PCTL model checking of symbolic probabilistic systems*

Marta Kwiatkowska[1], Gethin Norman[1] and Jeremy Sproston[2]
[1] School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, United Kingdom
[2] Dipartimento di Informatica, Università di Torino, 10149 Torino, Italy

April 4, 2003

Abstract

Probabilistic model checking is a method for automatically verifying that a probabilistic system satisfies a property with a given likelihood, with the probabilistic temporal logic PCTL being a common choice for the property specification language. In this paper, we explore methods for model checking PCTL properties of infinite-state systems in which probabilistic and nondeterministic behaviour coexist. Building on previous work on computing the maximum probability with which a state set is reached in such systems, we utilize symbolic operations on the state sets to generate a finite-state version of the system on which the PCTL model checking problem can be answered. As in the non-probabilistic case, our model checking algorithm is semi-decidable for infinite-state systems. We illustrate our technique using the formalism of probabilistic timed automata, for which previous PCTL model checking techniques were based on an unnecessarily fine subdivisions of the state space.

1 Introduction

Many systems, such as control, real-time, and embedded systems, give rise to *infinite-state* models. For instance, embedded systems can be modelled in formalisms characterised by a finite number of control states (representing a digital controller) interacting with a finite set of real-valued variables (representing an analogue environment). The standard approach is to express system behaviour purely in terms of nondeterminism. However, in many cases, particularly in the context of fault-tolerant systems, it may be desirable, or more faithful, to express the relative likelihood of the system exhibiting certain behaviour. Nondeterministic choice is nevertheless useful for the modelling of asynchronous systems and the underspecification of system behaviour, and therefore formalisms in which nondeterministic and probabilistic choice coexist are subject of much attention (see, for example, [Var85, dA97, BK98, KNSS02]). Nondeterminism

*Supported in part by the EPSRC grant GR/N22960.

and probabilistic behaviour are inherent real-world protocols such as the IEEE 1394 FireWire root contention and IEEE 802.11 MAC.

This paper continues our study of the verification problem for infinite-state systems featuring both nondeterminism and probabilistic behaviour. In [KNS01] we considered the *maximal reachability probability problem* for this class of system. We consolidate the results of that paper by studying model checking of the probabilistic temporal logic PCTL (Probabilistic Computation Tree Logic) [HJ94, BdA95]. In particular, this requires the study of properties which refer to the *minimal* probability with which certain state sets are reached. A contribution of this paper is to show how such properties can be reduced to properties referring to the maximal probability of either reaching a certain state set, or of staying within a certain set of states forever.

We apply our PCTL model checking algorithms to a class of *symbolic probabilistic systems*. Such systems can be infinite-state, feature both nondeterministic and probabilistic choice, and are subject to two assumptions to make our model checking algorithm feasible. The first assumption is the existence of a *symbolic theory* for a non-probabilistic version of the system. Symbolic theories were introduced in [HMR03] to enable the unified study of notions of state equivalence, state-space exploration algorithms and the decidability of temporal logic model checking of infinite-state systems. Such theories equip the infinite-state system with an abstract data type of *regions*, the elements of which represent state sets, and a set of *symbolic operations* on regions, which correspond to operations such as conjunction and disjunction of state sets, or the classical predecessor operation. As in [KNS01], we use such symbolic theories to analyse symbolic probabilistic systems, by converting the system to a non-probabilistic system, and then running state-space exploration algorithms to obtain a finite-state system which faithfully represents the behaviour of its infinite-state counterpart. The method is reliant on the encoding of sufficient information on the probabilistic behaviour of the system into the labels of the transitions of the non-probabilistic system in order to reconstruct the probabilistic behaviour at the finite-state level.

The second assumption that we make on symbolic probabilistic systems guarantees the presence of such an appropriate non-probabilistic representation of the system. We thoroughly overhaul the notation of [KNS01], and identify a class of infinite-state nondeterministic-probabilistic systems in which such a convenient representation can always be found. More precisely, this class of system is one in which transitions can be made according to a three-phase choice: the first phase comprises a nondeterministic choice over a finite set of alternatives, which then determines the probability distribution which is used in the second phase of choice, which in turn determines the possibly infinite set of target states that are available for nondeterministic choice in the third phase. We note that this class subsumes the class of infinite-state systems which make a transition by two-phase choice, either consisting of a finite nondeterministic choice followed by a finite probabilistic choice, or consisting of a finite probabilistic choice followed by a possibly infinite nondeterministic choice. It transpires that our class of system is adapted to the analysis of probabilistic timed and hybrid automata [KNSS02]. For example, in a location

of a probabilistic hybrid automaton, we first make a choice of either whether to let time advance or to leave the location using one of the enabled exiting probability distributions (finite choice); then, say a decision is made to leave the location, a probabilistic choice is made according to the chosen distribution (finite choice over edges between locations); and then, suppose an edge which resets nondeterministically the real-valued variable x in the interval $(1, 2]$ was chosen probabilistically in the second step, there is a nondeterministic choice of the new value of x (infinite choice).

