Department of Computer Science

Parameter Synthesis for Probabilistic Timed Automata Using Stochastic Game Abstractions

Aleksandra Jovanovic, Marta Kwiatkowska

CS-RR-14-06

Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD

Abstract. We propose a method to synthesise optimal values of timing parameters for probabilistic timed automata, in the sense that the probability of reaching some set of states is either maximised or minimised. Our first algorithm, based on forward exploration of the symbolic states, can only guarantee parameter values that correspond to upper (resp. lower) bounds on maximum (resp. minimum) reachability probability. To ensure precise reachability probabilities, we adapt the game-based abstraction refinement method. In the parametric setting, our method is able to determine all the possible maximum (or minimum) reachability probabilities that arise for different values of timing parameters, and yields optimal valuations represented as a set of symbolic constraints between parameters.

1 Introduction

Stochastic aspect is very important for modelling numerous classes of systems, such as communication and security protocols, due to component failures, unreliable channels or randomisation. The correctness of such systems can be guaranteed only with some probability. Many of them also operate under timing constraints. In such cases, the probability of a property being true depends on those timing aspects in the system: for example, increasing a certain delay might increase the maximum or minimum probability of reaching an error state.

Automatic synthesis of timing constraints to ensure the satisfaction of a given property has received a lot of attention lately. Its aim is to overcome the disadvantage of model checking, which requires complete knowledge of the system. This is often difficult to obtain, especially in the early design stages, when the whole environment is not yet known. The use of parameters instead of concrete values gives more freedom to the designers. A parametric timed model can specify that a transition is enabled for a time units or that a system can stay in a location for b time units, where a and b are parameters. The goal is then to automatically synthesize the values of model's parameters such that the specification is guaranteed. Parameterisation, however, makes verification more difficult, as most problems become undecidable.

In this paper, we are dealing with the synthesis of timing parameters for probabilistic real-time systems modelled as probabilistic timed automata (PTA) [18]. PTA have been introduced as an extension of timed automata (TA) [1] for modelling and analysing systems which exhibit real-time, nondeterministic and probabilistic behaviour. They are finite-state automata extended with clocks, real-valued variables which increase at the same, constant rate. The edge relation of a PTA differs from that of a TA, in the sense that non-deterministic choice over the set of edges is replaced by a set of discrete probability distributions, each of which is defined over a finite set of edges. A fundamental property of PTA is the maximum/minimum probability of reaching a certain set of states in the model (i.e. the reachability probabilities). These probabilities allow one to express a broad range of properties, from quality of service to reliability, for example, deadline properties: "the maximum probability of an airbag failing to deploy

within 0.02 seconds" or "the minimum probability that a packet is correctly delivered within 1 s". PTA have been successfully used to analyse protocols such as FireWire, Bluetooth, IEEE 802.11, etc. These are embedded in a networked environment and their properties are almost always expressed parametrically, as concrete values make sense only when the network environment is known. It is thus desirable to provide a tool to automatically derive the constraints on parameters for probabilistic systems, so that their correctness is ensured with optimal probability.

Contributions We propose an algorithm for parameter synthesis for PTA based on the symbolic state-space exploration. As the forward approach gives only upper (resp. lower) bounds on maximum (resp. minimum) reachability probability, we adapt the game-based abstraction refinement method. This method has been introduced in [13] for Markov decision processes, and extended in [15] for PTA, for the computation of exact max/min reachability probabilities. As we consider parametric models, these probabilities are not unique and depend on particular parameter valuations. Our algorithm allows us to choose the valuations for which these probabilities are either maximised or minimised, and to synthesise them as a finite set of symbolic constraints on parameters. To the best of our knowledge, this is the first paper dealing with optimal timing parameter synthesis for probabilistic timed automata.

Related work An orthogonal line of work on parameter synthesis for untimed probabilistic models is that of [8], where the authors consider Markov chains and transition probabilities as parameters. Regarding timed systems, parametric timed automata have been introduced in [2] as a means to specify parametric timing constraints. The *reachability-emptiness* problem, which asks whether there exists a parameter valuation such that the automaton has an accepting run, is undecidable. Subsequent research has thus concentrated on finding subclasses for which certain problems would be decidable by restricting the use of parameters [10] or by restricting the parameter domain [11]. In [7], the authors consider fully deterministic networks of timed automata with priorities and parametric guards, and extended MTL with counting formulas. They develop an algorithm based on constraint solving and Monte Carlo sampling to synthesise timing delays. There is little work, however, on timing parameter synthesis for probabilistic real-time systems. In [9], a technique is proposed to approximate parametric rate values for continuous-time Markov chains for bounded reachability probabilities. In [3], the authors apply their *Inverse* method for parameter synthesis for TA to PTA. The method starts from reference parameter values of a TA, and derives the constraints on parameters such that the obtained models are time-abstract equivalent. Time-abstract equivalence preserves untimed properties, and thus the parameter values derived on the non-probabilistic version of the model preserve reachability probabilities. Termination is not guaranteed and the derived constraints are not weakest in general. In [4], the authors consider a fully deterministic parametric model, where the remaining time in a node is unique and given as a parameter, and provide a method to compute the expected time to reach some node as a function of model's parameters.

2 Preliminaries

A discrete probability distribution over a set S is a function $\mu : S \mapsto [0, 1]$, such that $\sum_{s \in S} \mu(s) = 1$ and the set $\{s \mid s \in S \wedge \mu(s) > 0\}$ is finite. By $\mathsf{Dist}(S)$ we denote the set of such distributions. μ_p is a point distribution if $\mu_p(s) = 1$ for some $s \in S$. We now define *Markov decision processes*, a formalism for modelling systems which exhibit both nondeterministic and probabilistic behaviour.

Definition 1 (Markov decision processes). *An MDP is a tuple $\mathcal{M} = (S, s_0, \Sigma, \mathsf{Steps}_{\mathcal{M}})$, where S is a set of states, s_0 is a set of initial states, Σ is a set of actions and $\mathsf{Steps}_{\mathcal{M}} : S \times \Sigma \mapsto \mathsf{Dist}(S)$ is a probabilistic transition function.*

A transition in $\mathcal{M}$ from state s is first made by nondeterministically selecting an action $\delta \in \Sigma$ and then the successor state s' is chosen randomly according to the probability distribution $\mathsf{Steps}_{\mathcal{M}}(s, \delta)$. A *path* is a sequence of such transitions and represents a particular resolution of both nondeterminism and probability. A state s is *reachable* in $\mathcal{M}$ if there exists a path from the initial state of $\mathcal{M}$ to s. A strategy $\mathcal{A}$ is a function from finite paths to distributions which resolves nondeterminism in an MDP. For a fixed strategy $\mathcal{A}$, the behaviour of an MDP is purely probabilistic, and we can define the probability $p_s^{\mathcal{A}}(F)$ of reaching a target set $F \subseteq S$ from s under $\mathcal{A}$. By quantifying over all strategies in $\mathcal{M}$, we can define the minimum and maximum probability of reaching F:

$$p_{\mathcal{M}}^{min}(F) = \inf_{s \in s_0} \inf_{\mathcal{A}} p_s^{\mathcal{A}}(F) \text{ and } p_{\mathcal{M}}^{max}(F) = \sup_{s \in s_0} \sup_{\mathcal{A}} p_s^{\mathcal{A}}(F)$$

These values can be computed efficiently together with the corresponding strategies using, e.g., *value iteration*, which approximates the probability value.

Stochastic 2-player games [5] are turn-based games involving two players and probability. They generalise MDPs by allowing two types of nondeterministic choice, each controlled by a separate player.

Definition 2 (Stochastic games). *A stochastic game is a tuple $\mathcal{G} = (S, (S_1, S_2), s_0, \Sigma, \mathsf{Steps}_{\mathcal{G}})$, where S is a set of states partitioned into sets S_1 and S_2, s_0 is a set of initial states, Σ is a set of actions and $\mathsf{Steps}_{\mathcal{G}} : S_1 \times \Sigma \times S_2 \mapsto 2^{\mathsf{Dist}(S)}$ is a probabilistic transition function.*

S_1 and S_2 represent the sets of states controlled by player 1 and player 2, respectively. The behaviour of a game is as follows: first player 1, in state $s \in S_1$, selects an available action $\delta \in \Sigma$, which takes the game into a state $s' \in S_2$. Player 2 then selects a probability distribution μ from the set $\mathsf{Steps}_{\mathcal{G}}(s, \delta, s')$. Finally, the successor state s'' is chosen according to μ. A resolution of nondeterminism in $\mathcal{G}$ is a pair of strategies σ_1, σ_2 for player 1 and player 2, respectively, under which we can define the probability $p_s^{\sigma_1, \sigma_2}(F)$ of reaching a subset $F \subseteq S$ from state s.

Clocks and parameters. Let $\mathbb{R}$, $\mathbb{R}_{\geqslant 0}$ and $\mathbb{Z}$ be the sets of reals, non-negative reals and integers, respectively. Let X be a finite set. A linear expression on X is an expression of the form $\lambda := k \mid k \cdot x \mid \lambda + \lambda$, where $k \in \mathbb{Z}$ and $x \in X$.

Now let $X = \{x_1, ..., x_n\}$ be a finite set of clock variables. A clock valuation $u : X \mapsto \mathbb{R}_{\geqslant 0}$ is a function assigning a non-negative real number to each $x \in X$.

