing parameter valuation, what are other valuations preserving the same minimum and maximum probabilities for reachability properties as the reference valuation? Parametric probabilistic timed automata were then given a symbolic semantics in Jovanović and Kwiatkowska (2014); a method has been proposed in that same work to synthesize optimal parameter valuations to maximize or minimize the probability of reaching a discrete location.

In the purely probabilistic setting, Interval Markov Chains (IMCs for short) have been introduced by Jonsson and Larsen (1991) to take into account imprecision in the transition probabilities. IMCs extend Markov Chains by allowing to specify intervals of possible probabilities on transitions instead of exact values. Methods have then been developed to decide whether there exist Markov Chains with concrete probability values that match the intervals specified in a given IMC (see Delahaye et al. (2012)).

Contribution. In this paper, we propose to combine both abstraction approaches into a single specification theory: Parametric Interval Probabilistic Timed Automata (PIℙTAs for short). In this setting, parameters can be used in order to abstract timed constants on transition guards while intervals can be used to abstract imprecise transition probabilities. Allowing this higher level of freedom allow for incremental design, where one can first give large sets of values for which the system may be defined, and then further refine them. This refinement will take the form of an instance of a probabilistic interval, or the concrete instance of a timing parameter.

As for IMCs, it is important to be able to decide whether the probability intervals that are specified in a model allow defining consistent probability distributions (i. e., can be matched in a real-life implementation). This is called the consistency problem.

First, in the context of Interval Probabilistic Timed Automata with no timing parameters (IℙTAs for short), we propose an algorithm that solves this problem.

Second, in the parametric setting, since the behavior of the system is conditioned by the calibration of parameter values, it is necessary to decide whether there exist parameter values that ensure consistency of the resulting model (and synthesize these values when this is possible). We show that the existence of such parameter valuations is undecidable in the general context of PIℙTAs. Still, we exhibit a sufficient syntactic condition on the use of

the parameters to ensure decidability, when parameters are partitioned into lower-bound parameters and upper-bound parameters (in their comparisons with clocks). In addition, we propose a construction that characterizes, whenever the parametric probabilistic zone graph is finite, the set of parameter values that ensure consistency of the resulting IℙTA. We finally address the problem of parametric consistent reachability, i. e., of synthesizing valuations for which a given state is reachable and the model is consistent.

Example 1. The Root Contention Protocol, used for the election of a leader in the physical layer of the IEEE 1394 standard, consists in first drawing a random number (0 or 1), then waiting for some time according to the result drawn, followed by the sending of a message to the contending neighbor. This is repeated by both nodes until one of them receives a message before sending one, at which point the root is appointed. This protocol was modeled in Collomb-Annichini and Sighireanu (2001) using parametric timed automata, in Kwiatkowska et al. (2003) with probabilistic timed automata, and in André et al. (2013) using parametric probabilistic timed automata, i. e., parametric timed automata extended with (non-parametric) probabilistic distributions.

Figure 1 shows a PIℙTA model of the node i. The wire can be found in Kwiatkowska et al. (2003); André et al. (2013). Figure 1 features one clock x_i and four parameters f_min, f_max, s_min and s_max. In short, the goal of the protocol is that each node reaches either the child status, or the root status. In addition, observe that we use probabilistic interval distributions; they can be seen as an additional design freedom, allowing for incremental design. The one going out from ROOT_IDLE clearly admits no implementation, as no instance of the two intervals $[0.3, 0.4]$ can be such that their sum is equal to 1. This probabilistic interval distribution could be either disabled by setting other probabilities to 0 so that location ROOT_IDLE becomes unreachable; or by tuning the values of the four parameters (or the parameters in the other PIℙTAs in parallel) so that the guard going out from ROOT_IDLE becomes unsatisfiable. The rest of the this manuscript is dedicated to this problem.

Outline. We start Section 2 with preliminary definitions and then introduce the concepts of IℙTAs and PIℙTAs. In Section 3, we study the consistency problem for IℙTAs and propose a constructive algorithm based on the zone-graph construction that decides whether an IℙTA is consistent and produces

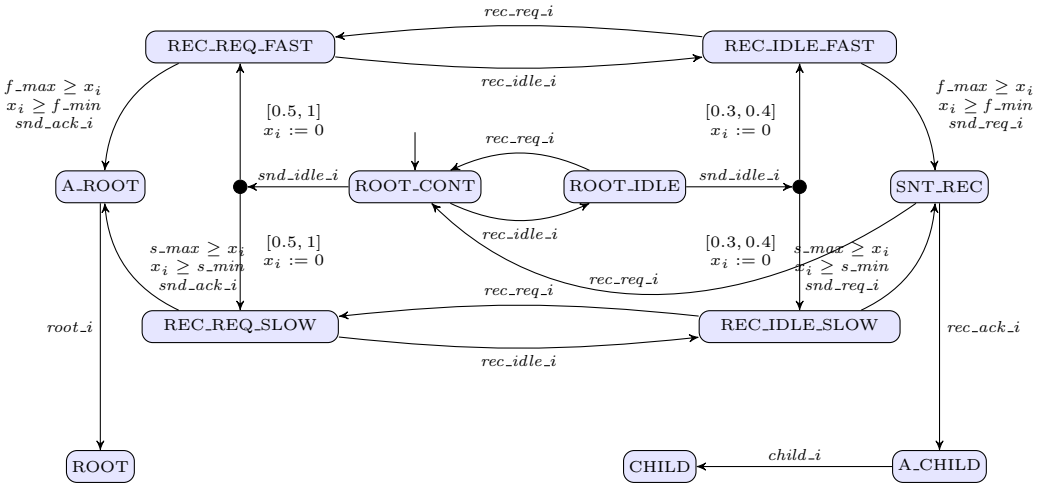


Figure 1: PIℙTA modeling node i in the Root Contention Protocol

an implementation if one exists. In Section 4, we move to the general problem of consistency of PIℙTAs. We first show that this problem is undecidable in general and then exhibit a decidable subclass. We then propose a construction that characterizes, whenever the parametric probabilistic zone graph is finite, the set of parameter values ensuring consistency of the resulting IℙTA. We also consider the problem of parametric consistent reachability. Finally, Section 5 concludes the paper.

2. Preliminaries

2.1. Clocks, parameters and constraints

Let $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}_+$ and $\mathbb{R}_+$ denote the sets of non-negative integers, integers, non-negative rational numbers and non-negative real numbers respectively. Given an arbitrary set S, we write $\mathsf{Dist}(S)$ for the set of probabilistic distributions over S.

Throughout this paper, let $X = \{x_1, \ldots, x_H\}$ be a set of *clocks*, i. e., real-valued variables that evolve at the same rate, and $\Gamma = \{\gamma_1, \ldots, \gamma_M\}$ be a set of *parameters*, i. e., unknown constants used in guards.

A clock valuation is a function $w : X \to \mathbb{R}_+$. We identify a clock valuation w with the *point* $(w(x_1), \ldots, w(x_H))$. We write $\vec{0}$ for the valuation that assigns 0 to each clock. Given $d \in \mathbb{R}_+$, $w + d$ denotes the valuation such that $(w + d)(x) = w(x) + d$, for all $x \in X$. Given $\rho \subseteq X$, we define $[w]_\rho$ as the

clock valuation obtained by resetting the clocks in ρ and keeping the other clocks unchanged.

A parameter *valuation* v is a function $v : \Gamma \to \mathbb{Q}_+$. We identify a parameter valuation v with the *point* $(v(\gamma_1), \ldots, v(\gamma_M))$.

In the following, we assume $\bowtie \in \{<, \leq, \geq, >\}$. Let *aft* range over affine terms over $X \cup \Gamma$, of the form $\sum_{1 \leq i \leq H} \alpha_i x_i + \sum_{1 \leq j \leq M} \beta_j \gamma_j + d$, with $x_i \in X$, $\gamma_j \in \Gamma$, and $\alpha_i, \beta_j, d \in \mathbb{Z}$. Similarly, let *paft* range over parametric affine terms over Γ, that is affine terms without clocks ($\alpha_i = 0$ for all i). A *constraint* C over $X \cup \Gamma$ is a conjunction of inequalities of the form *aft* $\bowtie 0$ (i. e., a convex polyhedron). Given a parameter valuation v, $v(C)$ denotes the constraint over X obtained by replacing each parameter γ in C with $v(\gamma)$. Likewise, given a clock valuation w, $w(v(C))$ denotes the expression obtained by replacing each clock x in $v(C)$ with $w(x)$. We say that v *satisfies* C, denoted by $v \models C$, if the set of clock valuations satisfying $v(C)$ is nonempty. Given a parameter valuation v and a clock valuation w, we denote by $w|v$ the valuation over $X \cup \Gamma$ such that for all clocks x, $w|v(x) = w(x)$ and for all parameters γ, $w|v(\gamma) = v(\gamma)$. We use the notation $w|v \models C$ to indicate that $w(v(C))$ evaluates to true. We say that C is *satisfiable* if $\exists w, v$ s. t.$w|v \models C$. We define the *time elapsing* of C, denoted by $C^{\nearrow}$, as the constraint over X and Γ obtained from C by delaying all clocks by an arbitrary amount of time. Given $\rho \subseteq X$, we define the *reset* of C, written $[C]_\rho$, as the constraint obtained from C by resetting the clocks in ρ, and keeping the other clocks unchanged. We denote by $C{\downarrow_\Gamma}$ the projection of C onto Γ, i. e., obtained by eliminating the clock variables (e. g., using the Fourier-Motzkin algorithm).

A *guard* g is a constraint over $X \cup \Gamma$ defined by inequalities of the form $x \bowtie z$, where $x \in X$ and z is either a parameter or a constant in $\mathbb{Z}$.

A *zone* is a polyhedron over a set of clocks in which all constraints on variables are of the form $x \bowtie k$ (rectangular constraints) or $x_i - x_j \bowtie k$ (diagonal constraints), where $x_i \in X$, $x_j \in X$ and k is an integer. Operations on zones are well-documented (see e. g., Bengtsson and Yi (2003)).

A *parametric zone* is a convex polyhedron over $X \cup \Gamma$ in which all constraints on variables are of the form $x \bowtie$ *paft* (parametric rectangular constraints) or $x_i - x_j \bowtie$ *paft* (parametric diagonal constraints), where $x_i \in X$, $x_j \in X$ and *paft* is a parametric affine term over Γ. We denote the set of all parametric zones by $\mathcal{Z}$.

2.2. Probabilistic timed automata

We start by reviewing the definition of timed probabilistic systems, as defined in Kwiatkowska et al. (2002). A *timed probabilistic system (TPS)* is a tuple $\mathcal{T} = (S, s_0, \Sigma, \Rightarrow)$ where S is a set of *states*, $s_0 \in S$ is the *initial state*, Σ is a finite set of *actions*, and $\Rightarrow \subseteq S \times \mathbb{R}_+ \times \Sigma \times \mathsf{Dist}(S)$ is a *probabilistic transition relation* that associates a probabilistic distribution over S to triples made of a source state in S, a time in $\mathbb{R}_+$ and an action in Σ.

Probabilistic timed automata (defined by Gregersen and Jensen (1995); Kwiatkowska et al. (2002)) are an extension of classical timed automata (defined in Alur and Dill (1994)) with discrete probability distributions.

2.2.1. Syntax

Definition 1. A Probabilistic Timed Automaton (PTA) $\mathcal{P}$ is a tuple $(\Sigma, L, l_0, X, prob)$, where: *i*) Σ is a finite set of actions, *ii*) L is a finite set of locations, *iii*) $l_0 \in L$ is the initial location, *iv*) X is a finite set of clocks, *v*) *prob* is a *probabilistic edge relation* consisting of elements of the form (l, g, a, υ), where $l \in L$, g is a zone over the clocks X, $a \in \Sigma$, and $\upsilon \in \mathsf{Dist}(2^X \times L)$.

Note that we use no invariant; this is an important condition for the correctness of our techniques. However, invariants can be eliminated (moved to the guards prior to the transition), following classical techniques defined for (probabilistic) timed automata.

We use the following conventions for the graphical representation of probabilistic timed automata: locations are represented by nodes, within which name of the location is written; probabilistic edges are represented by arcs from locations, labeled by the associated guard and action, and which split into multiple arcs, each of which leads to a location and which is labeled by a set of clocks to be reset to 0 and a probability (probabilistic edges which correspond to probability 1 are illustrated by a single arc from location to location).

Example 2. Figure 2a presents an example of a PTA with two clocks x and y. For example, l_0 can be exited whenever $y < 2$; then, with probability 0.4 the target location becomes l_2, resetting x; or with probability 0.6 the target location is l_1, resetting y. The transition from l_2 can be explained similarly.

2.2.2. Semantics of $\mathbb{P}$TAs

A $\mathbb{P}$TA can be interpreted as an infinite TPS. Due to the continuous nature of clocks, the underlying TPS has uncountably many states, and is uncountably branching.

Definition 2 (Concrete semantics of a $\mathbb{P}$TA)**.** Given a $\mathbb{P}$TA $\mathcal{P} = (\Sigma, L, l_0, X, \mathit{prob})$, where $H = |X|$, the concrete semantics of $\mathcal{P}$ is given by the timed probabilistic system $\mathcal{T}_{\mathcal{P}} = (S, s_0, \Sigma, \Rightarrow)$, with

- $S = \{(l, w) \in L \times \mathbb{R}_+^H\}$, $s_0 = (l_0, \vec{0})$
- $((l, w), d, a, \eta) \in \Rightarrow$ if both of the following conditions hold:
  1. time elapse: $\forall d' \in [0, d], (l, w + d') \in S$, and
  2. edge traversal: there exists a probabilistic edge $e = (l, g, a, \upsilon) \in \mathit{prob}$ such that $w + d \models g$ and, for each $l' \in L$ and $\rho \subseteq X$, $\eta(l', [w + d]_\rho) = \upsilon(\rho, l')$.