An advantage of our approach of verifying PCTL properties of symbolic probabilistic systems over an *iterative* method, in which a "quantitative" predecessor operation which refers directly to probabilities is used (see, for example, [dAM01]), is that we clearly separate issues of convergence of probabilities from issues of non-convergence that may arise because the system is infinite-state. For some classes of system, such as probabilistic timed automata, guarantees of termination of our approach are immediate.

In Section 2, we revisit the definitions of nondeterministic-probabilistic systems, PCTL and probabilistic timed automata. We describe symbolic probabilistic systems in Section 3, detailing the three-phase systems discussed above, and also the non-probabilistic representations of such systems. In Section 4, we present the PCTL model checking, in addition to the two key sub-algorithms which generate finite-state representations of the symbolic probabilistic system used for resolving probabilistic properties. In Section 5, we conclude the paper. Proofs of the key results can be found in the appendix.

Related work. Approaches to infinite-state systems with discrete probability distributions include model checking methods for probabilistic lossy channel systems [AR03, BS03]. In the case of probabilistic timed automata, methods for computing *exact* reachability probabilities are presented in [KNSS02] and [KNS02] based on the *region graph* [AD94] and *digital clocks* [HMP92] respectively. However, both suffer from the state explosion problem (in particular, the size of the verification problem is sensitive to the magnitudes of the model's timing constraints, which is not true of our technique). An alternative in [KNSS02] uses forwards reachability, however it is only able to compute *upper bounds* on the maximal reachability probabilities. We also mention [DJJL01] which uses abstraction and refinement methods to calculate bounds on the minimal and maximal reachability probabilities for finite state probabilistic systems. Verification methodologies for infinite-state systems with *continuous* distributions are given in [BHHK00, DGJP00, KNSS00].

2 Preliminaries

A (discrete probability) *distribution* over a finite set Q is a function $\mu : Q \to [0, 1]$ such that $\sum_{q \in Q} \mu(q) = 1$. Let $\mathsf{support}(\mu)$ be the subset of Q such that $q \in \mathsf{support}(\mu)$ if and only if $\mu(q) > 0$. For a possibly uncountable set Q', let $\mathsf{Dist}(Q')$ be the set of distributions over finite subsets of Q'.

2.1 Nondeterministic-probabilistic systems

A *nondeterministic-probabilistic system* $\mathsf{NP} = (S, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ comprises a set S of *states*, a *nondeterministic-probabilistic transition function* $\mathit{Steps} : S \to 2^{\mathsf{Dist}(S)}$, a set P of *observations*, and an observation function $\langle\!\langle \cdot \rangle\!\rangle : P \to 2^S$ which maps every observable to the set of states in which it is observed. A *nondeterministic-probabilistic transition* $s \xrightarrow{\mu} t$ consists of a two-phase choice:

1. the first phase comprises a nondeterministic selection of a distribution μ from the (possibly infinite) set $\mathit{Steps}(s)$;

2. the second phase comprises a probabilistic choice of target state t according to μ (hence, we must have $\mu(t) > 0$).

The definition of nondeterministic-probabilistic systems follows the classical, Markov decision process-based definitions previously introduced for finite-state systems [BdA95, BK98]. We highlight the fact that infinite choice is made over the nondeterministic alternatives *only*; instead, every probabilistic choice is made over a finite number of alternatives. Hence, we use the notation $\bar{\mathrm{N}}\mathrm{P}$ (where the overbar denotes that the nondeterministic choice could be infinite, whereas the absence of a bar over P indicates that probabilistic branching is only finitary) to denote the set of such nondeterministic-probabilistic systems, and henceforth refer to $\bar{\mathrm{N}}\mathrm{P}$ systems.

We consider two ways in which a probabilistic system's computation may be represented. A *path* represents a particular resolution of both nondeterminism *and* probability. Formally, a path is a finite or infinite sequence of probabilistic transitions of the form $\omega = s_0 \xrightarrow{\mu_0} s_1 \xrightarrow{\mu_1} \cdots$. We denote by $\omega(i)$ the $(i+1)$th state of ω and $\mathit{last}(\omega)$ the last state of ω if ω is finite.

On the other hand, an *adversary* represents a particular resolution of nondeterminism *only*. Formally, an *adversary* of NP is a function A mapping every finite path ω to a distribution $\mu \in \mathit{Steps}(\mathit{last}(\omega))$. Let $\mathit{Adv}_{\mathsf{NP}}$ be the set of adversaries of NP. For any $A \in \mathit{Adv}_{\mathsf{NP}}$, let $\mathit{Path}^A_{\mathit{ful}}(s)$ denote the set of infinite paths associated with A starting in the state $s \in S$. Then, in the standard way, we define the measure Prob^A over $\mathit{Path}^A_{\mathit{ful}}(s)$ for each state $s \in S$ [KSK76].