Let $\mathbf{0}$ be a valuation that assigns 0 to all clocks in X. For any $R \subseteq X$ and any valuation u on X, we write $u[R]$ for the valuation on X such that $u[R](x) = 0$ if $x \in R$ and $u[R](x) = u(x)$ otherwise. For $t \geqslant 0$, $u + t$ denotes the valuation which assigns $(u + t)(x) = u(x) + t$ to all $x \in X$. Let $P = \{p_1, ..., p_m\}$ be a finite set of parameters. A (linear parametric) constraint on $X \cup P$ is an expression of the form $\gamma := x_i \sim c \mid x_i - x_j \sim c \mid \gamma \wedge \gamma$ where $1 \leqslant i \neq j \leqslant n$, $x_i, x_j \in X$, $\sim \in \{<, \leqslant\}$ and c is a linear expression on P. By $\mathcal{C}(X, P)$ we denote the set of such parametric constraints and by $\mathcal{C}'(X, P)$ we denote the set of (non-diagonal) constraints of the form: $\gamma' := x_i \sim c \mid \gamma' \wedge \gamma'$. For any valuation v on P and any linear constraint γ on $X \cup P$, $v(\gamma)$ is the linear constraint on X obtained by replacing each parameter $p \in P$ by the (concrete) value $v(p)$. Given some arbitrary order on $X \cup P$, a valuation can be viewed as a real-valued vector of size $|X \cup P|$. The set of valuations satisfying some linear constraints is then a convex polyhedron of $\mathbb{R}^{|X \cup P|}$. A zone is a polyhedron defined only by conjunctions of the constraints of the form $x - y \sim c$ or $x \sim c$ with $x, y \in X$, $c \in \mathbb{Z}$ and $\sim \in \{<, \leqslant\}$. If v is a valuation on both clocks and parameters $X \cup P$ (as v is used throughout the paper, unless specified otherwise) then by $v_{|P}$ (resp. $v_{|X}$) we denote the projection of v onto P (resp. X). We now give a formal definition of *Parametric Probabilistic Timed Automata* (PPTA), which are PTA extended with timing parameters.

Definition 3 (PPTA). *A PPTA is a tuple $\mathcal{P} = (L, l_0, X, P, \Sigma, \mathsf{prob}, \mathsf{Inv})$ where: L is a finite set of locations; $l_0 \in L$ is the initial location; X is a finite set of clocks; P is a finite set of parameters; Σ is a finite set of actions; $\mathsf{prob} : L \times \Sigma \times \mathcal{C}(X, P) \mapsto \mathsf{Dist}(2^X \times L)$ is a probabilistic transition function; and $\mathsf{Inv} : L \mapsto \mathcal{C}'(X, P)$ is a function that assigns an invariant to each location.*

For any rational valuation v on P, the structure $v(\mathcal{P})$ obtained from $\mathcal{P}$ by replacing every constraint γ by $v(\gamma)$ is a PTA. The behaviour of a PPTA $\mathcal{P}$ is described by the behaviour of all PTA $v(\mathcal{P})$ obtained by considering all possible parameter valuations. A (concrete) state of $v(\mathcal{P})$ is a pair $(l, u) \in L \times \mathbb{R}_{\geqslant 0}^X$ such that the clock valuation u satisfies the invariant (notation $u \models v(\mathsf{Inv}(l))$). A transition in the semantics of $v(\mathcal{P})$ is a timed-action pair (t, δ). In each state certain amount of time $t \in \mathbb{R}_{\geqslant 0}$ can elapse, as long as $u + t \models v(\mathsf{Inv}(l))$. Time elapse is followed by the choice of an action $\delta \in \Sigma$, for which the set of clocks R to reset and successor locations l' are selected randomly according to the probability distribution $\mathsf{prob}(l, \delta, \gamma)$. The action δ can only be taken once its constraint $v(\gamma)$ (called guard) is satisfied by the current clock valuation. Each element $(R, l') \in 2^X \times L$, such that $\mathsf{prob}(l, \delta, \gamma)(R, l') > 0$, is called an edge and the set of all such edges, denoted $\mathsf{edges}(l, \delta, \gamma)$, is assumed to be an ordered list $\langle e_1, ..., e_n \rangle$. We now formally define the semantics of a PPTA under a parameter valuation v.

Definition 4 (Semantics of a PPTA). *Let $\mathcal{P} = (L, l_0, X, P, \Sigma, \mathsf{prob}, \mathsf{Inv})$ be a PPTA and v be a $\mathbb{R}$-valuation on P ($v : P \mapsto \mathbb{R}$) . The semantics of $v(\mathcal{P})$ is given by the infinite-state MDP $\mathcal{M}_{v(\mathcal{P})} = (Q, q_0, \mathbb{R}_{\geqslant 0} \times \Sigma, \mathsf{Steps}_{\mathcal{M}_{v(\mathcal{P})}})$ where:*

    - $Q = \{(l, u) \in L \times X \mapsto \mathbb{R}_{\geqslant 0} \mid u \models v(\mathsf{Inv}(l))\}$, $q_0 = (l_0, \mathbf{0})$

- $\mathsf{Steps}_{\mathcal{M}_{v(\mathcal{P})}}((l,u),(t,\delta)) = \mu$ *iff* $\exists (R,l') \in \mathsf{edges}(l,\delta,\gamma)$ *such that* $u+t \models v(\gamma) \wedge u+t' \models \mathsf{Inv}(l)$ *for all* $0 \leqslant t' \leqslant t$, *and for any* $(l',u') \in Q$:
$$\mu(l',u') = \sum\{| \, \mathsf{prob}(l,\delta,\gamma)(R,l') \mid R \in 2^X \wedge u' = (u+t)[R] \, |\}$$

Note that the definition of μ involves summation over the cases in which multiple clock resets result in the same target state (l',u'), expressed as a multiset, since some of the probabilities might be the same.

We study the *optimal timing parameter synthesis* problem for PPTA, i.e., automatically finding values of parameters such that the probability (either maximum or minimum) of reaching a certain set of locations is *optimised*. For example, in the case of property "the maximum probability of an airbag failing to deploy", we would want to choose the timing parameters that minimise this probability value. On the other hand, we would want to maximise "the maximum probability that the protocol successfully terminates". We therefore call such maximisation/minimisation - an optimisation, and refer as so hereinafter.

3 Synthesis with Forward Reachability

A naive approach to parameter synthesis for PTA is to restrict parameter values to bounded intervals of integers (or rationals that can be scaled to integers) and perform verification for each such (non-parametric) model using a probabilistic model checker, e.g. PRISM [16]. However, this approach is shown to be inefficient for (non-probabilistic) TA compared to symbolic techniques, especially when the sets of possible parameter values are large [11]. This is why we aim to formulate a symbolic algorithm for deriving constraints on parameters that ensure the optimisation of some reachability probability in the model. For the symbolic exploration of the state-space, we use the notion of *parametric symbolic state* and forward symbolic operations on valuation sets given below, defined in [11].

Definition 5 (Parametric symbolic state). *A (parametric) symbolic state of a PPTA $\mathcal{P}$, with set of clocks X and set of parameters P, is a pair $S = (l, \zeta)$ where l is a location of $\mathcal{P}$ and ζ is a set of valuations v on $X \cup P$.*

- *future* (time successors): $\zeta^{\nearrow} = \{v' \mid v \in \zeta \wedge v'(x) = v(x) + d, d \geqslant 0$ if $x \in X; v'(x) = v(x)$ if $x \in P\}$
- *reset of clocks* in $R \subseteq X$: $\zeta[R] = \{v[R] \mid v \in \zeta\}$
- *successor by edge* $e = (R, l')$ in the distribution $\mathsf{prob}(l,\delta,\gamma)$: $\mathsf{Succ}((l,\zeta),e) = (l', (\zeta \cap \gamma)[R]^{\nearrow} \cap \mathsf{Inv}(l'))$
- *initial symbolic state:* $Init(\mathcal{P}) = (l_0, \{v \in \mathbb{R}^{X \cup P} \mid v_{|X} \in \{\mathbf{0}_X\}^{\nearrow} \wedge v(\mathsf{Inv}(l_0))\})$.

The sets of valuations of all reachable symbolic states of a PPTA are convex polyhedra [10], since the set of valuations of the initial symbolic state is a convex polyhedron and all the operations preserve convexity.

Forward reachability exploration The forward exploration, which builds an MDP-based abstraction of a given PTA [18], is an extension of the well-known zone-based forward reachability algorithm, ubiquitous for model-checking TA, and implemented in tools such as UPPAAL [19] and KRONOS [6]. This algorithm performs the exploration of the state-space by successively computing symbolic states using Succ, starting from the initial state. For probabilistic models, on-the-fly techniques are not used, as the goal is to compute the probability of reaching a state, instead of just checking the existence of a path.

In Fig. 1 we present our extension of the forward reachability algorithm from [18] to *parametric* probabilistic timed automata. It takes a PPTA $\mathcal{P}$ and some subset of its locations F as input, and returns the *reachability graph* (Sym, $Trans$). Sym is the set of all reachable parametric symbolic states S of the model and $Trans$ is the set of symbolic transitions. $Waiting$ is the set of those symbolic states which have not yet been explored. As long as there are symbolic states unexplored ($Waiting \neq \varnothing$), successor states are computed for each possible edge using Succ operator. Each symbolic transition $T \in Trans$ is of the form $T = ((l, \zeta), \delta, \langle (l_1, \zeta_1), ..., (l_n, \zeta_n) \rangle)$, where $n = |\mathsf{edges}(l, \delta, \gamma)|$. A symbolic transition T induces probability distribution μ_T over symbolic states Sym. For any $(l', \zeta') \in Sym$: $\mu_T(l', \zeta') = \sum \{ | \mathsf{prob}(l, \delta, \gamma) e_i \mid e_i \in \mathsf{edges}(l, \delta, \gamma) \wedge \zeta_i = \zeta' | \}$.