Note that, due to the fact that we have no invariants, the first condition (time elapse) is always trivially true.

2.3. Parametric interval probabilistic timed automata

In this section, we introduce basic definitions for *(parametric) interval probabilistic timed automata*, that extend (parametric) probabilistic timed automata by providing *intervals* for transition probabilities instead of exact probability values. In the spirit of (parametric) Interval Markov Chains defined in Delahaye (2015); Delahaye et al. (2016), (parametric) interval probabilistic timed automata are used for specifying potentially infinite families (sets) of probabilistic timed automata—those whose exact probability values match the specified intervals—with a finite structure of similar form.

2.3.1. Syntax

Given an arbitrary measurable set S, we call an *interval distribution* over S a function Υ that assigns to each element of S an interval of probabilities $[a, b] \subseteq [0, 1]$. Intuitively, an interval distribution Υ over S represents the set of all distributions $\mu \in \mathsf{Dist}(S)$ that assign to each element $s \in S$ a probability $\mu(s)$ such that $\mu(s) \in \Upsilon(s)$. Formally, let $\mathsf{IntDist}(S)$ denote the set of all interval distributions over S; we define the implementation of an interval distribution as follows.

Definition 3 (Implementation of an interval distribution)**.** Let S be an arbitrary set. Given an interval distribution $\Upsilon \in \mathsf{IntDist}(S)$, $\upsilon \in \mathsf{Dist}(S)$ is an *implementation* of Υ, written $\upsilon \in \Upsilon$ iff, for all $s \in S$, we have $\upsilon(s) \in \Upsilon(s)$.

We now move to the definition of (parametric) interval probabilistic timed automata.

Definition 4. A Parametric Interval Probabilistic Timed Automaton (PI$\mathbb{P}$TA) $\mathcal{PIP}$ is a tuple $(\Sigma, L, l_0, X, \Gamma, \mathbb{I})$, where: *i*) Σ is a finite set of actions, *ii*) L is a finite set of locations, *iii*) $l_0 \in L$ is the initial location, *iv*) X is a finite set of clocks, *v*) Γ is a finite set of parameters, *vi*) $\mathbb{I}$ is an *interval-valued probabilistic edge relation* consisting of elements of the form (l, g, a, Υ), where $l \in L$, g is a guard, $a \in \Sigma$, and $\Upsilon \in \mathsf{IntDist}(2^X \times L)$ is an interval distribution.

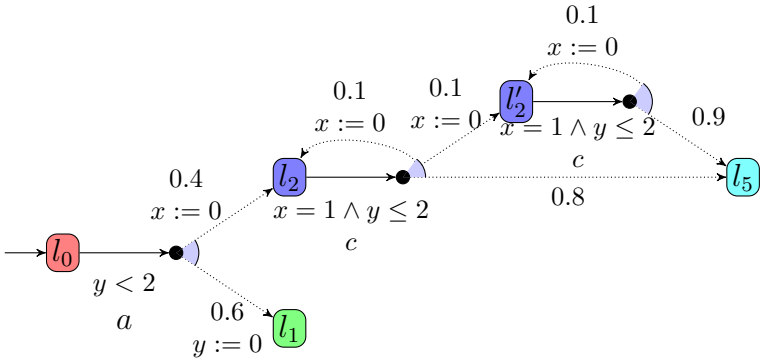
Given a PI$\mathbb{P}$TA $\mathcal{PIP} = (\Sigma, L, l_0, X, \Gamma, \mathbb{I})$ and a parameter valuation v, the *valuation* of $\mathcal{PIP}$ with v, written $v(\mathcal{PIP})$, is an Interval Probabilistic Timed Automaton (I$\mathbb{P}$TA) $\mathcal{IP} = (\Sigma, L, l_0, X, \mathbb{I}')$, where $\mathbb{I}'$ is obtained by replacing within $\mathbb{I}$ any occurrence of a parameter γ with $v(\gamma)$ and removing all transitions (l, g, a, Υ) such that $v(g) \equiv \bot$ (technically, this latter part is not strictly speaking necessary, but it syntactically reduces the model a bit).

Remark that I$\mathbb{P}$TAs are very similar to $\mathbb{P}$TAs: the only difference is that probabilistic edges are labeled with intervals instead of exact probability values.

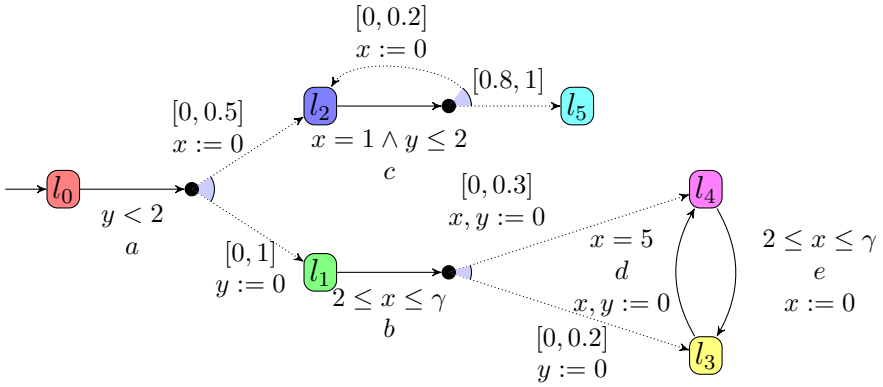
In our graphical representations, when the interval associated with a distribution is reduced to a point (e. g., $[0.5, 0.5]$, we simply represent it using its punctual value (i. e., 0.5). Also, when a distribution is made of a single target location with probability 1, we simply omit the distribution.

Once a parameter valuation is fixed, the resulting I$\mathbb{P}$TA represents a potentially infinite set of $\mathbb{P}$TAs. In order to relate a given I$\mathbb{P}$TA with the $\mathbb{P}$TAs it represents, we use the notion of *implementation* defined hereafter. This notion is similar to the one defined in the context of (parametric) Interval Markov Chains in Delahaye (2015); Delahaye et al. (2016). Remark that a $\mathbb{P}$TA implementing an I$\mathbb{P}$TA needs to conserve the exact same clocks, guards and resets.

Definition 5 (Implementation of an I$\mathbb{P}$TA)**.** Let $\mathcal{P} = (\Sigma, L, l_0, X, prob)$ be a $\mathbb{P}$TA and $\mathcal{IP} = (\Sigma, L', l'_0, X, \mathbb{I})$ be an I$\mathbb{P}$TA.



(a) A $\mathbb{P}$TA



(b) A PI$\mathbb{P}$TA

Figure 2: Examples

We say that $\mathcal{P}$ is an implementation of $\mathcal{IP}$, written $\mathcal{P} \models \mathcal{IP}$, iff there exists a relation $\mathcal{R}_P \subseteq L \times L'$, called an *implementation relation* s. t. $(l_0, l'_0) \in \mathcal{R}_P$ and, whenever $(l, l') \in \mathcal{R}_P$, we have

- $\forall (l, g, a, \upsilon) \in \mathit{prob}, \exists (l', g, a, \Upsilon) \in \mathbb{I}$ s. t. $\upsilon \preceq_{\mathcal{R}_P} \Upsilon$, and
- $\forall (l', g', a, \Upsilon) \in \mathbb{I}, \exists (l, g', a, \upsilon) \in \mathit{prob}$ s. t. $\upsilon \preceq_{\mathcal{R}_P} \Upsilon$,

where $\upsilon \preceq_{\mathcal{R}_P} \Upsilon$ iff $\exists \delta \in \mathsf{Dist}(L \times L')$ s. t.

- $\forall (\rho, l) \in 2^X \times L, \upsilon(\rho, l) > 0 \Rightarrow \sum_{l' \in L'} (\delta(l, l')) = 1$,
- $\forall (\rho', l') \in 2^X \times L', \sum_{l \in L} (\upsilon(\rho', l) \cdot \delta(l, l')) \in \Upsilon(\rho', l')$, and
- $\delta(l, l') > 0 \Rightarrow (l, l') \in \mathcal{R}_P$.

In the above definition, the relation $\mathcal{R}_P$ encodes the pairs of states $(l, l') \in L \times L'$ where l is an *implementation* of l'. On the other hand, the relation $\preceq_{\mathcal{R}_P}$ is a lifting of the relation $\mathcal{R}_P$ to distributions over locations (also called a *coupling*), and therefore represents *compatible* distributions w.r.t. $\mathcal{R}_P$. This notion of satisfaction has been adapted from the notion of "weak weak" satisfaction in the context of Abstract Probabilistic Automata, for which several notions of satisfaction exist. We have chosen this particular notion because it is the most permissive among those presented in Delahaye et al. (2013). For a detailed discussion on this topic, we refer the interested reader to Delahaye et al. (2013).

Given an IPTA, deciding whether the family it represents is nonempty is a nontrivial problem. Indeed, the interval distributions used throughout its structure could represent contradictory constraints on the transition probabilities, therefore preventing any PTA from implementing it.

In the following, we say that a PTA $\mathcal{P}$ has *the same structure* as an IPTA $\mathcal{IP}$ if and only if the underlying directed graph of $\mathcal{P}$ is a subgraph (up to renaming and removal of unreachable states) of the underlying directed graph of $\mathcal{IP}$. This notion trivially extends to PIPTA and other models we use in the rest of the paper.

Definition 6 (Consistency of an IPTA)**.** An IPTA is consistent if it admits at least one implementation.

Example 3. Consider the PIPTA $\mathcal{PIP}$ given in Figure 2b, and containing a single parameter γ. Let v_1 be the parameter valuation such that $v_1(\gamma) = 1$.

In the IPTA $v_1(\mathcal{PIP})$, the transition outgoing from l_1 can never be taken, as its guard becomes $2 \leq x \leq 1$, which is unsatisfiable. Then, it is clear that the PTA $\mathcal{P}$ given in Figure 2a is an implementation of $v_1(\mathcal{PIP})$. We emphasize the fact that location l_2 from $v_1(\mathcal{PIP})$ has been "unfolded" in $\mathcal{P}$, yielding two locations l_2 and l_2'. As a consequence, the underlying structure of $\mathcal{P}$ is not identical to the one of $v_1(\mathcal{PIP})$. Nevertheless, both locations l_2 and l_2' of $\mathcal{P}$ obviously satisfy the original l_2 from $v_1(\mathcal{PIP})$, which allows $\mathcal{P}$ to satisfy $v_1(\mathcal{PIP})$ despite their distinct structures. As a consequence, $v_1(\mathcal{PIP})$ is a consistent IPTA.

An important problem is therefore to decide whether a given IPTA is consistent, which we address in the next section.

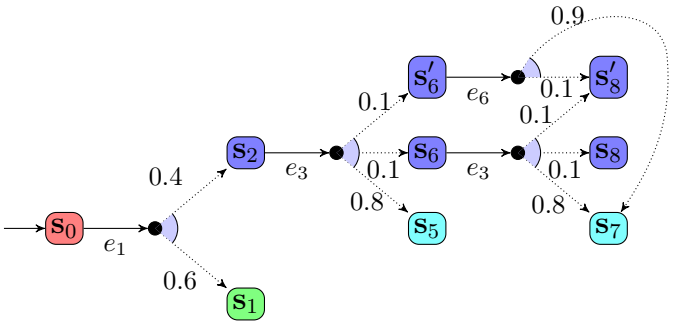
3. The consistency problem for IPTAs

In this section, we address the problem of deciding whether a given IPTA is consistent. Unlike in the context of IMCs, where it is proven that a given IMC is consistent iff it admits an implementation with the same structure, a given IPTA can be consistent but still not admit any implementation that respects its structure. Indeed, the structure of implementations depends on the structure of the zone graph rather than on the structure of the IPTA itself which can be different. Algorithms such as those proposed for deciding consistency of (p)IMCs in Delahaye et al. (2016) therefore cannot be directly adapted to the IPTAs setting as they are dependent on this property.

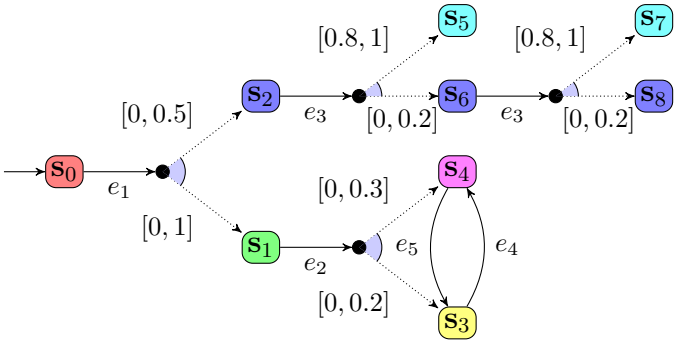
Fortunately, the operational semantics of IPTAs can be expressed in terms of Interval Markov Decision Processes (IMDPs), which are similar to IMCs and satisfy the same structural properties regarding consistency. We therefore propose an algorithm for deciding consistency of IPTAs based on the consistency of their symbolic IMDP semantics. An alternative solution would be to "normalize" IPTAs into special IPTAs where all edges can fire, via a region construction. For the sake of simplicity, we only explore the first solution. We start with preliminary definitions on IMDPs, then formally define the symbolic semantics of IPTAs and finally propose an algorithm for deciding whether a given IPTA is consistent.

3.1. Preliminary definitions

An IMDP is a tuple (S, s_0, Σ, T) where S is a set of states, $s_0 \in S$ is the initial state, Σ is a finite set of actions and $T \subseteq S \times \Sigma \times \mathsf{IntDist}(S)$ is a probabilistic (interval) transition relation.



(a) An example of an MDP



(b) An example of an IMDP

Figure 3: Examples

Example 4. Figure 3b depicts an example of an IMDP. Just as for IℙTAs, when the interval associated with a distribution is reduced to a point (which is not the case here), we simply represent it using its punctual value. When a distribution is made of a single target location with probability 1, we simply omit the distribution (e. g., from $\mathbf{s}_3$ to $\mathbf{s}_4$).

Definition 7 (MDP)**.** An MDP is an IMDP such that for each $(s, a, I) \in T$, and for all $s' \in S$, we have $I(s') = [m, m]$ is a singleton. In addition, for each $(s, a, I) \in T$, we have

$$\sum_{s' \in S} I(s') = 1.$$

Example 5. Figure 3a depicts an example of an MDP.

Definition 8 (Implementation of an IMDP)**.** Let $\mathcal{IM} = (S, s_0, \Sigma, T)$ be an IMDP. Let $\mathcal{M} = (S', s'_0, \Sigma, T')$ be an MDP. We say that $\mathcal{M}$ is an implementation of $\mathcal{IM}$, written $\mathcal{M} \models \mathcal{IM}$, if $\exists \mathcal{R}_M \subseteq S' \times S$ s. t. $(s'_0, s_0) \in \mathcal{R}_M$ and $(s', s) \in \mathcal{R}_M$ if

- $\forall (s', a, \iota) \in T', \exists (s, a, I) \in T$ s. t. $\iota \preceq_{\mathcal{R}_M} I$, and
- $\forall (s, a, I) \in T, \exists (s', a, \iota) \in T'$ s. t. $\iota \preceq_{\mathcal{R}_M} I$,

where $\iota \preceq_{\mathcal{R}_M} I$ iff $\exists \delta \in \mathsf{Dist}(S' \times S)$ s. t.

- $\forall s' \in S', \iota(s') > 0 \Rightarrow \sum_{s \in S}(\delta(s', s)) = 1$,
- $\forall s \in S, \sum_{s' \in S'}(\iota(s') \cdot \delta(s', s)) \in I(s)$, and
- $\delta(s', s) > 0 \Rightarrow (s', s) \in \mathcal{R}_M$.

As for IℙTAs, we say that an IMDP is *consistent* iff it admits at least one implementation.

Example 6. The IMDP given in Figure 3b admits no implementation: indeed, on the (single) transition labeled with e_2, no valuation of the two intervals $[0, 0.3]$ and $[0, 0.2]$ is such that the sum of both valuations is equal to 1. Nevertheless, it could be that the IMDP is still consistent if one assigns a 0-probability on the transition from s_0 to s_1. However, although this would be compatible with the interval $(0 \in [0, 1])$, the second interval (to s_2) does not accept a 1-probability since its probability must be within $[0, 0.5]$.

As said above IMDPs satisfy the same structural property as IMCs concerning implementations: they are consistent iff they admit at least one implementation that respects their structure. This result is formalized in the following lemma.

Lemma 1 (structure of an implementation)**.** *An IMDP $\mathcal{IM}$ is consistent iff there exists an MDP $\mathcal{M}$ with the same structure s. t. $\mathcal{M} \models \mathcal{IM}$.*

Proof. Let $\mathcal{IM} = (S, s_0, \Sigma, T)$ be an IMDP.

One direction of this result is trivial: if there exists an MDP $\mathcal{M}$ with the same structure as $\mathcal{IM}$ s. t. $\mathcal{M} \models \mathcal{IM}$, then $\mathcal{IM}$ is clearly consistent.

The reverse implication is more involved. Assume that $\mathcal{IM}$ is consistent, i. e., there exists an MDP $\mathcal{M} = (S', s'_0, \Sigma, T')$, with no assumption on its structure, such that $\mathcal{M} \models \mathcal{IM}$. We then have to build an MDP $\mathcal{M}^* = (S, s_0, \Sigma, T^*)$ such that $\mathcal{M}^* \models \mathcal{IM}$. Observe that S and s_0 must be identical to that of $\mathcal{IM}$ because they have the same structure.

Let $\mathcal{R}_M$ be the relation witnessing that $\mathcal{M} \models \mathcal{IM}$ and let $f : S \to S' \cup \{\bot\}$ be a function that associates to all states in $\mathcal{IM}$ one of the states from $\mathcal{M}$ that contributes to its implementation, if there is any, and $\bot$ otherwise. Formally, for all $s \in S$, if $f(s) \neq \bot$ then $(f(s), s) \in \mathcal{R}_M$, and whenever there exists $s' \in S'$ such that $(s', s) \in \mathcal{R}_M$, we have $f(s) \neq \bot$.

The transition relation T^* of $\mathcal{M}^*$ is constructed as follows: For each state s that is implemented, i. e., such that $f(s) \neq \bot$, and probabilistic interval transition $(s, a, I) \in T$ in $\mathcal{IM}$, we build a corresponding transition (s, a, ι^I) in $\mathcal{M}^*$ from the transitions in $\mathcal{M}$ that implement (s, a, I). In other words, we pick one of the states that satisfy s (using function f) and mimic its outgoing transitions in $\mathcal{M}^*$. All the other states that satisfy s are simply removed. States that are not implemented do not serve for consistency and are therefore not considered.

Formally, let $(s_1, a, I) \in T$ be a probabilistic interval transition in $\mathcal{IM}$. From Definition 8, we know that there exists $(f(s_1), a, \iota) \in T'$ s. t. $\iota \preceq_{\mathcal{R}_M} I$. According to the definition, there exists at least one function δ that witnesses $\iota \preceq_{\mathcal{R}_M} I$. In the following we pick one such function and name it $\delta_{(\iota,I)}$. The distribution ι^I is then constructed as follows: for all $s_2 \in S$, let $\iota^I(s_2) = \sum_{s' \in S'} \iota(s') \cdot \delta_{(\iota,I)}(s', s_2)$.

By definition of $\delta_{(\iota,I)}$, observe that $\iota^I(s_2) \in I(s_2)$ for all $s_2 \in S$ and that, whenever $\iota^I(s_2) > 0$, $f(s_2) \neq \bot$.

Clearly, $\mathcal{M}^*$ is therefore an implementation of $\mathcal{IM}$, with witnessing relation $\mathcal{R}^*_M$ defined as the identity relation on the set of states $s \in S$ such that

$f(s) \neq \bot$. □

3.2. A symbolic semantics for IPTAs

We equip IPTAs with a symbolic semantics, defined below. Basically, it is inline with the symbolic semantics defined for timed automata in the form of a zone graph, with the addition of probabilistic intervals on the edges; as a consequence, the semantics becomes not an LTS, but an IMDP.

Definition 9 (Symbolic semantics of an IPTA)**.** Given an IPTA $\mathcal{IP} = (\Sigma, L, l_0, X, \mathbb{I})$, the symbolic semantics of $\mathcal{IP}$ is given by the IMDP $(\mathbf{S}, \mathbf{s}_0, \mathbb{I}, T)$, with

- $\mathbf{S} = \{(l, C) \in L \times \mathcal{Z}\}$, $\mathbf{s}_0 = (l_0, (\bigwedge_{1 \leq i \leq H} x_i = 0)^{\nearrow})$, where l is the location and C the associated zone,
- $((l, C), e, I) \in T$ if $e = (l, g, a, \Upsilon) \in \mathbb{I}$ and for all $l' \in L$, for all $\rho \subseteq X$ such that $\Upsilon(\rho, l') > 0$, $C' = \left([C \wedge g]_\rho\right)^{\nearrow}$, and $I((l', C')) = \Upsilon(\rho, l')$.

Given a symbolic state $\mathbf{s} = (l, C)$, we denote by $\mathbf{s}.l$ and $\mathbf{s}.C$ its location and its associated zone (symbolic constraint), respectively.

Observe that, whenever an IPTA has no probabilistic choice, then the IMDP becomes a labeled transition system, and the symbolic semantics matches that of timed automata given in the form of a zone graph (see e. g., Bengtsson and Yi (2003)). It is well-known that the zone graph of a timed automaton can have an infinite number of states; however, applying the classical k-extrapolation (that basically splits zones between a part where the clock constraints are smaller or equal to k and a part where constraints are larger than k, where k is the largest integer-constant in the timed automaton) yields finiteness (see, e. g., Behrmann et al. (2006)). In the following, we apply the classical k-extrapolation to the symbolic constraints of the semantics of an IPTA $\mathcal{IP}$, and therefore the number of states in the IMDP described in Definition 9 is finite. We refer to the symbolic semantics of $\mathcal{IP}$ as the *probabilistic zone graph* of $\mathcal{IP}$.

Remark that the probabilistic zone graph is defined for IPTAs in the form of an IMDP; a PTA can be understood as an IPTA, and its associated zone graph becomes an MDP.

Example 7. The probabilistic zone graph of the PTA in Figure 2a is the MDP given in Figure 3a. The symbolic states $\mathbf{s}_i = (l_i, C_i)$ are expanded in Table 1.

State	Location	C
$\mathbf{s}_0$	l_0	$x = y \wedge x \geq 0$
$\mathbf{s}_1$	l_1	$0 \leq x - y < 2 \wedge y \geq 0$
$\mathbf{s}_2$	l_2	$0 \leq y - x < 2 \wedge x \geq 0$
$\mathbf{s}_5$	l_5	$0 \leq y - x \leq 1 \wedge x \geq 1$
$\mathbf{s}_6$	l_2	$1 \leq y - x \leq 2 \wedge x \geq 0$
$\mathbf{s}'_6$	l'_2	$1 \leq y - x \leq 2 \wedge x \geq 0$
$\mathbf{s}_7$	l_5	$y \geq 2 \wedge y = x + 1$
$\mathbf{s}_8$	l_2	$y \geq 2 \wedge y = x + 2$
$\mathbf{s}'_8$	l'_2	$y \geq 2 \wedge y = x + 2$

Table 1: Description of the states in Figure 3a

3.3. Reconstructing an IPTA from a Probabilistic Zone Graph

It is well-known that, given a timed automata $\mathcal{A}$ and its zone graph, a second timed automaton $\mathcal{A}'$ can be reconstructed from the zone graph, with the same structure as the zone graph, and such that the zone graph of $\mathcal{A}'$ is the same as that of $\mathcal{A}$. We extend this technique here to IPTAs.

The construction. Let $\mathcal{IP} = (\Sigma, L, l_0, X, \mathbb{I})$ be an IPTA; let $\mathcal{IM} = (\mathbf{S}, \mathbf{s}_0, \mathbb{I}, T)$ be its probabilistic zone graph. Let us build a second IPTA $\mathcal{IP}' = (\Sigma, L', l'_0, X, \mathbb{I}')$ as follows.

First, each state of $\mathcal{IM}$ is translated into a location of $\mathcal{IP}'$, i. e., we have $L' = \mathbf{S}$.

Second, the initial location of $\mathcal{IP}'$ is the initial state of $\mathcal{IM}$, i. e., we have $l'_0 = \mathbf{s}_0$.

Third, for each transition $(\mathbf{s}, e, I) \in T$ in $\mathcal{IM}$, with $e = (l, g, a, \Upsilon)$, we create in $\mathcal{IP}'$ a transition $(\mathbf{s}, g, a, \Upsilon')$, where Υ' is defined as follows: for each $\mathbf{s}'$ such that $I(\mathbf{s}') > 0$, then $\Upsilon'(\rho', \mathbf{s}') = I(\mathbf{s}')$, where ρ' is the set of clocks to be reset from $\mathbf{s}.l$ to $\mathbf{s}'.l$ via edge e in $\mathcal{IP}$.

Given an IPTA $\mathcal{IP}$ with probabilistic zone graph $\mathcal{IM}$. We denote by $\mathsf{Reconstruct}(\mathcal{IM})$ the IPTA $\mathcal{IP}'$ reconstructed from $\mathcal{IM}$ following the above technique. Obviously, this construction also applies to PTA, which are just IPTA where intervals are reduced to single points.

An equivalence result. As should be expected, the probabilistic zone graph $\mathcal{IM}'$ of the IPTA $\mathcal{IP}'$ reconstructed from the probabilistic zone graph $\mathcal{IM}$ of a IPTA $\mathcal{IP}$ is equivalent to $\mathcal{IM}$.

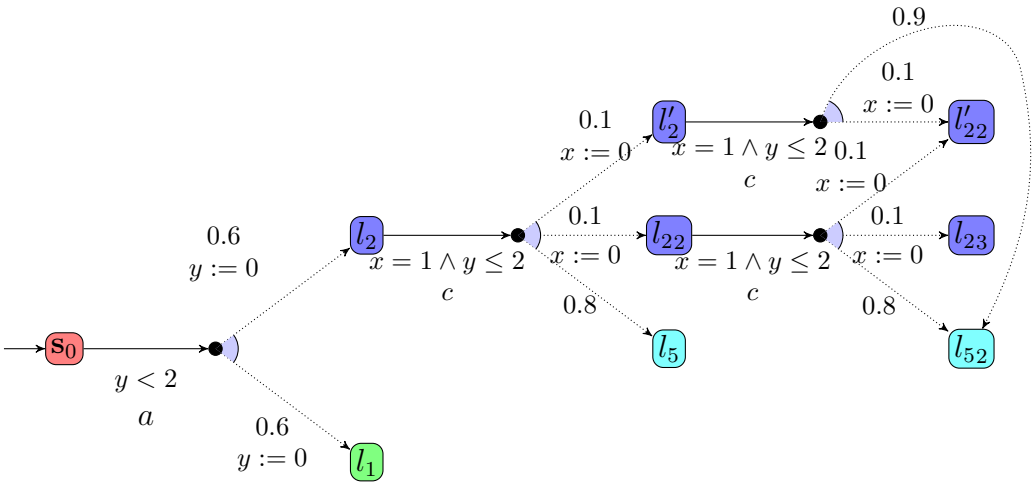


Figure 4: A PTA reconstructed from the probabilistic zone graph in Figure 3a

Proposition 1. *Let $\mathcal{IP}$ be an* IPTA and $\mathcal{IM}$ be its probabilistic zone graph. Let $\mathcal{IP}' = \mathsf{Reconstruct}(\mathcal{IM})$. Let $\mathcal{IM}'$ be the probabilistic zone graph of $\mathcal{IP}'$.