Using these distributions, the algorithm BUILDMDP(Sym, $Trans$) constructs an MDP similarly to that of [18] for PTA, which can then be analysed to compute the reachability probabilities. For PTA, and therefore for PPTA, this approach only gives upper (resp. lower) bounds on maximum (resp. minimum) reachability probability in the model. This is because the reachability graph is too coarse to preserve precise time the actions can be taken, and thus constructs an over-approximation of the possible strategies.

Let us highlight the differences between our algorithm and its non-parametric counterpart from [18]. In the non-parametric case, all the symbolic states (l, ζ) containing some location $l \in F$ are collected into a set $Reached$. Then, in the constructed MDP, the max. (or min.) probability of ending up in $Reached$ is calculated. In our setting, we are interested in finding the *optimal* parameter valuations (that maximise or minimise some reachability probability). We thus need to keep separate those symbolic states containing different parameter valuations and calculate the max/min reachability probability for each one. We divide the set $Reached$ into subsets $Reached_i$, each of which contains the symbolic states (l_i, ζ_i) with equivalent parameter values (obtained by projection onto parameters $\zeta_{i|P}$). Another difference arises when building symbolic transitions $Trans$. This follows from the property of TA (and therefore PTA) proven in [10], which states that weakening (resp. strengthening) the guards in any TA $\mathcal{T}$, e.g decreasing lower and increasing upper (resp. increasing lower and decreasing upper) bounds on clocks, yields an automaton whose reachable states include (resp. are subset of) those of $\mathcal{T}$. We therefore add, for any two symbolic states $(l_i, \zeta_i), (l_j, \zeta_j) \in Sym$ which satisfy $\zeta_{i|X} \subseteq \zeta_{j|X} \wedge \zeta_{i|P} \subseteq \zeta_{j|P} \wedge l_i = l_j$, a transition (point distribution) from (l_j, ζ_j) to (l_i, ζ_i), in order to obtain the correct

probabilities in the MDP. By $\{Reached_i\}_{|P}$ in Fig. 1, we denote the parameter values contained in $Reached_i$.

$\textsc{ParReach}(\mathcal{P}, F)$

$Sym := \varnothing;\ Trans := \varnothing;\ Reached := \varnothing;\ Waiting := \{Init(\mathcal{P})\};\ n := 0;\ Reached_0 := \varnothing$
while $Waiting \neq \varnothing$
    choose and remove (l, ζ) **from** $Waiting$
    $Sym := Sym \cup \{(l, \zeta)\}$
    for $\delta \in \Sigma$ **such that** $\mathsf{edges}(l, \delta, \gamma) \neq \varnothing$
        for each $e_i \in \mathsf{edges}(l, \delta, \gamma) = \langle e_1, ..., e_n \rangle$
            $(l'_i, \zeta'_i) := \mathsf{Succ}((l, \zeta), e_i)$
            if $(l'_i, \zeta'_i) \notin Sym \wedge \zeta'_i \neq \varnothing \wedge l'_i \notin F$ **then** $Waiting := Waiting \cup \{(l'_i, \zeta'_i)\}$
            else if $(l'_i, \zeta'_i) \notin Sym \wedge \zeta'_i \neq \varnothing$ **then** $Reached := Reached \cup \{(l'_i, \zeta'_i)\}$
        $Trans := Trans \cup \{((l, \zeta), \delta, \langle (l_1, \zeta_1), ..., (l_n, \zeta_n) \rangle)\}$
$//Additional\ transitions\ from\ a\ state\ to\ its\ subsets$
for each $(l, \zeta) \in Sym$
    if $\exists (l', \zeta') \in Sym$ **such that** $l = l' \wedge \zeta_{|X} \subseteq \zeta'_{|X} \wedge \zeta_{|P} \subseteq \zeta'_{|P}$ **then**
        $Trans := Trans \cup \{(l', \zeta'), \varnothing, \langle (l, \zeta) \rangle\}$
$//Divide\ Reached\ into\ subsets\ Reached_i\ according\ to\ different\ parameter\ valuations$
for each $(l, \zeta) \in Reached$
    if $(\zeta_{|P} = \{Reached_i\}_{|P}$ for some $Reached_i$ where $i \in [0..n])$ **then**
        $Reached_i := Reached_i \cup \{(l, \zeta)\}$
    else $Reached_n := Reached_n \cup \{(l, \zeta)\};\ n++;$
return $(Sym, Trans)$
// $\textsc{BuildParMDP}(Sym, Trans)$
$sym_0 = \{(l, \zeta) \in Sym \mid l = l_0\}$
for $(l, \zeta) \in Sym$ **and** $T \in Trans(l, \zeta)$
    $\mathsf{Steps}_{\mathcal{M}}((l, \zeta), T) := \mu_T$
return $\mathcal{M} = (Sym, sym_0, Trans, \mathsf{Steps}_{\mathcal{M}})$

Fig. 1. Parametric forward reachability and construction of the corresponding MDP

Example 1. Let us consider a PPTA shown in Fig. 2, which represents a communication protocol operating over a lossy channel. We have simplified the model from [18] and added a parameter b to the guard of the first transition. The protocol starts with the delivery of new data which resets both clocks x and y. The data are retained at least b time units before being send onto the medium, which resets clock x to zero. The medium then sends the data to the receiver, and the attempt is successful with probability 0.65. The medium re-sends the data at the time varying between 3 and 7 time units. If exactly 7 time units are elapsed since the delivery of the data, the system aborts. We are interested in the values of the parameter b which maximise the probability of the medium successfully *sending* the data (reaching location l_2). The MDP constructed from the reachability graph is shown in Fig. 3. There are three symbolic states holding l_2 with different parameter valuations, $Reached_1 = \{(l_2, x = y \wedge b \leqslant 1)\}$, $Reached_2 = \{(l_2, x = y \wedge b \leqslant 3)\}$ and $Reached_3 = \{(l_2, x = y \wedge b \leqslant 5)\}$. Using PRISM, we calculated maximal probabilities of reaching those states in the MDP: $p^{max}(\lozenge Reached_1) = 0.65$,

$p^{max}(\lozenge Reached_2) = 0.8775$, and $p^{max}(\lozenge Reached_3) = 0.957125$, where $\lozenge\phi$ means that ϕ must hold eventually. If we want to *maximise* the probability of reaching l_2, it is clear that we should choose $b \leqslant 1$. Note that, if we were to consider reachability via action *time_out* or *abort*, instead of *send*, we would be interested in actually *minimising* the maximum probability of reaching l_2. In that case we would choose $3 < b \leqslant 5$.
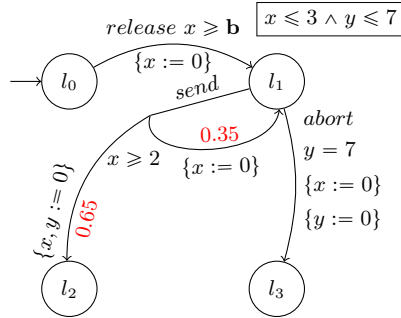

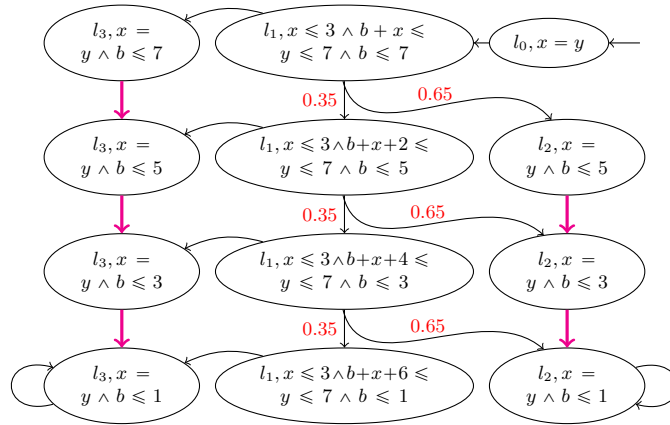**Fig. 2.** An example of a Parametric Probabilistic Timed Automata

Fig. 3. MDP for PPTA of Fig. 2

The forward reachability algorithm provides only upper (resp. lower) bound on the max. (resp. min.) reachability probability. In Example 1, this method actually gives the correct values, but consider now the automaton of Fig. 4,
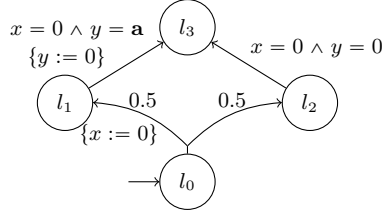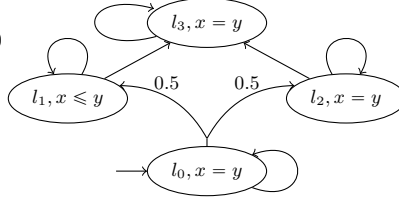
Fig. 4. Another example of a PPTA   **Fig. 5.** MDP for PPTA of Fig. 4

inspired by [18]. The probability of reaching l_3 obtained using forward approach (the resulting MDP is shown in Fig. 5) is 1, regardless of the value of a. By careful inspection, we observe that the max. probability is 1 only if $a = 0$ (when the transition from l_0 is taken at $x = y = 0$), and otherwise it is at most 0.5.