Then $\mathcal{IM}'$ is equivalent to $\mathcal{IM}$ up to location renaming.

Example 8. We apply the above procedure to the probabilistic zone graphs from Figure 3a and Figure 3b. The PTA and IPTA reconstructed from these zone graphs are given in Figure 4 and Figure 5, respectively. Remark that their probabilistic zone graphs are again that of Figure 3a and Figure 3b.

3.4. An algorithm for the consistency of IPTAs

We start with the following observation: by construction, the purpose of the symbolic semantics of IPTAs is to represent, at a lower level of abstraction, the same set of objects. Intuitively, the symbolic IMDP semantics of a given IPTA should therefore be consistent iff the original IPTA is itself consistent. This result is formally proven in Proposition 2.

Proposition 2. *An* IPTA $\mathcal{IP}$ is consistent iff its probabilistic zone graph is consistent.

Proof. Let $\mathcal{IP} = (\Sigma, L, l_0, X, \mathbb{I})$ be an IPTA. Let $\mathcal{IM} = (\mathbf{S}, \mathbf{s}_0, \mathbb{I}, T)$ be the probabilistic zone graph of $\mathcal{IP}$.

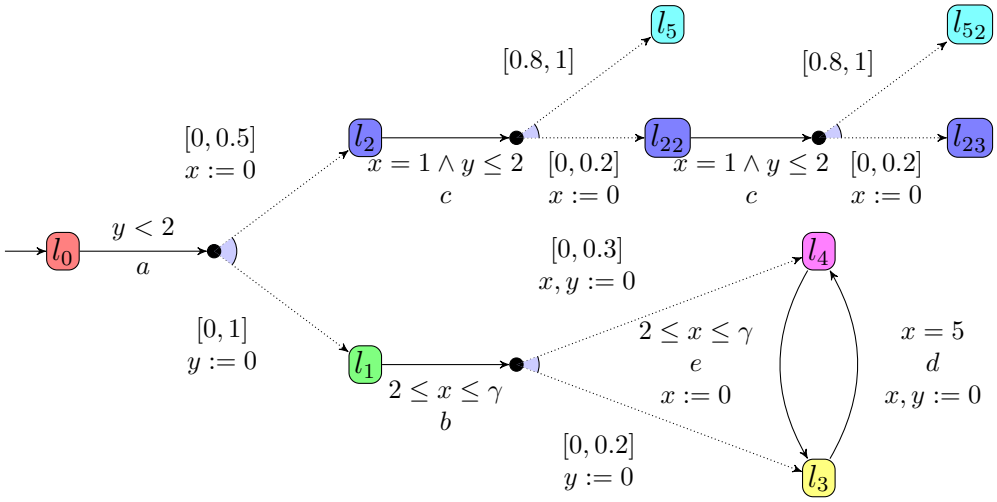


Figure 5: An IℙTA reconstructed from the probabilistic zone graph in Figure 3b

⇒ Assume $\mathcal{IP}$ is consistent, and let us show that its probabilistic zone graph is consistent. From the definition of consistency, there exists a ℙTA $\mathcal{P} = (\Sigma, L', l'_0, X, prob)$ such that $\mathcal{P} \models \mathcal{IP}$, with implementation relation $\mathcal{R}_P$. Let $\mathcal{M} = (\mathbf{S}', \mathbf{s}'_0, prob, T')$ be the probabilistic zone graph of $\mathcal{P}$. Let us show that $\mathcal{M} \models \mathcal{IM}$.

We therefore define a relation $\mathcal{R}_M$, and show that it is an implementation relation. We define $\mathcal{R}_M = \{((l, C), (l', C')) \mid (l, l') \in \mathcal{R}_P \wedge C = C'\}$.

- From Definition 9, the initial state of $\mathcal{M}$ is $\mathbf{s}'_0 = (l'_0, (\bigwedge_{1 \leq i \leq H} x_i = 0)^{\nearrow})$; the initial state of $\mathcal{IM}$ is $\mathbf{s}_0 = (l_0, (\bigwedge_{1 \leq i \leq H} x_i = 0)^{\nearrow})$. Since $\mathcal{P} \models \mathcal{IP}$ then from Definition 5 we have $(l'_0, l_0) \in \mathcal{R}_P$, and therefore $(\mathbf{s}'_0, \mathbf{s}_0) \in \mathcal{R}_M$.
- Let $((l, C), (l', C)) \in \mathcal{R}_M$.
  - Let $((l', C), a, \iota) \in T'$. Since Definition 9, there exists an edge $e' = (l', g, a, \upsilon) \in prob$. Therefore, by $\mathcal{R}_P$,there exists an edge $e = (l, g, a, \Upsilon) \in \mathbb{I}$ such that $\upsilon \preceq_{\mathcal{R}_P} \Upsilon$.
    As a consequence, by Definition 9 and since the guards are the same in $\mathcal{P}$ and $\mathcal{IP}$, there exists $((l, C), a, I) \in T$.
    Moreover, by Definition 9, we have $\iota \preceq_{\mathcal{R}_M} I$, with $\delta_{\mathcal{R}_M}((l', C'), (l, C)) = \delta_{\mathcal{R}_P}(l', l)$ if $C = C'$ and 0 otherwise.

* Similarly, for all $((l, C), a, I) \in T$, there exists $((l', C), a, \iota) \in T'$ such that $\iota \preceq_{\mathcal{R}_M} I$ by $\mathcal{R}_P$ and Definition 9.

Therefore $\mathcal{M} \models \mathcal{IM}$.

$\Leftarrow$ Assume the probabilistic zone graph $\mathcal{IM}$ of $\mathcal{IP}$ is consistent, and let us show that $\mathcal{IP}$ is consistent. From Lemma 1, there exists $\mathcal{M} = (\mathbf{S}, \mathbf{s}_0, \mathbb{I}, T')$ such that $\mathcal{M} \models \mathcal{IM}$ with the same structure as $\mathcal{IM}$, and with implementation relation $\mathcal{R}_M$ (note that $\mathcal{R}_M$ is the identity because they have the same structure).

Let us reconstruct an IℙTA $\mathcal{IP}' = (\Sigma, L', l'_0, X, \mathbb{I}')$ from the probabilistic zone graph $\mathcal{IM}$, using the procedure Reconstruct from Section 3.3. Now, let $\mathcal{P} = (\Sigma, L', l'_0, X, prob)$ with the same structure as $\mathcal{IP}'$, and where $prob$ is obtained by replacing every occurrence of I in $\mathbb{I}'$ by ι taken from T in $\mathcal{M}$. Note that there is a one-to-one correspondence between T' and T since they have the same structure, which is a key point here.

Recall that, during Reconstruct, for each transition $(\mathbf{s}, e, I) \in T$ in $\mathcal{IM}$, with $e = (l, g, a, \Upsilon)$, we create in $\mathcal{IP}'$ a transition $(\mathbf{s}, g, a, \Upsilon')$, where Υ' is defined as follows: for each $\mathbf{s}'$ such that $I(\mathbf{s}') > 0$, then $\Upsilon'(\rho', \mathbf{s}') = I(\mathbf{s}')$, where ρ' is the set of clocks to be reset from $\mathbf{s}.l$ to $\mathbf{s}'.l$ via edge e in $\mathcal{IP}$. Here, we simply replace I with ι, where ι is the distribution corresponding to I in $\mathcal{M}$.

Now, let us show that $\mathcal{P} \models \mathcal{IP}$. Recall from Section 3.3 that the locations in $\mathcal{P}$ are of the form (l, C). We thus define $\mathcal{R}_P = \{((l', C'), l \mid l' = l\}$.

- From the reconstruction Reconstruct, the initial location of $\mathcal{P}$ is (l_0, C_0). Therefore, $((l_0, C_0), l_0) \in \mathcal{R}_P$.
- Let $((l, C), l') \in \mathcal{R}_P$.
  * Let $e = ((l, C), g, a, \upsilon) \in prob$. By Reconstruct, there must exist $((l, C), e, \iota) \in T'$ in $\mathcal{M}$ with $\upsilon(\rho, (l', C')) = \iota((l', C'))$, for all (l', C'), where ρ is the set of clocks to be reset from l to l' via edge e in $\mathcal{P}$. Therefore, from $\mathcal{R}_M$, there exists $((l, C), e, I) \in T$ of $\mathcal{IM}$ s. t. $\iota \preceq_{\mathcal{R}_M} I$.

As a consequence, by the zone graph construction (Definition 9), there exists a transition $(l, g, a, \Upsilon) \in \mathbb{I}$ in $\mathcal{IP}$ such that $I((l', C')) = \Upsilon(\rho, l')$, for all ρ, l', C'.
Let $\delta_{\mathcal{R}_P}$ be such that $\delta_{\mathcal{R}_P}((l', C'), l'') = 1$ if $l' = l''$ and 0 otherwise. Finally, by $\mathcal{R}_M$, we obtain $\upsilon \preceq_{\mathcal{R}_P} \Upsilon$.

- $*$ Similarly, for all $(l, g, a, \Upsilon) \in \mathbb{I}$ in $\mathcal{IP}$, there exists $((l, C), g, a, \upsilon) \in prob$ such that $\upsilon \preceq_{\mathcal{R}_P} \Upsilon$.

Therefore $\mathcal{P} \models \mathcal{IP}$.

□

Given the results presented in Lemma 1 and Proposition 2, deciding whether a given IPTA $\mathcal{IP}$ is consistent can be done by deciding whether its probabilistic zone graph admits at least one implementation that preserves its structure.

Such an algorithm was provided in Delahaye (2015) in the context of IMCs instead of IMDPs. We show how this algorithm can be adapted to our context. As for IMCs, we say that a state is *locally inconsistent* in a given IMDP iff one of its outgoing probabilistic (interval) transitions cannot be implemented, i. e., if there is no distribution that matches the specified intervals. Let $\mathcal{IM} = (S, s_0, \mathbb{I}, T)$ be the IMDP symbolic semantics of a given IPTA. Our algorithm is given in Algorithm 1.

Algorithm 1: Consistency of IMDPs

Let Inc be the set of locally inconsistent states in $\mathcal{IM}$ and $\mathsf{Passed} = \emptyset$.
while $s_0 \notin \mathsf{Passed}$ *and* $\mathsf{Inc} \neq \emptyset$ **do**
  Let $s \in \mathsf{Inc}$ and $\mathsf{Passed} = \mathsf{Passed} \cup \{s\}$.
  Replace all transitions (s', a, I) such that $I(s) \neq [0, 0]$ with (s', a, I') where
  - $I'(s'') = I(s'')$ for all $s'' \neq s$, and
  - $I'(s) = I(s) \cap [0, 0]$

  Update $\mathsf{Inc} \subseteq (S \setminus \mathsf{Passed})$.

Algorithm 1 is based on the following principle: as soon as a locally inconsistent state is detected, it is made unreachable by modifying the incoming interval probabilities to $I(s) \cap [0,0]$. Remark that if 0 is not an admissible transition probability, the inconsistency is transfered to the predecessor states because $I(s) \cap [0,0] = \emptyset$.

In the context of IMCs, it is proven in Delahaye (2015) that this algorithm converges and that the original IMC is consistent iff the initial state is not locally inconsistent in the resulting IMC. The proof from Delahaye (2015) can be trivially adapted to the context of IMDPs.

Proposition 2 together with Algorithm 1 and the above discussion on termination give the following theorem:

Theorem 1. *The consistency problem for* IPTAs is decidable.

4. Consistency-emptiness and synthesis for PIPTAs

We now move to the parametric setting and consider the following two problems:

Consistency-emptiness problem:
INPUT: A PIPTA $\mathcal{PIP}$
PROBLEM: does there exist a parameter valuation v such that $v(\mathcal{PIP})$ is consistent?

Consistency-synthesis problem:
INPUT: A PIPTA $\mathcal{PIP}$
PROBLEM: find all parameter valuations v for which $v(\mathcal{PIP})$ is consistent.

In the following, we first address the consistency-emptiness problem and show that, while this problem is undecidable in the general context of PIPTAs, it becomes decidable for a syntactic subclass (Section 4.1). We then introduce an adaptation of the parametric zone-graph construction for parametric timed automata (Section 4.2), and propose in Section 4.3 a construction for the consistency-synthesis problem. This construction can only be applied when the parametric probabilistic zone-graph construction of the original PIPTA is finite. When this is the case, the set of parameter values that are

synthesized is exactly those that ensure consistency of the resulting IℙTA. We finally address the more general problem of consistent reachability in Section 4.4.

4.1. The emptiness problem

4.1.1. Undecidability in the general case

The undecidability of the consistency-emptiness for PIℙTAs follows from the undecidability of the reachability emptiness for parametric timed automata.

Theorem 2. *The consistency-emptiness for* PIℙTAs is undecidable.

Proof. The reachability emptiness for parametric timed automata (i. e., the existence of at least one parameter valuation for which a given location is reachable) is undecidable (see e. g., Alur et al. (1993); Jovanović et al. (2015); André and Markey (2015); Beneš et al. (2015), and André (2018) for a complete survey). In particular, it is undecidable even without invariant as shown in Alur et al. (1993); Beneš et al. (2015), which is inline with our setting.

We prove our result by reducing from the reachability emptiness for parametric timed automata. Assume a PTA (without probability), with a special goal location. From that goal location, let us add an unguarded transition to a new location for which no implementation exists (for example a single transition labeled with $[0.5, 0.5]$). Hence there exists a parameter valuation for which the underlying IℙTA admits no implementation iff there exists a parameter valuation for which the goal location is reachable—which is undecidable. □

The undecidability of the consistency-emptiness problem rules out the possibility to, in general, compute a solution to the consistency-synthesis problem. In the following, we will still address this computation problem by proposing a synthesis procedure that can be used when the parametric probabilistic zone graph is finite.

4.1.2. A decidability result

Despite the negative result of Theorem 2, we can exhibit a decidability result for a syntactic subclass of PIℙTAs. In Hune et al. (2002), a syntactic subclass, namely lower-bound/upper-bound parametric timed automata (*L/U-PTAs*) is introduced that restricts the use of parameters in parametric timed automata. Basically, in an L/U-PTA, any parameter must be either

always used as an upper-bound (a constraint $x \leq \gamma$ or $x < \gamma$) or always as a lower-bound ($x \geq \gamma$ or $x > \gamma$) in the guards and invariants of the parametric timed automaton. L/U-PTAs benefit from several main decidability results: the EF-*emptiness*, or *reachability-emptiness*, problem ("is the set of parameter valuations for which a given location is reachable empty?") is shown to be decidable in Hune et al. (2002). Then, the infinite acceptance emptiness ("is the set of parameter valuations for which a given set of locations is visited infinitely often along some run empty?") and universality ("is a given set of locations visited infinitely often along some run for all parameter valuations?") have been proved to be decidable for L/U-PTAs with integer-valued parameters in Bozzelli and La Torre (2009). Unavoidability was studied in Jovanović et al. (2015) while liveness and deadlocks were studied in André and Lime (2017) with a thin frontier between decidability and undecidability. Finally, the EF-universality problem was shown to be decidable for L/U-PTAs over rational parameters by André (2018).

In the following, we reuse the concept of lower-bound and upper-bound parameters in the setting of PIℙTAs.

Definition 10 (L/U-PIℙTA)**.** An L/U-PIℙTA is a PIℙTA whose set of parameters is partitioned into lower-bound parameters and upper-bound parameters, where an upper-bound (resp. lower-bound) parameter γ_i is such that, for every guard constraint $x \bowtie z$, we have: $z = \gamma_i$ implies $\bowtie \in \{\leq, <\}$ (resp. $\bowtie \in \{\geq, >\}$).

Example 9. Consider the PIℙTA in Figure 7a. Then it is an L/U-PIℙTA with one upper-bound parameter γ_1 and one lower-bound parameter γ_2.

L/U-PTAs enjoy a well-known monotonicity property recalled in the following lemma (that corresponds to a reformulation of (Hune et al., 2002, Prop 4.2)), stating that increasing upper-bound parameters or decreasing lower-bound parameters can only add behaviors.

Lemma 2 (Hune et al. (2002))**.** *Let $\mathcal{A}$ be an L/U-PTA and v be a parameter valuation. Let v' be a valuation such that for each upper-bound parameter γ^+, $v'(\gamma^+) \geq v(\gamma^+)$ and for each lower-bound parameter γ^-, $v'(\gamma^-) \leq v(\gamma^-)$. Then any run of $v(\mathcal{A})$ is a run of $v'(\mathcal{A})$.*

Remark 1. Unfortunately, we cannot directly use the monotonicity property of L/U-PTAs to prove the decidability of the consistency-emptiness for L/U-PIℙTAs. It could have been handful to consider the IℙTA, say $\mathcal{IP}_{\infty,0}$,

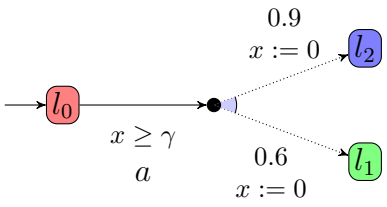


Figure 6: An L/U-PIℙTA inconsistent for all parameter valuations

obtained by replacing every lower-bound parameter (resp. upper-bound parameter) in the guards of a PIℙTA $\mathcal{PIP}$ with ∞ (resp. 0).[1] $\mathcal{IP}_{\infty,0}$ is the most restrictive IℙTA obtained from $\mathcal{PIP}$ (as any other valuation will have more behaviors thanks to Lemma 2). If $\mathcal{IP}_{\infty,0}$ is inconsistent, then any other parameter valuation is clearly inconsistent as well thanks to the monotonicity of L/U-PTAs; however, if $\mathcal{IP}_{\infty,0}$ is consistent, it is not possible to conclude that a (non-infinite) parameter valuation is consistent. In fact, it is easy to exhibit a counter-example for which $\mathcal{IP}_{\infty,0}$ is consistent, but for which $v(\mathcal{PIP})$ is inconsistent for any (non-infinite) valuation v. This is the case of the PIℙTA depicted in Figure 6: when γ is replaced with ∞, the system is stuck forever in l_0, and is therefore consistent. For any (non-infinite) parameter valuation, the system can take the transition labeled with a, which is inconsistent due to the sum of the probabilities.

Still, we show in Theorem 3 below that the consistency-emptiness problem is decidable in the context of L/U-PIℙTAs.

To prove decidability, we use the following reasoning. An L/U-PIℙTA is consistent if we can "block" the inconsistent edges, i. e., those who cannot admit any implementation because the sum of their probabilities cannot be 1. There are two ways of achieving this goal: either set to 0 some of the probabilities on all paths leading to a given inconsistent edge, or tune the timing parameters so as to forbid this edge because the guard can never be satisfied. The first way can be achieved by enumerating all possible combi-

[1] Valuating a parameter with ∞ is achieved as follows: for each upper-bound parameter γ for which $v(\gamma) = \infty$, we delete any comparison of a clock with γ (i. e., the clock constraint becomes the most permissive); for each lower-bound parameter γ for which $v(\gamma) = \infty$, we replace any constraint in which γ appears by false (i. e., the transition labeled by the guard is deleted). Therefore the result of this valuation is an IℙTA as expected.

nations to set to 0 some probabilities. The second way can be achieved by parametric model checking: for a given combination of probabilities set to 0, if we can find at least one valuation for which none of the inconsistent edges is reachable, then we can answer false to the consistency-emptiness problem for L/U-PIℙTAs. Finding at least one such valuation is equivalent to answering no to the EF-universality problem—which is decidable for L/U-PTAs as shown in André (2018). In the following, we explain this reasoning step by step.

We first need a notation, used in the proof of Theorem 3, and later in Section 4.3.

Definition 11 (feasible supports)**.** Given an interval distribution I in an IMDP, let $FS(I)$ denote the *feasible supports* of I i. e., set of sets of target states for which a consistent distribution can be assigned. Formally, $FS(I) = \{S' \subseteq S \mid \exists \mu \in \mathsf{Dist}(S) \text{ s.t. } \forall s \in S : \mu(s) \in I(s) \text{ and } \forall s \in S : \mu(s) > 0 \text{ iff } s \in S'\}$.

That is, each set of states S' is such that there exists a distribution I' for which the probability of reaching each state in S' is not zero, such that this distribution is a punctual distribution, is an implementation of I and is consistent.

Definition 12. An interval distribution I is *inconsistent* if $FS(I) = \emptyset$.

By extension, we say that an edge is inconsistent if its interval distribution is inconsistent.

We also use the same notions for interval distributions in PIℙTAs.

Example 10. In Figure 2b, let Υ_1 be the (unique) interval distribution outgoing from l_1. We have $FS(\Upsilon_1) = \{\}$, as no implementation can make this distribution consistent. That is, Υ_1 is an inconsistent edge. Let Υ_2 be the (unique) interval distribution outgoing from l_2. We have $FS(\Upsilon_2) = \{\{l_5\}, \{l_2, l_5\}\}$.

We now define the set of PIℙTAs obtained by taking all possible combinations of feasible supports.

Definition 13. Given a PIℙTA $\mathcal{PIP}$, let $CombFS(\mathcal{PIP})$ denote the set of all possible PIℙTAs obtained from $\mathcal{PIP}$ by selecting for each interval distribution Υ exactly one element from $FS(\Upsilon)$ when $FS(\Upsilon) \neq \emptyset$, or by keeping the original distribution if $FS(\Upsilon) = \emptyset$.

Intuitively, $CombFS(\mathcal{PIP})$ contains all possible ways to remove transitions by setting some probabilities to 0 while keeping the sum of the other probabilities possibly equal to 1.

Example 11. Consider the PIℙTA $\mathcal{PIP}$ in Figure 2b. $CombFS(\mathcal{PIP})$ contains 4 PIℙTAs. All are such that l_1 has the same outgoing distribution to l_4 and l_3 as in Figure 2b (as FS is empty for this distribution). Two of these 4 PIℙTAs (say 1 and 2) are such that the distribution outgoing from l_0 goes to both l_1 and l_2 (with the same probabilities as in Figure 2b), while two others (say 3 and 4) are such that this distribution is only going to l_1. In addition, two of these 4 PIℙTAs (say 1 and 3) are such that the distribution outgoing from l_2 goes to both l_2 and l_5, while two others (say 2 and 4) are such that this distribution is only going to l_5.

Example 12. Consider now the PIℙTA $\mathcal{PIP}$ in Figure 7a. Then $CombFS(\mathcal{PIP})$ contains the 2 PIℙTAs in Figure 7a (i. e., itself) and in Figure 7b.

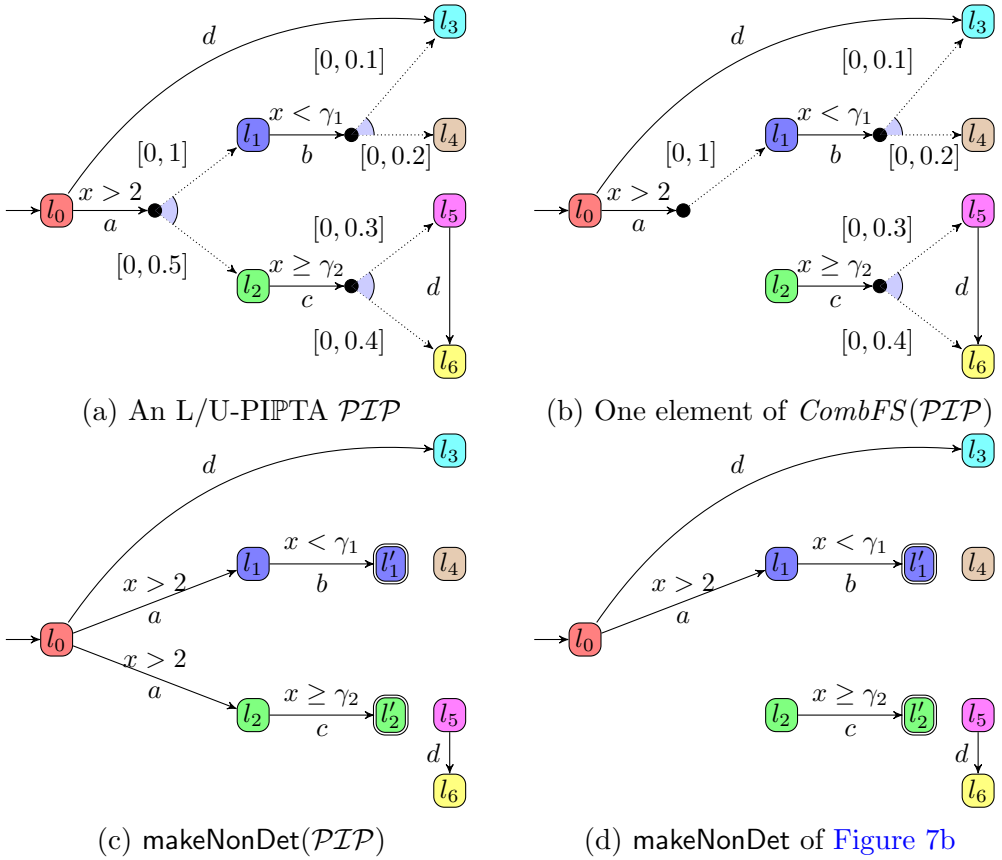
Given an L/U-PTA $\mathcal{A}$ and a subset G of its locations, let us denote by $\mathsf{EFuniv}(\mathcal{A}, G)$ the result of EF-universality for locations G in $\mathcal{A}$, i. e., the answer to the following question: "is the set of valuations v such that at least one location of G is reachable in $v(\mathcal{A})$ universal?" Or put differently, do *all* valuations reach at least one location of G? Recall that EF-universality is decidable for L/U-PTAs as shown in André (2018).

We need two additional notations before introducing our decision procedure. First, given a PIℙTA $\mathcal{PIP}$, let $\mathsf{makeNonDet}(\mathcal{PIP})$ denote the PTA obtained from $\mathcal{PIP}$ by performing the following three operations:

1. for each location l, for each inconsistent edge Υ from l, add a new non-probabilistic transition from l to a fresh location, with the same guard as on Υ;
2. remove all inconsistent edges;
3. replace all probabilistic distributions with non-determinism (see e. g., André et al. (2013)).

Observe that, if $\mathcal{PIP}$ is an L/U-PIℙTA, then $\mathsf{makeNonDet}(\mathcal{PIP})$ is an L/U-PTA. Second, $\mathsf{makeAcc}(\mathcal{PIP})$ returns the set of locations added at step 1.

Beyond transforming the L/U-PIℙTA into a non-probabilistic L/U-PTA, the rationale behind makeNonDet is that we need to test for reachability



(a) An L/U-PIℙTA $\mathcal{PIP}$ (b) One element of $\mathit{CombFS}(\mathcal{PIP})$

(c) makeNonDet($\mathcal{PIP}$) (d) makeNonDet of Figure 7b

Figure 7: Exemplifying *CombFS*, makeNonDet and makeAcc

of an *edge*, which is not possible natively in L/U-PTAs; therefore, we add new locations target of these edges. In addition, we cannot test for the reachability of an arbitrary existing location target of these edges, as they may be reached via other paths too. In Figure 7a, l_3 is reachable via the inconsistent edge outgoing from l_1, but also directly from l_0: only reaching l_3 via l_1 should be avoided, which justifies the creation of l'_1 in Figure 7c.

Example 13. Consider again the PIℙTA $\mathcal{PIP}$ in Figure 7a. Then the L/U-PTA result of makeNonDet($\mathcal{PIP}$) is given in Figure 7c, while makeAcc($\mathcal{PIP}$) is $\{l'_1, l'_2\}$.