Theorem 1. *For a PPTA $\mathcal{P}$ and a subset of its locations F, if $(Sym, Trans) =$ PARREACH$(\mathcal{P}, F)$ and $\mathcal{M} =$ BUILDMDP$(Sym, Trans)$, then:*

- $p_{\mathcal{M}}^{min}(Reached) \leqslant p_{\mathcal{P}}^{min}(F)$ and $p_{\mathcal{M}}^{max}(Reached) \geqslant p_{\mathcal{P}}^{max}(F)$;

- if $\mathcal{M}$ gives the precise reachability probabilities in $\mathcal{P}$ and if some $(l_k, \zeta_k) \in$ Reached has the optimum (max. or min.) reachability probability, among all $(l_j, \zeta_j) \in$ Reached, then $\{\zeta_{k|P} \backslash (\bigcup_{\forall j \neq k, l_j \in F} \zeta_{j|P})\}$ is the solution to the optimal parameter synthesis problem.

The *reachability-emptiness* problem for parametric timed automata is undecidable [2], and the algorithm for forward reachability exploration for this model might not terminate [11,10]. Since our algorithm for PPTA can be viewed as its extension, termination cannot be guaranteed either.

To resolve the limitation of the forward approach, namely, that it can only compute bounds on the reachability probabilities, in the next section we adapt the *game-based abstraction refinement* method from [15] to synthesise the optimal timing parameter values for PPTA. We choose this approach as it can compute precise min. and max. probabilities and is shown to perform better then the alternative model checking technique for PTA, digital clocks [17].

4   Synthesis with Game-based Abstraction Refinement

In [14], stochastic two-player games are used as abstractions for MDPs. In such a game, the two players represent nondeterminism introduced by the abstraction (player 1) and nondeterminism from the original model (player 2). By quantifying over all possible strategies for players 1 and 2, we can obtain both the lower bound (lb) and upper bound (ub) on either maximum or minimum reachability probability in the original MDP. If a game $\mathcal{G}$ is constructed from an MDP $\mathcal{M}$ using the approach from [14], where F is a subset of states of $\mathcal{M}$, we have:

$p_{\mathcal{G}}^{lb,min}(F) \leqslant p_{\mathcal{M}}^{min}(F) \leqslant p_{\mathcal{G}}^{ub,min}(F)$ and $p_{\mathcal{G}}^{lb,max}(F) \leqslant p_{\mathcal{M}}^{max}(F) \leqslant p_{\mathcal{G}}^{ub,max}(F)$.

In case of maximum probability we have: $p_{\mathcal{G}}^{lb,max}(F) \stackrel{\text{def}}{=} \sup_{s \in s_0} \inf_{\sigma_1} \sup_{\sigma_2} p_s^{\sigma_1, \sigma_2}(F)$

and $p_{\mathcal{G}}^{ub,max}(F) \stackrel{\text{def}}{=} \sup_{s \in s_0} \sup_{\sigma_1} \sup_{\sigma_2} p_s^{\sigma_1,\sigma_2}(F)$. Using similar techniques to value iteration for MDPs [5], these probabilities can be efficiently approximated, together with the corresponding strategy pairs which achieve them.

In [15], the concept of gamed-based abstractions is used for PTA in order to compute the maximum and minimum reachability probabilities. The method starts from the MDP obtained via forward reachability algorithm, and subsequently builds and refines the stochastic game abstraction. In this section, we generalise this method by taking into account timing parameters.

Game-based abstraction for PPTA The game-based abstraction is constructed by analysing transitions outgoing from each location in a PPTA. The transitions are divided into subsets according to the common part of the symbolic state in which they are enabled. This analysis is based on the *validity* operator [15]. In the non-parametric case, this operator takes the symbolic transition $T = ((l,\zeta),\delta,\langle(l_1,\zeta_1),...,(l_n,\zeta_n)\rangle)$ and returns false if the part of ζ from which it is possible to let time pass and then perform action δ, such that taking the ith edge (R_i, l_i) gives some state $(l_i, v) \in (l_i, \zeta_i)$, is empty. Such analysis requires several backward operators, defined for the parametric domain in [12]:

- *past* (time predecessors): $\zeta^{\swarrow} = \{v' \mid v \in \zeta \wedge v'(x) \geqslant 0, v'(x) + d = v(x), d \geqslant 0 \text{ if } x \in X; v'(x) = v(x) \text{ if } x \in P\}$
- *inverse reset of clocks* in set $R \subseteq X$: $\zeta[R]^{-1} = \{v' \mid \exists v \in \zeta \text{ s.t. } v'(x) = 0 \text{ if } x \in R \wedge v'(x) = v(x) \text{ otherwise}\}$

We extend the validity operator to parametric domain and replace Boolean operations with the corresponding set-theoretic operations, in order to obtain the valuations on $X \cup P$ from which it is possible to perform such a transition: $valid(T) = \zeta \cap ((\gamma \cap (\cap_{i=1}^{n}(\zeta_i[R]^{-1})))^{\swarrow})$. The transition T is therefore *valid* if the set of valuations (polyhedron) $valid(T)$ is non-empty. The projection of these valuations onto parameters gives the corresponding values of parameters. In order to construct a stochastic game, the notion of validity is extended to sets of symbolic transitions with the same source. Again, we replace Boolean with set-theoretic operators: $valid(\mathbb{T}) = (\cap_{T \in \mathbb{T}} valid(T)) \cap (\cap_{T \in Trans(l,\zeta) \setminus \mathbb{T}} \neg valid(T))$. $valid(\mathbb{T})$ defines the set of valuations $v \models \zeta$ on $X \cup P$, such that from (l,v) it is possible to perform any symbolic transition $T \in \mathbb{T}$, but it is not possible to perform any other transition of $Trans(l,\zeta)$. In a symbolic state (l,ζ) of a stochastic game abstraction of a PPTA, player 1 first picks a subset $\mathbb{T}$ of symbolic transitions (in other words, part of the symbolic state in which these transitions are valid), and player 2 then picks a transition $T \in \mathbb{T}$. Fig. 6 shows the algorithm for the construction of a stochastic game from a given reachability graph, which yields (by quantifying over all possible strategies for player 1 and player 2) upper and lower bounds on the max/min reachability probabilities in a PPTA.

Theorem 2. *If* $(Sym, Trans) = \textsc{ParReach}(\mathcal{P}, F)$, $\mathcal{G} = \textsc{BuildGame}(Sym, Trans)$ *and* $* \in \{min, max\}$ *then:* $p_{\mathcal{G}}^{lb,*}(Reached) \leqslant p_{\mathcal{P}}^{*}(F) \leqslant p_{\mathcal{G}}^{ub,*}(Reached)$.

BuildGame(Sym, $Trans$)

$sym_0 = \{(l, \zeta) \in S \mid l = l_0\}$
for $(l, \zeta) \in S$
    for $\mathbb{T} \subseteq Trans(l, \zeta)$ s.t. $\mathbb{T} \neq \varnothing$ and $valid(\mathbb{T}) \neq \varnothing$
        $\mathsf{Steps}_{\mathcal{G}}((l, \zeta), \mathbb{T}) := \{\mu_T \mid T \in \mathbb{T}\}$
return $\mathcal{G} = (Sym, sym_0, 2^{Trans}, \mathsf{Steps}_{\mathcal{G}})$

Fig. 6. Algorithm for stochastic game abstraction

Refine(Sym, $Trans$, (l, ζ), $\mathbb{T}_{lb}$, $\mathbb{T}_{ub}$)

$\zeta_{lb} := valid(\mathbb{T}_{lb})$; $\zeta_{ub} := valid(\mathbb{T}_{ub})$
$Sym^{new} := \{(l, \zeta_{lb}), (l, \zeta_{ub}), (l, \zeta \wedge \neg(\zeta_{lb} \vee \zeta_{ub}))\} \backslash \{\varnothing\}$
$Sym^{ref} := (Sym \backslash \{(l, \zeta)\}) \uplus S^{new}$; $Trans^{ref} := \varnothing$
for each $T = (S_0, \delta, \langle S_1, ..., S_n \rangle) \in Trans$
    if $(l, \zeta) \notin \{S_0, S_1, ..., S_n\}$ **then**
        $Trans^{ref} := Trans^{ref} \cup \{T\}$
    else $\mathbb{T}^{new} := \{(S'_0, \delta, \langle S'_1, ..., S'_n \rangle) \mid S'_i \in Sym^{new}$ if $S_i = (l, \zeta) \wedge S'_i = S_i$ otherwise$\}$
        for $T^{new} \in \mathbb{T}^{new}$ **such that** $valid(T^{new}) \neq \varnothing$
            $Trans^{ref} := Trans^{ref} \cup \{T^{new}\}$
return $(Sym^{ref}, Trans^{ref})$

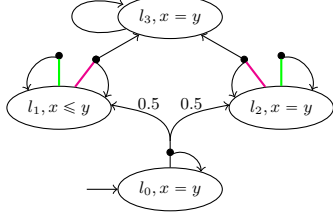**Fig. 7.** Algorithm for parametric abstraction refinement

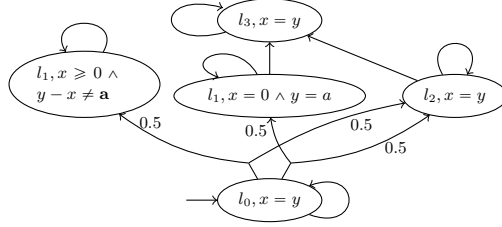**Fig. 8.** Game-based abstraction   **Fig. 9.** Refinement of a symbolic state

Example 2. A game constructed from the forward reachability graph of a PPTA in Fig. 2 is shown in Fig. 8. We represent player 1 states by ellipses containing symbolic states (l, ζ), and player 2 states by a black dot. In two of its states (containing l_1 and l_2), player 1 can choose between the part of the state where both transitions are valid and the part where only one transition is valid (a self-loop). The analysis of this game, however, gives values 0 and 1 for lower and upper bound, respectively, on the maximum probability of reaching l_3. We address this issue below by applying a method to refine the abstraction.