We can now give below the main equation to solve consistency-emptiness for L/U-PIℙTAs.

$$\bigwedge_{\mathcal{PIP}' \in CombFS(\mathcal{PIP})} \mathsf{EFuniv}\big(\mathsf{makeNonDet}(\mathcal{PIP}'), \mathsf{makeAcc}(\mathcal{PIP}')\big) \quad (1)$$

The idea is that consistency-emptiness holds for an L/U-PIℙTA if, for each combination of probabilities set to 0 ($CombFS$), for all parameter valuations (EFuniv), some of the locations target of an inconsistent distribution (makeAcc) are always reachable. In other words, there is no way to set some probabilities to 0 and to exhibit some parameter valuations that would avoid an inconsistent distribution.

Note that this procedure can be easily implemented by enumerating all PIℙTAs in $CombFS(\mathcal{PIP})$, replacing probabilities with non-determinism as in makeNonDet, and testing EFuniv on each resulting L/U-PTA using the procedures given in Bozzelli and La Torre (2009); André (2018).

Example 14. Consider again the PIℙTA $\mathcal{PIP}$ in Figure 7a. Recall that $CombFS(\mathcal{PIP})$ is given in Figures 7a and 7b, and $\mathsf{makeAcc}(\mathcal{PIP}) = \{l'_1, l'_2\}$. Therefore, checking consistency-emptiness for $\mathcal{PIP}$ amounts to checking EF-universality of locations $\{l'_1, l'_2\}$ in the L/U-PTAs in Figures 7c and 7d. For the L/U-PTA in Figure 7c, EFuniv gives true, as l'_2 can be reached for any valuation of γ_2 and regardless of γ_1 (it suffices to wait enough time in l_2 so that the guard $x \geq \gamma_2$ becomes enabled). However, for the L/U-PTA in Figure 7d, EFuniv gives false: indeed, while l'_2 is clearly unreachable, l'_1 can only be reached if $\gamma_1 > 2$. Therefore, there exist valuations (typically $\gamma_1 \in [0, 2]$) for which locations $\{l'_1, l'_2\}$ are unreachable.

In fact, it can be shown that, for the PIℙTA $\mathcal{PIP}$ in Figure 7a, the set of valuations for which the IℙTA is consistent is $\gamma_1 \in [0, 2] \wedge \gamma_2 \geq 0$. The idea is to disable the transition to l_2 using probabilities (i. e., assigning 1 to l_1 and 0 to l_2), and to disable the transitions to l_3 and l_4 by tuning γ_1.

Theorem 3. *The consistency-emptiness for L/U-PIℙTAs is decidable.*

Proof. Given an L/U-PIℙTA $\mathcal{PIP}$, we show that Equation 1 holds iff the consistency-emptiness holds for $\mathcal{PIP}$, i. e., no parameter valuation v is such that $v(\mathcal{PIP})$ is consistent.

⇒ Assume Equation 1 holds. Then EF-universality is true for all possible combinations of probabilities set to 0 (given by $CombFS(\mathcal{PIP})$). That is, for each of these potentially consistent models, for any valuation v,

it is always possible to reach at least one of the new locations added by makeNonDet, and therefore one of the inconsistent edges in the original model. Therefore, from Definition 6, $v(\mathcal{PIP})$ is inconsistent for all v. Therefore the consistency-emptiness holds for $\mathcal{PIP}$.

$\Leftarrow$ Assume consistency-emptiness holds for $\mathcal{PIP}$, i. e., no parameter valuation v is such that $v(\mathcal{PIP})$ is consistent. Then there is no way to tune the probabilities and to tune the timing parameters to avoid the inconsistent edges, and therefore to avoid the new locations added by makeNonDet. Then for any $\mathcal{PIP}' \in \mathit{CombFS}(\mathcal{PIP})$, we have that

$$\mathsf{EFuniv}(\mathsf{makeNonDet}(\mathcal{PIP}'), \mathsf{makeAcc}(\mathcal{PIP}'))$$

holds. Then Equation 1 holds.

The result then follows from the decidability of the EF-universality problem for L/U-PTAs proved in André (2018). □

The decidability of the emptiness in Theorem 3 does not necessarily mean that the exact synthesis can be achieved. In fact, we show in the following result that the consistency-synthesis for L/U-PIℙTAs is intractable in practice, as the set of valuations cannot be represented using, e. g., a finite union of polyhedra.

Proposition 3. *The result of the consistency-synthesis for L/U-*PIℙTAs cannot be represented using any formalism for which the emptiness of the intersection is decidable.

Proof. We adapt a reasoning from Jovanović et al. (2015) originally showing that the synthesis for L/U-PTAs is intractable. Let us assume an arbitrary PIℙTA (not necessarily L/U). For each parameter γ_i, let us create a parameter γ_i^l and a parameter γ_i^u (and delete γ_i). Then, let us replace each constraint $x \leq \gamma_i$ with $x \leq \gamma_i^u$, each constraint $x < \gamma_i$ with $x < \gamma_i^u$, each constraint $x \geq \gamma_i$ with $x \geq \gamma_i^l$, each constraint $x > \gamma_i$ with $x > \gamma_i^l$, and each constraint $x = \gamma_i$ with $x \geq \gamma_i^l \wedge x \leq \gamma_i^u$. We obtain an L/U-PIℙTA. Clearly, if $\gamma_i^l = \gamma_i^u$ for all i, then the behavior of the L/U-PIℙTA is identical to that of the original PIℙTA.

Now, assume that the result of the consistency-synthesis for L/U-PIℙTAs can be represented using a formalism for which the emptiness of the intersection is decidable. We can therefore synthesize all valuations for which the

L/U-PIℙTAs is consistent using such a formalism. Then, let us intersect this result with $\bigwedge_{1\leq i\leq M} \gamma_i^l = \gamma_i^u$. Finally, let us check whether this intersection is empty. We are thus able to test the consistency-emptiness of the original PIℙTA—which contradicts Theorem 2. □

In the rest of the section, despite the negative results of Theorem 2 and Proposition 3, we will still attempt to address synthesis for the full class of PIℙTAs.

4.2. A symbolic semantics for PIℙTAs

We equip PIℙTAs with a symbolic semantics, defined below. Basically, it is inline with the symbolic semantics defined for parametric timed automata (see e. g., André et al. (2009); Jovanović et al. (2015)), with the addition of probabilistic intervals on the edges; as a consequence, the semantics becomes not an LTS, but an IMDP. Remark that this is a conservative extension of the symbolic semantics of IℙTA presented in Definition 9.

Definition 14 (Symbolic semantics of a PIℙTA)**.** Given a PIℙTA $\mathcal{PIP} = (\Sigma, L, l_0, X, \Gamma, \mathbb{I})$, the symbolic semantics of $\mathcal{PIP}$ is given by the IMDP $(\mathbf{S}, \mathbf{s}_0, \mathbb{I}, T)$, with

- $\mathbf{S} = \{(l, C) \in L \times \mathcal{Z}\}$, $\mathbf{s}_0 = (l_0, (\bigwedge_{1\leq i\leq H} x_i = 0)^{\nearrow})$,
- $((l, C), e, \Upsilon') \in T$ if there exists $e = (l, g, a, \Upsilon) \in \mathbb{I}$ such that for all $l' \in L$, for all $\rho \subseteq X$ such that $\Upsilon(\rho, l') > 0$, $C' = \left([C \wedge g]_\rho\right)^{\nearrow}$, and $\Upsilon'((l', C')) = \Upsilon(\rho, l')$.

Observe that, whenever a PIℙTA has no probabilistic choice (i. e., is a PTA), then the IMDP becomes a labeled transition system, and the symbolic semantics matches that of parametric timed automata. We refer to the symbolic semantics of $\mathcal{PIP}$ as the *parametric probabilistic zone graph* of $\mathcal{PIP}$.

Just as in parametric timed automata, the number of symbolic states in a PIℙTA can be infinite in general.

In parametric timed automata, the *reachability condition* is the projection onto the parameters of a parametric zone (see Jovanović et al. (2015)). It is well-known that, given a symbolic run of a parametric timed automaton leading to a symbolic state (l, C), there exists an equivalent concrete run iff $\gamma \models C{\downarrow_\Gamma}$ (see e. g., Hune et al. (2002)). Since our definition of zones matches

State	Location	C	$C{\downarrow_\Gamma}$
$\mathbf{s}_0$	l_0	$x = y \wedge x \geq 0 \wedge \gamma \geq 0$	$\gamma \geq 0$
$\mathbf{s}_1$	l_1	$0 \leq x - y < 2 \wedge y \geq 0 \wedge \gamma \geq 0$	$\gamma \geq 0$
$\mathbf{s}_2$	l_2	$0 \leq y - x < 2 \wedge x \geq 0 \wedge \gamma \geq 0$	$\gamma \geq 0$
$\mathbf{s}_3$	l_3	$2 \leq x - y \leq \gamma \wedge y \geq 0$	$\gamma \geq 2$
$\mathbf{s}_4$	l_4	$x = y \wedge x \geq 0 \wedge \gamma \geq 2$	$\gamma \geq 2$
$\mathbf{s}_5$	l_5	$0 \leq y - x \leq 1 \wedge x \geq 1 \wedge \gamma \geq 0$	$\gamma \geq 0$
$\mathbf{s}_6$	l_2	$1 \leq y - x \leq 2 \wedge x \geq 0 \wedge \gamma \geq 0$	$\gamma \geq 0$
$\mathbf{s}_7$	l_5	$y \geq 2 \wedge y = x + 1 \wedge \gamma \geq 0$	$\gamma \geq 0$
$\mathbf{s}_8$	l_2	$y \geq 2 \wedge y = x + 2 \wedge \gamma \geq 0$	$\gamma \geq 0$

Table 2: Description of the states in Figure 3b

that of Hune et al. (2002), this results extends to PIℙTAs in a straightforward manner.

Lemma 3. *Let $\mathcal{PIP}$ be a* PIℙTA. Consider a run in the parametric probabilistic zone graph of $\mathcal{PIP}$ reaching state (l, C). Let v be a parameter valuation. Then, there exists an equivalent run in $v(\mathcal{PIP})$ iff $v \models C{\downarrow_\Gamma}$.

By equivalent run, we mean (just as for parametric timed automata) an identical discrete structure (locations and edges).

Example 15. The parametric probabilistic zone graph of the PIℙTA in Figure 2b is the IMDP given in Figure 3b. The symbolic states $\mathbf{s}_i = (l_i, C_i)$ are expanded in Table 2. In addition, we also give the reachability condition of each state, i. e., the projection onto the parameters of the zone ($C{\downarrow_\Gamma}$).

4.3. A construction for consistency-synthesis for PIℙTAs

Unlike for IℙTAs / IMDPs where inconsistent states can only be avoided by enforcing their incoming probabilities to 0, there are two ways of avoiding inconsistent states in PIℙTAs. Indeed, while imposing a 0 probability to all transitions going to inconsistent states is a safe choice, it is also possible to avoid inconsistent states by cleverly choosing parameter values such that the guards of transitions potentially going to these states are never satisfied.

The construction we propose for synthesizing parameter valuations ensuring consistency of a given PIℙTA is based on the following observation: Since parameters only occur in transition guards, the choice of parameter values cannot interfere with the choice of probability distributions matching (or not) the specified intervals. That comes from the fact that, given a state $\mathbf{s}$,

all successors of this state via a given transition have the same parameter constraint (this would not hold with invariants). As a consequence, states that can be made unreachable through probabilistic choice can be made so regardless of the choice of parameter values.

Notations. We first introduce a few notations to make our construction more compact. Given an IMDP $\mathcal{IM} = (S, s_0, \mathbb{I}, T)$ (representing the semantics of a PIℙTA $\mathcal{PIP}$), let $T_{out}(\mathbf{s})$ denote the set of transitions of source $\mathbf{s}$, i. e., $T_{out}(\mathbf{s}) = \{(\mathbf{s}, e, I) \in T\}$.

Given a transition $(\mathbf{s}, e, I) \in T$, we may want to forbid this transition; recall that the guard (in the original PIℙTA) is the same for all targets, as there is a single guard per interval distribution. As we have no invariants, all target states of a given transition have the same reachability condition (i. e., $C'{\downarrow_\Gamma}$, for a target $\mathbf{s}' = (l', C')$). Therefore, in order to forbid a transition, it suffices to negate the reachability condition of any of the target states of this transition. Let $\mathit{ForbidD}(I)$ denote this result, i. e., $\mathit{ForbidD}(I) = \neg C'{\downarrow_\Gamma}$, where (l', C') is an (arbitrary) target state of I.

Finally, recall that a disjunction over an empty set of clauses is by definition false. Therefore, we use $\bigvee_i K_i$ to denote the union over a set that returns the usual union of K_i for all i in the set if the set is non-empty, or $\bot$ if the set is empty. Similarly, $\bigwedge_i K_i$ denotes the intersection over a set that returns the usual intersection of K_i for all i in the set if the set is non-empty, or $\top$ if the set is empty.

Example 16. Consider the IMDP in Figure 3b, which is the zone graph of the PIℙTA from Figure 2b. Recall that the description of the symbolic states of the IMDP from Figure 3b is given in Table 2. We illustrate the constructions for T_{out} and $\mathit{ForbidD}$ given above.

Clearly, there is only one outgoing transition from state $\mathbf{s}_1$, which is labeled with e_2. As a consequence, we have $T_{out}(\mathbf{s}_1) = (\mathbf{s}_1, e_2, I)$ with $I(\mathbf{s}_3) = [0, 0.2]$, $I(\mathbf{s}_4) = [0, 0.3]$, and $I(\mathbf{s}_i) = [0, 0]$ for $i \notin \{3, 4\}$.

Remark that, as explained above, all the states that are reachable through I have the same reachability condition (given in Table 2). As a consequence, we have $\mathit{ForbidD}(I) = \neg(\gamma \geq 2) \equiv \gamma < 2$.

We now propose a characterization of the set of parameter valuations that ensure consistency of a given PIℙTA under the assumption that its parametric probabilistic zone graph is finite.

Let $\mathcal{PIP}$ be a PIℙTA, and let $\mathcal{IM}$ be its parametric probabilistic zone graph. Assume $\mathcal{IM}$ is finite with state space $\mathbf{S} = \{\mathbf{s}_0, \ldots, \mathbf{s}_n\}$. Consider the formula $cons(c_{s_0}, ..., c_{s_n})$ defined as:

$$(c_{s_0} = \top) \wedge \bigwedge_{\mathbf{s} \in \mathbf{S}} \bigwedge_{(\mathbf{s},e,I) \in T_{out}(\mathbf{s})} \left(\neg c_{\mathbf{s}} \vee \mathit{ForbidD}(I) \vee \bigvee_{S' \in FS(I)} \bigwedge_{\mathbf{s}' \in S' \setminus \{s\}} c_{\mathbf{s}'} \right).$$

Intuitively in this formula the variable $c_{\mathbf{s}}$ represents whether state $\mathbf{s}$ can be reachable in an implementation. Recall that this can only be true if $\mathbf{s}$ is consistent. As a consequence, the formula can only be true when the valuation of the parameters is coherent with the consistent states. Indeed, this formula ensures that the initial state is reachable and that, for any state $\mathbf{s}$ and any outgoing transition of this state, either:

- the source state $\mathbf{s}$ is not reachable ($\neg c_{\mathbf{s}}$), or
- the transition is disabled due to the valuation of the parameters ($\mathit{ForbidD}(I)$), or
- the transition is enabled and thus there must exist a feasible support for which all reachable states are also consistent.

The set of all solutions for the consistency synthesis problem is thus given as the set of solutions (in terms of parameter valuations) of the equation:

$$\bigvee_{(c_{\mathbf{s}_0}, \ldots, c_{\mathbf{s}_n}) \in \{\top, \bot\}^{n+1}} cons(c_{s_0}, \ldots, c_{\mathbf{s}_n}) \tag{2}$$

In the following, this procedure (i. e., solving equation (2)) is called ConstSynth. The intuition behind procedure ConstSynth is that we "guess" the states that will be present in the implementation through the first disjunction (states for which $c_{\mathbf{s}} = \top$), and then verify using $cons(c_{s_0}, \ldots, c_{\mathbf{s}_n})$ that the resulting implementation is well-defined.

Obviously, the empty set of parameter valuations is always a solution to equation (2). Indeed, in this case, one can set $c_{\mathbf{s}_0} = \top$ and $c_{\mathbf{s}} = \bot$ for all $\mathbf{s} \neq \mathbf{s}_0$ and then $\mathit{ForbidD}(I)$ is true for all outgoing transitions of $\mathbf{s}_0$. If this is the only solution, then the PIℙTA $\mathcal{PIP}$ is inconsistent. Otherwise, $\mathcal{PIP}$ is consistent.

We first illustrate our construction on an example and then show that it is sound and complete when the parametric probabilistic zone graph of $\mathcal{PIP}$ is finite.

Example 17. We now apply our construction to the IMDP from Figure 3b. Recall that parameters are non-negative, therefore $\gamma < 0 \equiv \bot$. First observe that either $\mathbf{s}_1$ has to be non-reachable ($c_{\mathbf{s}_1} = \bot$) or its outgoing transition needs to be forbidden, because there is no set $S' \in FS(I)$. As a consequence, we obtain the following constraint: $\neg c_{\mathbf{s}_1} \vee (\gamma < 2)$. There are no constraints for states $\mathbf{s}_5, \mathbf{s}_7$ and $\mathbf{s}_8$ as they have no outgoing transitions. For state $\mathbf{s}_6$, we have the following constraint:

$$\neg c_{\mathbf{s}_6} \vee (\gamma < 0) \vee c_{\mathbf{s}_7} \vee (c_{\mathbf{s}_7} \wedge c_{\mathbf{s}_8}) \equiv \neg c_{\mathbf{s}_6} \vee (\gamma < 0) \vee c_{\mathbf{s}_7} \equiv \neg c_{\mathbf{s}_6} \vee c_{\mathbf{s}_7}$$

Similarly, for state $\mathbf{s}_2$, we obtain:

$$\neg c_{\mathbf{s}_2} \vee (\gamma < 0) \vee c_{\mathbf{s}_5} \equiv \neg c_{\mathbf{s}_2} \vee c_{\mathbf{s}_5}$$

Finally, state $\mathbf{s}_0$ yields the following:

$$\neg c_{\mathbf{s}_0} \vee (\gamma < 0) \vee c_{\mathbf{s}_1} \equiv \neg c_{\mathbf{s}_0} \vee c_{\mathbf{s}_1}$$

Clearly, when put together in equation (2), we obtain the following (after simplifications):

$$\bigvee_{(c_{\mathbf{s}_0},\dots,c_{\mathbf{s}_n}) \in \{\top,\bot\}^{n+1}} (c_{\mathbf{s}_0}) \wedge (c_{\mathbf{s}_1}) \wedge (\gamma < 2) \wedge (\neg c_{\mathbf{s}_2} \vee c_{\mathbf{s}_5}) \wedge (\neg c_{\mathbf{s}_6} \vee c_{\mathbf{s}_7})$$

The solutions are therefore all parameter valuations such that $\gamma < 2$, and can be obtained for all assignments of $c_{\mathbf{s}}$ such that $c_{\mathbf{s}_0} = c_{\mathbf{s}_1} = \top$, $c_{\mathbf{s}_2} \Rightarrow c_{\mathbf{s}_5}$ and $c_{\mathbf{s}_6} \Rightarrow c_{\mathbf{s}_7}$.

We now prove that our construction is indeed correct whenever the parametric probabilistic zone graph of the given PIℙTA $\mathcal{PIP}$ is finite.

Proposition 4 (Correctness)**.** *Let $\mathcal{PIP}$ be a* PIℙTA, and let $\mathcal{IM}$ be its parametric probabilistic zone graph. Assume $\mathcal{IM}$ is finite. Assume that the set of parameter valuations satisfying equation (2) is not empty and let v be such a parameter valuation.

Then $v(\mathcal{PIP})$ is consistent.

Proof. Let v be a solution of equation (2). As a consequence, there must exist an assignment $\kappa_{s_0}, ..., \kappa_{s_n}$ of the variables $c_{\mathbf{s}}$ such that $v(cons(\kappa_{s_0}, \ldots, \kappa_n))$. Moreover, for each state $\mathbf{s}$ such that $\kappa_{\mathbf{s}} = \top$, the following equation is satisfied:

$$\bigwedge_{(\mathbf{s},e,I)\in T_{out}(\mathbf{s})} \left(v(ForbidD(I)) \vee \bigvee_{S'\in FS(I)} \bigwedge_{\mathbf{s}'\in S'\setminus\{s\}} \kappa_{\mathbf{s}'} \right)$$

Therefore, for each $(\mathbf{s}, e, I) \in T_{out}(\mathbf{s})$, either $v(ForbidD(I))$ is true (in this case the transition cannot be taken due to timing parameters and is therefore absent from $v(\mathcal{IM})$), or there exists a distribution ι matching I such that all states $\mathbf{s}'$ such that $\iota(\mathbf{s}') > 0$ are such that $\kappa_{\mathbf{s}'} = \top$.

We can therefore construct an MDP whose states are exactly the states $\mathbf{s}$ such that $\kappa_{\mathbf{s}} = \top$, and whose transitions are given the distributions ι defined above, that clearly satisfies the IMDP $v(\mathcal{IM})$.

By Proposition 2, we can therefore conclude that $\mathcal{PIP}$ is consistent. □

We now show that our construction is complete whenever the parametric probabilistic zone graph of the given PIℙTA $\mathcal{PIP}$ is finite.

Proposition 5 (Completeness)**.** *Let $\mathcal{PIP}$ be a* PIℙTA, and let $\mathcal{IM}$ be its parametric probabilistic zone graph. Assume $\mathcal{IM}$ is finite. Let v be such that $v(\mathcal{PIP})$ is consistent. Then v is a solution of equation (2).

Proof. Since $v(\mathcal{PIP})$ is consistent, there must exist, by Proposition 2 and Lemma 1, an MDP $\mathcal{M}$ with the same structure as $v(\mathcal{IM})$ that satisfies $v(\mathcal{IM})$.

We now propose a valuation of the variables $c_{\mathbf{s}}$ and show that, for this valuation, equation (2) is true.

For all state $\mathbf{s}$ in $\mathcal{IM}$, let $c_{\mathbf{s}} = \top$ if $\mathbf{s}$ is reachable (and present) in $\mathcal{M}$, and $\bot$ otherwise. We now show that the equation $v(cons(c_{\mathbf{s}_0}, \ldots, c_{\mathbf{s}_n}))$ is true. Clearly, we have $c_{\mathbf{s}_0} = \top$, so we just have to show that for all state $\mathbf{s}$ in $\mathcal{IM}$,

$$\bigwedge_{(\mathbf{s},e,I)\in T_{out}(\mathbf{s})} \left(\neg c_{\mathbf{s}} \vee v(ForbidD(I)) \vee \bigvee_{S'\in FS(I)} \bigwedge_{\mathbf{s}'\in S'\setminus\{s\}} c_{\mathbf{s}'} \right) = \mathsf{true}$$

If $\mathbf{s}$ is such that $c_{\mathbf{s}} = \bot$, then this is trivial. Otherwise, let $(\mathbf{s}, e, I) \in T_{out}(\mathbf{s})$. Clearly, since the transition is present in $v(\mathcal{IM})$, we have $v(ForbidD(I)) = \mathsf{false}$. Moreover, since $\mathbf{s}$ is present (and reachable) in $\mathcal{M}$, there is a transition

$(\mathbf{s}, e, \iota)$ in $\mathcal{M}$ such that $\iota \preceq_{\mathcal{R}_M} I$ for the witnessing relation $\mathcal{R}_M$. As a consequence, the set $S' = \{\mathbf{s}' \mid \iota(\mathbf{s}') > 0\}$ is such that $S' \in FS(I)$. Moreover, all states in S' are reachable by construction, thus $\mathbf{s}' \in S' \Rightarrow c_{\mathbf{s}'} = \top$. Therefore, $v(cons(c_{\mathbf{s}_0}, \ldots, c_{\mathbf{s}_n}))$ is true. □

Remark 2. Our construction is based on the parametric probabilistic zone graph. It is sound and complete when this zone graph is finite. However, the resulting equation contains infinite conjunctions and disjunctions when the parametric probabilistic zone graph is infinite, rendering it useless in practice in this case.

However, in practice, one could truncate the parametric probabilistic zone graph up to a certain depth, which would allow computing an approximation of the set of parameter valuations ensuring consistency.

4.4. Parametric Consistent Reachability

A model that is inconsistent is a model that can be considered as ill-formed; therefore, synthesizing valuations for a model to be consistent is an important problem. However, it may not be seen as the final problem a system designer aims at solving. More common problems are reachability, safety, unavoidability, or more complex properties expressed, e. g., using logic formulas.

In this section, we illustrate how consistency synthesis can be combined with existing synthesis algorithms. As a proof of concept, we consider the following parametric consistent reachability synthesis problem:

parametric consistent reachability synthesis problem:
INPUT: A PIℙTA $\mathcal{PIP}$, a set of goal locations G
PROBLEM: find all parameter valuations v for which $v(\mathcal{PIP})$ is consistent and at least one location in G is reachable in $v(\mathcal{PIP})$.

The corresponding emptiness problem, i. e., the emptiness of the valuation set for which a PIℙTA is consistent and at least one goal location is reachable, is clearly undecidable: it suffices to consider a PIℙTA with no probabilities. This gives a PTA, for which reachability emptiness is undecidable: so, clearly, a PTA is always consistent and therefore consistent reachability emptiness reduces to reachability emptiness, which is undecidable, as shown in Alur et al. (1993).

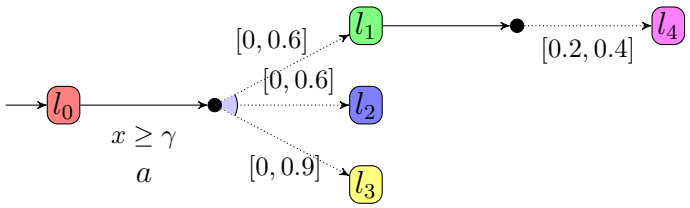


Figure 8: A PIℙTA for which no valuation allows for consistent reachability of l_1

Still, we will propose a method to perform parametric consistent reachability synthesis for PIℙTA; again, this method only works when the parametric probabilistic zone graph is finite.

First, let us rule out the following naive method. We could have considered the PTA obtained from a PIℙTA by removing all probabilities, then we could have synthesized valuations for which reachability of location G is ensured, which can be obtained using the algorithm described in, e. g., Jovanović et al. (2015), and that we will call EFsynth. This gives a constraint K_{reach}. Then, we could synthesize the constraint K_{cons} obtained from ConstSynth. Finally, we could have considered the intersection $K_{reach} \wedge K_{cons}$. However this is not satisfactory (and wrong), as shown in the example below.

Example 18. Consider the PIℙTA in Figure 8. Assume $G = \{l_1\}$. Clearly, l_1 is inconsistent, as its successor has an interval distribution that admits no implementation. l_1 can easily be discarded by assigning it a 0-probability from l_0 while keeping the interval consistent.

On this PIℙTA without probabilities, EFsynth will output $\top$ as any parameter valuation may reach l_1. ConstSynth will also output $\top$. The intersection gives $\top$, while the set of valuations for which l_1 is reachable and the system is consistent is empty.

We propose the following construction, which we adapt from our construction ConstSynth. Recall that in this construction, we use for each state $\mathbf{s}$ a variable $c_{\mathbf{s}}$ that encodes the potential presence of state $\mathbf{s}$ in an implementation (and therefore imposes that this state is consistent). Unfortunately, the presence of such a state in an implementation is not sufficient to ensure that this state is reachable from the initial state. In order to guarantee reachability, we therefore have to add variables and constraints to ConstSynth.