Parametric abstraction refinement   Stochastic game abstractions might be too imprecise for reachability probabilities, as shown in Example 2. The abstraction refinement method proceeds by iteratively computing refined abstractions until suitable precision is obtained. The game-based abstraction refinement for

MDPs from [13] uses the difference between lower and upper bounds on max/min reachability probability computed thus far as the quantitative measure of precision. This method has been subsequently used in [15] for the abstraction refinement for PTA. We now explain our extension for the parametric case, which leads to parameter values corresponding to precise probabilities in the model.

After the construction and analysis of a stochastic game, the refinement algorithm, presented in Fig. 7, takes the reachability graph $(Sym, Trans)$, splits one symbolic state per iteration and modifies symbolic transitions accordingly. The split of a symbolic state (l, ζ) is done with respect to player 1 strategy choices, $\mathbb{T}_{ub}$ and $\mathbb{T}_{lb}$, in (l, ζ), which achieve lower and upper bounds (such choices must exist in a state where these bounds differ, [14]). The symbolic state (l, ζ) is therefore split into $(l, valid(\mathbb{T}_{lb})), (l, valid(\mathbb{T}_{ub}))$, and $(l, \zeta \wedge \neg(valid(\mathbb{T}_{lb}) \vee valid(\mathbb{T}_{ub})))$. By construction, both $valid(\mathbb{T}_{lb})$ and $valid(\mathbb{T}_{ub})$ are non-empty and $valid(\mathbb{T}_{lb}) \neq valid(\mathbb{T}_{ub})$, and thus the split produces strict refinement. The MDP of Fig. 5, after a refinement of one symbolic state, is shown in Fig. 9.

The complete game-based abstraction refinement scheme, shown in Fig. 10, provides a means to compute the precise values for max/min reachability probability, each corresponding to a particular parameter valuation. We can then choose those valuations that optimise (either maximise or minimise) these probabilities. Algorithm SYNTH uses function ANALYZEGAME of [5] to compute bounds on max/min probability of reaching some set of locations in a stochastic game and the corresponding strategies. The choice $\mathbb{T}_i$ of player 1, in some (l, ζ), is a set of symbolic transitions T and also represents the part of ζ in which these transitions are valid. To find the optimal parameter valuations, we first need to take the projection onto the parameters for each $valid(\mathbb{T}_i)$ in the optimal strategy of player 1 (the strategy for reaching some $Reached_k$ which gives the optimal probability), and take their intersection. Then, for some $(l_k, \zeta_k) \in Reached_k$ (all of them have the equivalent $\zeta_{k|P}$), we obtain the solution as $\{\bigcap_i valid(\mathbb{T}_i)_{|P} \cap (\zeta_{k|P} \backslash (\bigcup_{\forall j \neq k, l_j \in F} \zeta_{j|P}))\}$.

Theorem 3. *For a PPTA $\mathcal{P}$, a subset of its location F and $* \in \{min, max\}$, let $(Sym, Trans) = $ PARREACH$(\mathcal{P}, F)$. If $(Sym^{ref}, Trans^{ref})$ is the result returned by applying* REFINE *to $(Sym, Trans)$, $\mathcal{G}$ by* BUILDGAME$(Sym, Trans)$ *and $\mathcal{G}^{ref}$ by* BUILDGAME$(Sym^{ref}, Trans^{ref})$ *then:*

- *$(Sym^{ref}, Trans^{ref})$ is a reachability graph for $(\mathcal{P}, F)$;*
- *$p_{\mathcal{G}}^{lb,*}(Reached) \leqslant p_{\mathcal{G}^{ref}}^{lb,*}(Reached)$ and $p_{\mathcal{G}}^{ub,*}(Reached) \geqslant p_{\mathcal{G}^{ref}}^{ub,*}(Reached)$;*
- *If $p^* = p_{\mathcal{G}^{ref}}^{lb,*}(l_k, \zeta_k) = p_{\mathcal{G}^{ref}}^{ub,*}(l_k, \zeta_k)$, for some $(l_k, \zeta_k) \in Reached$, is the optimum $*$ reachability probability, among all $(l_j, \zeta_j) \in Reached$, then the solution to the optimal parameter synthesis can be extracted from the strategy σ_1 of Player 1 (which achieves p^*) and ζ_k.*

The algorithm is designed to terminate when the difference between the upper and lower bound falls below some threshold ϵ for reasons of computational efficiency. We show that this is, however, not necessary. If the initial forward reachability exploration terminates (PARREACH), then our algorithm, similarly

$\text{SYNTH}(\mathcal{P}, F, *, \varepsilon, \star)$

$(Sym, Trans) = \text{PARREACH}(\mathcal{P}, F); \mathcal{G} = \text{BUILDGAME}(Sym, Trans); p^* := 0; \sigma_{p*} := \varnothing$
for each $Reached_i \in Reached$
    $(p_{\mathcal{G}}^{lb,*}, p_{\mathcal{G}}^{ub,*}, \sigma_1^{lb}, \sigma_1^{ub}) := \text{ANALYSEGAME}(\mathcal{G}, Reached_i, *)$
    while $p_{\mathcal{G}}^{ub,*} - p_{\mathcal{G}}^{lb,*} > \varepsilon$
        choose $(l, \zeta) \in Sym$
            $(Sym^{ref}, Trans^{ref}) = \text{REFINE}(Sym, Trans, (l, \zeta), \sigma_1^{lb}(l, \zeta), \sigma_1^{ub}(l, \zeta))$
            $\mathcal{G} = \text{BUILDGAME}(Sym^{ref}, Trans^{ref})$
            $(p_{\mathcal{G}}^{lb,*}, p_{\mathcal{G}}^{ub,*}, \sigma_1^{lb}, \sigma_1^{ub}) := \text{ANALYSEGAME}(\mathcal{G}, Reached_i, *)$
    if $p^* \sim p_{\mathcal{G}}^{lb,*}$ **then** // *put* $<$ *(resp.* $>$*) instead of* $\sim$ *when* $\star$ *is maximisation*
        $p^* := p_{\mathcal{G}}^{lb,*}; \sigma_{p*} := \sigma_1^{lb}$ *(resp. minimisation)*
return $[p^*, \sigma_{p*}]$

Fig. 10. Parameter synthesis using game-based abstraction refinement loop

to its non-parametric counterpart from [15], is guaranteed to terminate in a finite number of steps with a precise answer.

Theorem 4 (Termination). *Let* $* \in \{min, max\}$*. If forward reachability algorithm (*PARREACH*) terminates, then the algorithm for parameter synthesis* SYNTH *terminates after a finite number of steps and returns* $p^* = p^{lb,*} = p^{ub,*}$*.*

Example 3. We return to the PPTA in Fig. 4. The final stochastic game, after two refinement iterations, is shown in Fig. 11. The validity of each new symbolic transition T_i, obtained in the refinement process, gives the following parameter valuations:

- $T_1 = ((l_0, x = y), \varnothing, \langle(l_1, x = 0 \land y = a), (l_2, x = y = 0)\rangle) \neq \varnothing$ if $a = 0$
- $T_2 = ((l_0, x = y), \varnothing, \langle(l_1, x = 0 \land y = a), (l_2, x = y > 0)\rangle) \neq \varnothing$ if $a \neq 0$
- $T_3 = ((l_0, x = y), \varnothing, \langle(l_1, x \geqslant 0 \land y \neq a), (l_2, x = y = 0)\rangle) \neq \varnothing$ if $a \neq 0$
- $T_4 = ((l_0, x = y), \varnothing, \langle(l_1, x \geqslant 0 \land y \neq a), (l_2, x = y > 0)\rangle) \neq \varnothing$ for $a \in \mathbb{R}_{\geqslant 0}$.

The set of transitions $\mathbb{T}_1 = \{T_2, T_3, T_4\}$ is valid if $a \neq 0$, in which case the max. probability of reaching l_3 is 0.5, and $\mathbb{T}_2 = \{T_1, T_4\}$, is valid if $a = 0$, in which case the max. probability of reaching l_3 is 1. If we wish to maximise this probability, the algorithm obtains the constraint $a = 0$.

5 Conclusion

We presented a technique for PPTA which derives symbolic constraints on parameters of the model, such that the max/min probability of reaching some set of locations is optimised. We focused on probabilistic reachability, but can easily consider more expressive target sets that refer to locations and clocks by syntactically modifying the model as in [18]. Computing expected time properties using game abstractions is still open for PTA. Termination of our algorithm depends on whether the forward reachability exploration terminates. Unlike for
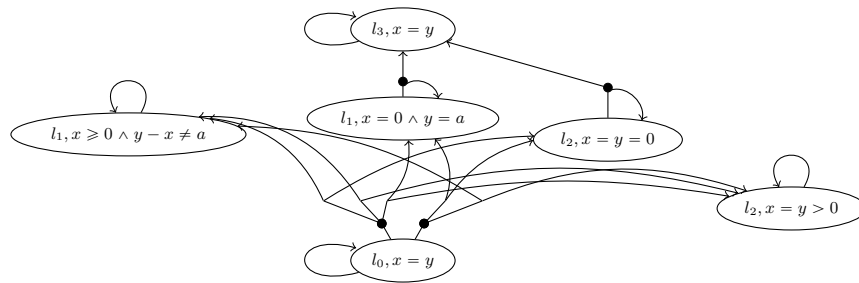
Fig. 11. Final stochastic game for PPTA of Fig. 4 (probability of 0.5 assumed on each edge of probabilistic transitions)

TA/PTA, where the extrapolation operator on zones can be used, in the parametric case we need to impose certain restrictions to ensure termination. One possibility is to restrict the parameter domain to bounded integers as in [11]. We are currently implementing the algorithm in PRISM and plan to explore expected time properties for PTA/PPTA.

References

1. R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126:183–235, 1994.
2. R. Alur, T. A. Henzinger, and M. Y. Vardi. Parametric real-time reasoning. In *STOC'93*, pages 592–601. ACM Press, 1993.
3. É. André, L. Fribourg, and J. Sproston. An extension of the inverse method to probabilistic timed automata. *FMSD*, 42:119–145, 2013.
4. N. Chamseddine, M. Duflot, L. Fribourg, C. Picaronny, and J. Sproston. Computing expected absorption times for parametric determinate probabilistic timed automata. In *QEST'08*, pages 254–263. IEEE CS Press, 2008.
5. A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.
6. C. Daws, A. Olivero, S. Tripakis, and S. Yovine. The tool kronos. In *In Proc. of Hybrid Systems III, LNCS 1066*, pages 208–219. Springer Verlag, 1996.
7. M. Diciolla, C. H. P. Kim, M. Kwiatkowska, and A. Mereacre. Synthesising optimal timing delays for timed i/o automata. In *EMSOFT'14*. ACM, 2014.
8. E. M. Hahn, H. Hermanns, and L. Zhang. Probabilistic reachability for parametric markov models. In *SPIN*, pages 88–106, 2009.
9. T. Han, J.-P. Katoen, and A. Mereacre. Approximate parameter synthesis for probabilistic time-bounded reachability. In *RTSS*, pages 173–182. IEEE Computer Society, 2008.
10. T. Hune, J. Romijn, M. Stoelinga, and F. W. Vaandrager. Linear parametric model checking of timed automata. *Journal of Logic and Algebraic Programming*, 52-53:183–220, 2002.
11. A. Jovanović, D. Lime, and O. H. Roux. Integer parameter synthesis for timed automata. In *TACAS 2013*, volume 7795 of *LNCS*, pages 401–415. Springer, 2013.

12. A. Jovanović, D. Lime, and O. H. Roux. Synthesis of bounded integer parameters for parametric timed reachability games. In *ATVA 2013*, volume 8172 of *LNCS*. Springer, 2013.

13. M. Kattenbelt, M. Kwiatkowska, G. Norman, and D. Parker. A game-based abstraction-refinement framework for Markov decision processes. *FMSD*, 36(3):246–280, 2010.

14. M. Kwiatkowska, G. Norman, and D. Parker. Game-based abstraction for Markov decision processes. In *QEST'06*, pages 157–166. IEEE CS Press, 2006.

15. M. Kwiatkowska, G. Norman, and D. Parker. Stochastic games for verification of probabilistic timed automata. In *FORMATS'09*, volume 5813 of *LNCS*, pages 212–227. Springer, 2009.

16. M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV'11*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

17. M. Kwiatkowska, G. Norman, D. Parker, and J. Sproston. Performance analysis of probabilistic timed automata using digital clocks. *FMSD*, 29:33–78, 2006.

18. M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *TCS*, 282:101–150, 2002.

19. K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a Nutshell. *Int. Journal on Software Tools for Technology Transfer*, 1:134–152, 1997.

A   Proofs

Theorem 1. *For a PPTA $\mathcal{P}$ and a subset of its locations F, if $(Sym, Trans) =$ PARREACH$(\mathcal{P}, F)$ and $\mathcal{M} =$ BUILDMDP$(Sym, Trans)$, then:*
   - $p_{\mathcal{M}}^{min}(Reached) \leqslant p_{\mathcal{P}}^{min}(F)$ and $p_{\mathcal{M}}^{max}(Reached) \geqslant p_{\mathcal{P}}^{max}(F)$.
   - if $\mathcal{M}$ gives the precise reachability probabilities in $\mathcal{P}$ and if some $(l_k, \zeta_k) \in$ Reached has the optimum (max. or min.) reachability probability, among all $(l_j, \zeta_j) \in$ Reached, then $\{\zeta_{k|P} \backslash (\bigcup_{\forall j \neq k, l_j \in F} \zeta_{j|P})\}$ is the solution to the optimal parameter synthesis problem.

Proof. Let us fix PPTA $\mathcal{P} = (L, l_0, X, P, \Sigma, \mathsf{prob}, \mathsf{Inv})$. $(Sym, Trans)$ is the reachability graph of $\mathcal{P}$ and $\mathcal{M} = (Sym, sym_0, Trans, \mathsf{Steps}_{\mathcal{M}})$ the MDP obtained with BUILDMDP$(Sym, Trans)$. *Reached* is the set of all symbolic states (l, ζ) from Sym such that $l \in F$.

For any parameter valuation v, we obtain a PTA $v(\mathcal{P})$. The semantics of $v(\mathcal{P})$ is given by an (infinite state) MDP $\mathcal{M}_{v(\mathcal{P})} = (Q, q_0, \mathbb{R}_{\geqslant 0} \times \Sigma, \mathsf{Steps}_{\mathcal{M}_{v(\mathcal{P})}})$.

For any such $v(\mathcal{P})$, the first part of the theorem is a direct consequence of the following Lemma 1, proven in [15]. By $v(Sym)$ we denote the set of all reachable (non-parametric) symbolic states obtained using the forward reachability algorithm applied to $v(\mathcal{P})$, and by $v(\mathcal{M})$ the corresponding MDP. In a similar way, we define $v(S) \in v(Sym)$ as $(l, v(\zeta))$, a (non-parametric) symbolic state, obtained in a forward reachability algorithm applied to $v(\mathcal{P})$, and $v(Reached)$ is the set of all symbolic states $(l, v(\zeta))$ such that $l \in F$. For the sake of simplicity, we redefine an edge $e = (R, l')$ in the distribution $\mathsf{prob}(l, \delta, \gamma)$ as $e = (l, \delta, \gamma, R, l')$ and $v(e) = (l, \delta, v(\gamma), R, l')$.

Lemma 1. *For any strategy $\mathcal{A}$ of $\mathcal{M}_{v(\mathcal{P})}$ and $q \in Q$, there exists a strategy $\mathcal{B}_{\mathcal{A}}$ of $v(\mathcal{M})$ where $p_q^{\mathcal{A}}(Q^F) = p_{v(S)}^{\mathcal{B}_{\mathcal{A}}}(v(Reached))$, for all $v(S) \in v(Sym)$ such that $q \in v(S)$, where Q^F is a set of states $(l, u) \in Q$ such that $l \in F$.*

For the second part of the theorem, we first define a run as a finite sequence $\rho = q_1 a_1 q_2 a_2 ... a_{n-1} q_n$ such that for all $i, q_i \in Q, a_i \in \Sigma \cup \mathbb{R}_{\geqslant 0}$ and $q_i \xrightarrow{a_i} q_{i+1}$. A sequence of symbolic successor by edges $e_1, ..., e_n$: $\mathsf{Succ}(S, e_1, ... e_n)$ is defined as $\mathsf{Succ}(...\mathsf{Succ}(\mathsf{Succ}((l, \zeta), e_1), e_2)..., e_n)$. Moreover, we need the following lemma proven in [11].

Lemma 2. *For any symbolic state (l, ζ), for all edges e and valuations v, $v(\mathsf{Succ}((l, \zeta), e)) = \mathsf{Succ}((l, v(\zeta)), v(e))$.*

We can safely use Lemma 2 as the Succ operator is not impacted by probabilities. The corollary of Lemma 2 is that for any sequence of edges $e_1, ..., e_n$: $v(\mathsf{Succ}(S, e_1, ..., e_n)) = \mathsf{Succ}(v(S), v(e_1), ..., v(e_n))$.

Let us assume that the set *Reached* is divided into subsets $Reached_i$ for $i \in [1..n]$. According to our forward reachability algorithm, $\forall (l, \zeta), (l', \zeta') \in Reached_i$, for any $i \in [1..n]$, $\zeta_{|P} = \zeta'_{|P}$.

Let us assume that some $Reached_k = \{(l_k, \zeta_k)\}, l_k \in F$ has the optimum reachability probability among all $Reached_i \in Reached$ in the MDP $\mathcal{M}$.

To prove the second part of the theorem, let us first prove that $\zeta_{k|P}$ is a solution to the *reachability synthesis problem*, i.e., for any parameter valuation $v \in \zeta_{k|P}$, l_k is reachable in $v(\mathcal{P})$.

Let us consider the possibly infinite directed labeled tree, whose root is labeled by $Init(\mathcal{P})$ and for every node n, if n is labeled by a symbolic state S, then, for all edges e of the PPTA, there exists a child n' labeled by $\mathsf{Succ}(S, e)$ iff $\mathsf{Succ}(S, e)$ is not empty. For easier reference, we also label the arc from n to n' by e.

Now, consider that the forward reachability algorithm has terminated. Then only a finite prefix (a subset closed under the parent relation) T of the infinite tree has been visited and each leaf must correspond to one of the leaf conditions of the algorithm or to the absence of children. This means that all leaves n of the tree are labeled by symbolic states S such that:

- either $S = (l, \zeta)$ and $l = l_k$;
- or $S \in Sym$;
- or S has no successor.

We first state the following lemma from [11].

Lemma 3. *Let n be a node of T, labeled by some symbolic state S, and such that the subtree rooted at n has depth N. We have that a parameter valuation $v \in \zeta_{k|P}$ is the solution to the reachability synthesis problem, starting the forward reachability algorithm from S, iff there exists a state q in $v(S)$ and a run ρ in $v(\mathcal{P})$, with fewer then N discrete steps, that starts in q and reaches l_k.*

Proof. We prove this by induction on N. Note that the tree T is always non-empty (it contains at least the root which is labeled by $Init(\mathcal{P})$).