We therefore add new variables $r_{\mathbf{s}}$ for all states $\mathbf{s}$ in the parametric probabilistic zone graph. These variables will be assigned values in $[0, N] \cup \{\infty\}$,

where N is the total number of states of the parametric probabilistic zone graph.

Then, in order to ensure reachability of the goal locations, we add the following constraints:

- $r_{(l,C)} = 0 \iff l \in G$
- for all states $\mathbf{s} = (l, C)$ in the parametric probabilistic zone graph such that $l \notin G$, we impose that either $(r_{\mathbf{s}} = \infty)$ or

$$\bigvee_{(\mathbf{s},e,I)\in T_{out}(\mathbf{s})} \left(\neg ForbidD(I) \wedge \bigvee_{S'\in FS(I)} \left(\bigwedge_{\mathbf{s}'\in S'\setminus\{s\}} c_{\mathbf{s}'} \wedge \bigvee_{\mathbf{s}'\in S'} (r_{\mathbf{s}} = r_{\mathbf{s}'} + 1) \right) \right)$$

Now, solving the conjunction of equation (2) from ConstSynth and the constraints presented above, while imposing that $r_{\mathbf{s}_0} < \infty$ will yield exactly the set of parameter valuations ensuring the consistent reachability of goal locations from G. We call this new procedure ConstEFSynth.

Proposition 6. *Let v be a parameter valuation satisfying the result of* ConstEFSynth*. Then, $v(\mathcal{PIP})$ is consistent and at least one location in G is reachable in $v(\mathcal{PIP})$.*

Proof. First observe that any parameter valuation v obtained through ConstEFSynth needs to satisfy ConstSynth. As a consequence, $v(\mathcal{PIP})$ is consistent. Moreover, the additional constraints provided in ConstEFSynth ensure that whenever $r_{\mathbf{s}} < \infty$, there is an execution of length at most $r_{\mathbf{s}}$ from $\mathbf{s}$ to a state (l, C) such that $l \in G$ in the parametric probabilistic zone graph of $v(\mathcal{PIP})$. Since we impose that $r_{\mathbf{s}_0} < \infty$, G is indeed reachable in $v(\mathcal{PIP})$. □

Example 19. Let us come back to Figure 8. ConstSynth yields the entire set of parameter valuations. By construction, the parametric probabilistic zone graph of this PIℙTA is almost identical to the PIℙTA itself (the states will be $\mathbf{s}_i = (l_i, \gamma \geq 0)$ for all i). However, in ConstEFSynth, it will be impossible to set $r_{\mathbf{s}_1}$ to a finite value as the feasible support of its outgoing transition is empty. As a consequence, ConstEFSynth will yield the empty set of parameter valuations, as expected.

Example 20. Assume now that we would like to synthesize the parameter values ensuring the consistent reachability of l_5 in the PIℙTA given in Figure 2b. Recall that the probabilistic zone graph is given in Figure 3b and the solutions of ConstSynth are given in Example 17. In the process of solving ConstEFSynth on this example, we are required to set $r_{\mathbf{s}_i} = \infty$ for $i \in \{1, 3, 4, 8\}$. We also have to set $r_{\mathbf{s}_5} = r_{\mathbf{s}_7} = 0$. Finally, in addition to the above constraints and those obtained in ConstSynth, ConstEFSynth yields the following:

- For state $\mathbf{s}_6$: either $(r_{\mathbf{s}_6} = \infty)$, or

  $$(\gamma \geq 0) \wedge ((c_{\mathbf{s}_7} \wedge (r_{\mathbf{s}_6} = r_{\mathbf{s}_7} + 1)) \vee (c_{\mathbf{s}_7} \wedge c_{\mathbf{s}_8} \wedge ((r_{\mathbf{s}_6} = r_{\mathbf{s}_7} + 1) \vee (r_{\mathbf{s}_6} = r_{\mathbf{s}_8} + 1))))$$

- For state $\mathbf{s}_2$: either $(r_{\mathbf{s}_2} = \infty)$, or

  $$(\gamma \geq 0) \wedge ((c_{\mathbf{s}_5} \wedge (r_{\mathbf{s}_2} = r_{\mathbf{s}_5} + 1)) \vee (c_{\mathbf{s}_5} \wedge c_{\mathbf{s}_6} \wedge ((r_{\mathbf{s}_2} = r_{\mathbf{s}_5} + 1) \vee (r_{\mathbf{s}_2} = r_{\mathbf{s}_6} + 1))))$$

- For state $\mathbf{s}_0$: either $(r_{\mathbf{s}_0} = \infty)$, or

  $$(\gamma \geq 0) \wedge ((c_{\mathbf{s}_1} \wedge (r_{\mathbf{s}_0} = r_{\mathbf{s}_1} + 1)) \vee (c_{\mathbf{s}_1} \wedge c_{\mathbf{s}_2} \wedge ((r_{\mathbf{s}_0} = r_{\mathbf{s}_1} + 1) \vee (r_{\mathbf{s}_0} = r_{\mathbf{s}_2} + 1))))$$

In the end, we can set $(r_{\mathbf{s}_6} = 1), (r_{\mathbf{s}_2} = 1)$, and $(r_{\mathbf{s}_0} = 2)$ for instance. In this case, we still obtain the same set of parameter valuations as in ConstSynth: all those satisfying $(\gamma < 2)$.

Remark 3. Observe that, for acyclic PIℙTAs (i. e., the underlying graph of which contains no cycle), the answer to the parametric consistent reachability synthesis problem can be effectively computed. Indeed, the procedure presented above consists in a procedure to be solved on a set of states. If that set is finite, the procedure can be effectively solved with an exact result.

This result can also extended to PIℙTAs the symbolic semantics of which is acyclic (i. e., the underlying IMDP contains no cycle). However, it may not be possible to decide whether an arbitrary PIℙTAs has a finite symbolic semantics.

5. Conclusion

In this work, we provided abstractions to reason on systems involving real-time constraints and probabilities: first, by allowing probabilities to

range in some intervals, and, second, by allowing timing constants to be abstracted in the form of parameters. Without parameters, we proposed an approach to decide whether an interval probabilistic timed automaton is consistent, i. e., admits an implementation based on a simulation relation. When adding parameters, the mere existence of a parameter valuation yielding consistency is undecidable. However, when the set of parameters is partitioned between lower-bound parameters and upper-bound parameters, decidability is ensured. We also proposed a procedure to synthesize valuations ensuring consistency for PI$\mathbb{P}$TAs whose parametric probabilistic zone graph is finite, as well as to ensure consistent reachability.

Discussion. We believe our definition of consistency allows for incremental design: one can first define range for probabilities and range for timing parameters. Then, depending on refined design choices, one will assign interval probabilities with punctual values, and valuate timing parameters. Clearly, inconsistent probabilistic distributions can be seen as ill-formed models—just as deadlocks, for examples. One could argue that, contrarily to deadlocks, one could statically detect such situations, or even forbid them statically. However, we see two reasons not to do so. First, we believe that allowing these situations could be used as an additional freedom, that can be then detected and corrected using the methods described in this manuscript. That is, inconsistent intervals do not need to be removed statically if there is another way to remove them (using other probabilities or timing parameters). Second, our work builds on top on works where *parametric* probabilistic bounds can be used (e. g., Delahaye (2015); Delahaye et al. (2016)). In this latter case, the static detection does not work. As our ultimate goal is to reintroduce parametric intervals in the future (see below), we believe our definition of consistency is worth exploring.

Future works. We envision several future works. First, exhibiting subclasses of PI$\mathbb{P}$TAs for which exact synthesis can be achieved is on our agenda. As the use of timing parameters seems critical in our undecidability results, relying on recent works exhibiting decidable subclasses of parametric timed automata, such as bounded integer parameters (see Jovanović et al. (2015)) or reset-parametric timed automata (see André et al. (2016, 2018)), can serve as a first basis for a probabilistic extension.

Finally, we are interested in considering higher-level abstractions of probabilities; notably, using parameters instead of intervals with constant bounds

(as in Delahaye et al. (2016) for parametric interval Markov chains) is of high interest, and makes the notion of consistency even more delicate, as tuning the parametric bounds in an interval may impact the consistency of other probabilistic distributions.

Acknowledgements

We would like to thank the anonymous reviewers for useful comments that helped us to improve the manuscript.

Bibliography

Alur, R., Dill, D. L., Apr. 1994. A theory of timed automata. Theoretical Computer Science 126 (2), 183–235.

Alur, R., Henzinger, T. A., Vardi, M. Y., 1993. Parametric real-time reasoning. In: Kosaraju, S. R., Johnson, D. S., Aggarwal, A. (Eds.), Proceedings of the twenty-fifth annual ACM symposium on Theory of computing. STOC'93. ACM, New York, NY, USA, pp. 592–601.

André, É., 2018. What's decidable about parametric timed automata? International Journal on Software Tools for Technology TransferTo appear.

André, É., Chatain, Th., Encrenaz, E., Fribourg, L., 2009. An inverse method for parametric timed automata. International Journal of Foundations of Computer Science 20 (5), 819–836.

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

André, É., Lime, D., 2017. Liveness in L/U-parametric timed automata. In: Legay, A., Schneider, K. (Eds.), ACSD. IEEE, pp. 9–18.

André, É., Lime, D., Ramparison, M., 2018. Timed automata with parametric updates. In: ACSD. To appear.

André, É., Lime, D., Roux, O. H., 2016. Decision problems for parametric timed automata. In: Ogata, K., Lawford, M., Liu, S. (Eds.), Proceedings of the 18th International Conference on Formal Engineering Methods

(ICFEM'16). Vol. 10009 of Lecture Notes in Computer Science. Springer, pp. 400–416.

André, É., Markey, N., Sep. 2015. Language preservation problems in parametric timed automata. In: Sankaranarayanan, S., Vicario, E. (Eds.), FORMATS. Vol. 9268 of Lecture Notes in Computer Science. Springer, pp. 27–43.

Behrmann, G., Bouyer, P., Larsen, K. G., Pelánek, R., 2006. Lower and upper bounds in zone-based abstractions of timed automata. International Journal on Software Tools for Technology Transfer 8 (3), 204–215.

Beneš, N., Bezděk, P., Larsen, K. G., Srba, J., Jul. 2015. Language emptiness of continuous-time parametric timed automata. In: Halldórsson, M. M., Iwama, K., Kobayashi, N., Speckmann, B. (Eds.), ICALP, Part II. Vol. 9135 of Lecture Notes in Computer Science. Springer, pp. 69–81.

Bengtsson, J., Yi, W., 2003. Timed automata: Semantics, algorithms and tools. In: Desel, J., Reisig, W., Rozenberg, G. (Eds.), Lectures on Concurrency and Petri Nets, Advances in Petri Nets. Vol. 3098 of Lecture Notes in Computer Science. Springer, pp. 87–124.

Bozzelli, L., La Torre, S., 2009. Decision problems for lower/upper bound parametric timed automata. Formal Methods in System Design 35 (2), 121–151.

Collomb-Annichini, A., Sighireanu, M., 2001. Parameterized reachability analysis of the IEEE 1394 root contention protocol using TReX. In: RT-TOOLS.

Delahaye, B., 2015. Consistency for parametric interval markov chains. In: 2nd International Workshop on Synthesis of Complex Parameters, SynCoP 2015, April 11, 2015, London, United Kingdom. Vol. 44 of OASICS. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, pp. 17–32.

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

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

Delahaye, B., Lime, D., Petrucci, L., 2016. Parameter synthesis for parametric interval markov chains. In: Verification, Model Checking, and Abstract Interpretation - 17th International Conference, VMCAI 2016, St. Petersburg, FL, USA, January 17-19, 2016. Proceedings. Vol. 9583 of Lecture Notes in Computer Science. Springer, pp. 372–390.

Doyen, L., 2007. Robust parametric reachability for timed automata. Information Processing Letters 102 (5), 208–213.

Gregersen, H., Jensen, H. E., 1995. Formal design of reliable real time systems. Master's thesis, Department of Mathematics and Computer Science, Aalborg University.

Hune, T., Romijn, J., Stoelinga, M., Vaandrager, F. W., 2002. Linear parametric model checking of timed automata. Journal of Logic and Algebraic Programming 52-53, 183–220.

Jonsson, B., Larsen, K., 1991. Specification and refinement of probabilistic processes. In: LICS. IEEE Computer, pp. 266–277.

Jovanović, A., Kwiatkowska, M. Z., 2014. Parameter synthesis for probabilistic timed automata using stochastic game abstractions. In: Ouaknine, J., Potapov, I., Worrell, J. (Eds.), Proceedings of the 8th International Workshop on Reachability Problems (RP 2014). Vol. 8762 of Lecture Notes in Computer Science. Springer, pp. 176–189.

Jovanović, A., Lime, D., Roux, O. H., 2015. Integer parameter synthesis for timed automata. IEEE Transactions on Software Engineering 41 (5), 445–461.

Kwiatkowska, M. Z., Norman, G., Parker, D., Sproston, J., 2006. Performance analysis of probabilistic timed automata using digital clocks. Formal Methods in System Design 29 (1), 33–78.

Kwiatkowska, M. Z., Norman, G., Segala, R., Sproston, J., 2002. Automatic verification of real-time systems with discrete probability distributions. Theoretical Computer Science 282, 101–150.

Kwiatkowska, M. Z., Norman, G., Sproston, J., 2003. Probabilistic model checking of deadline properties in the IEEE 1394 FireWire root contention protocol. Formal Aspects of Computing 14 (3), 295–318.

Miller, J. S., 2000. Decidability and complexity results for timed automata and semi-linear hybrid automata. In: Lynch, N. A., Krogh, B. H. (Eds.), HSCC. Vol. 1790 of Lecture Notes in Computer Science. Springer, pp. 296–309.