- Case of a leaf n labeled by S: the subtree rooted at n has depth 1.
  - If v is the solution to the reachability synthesis problem then the only leaf condition of the algorithm that can be verified is $S = (l_k, \zeta_k)$, so, for all states $q = (l_k, u)$ in $v(S)$, there is a run with no discrete steps that starts in q and reaches l_k.
  - If there exists a state $q \in v(S)$ and a run with no discrete steps that starts in q and reaches l_k, then if l is the location of q, we have $l = l_k$, and therefore v is the solution to the reachability synthesis problem.
- Case of a non-leaf node n labeled by S: suppose the subtree rooted at n has depth $k > 1$ and that, for all nodes n' with subtree rooted at n' of depth $k' < k$, the property holds.
  - If v is the solution to the reachability synthesis problem then, since n is not a leaf, the following condition must be true:
    there exists a successor n' of n, labeled by $S' = \mathsf{Succ}(S, e)$ for some edge e such that v is the solution to the reachability synthesis problem, starting the forward reachability algorithm from S'. Since it is a successor of n, n' has depth less than k. So we can use the induction hypothesis: there exists a run with less than $k - 1$ discrete steps, starting in some state $q' \in v(S')$ and reaching l_k in $v(\mathcal{P})$. By Lemma 2, $q' \in \mathsf{Succ}(v(S), v(e))$ so q' has a predecessor q by e in $v(S)$ and we get the expected result.

- If there exists a run ρ starting in some state $q \in v(S)$ and reaching l_k, with fewer than k discrete steps, then this run has at least 1 discrete step, as otherwise n would be a leaf of T. So we can write it $q \xrightarrow{d} q_d \xrightarrow{a} \rho'$ where a is the action of some edge e. Then ρ' is a run starting from some state $q' \in \mathsf{Succ}(v(S), v(e))$, reaching l_k and with less than $k-1$ discrete steps. Moreover $q' \in v(S')$ with $S' = \mathsf{Succ}(S, e)$ (by Lemma 2) So we can apply the induction hypothesis and v is the solution to the reachability synthesis problem, starting the forward reachability algorithm from S'.

With Lemma 3, we immediately have that if $\zeta_{k|P}$ is a solution to the reachability synthesis problem, then there exists a run in $v(\mathcal{P})$ that starts in the initial state and reaches l_k.

In the other direction, suppose there exists such a run ρ. Then ρ is finite and its last state has a location belonging to l_k. Let $e_1, ..., e_p$ be the edges taken in ρ and consider the branch in the tree T obtained by following this edge sequence on the labels of the arcs in the tree as long as possible. If the whole edge sequence is feasible in T, then the tree T has depth greater or equal to the size of the sequence and we can apply Lemma 3 to obtain that v is the solution to the reachability synthesis problem.

Otherwise, let $S = (l, \zeta)$ be the symbolic state labelling the last node of the branch, e_k be the first edge in $e_1, ..., e_p$ that is not present in the branch and q be the state of ρ just before taking e_k.

Using Lemma 2, $v(\mathsf{Succ}(S, e_k))$ is not empty so $\mathsf{Succ}(S, e_k)$ is not empty.

Since the node labeled by S has no children in T, it follows that either $l = l_k$ or there exists another node on the branch that is labeled by S. In the former case, we can apply Lemma 3 to the prefix of ρ ending in q and we obtain that v is the solution to reachability synthesis problem.

In the latter case, by the corollary of Lemma 2, there exists a run along edges $e_1, ..., e_m$, with $m < k$, that reaches q in $v(\mathcal{P})$.

From that run we can construct another run ρ' by merging with the suffix of ρ that starts from q. ρ' has strictly fewer discrete actions than ρ and also reaches l_k and we can repeat the same reasoning as we have just done. We can do this only a finite number of times (because the length of the considered run is strictly decreasing) so at some point we have to be in some of the other cases and we obtain the expected result.

We have proven that $\zeta_{k|P}$ is a solution for the reachability synthesis problem. The same stands for any $\zeta_{j|P}$, such that $(l_j, \zeta_j) \in Reached$. If we were interested in the reachability synthesis problem, we would take $\bigcup_j \zeta_{j|P}$, for all $(l_j, \zeta_j) \in Reached$, as a solution. As we are interested in the optimum parameter synthesis problem we must exclude the parameter valuations contained in the other symbolic states of $Reached$. We therefore take $\{\zeta_{k|P} \backslash (\bigcup_{\forall j \neq k, l_j \in F} \zeta_{j|P})\}$ as a solution.

Theorem 2. *If* $(Sym, Trans) = \textsc{ParReach}(\mathcal{P}, F)$, $\mathcal{G} = \textsc{BuildGame}(Sym, Trans)$ *and* $\divideontimes \in \{min, max\}$ *then:* $p_{\mathcal{G}}^{lb, \divideontimes}(Reached) \leqslant p_{\mathcal{P}}^{\divideontimes}(F) \leqslant p_{\mathcal{G}}^{ub, \divideontimes}(Reached)$.

Proof. For any parameter valuation v, we obtain PTA $v(\mathcal{P})$. As in the proof of Theorem 1 its semantics is given by an MDP $\mathcal{M}_{v(\mathcal{P})}$. We now recall the following lemmas, whose proofs can be found in [15]. $v(S)$, $v(Reached)$ and $v(Sym)$ are defined as in the proof of Theorem 1, and $v(\mathcal{G})$ is the stochastic game obtained from $v(\mathcal{P})$.

Lemma 4. *For any strategy $\mathcal{A}$ of $\mathcal{M}_{v(\mathcal{P})}$ and $q \in Q$, there exists a strategy pair (σ_1, σ_2) of $v(\mathcal{G})$ where $p_q^{\mathcal{A}}(Q^F) = p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached))$ for all $v(S) \in v(Sym)$ such that $q \in v(S)$.*

Lemma 5. *For any symbolic state $v(S) \in v(Sym)$ and player 2 strategy σ_2 of $v(\mathcal{G})$, there exists a strategy $\mathcal{A}$ of $\mathcal{M}_{v(\mathcal{P})}$ where: $\inf_{\sigma_1} p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached)) \leqslant p_q^{\mathcal{A}}(Q^F)$ and $\sup_{\sigma_1} p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached)) \geqslant p_q^{\mathcal{A}}(Q^F)$*

We now briefly present the proof of Theorem 2. From Lemma 4 we have:

- $\inf_{\sigma_1,\sigma_2} p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached)) \leqslant \inf_{\mathcal{A}} p_q^{\mathcal{A}}(Q^F)$
- $\sup_{\sigma_1,\sigma_2} p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached)) \geqslant \sup_{\mathcal{A}} p_q^{\mathcal{A}}(Q^F)$

for all $v(S) \in v(Sym)$ such that $q \in v(S)$, and therefore:
$p_{v(\mathcal{P})}^{min}(F) \geqslant p_{v(\mathcal{G})}^{lb,min}(v(Reached))$ and $p_{v(\mathcal{P})}^{max}(F) \leqslant p_{v(\mathcal{G})}^{ub,max}(v(Reached))$.
Using Lemma 5, we have that for any $q \in Q$ and $v(S) \in v(Sym)$ such that $q \in v(S)$: $\inf_{\mathcal{A}} p_q^{\mathcal{A}}(Q^F) \leqslant \inf_{\sigma_2}\sup_{\sigma_1} p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached)) = \sup_{\sigma_1}\inf_{\sigma_2} p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached))$.
The last equality follows from properties of stochastic games [5]. In a similar way, we have: $\sup_{\mathcal{A}} p_q^{\mathcal{A}}(Q^F) \geqslant \inf_{\sigma_1}\sup_{\sigma_2} p_{v(S)}^{\sigma_1,\sigma_2}(v(Reached))$.
It follows: $p_{v(\mathcal{P})}^{min}(F) \leqslant p_{v(\mathcal{G})}^{ub,min}(v(Reached))$ and $p_{v(\mathcal{P})}^{max}(F) \geqslant p_{v(\mathcal{G})}^{lb,max}(v(Reached))$.
As the behaviour of $\mathcal{P}$ is described by the behaviour of all $v(\mathcal{P})$, obtained by considering all possible parameter valuations v, we can complete the proof with:

- $p_{\mathcal{P}}^{min}(F) \leqslant p_{\mathcal{G}}^{ub,min}(Reached)$ and $p_{\mathcal{P}}^{max}(F) \geqslant p_{\mathcal{G}}^{lb,max}(Reached)$
- $p_{\mathcal{P}}^{min}(F) \geqslant p_{\mathcal{G}}^{lb,min}(Reached)$ and $p_{\mathcal{P}}^{max}(F) \leqslant p_{\mathcal{G}}^{ub,max}(Reached)$

Theorem 3. *For a PPTA $\mathcal{P}$, a subset of its location F and $* \in \{min, max\}$, let $(Sym, Trans) = \textsc{ParReach}(\mathcal{P}, F)$. If $(Sym^{ref}, Trans^{ref})$ is the result returned by applying $\textsc{Refine}$ to $(Sym, Trans)$, $\mathcal{G}$ by $\textsc{BuildGame}(Sym, Trans)$ and $\mathcal{G}^{ref}$ by $\textsc{BuildGame}(Sym^{ref}, Trans^{ref})$ then:*
   - *$(Sym^{ref}, Trans^{ref})$ is a reachability graph for $(\mathcal{P}, F)$;*
   - *$p_{\mathcal{G}}^{lb,*}(Reached) \leqslant p_{\mathcal{G}^{ref}}^{lb,*}(Reached)$ and $p_{\mathcal{G}}^{ub,*}(Reached) \geqslant p_{\mathcal{G}^{ref}}^{ub,*}(Reached)$;*
   - *If $p* = p_{\mathcal{G}^{ref}}^{lb,*}(l_k, \zeta_k) = p_{\mathcal{G}^{ref}}^{ub,*}(l_k, \zeta_k)$, for some $(l_k, \zeta_k) \in Reached$, is the optimum $*$ reachability probability, among all $(l_j, \zeta_j) \in Reached$, then the solution to the optimal parameter synthesis can be extracted from the strategy σ_1 of Player 1 (which achieves $p*$) and ζ_k.*

Proof.   − We first prove that $(Sym^{ref}, Trans^{ref})$ is the reachability graph for $(\mathcal{P}, F)$. We state the following lemma, modified from [15] for the parametric case.

Lemma 6. *If $S^{ref} \in Sym^{ref}$, $(S^{ref}, \delta, \langle S_1^{ref}, ..., S_n^{ref} \rangle) \in Trans(S^{ref})$ and $S \in Sym$ such that $S^{ref} \subseteq S$, then there exists $(S, \delta, \langle S_1, ..., S_n \rangle) \in Trans$ such that $S_i^{ref} \subseteq S_i$ for all $1 \leqslant i \leqslant n$.*

Let us consider any $S^{ref} \in Sym^{ref}$, $(S^{ref}, \delta, \langle S_1^{ref}, ..., S_n^{ref} \rangle) \in Trans(S^{ref})$ and $S \in Sym$ such that $S^{ref} \in S$. The proof is split into two cases:

- If $S^{ref} \in Sym$, then by construction $S^{ref} = S$, and therefore we have that either $(S^{ref}, \delta, \langle S_1^{ref}, ..., S_n^{ref} \rangle) \in Trans(S)$, in which case the lemma holds, or there exists $(S, \delta, \langle S_1, ...S_n \rangle) \in Trans(S)$ from which $(S^{ref}, \delta, \langle S_1^{ref}, ...S_n^{ref} \rangle)$ was constructed. In the second case, it follows from REFINE, Fig. 7, that $S_i^{ref} \in S_i$, for all $1 \leqslant i \leqslant n$, as required.
- If $S^{ref} \notin Sym$, then for $S^{ref} \subseteq S$ it follows that S^{ref} was formed by splitting S. Thus, there exists a symbolic transition $(S, \delta, \langle S_1, ..., S_n \rangle) \in Trans$ which was used to construct $(S^{ref}, \delta, \langle S_1^{ref}, ..., S_n^{ref} \rangle)$. It follows from this construction that $S_i^{ref} \in S_i$ for all $1 \leqslant i \leqslant n$, as required.

The fact that $(Sym^{ref}, Trans^{ref})$ is the reachability graph for $\mathcal{P}$ follows from the fact that we split only symbolic states and remove transitions which are not valid.

- We now prove the second part of the theorem which states:
$p_{\mathcal{G}}^{lb,*}(Reached) \leqslant p_{\mathcal{G}^{ref}}^{lb,*}(Reached)$ and $p_{\mathcal{G}}^{ub,*}(Reached) \geqslant p_{\mathcal{G}^{ref}}^{ub,*}(Reached)$;
We consider $v(\mathcal{P})$ for any parameter valuation v, and the following Lemma 7 and Lemma 8, from [15]. $v(Sym)$, $v(S)$, $v(Reached)$ and $v(\mathcal{G})$ are defined as in proofs of Theorem 1 and Theorem 2.

Lemma 7. *For any $v(S) \in v(Sym)$ and player 2 strategy σ_2 of $v(\mathcal{G})$ there exist a strategy pair $(\sigma_1^{ref}, \sigma_2^{ref})$ of $v(\mathcal{G}^{ref})$ where: $\inf_{\sigma_1} p_{v(S)}^{\sigma_1, \sigma_2}(v(Reached)) \leqslant p_{v(S^{ref})}^{\sigma_1^{ref}, \sigma_2^{ref}}(v(Reached))$ and $\sup_{\sigma_1} p_{v(S)}^{\sigma_1, \sigma_2}(v(Reached)) \geqslant p_{v(S^{ref})}^{\sigma_1^{ref}, \sigma_2^{ref}}(v(Reached))$.*

Lemma 8. *Given a player 2 strategy σ_2^{ref} of $v(\mathcal{G}^{ref})$ the proof follows by constructing a player 1 strategy σ_1^{ref} of $v(\mathcal{G}^{ref})$ and strategy pair (σ_1, σ_2) of $v(\mathcal{G})$ such that: $p_{v(S)}^{\sigma_1, \sigma_2}(v(Reached)) = p_{v(S^{ref})}^{\sigma_1^{ref}, \sigma_2^{ref}}(v(Reached))$ for all $v(S^{ref})$ such that $v(S^{ref}) \subseteq v(S)$.*

The proof of the theorem then follows similarly to the proof of Theorem 2, using Lemma 7 and Lemma 8 instead of Lemma 4 and Lemma 5. We refer to [15] for the complete proof.

- We now focus on the last part of the theorem:
If $p^* = p_{\mathcal{G}^{ref}}^{lb,*}(l_k, \zeta_k) = p_{\mathcal{G}^{ref}}^{ub,*}(l_k, \zeta_k)$, for some $(l_k, \zeta_k) \in Reached$, is the optimum $*$ reachability probability, among all $(l_j, \zeta_j) \in Reached$, then the solution to the optimal parameter synthesis can be extracted from the strategy σ_1 of Player 1, (that corresponds to p^*), and ζ_k.
Combining the proof of Theorem 1 and the first part of Theorem 3, which states that $(Sym^{ref}, Trans^{ref})$ is a reachability graph of $\mathcal{P}$, we obtain that $\zeta_{k|P}$ is a solution to the reachability synthesis problem.

Upon the termination of the iterative procedure of the algorithm SYNTH, we are sure that no state $(l, \zeta) \in Reached$ has been divided, therefore parameter valuations $\zeta_{|P}$ of any $(l, \zeta) \in Reached$ stay unchanged. On the other hand, some of the symbolic states from Sym are divided and therefore their parameter valuations are refined. Optimal strategy σ_1 of player 1 (a strategy to reach (l_k, ζ_k) which has the optimum reachability probability) consists of choices $\mathbb{T}$ in symbolic states of the final set of symbolic states Sym^{ref}.

For each such $\mathbb{T}$, $valid(\mathbb{T})$ gives the set of valuations on $X \cup P$, in which $\mathbb{T}$ is valid. $valid(\mathbb{T})_{|P}$ then gives the corresponding parameter values. To allow all choices $\mathbb{T}_i$ in the optimal strategy σ_1, we need to take $\bigcap_i valid(\mathbb{T}_i)_{|P}$. Finally, the optimal solution is obtained as $\{\bigcap_i valid(\mathbb{T}_i)_{|P} \cap (\zeta_{k|P} \backslash (\bigcup_{\forall j \neq k, l_j \in F} \zeta_{j|P}))\}$.

Theorem 4 (Termination). *Let $\circledast \in \{min, max\}$. If forward reachability algorithm (PARREACH) terminates, then the algorithm for parameter synthesis SYNTH terminates after a finite number of steps and returns $p\circledast = p^{lb,\circledast} = p^{ub,\circledast}$.*

Proof. As stated in Section 3, for stochastic game $\mathcal{G} = (S, (S_1, S_2), s_0, \Sigma, Steps_\mathcal{G})$ and maximum reachability probabilities, for some $F \subseteq S$ we have:

- $p_\mathcal{G}^{lb,max}(F) \overset{\text{def}}{=} \sup_{s \in s_0} \inf_{\sigma_1} \sup_{\sigma_2} p_s^{\sigma_1, \sigma_2}(F)$
- $p_\mathcal{G}^{ub,max}(F) \overset{\text{def}}{=} \sup_{s \in s_0} \sup_{\sigma_1} \sup_{\sigma_2} p_s^{\sigma_1, \sigma_2}(F)$

For the minimum reachability probabilities, similar equations hold:

- $p_\mathcal{G}^{lb,min}(F) \overset{\text{def}}{=} \inf_{s \in s_0} \inf_{\sigma_1} \inf_{\sigma_2} p_s^{\sigma_1, \sigma_2}(F)$
- $p_\mathcal{G}^{ub,min}(F) \overset{\text{def}}{=} \inf_{s \in s_0} \sup_{\sigma_1} \inf_{\sigma_2} p_s^{\sigma_1, \sigma_2}(F)$

At each iteration, a split of a symbolic state provides a strict refinement. At the point where each symbolic states (l, ζ) in a game $\mathcal{G}^{ref}$ has only one outgoing subset of edges $\mathbb{T}$ (thus, only one possible choice for player 1), we have $p_{\mathcal{G}^{ref}}^{lb,\circledast}(Reached) = p_{\mathcal{G}^{ref}}^{ub,\circledast}(Reached)$, because the upper and the lower bounds, for either maximum or minimum reachability probability, are obtained for the different strategy choice of player 1 (player 2 plays either its best or its worst choice (in terms of probability), depending on whether we are considering maximum or minimum reachability probabilities, respectively). The algorithm then terminates with the precise answer, as, according to Theorem 3, $p_{\mathcal{G}^{ref}}^{lb,\circledast}(Reached) \leqslant p_\mathcal{P}^{\circledast}(F) \leqslant p_{\mathcal{G}^{ref}}^{ub,\circledast}(Reached)$.