2.2 Probabilistic Computation Tree Logic

The syntax of PCTL (Probabilistic Computation Tree Logic) [HJ94, BdA95] is defined as follows:

$$\phi ::= p \mid \neg p \mid \phi \vee \phi \mid \phi \wedge \phi \mid \mathbb{P}_{\sim\lambda}(\phi \mathcal{U} \phi) \mid \mathbb{P}_{\sim\lambda}(\Box\phi)$$

where p is an observable from some set P, $\sim \in \{<, \leq, \geq, >\}$ is a comparison operator, and λ is probability threshold. The sub-formula $\phi_1 \mathcal{U} \phi_2$ is the classical "until" path formula, with the usual abbreviation $\Diamond\phi \equiv \mathtt{true}\mathcal{U}\phi$, and $\Box\phi$ is the classical "globally" path formula. The symbol $\mathbb{P}_{\sim\lambda}$ is a quantifier over adversaries that permits reference to probability thresholds within the formula. The logic PCTL is useful to verify properties of the form "with probability at least 0.99, a data packet is delivered" ($\mathbb{P}_{\geq 0.99}(\Diamond \mathit{deliver})$), or "an error state is

reached with probability less than 0.01" ($\mathbb{P}_{<0.01}(\Diamond \mathit{error})$). In addition, for real-time systems, it can be used to verify time-bounded reachability properties, also known as *soft* deadlines, such as "with probability 0.975 or greater, a leader is elected within 100 time units" ($\mathbb{P}_{\geq 0.975}((\mathit{time} \leq 100)\, \mathcal{U}\ \mathit{leader})$).

Given an $\bar{\mathrm{N}}$P system $\mathsf{NP} = (S, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ and a set $\mathcal{A}$ of adversaries of NP, we define the satisfaction relation $\models_{\mathcal{A}}$ of PCTL as follows. Note that we write $\sqsubseteq \in \{<, \leq\}$ and $\sqsupseteq \in \{\geq, >\}$. Let $s \in S$ be a state of NP. The satisfaction relation for the observables and Boolean combinators is standard: that is, $s \models_{\mathcal{A}} p$ if and only if $s \in \langle\!\langle p \rangle\!\rangle$, $s \models_{\mathcal{A}} \neg p$ if and only if $s \in \langle\!\langle \bar{p} \rangle\!\rangle$, $s \models_{\mathcal{A}} \phi_1 \vee \phi_2$ if and only if $s \models_{\mathcal{A}} \phi_1$ or $s \models_{\mathcal{A}} \phi_2$, and $s \models_{\mathcal{A}} \phi_1 \wedge \phi_2$ if and only if $s \models_{\mathcal{A}} \phi_1$ and $s \models_{\mathcal{A}} \phi_2$. The satisfaction relation for the probabilistically quantified formulae is as follows:

$$\begin{array}{lllrcl}
s & \models_{\mathcal{A}} & \mathbb{P}_{\sqsubseteq \lambda}(\phi_1 \mathcal{U} \phi_2) & \Leftrightarrow & \mathit{Max}\mathcal{U}(\phi_1, \phi_2, \mathcal{A}, s) & \sqsubseteq \ \lambda \\
s & \models_{\mathcal{A}} & \mathbb{P}_{\sqsubseteq \lambda}(\Box \phi) & \Leftrightarrow & \mathit{Max}\Box(\phi, \mathcal{A}, s) & \sqsubseteq \ \lambda \\
s & \models_{\mathcal{A}} & \mathbb{P}_{\sqsupseteq \lambda}(\phi_1 \mathcal{U} \phi_2) & \Leftrightarrow & \mathit{Min}\mathcal{U}(\phi_1, \phi_2, \mathcal{A}, s) & \sqsupseteq \ \lambda \\
s & \models_{\mathcal{A}} & \mathbb{P}_{\sqsupseteq \lambda}(\Box \phi) & \Leftrightarrow & \mathit{Min}\Box(\phi, \mathcal{A}, s) & \sqsupseteq \ \lambda\ .
\end{array}$$

The *maximal (minimal) until probability* $\mathit{Max}\mathcal{U}(\phi_1, \phi_2, \mathcal{A}, s)$ ($\mathit{Min}\mathcal{U}(\phi_1, \phi_2, s)$) is defined as the maximal (respectively, minimal) probability with which "ϕ_1 until ϕ_2" can be satisfied by any adversary in $\mathcal{A}$, starting from the state s. The maximal and minimal globally probabilities are defined in an analogous manner. More formally, and using $\phi_1 \mathcal{U} \phi_2$ and $\Box \phi$ to express the until and globally path formulas, respectively, the semantics of which are defined in the usual way (see, for example, [CGP99]), we have:

$$\begin{array}{rcl}
\mathit{Max}\mathcal{U}(\phi_1, \phi_2, \mathcal{A}, s) & = & \sup_{A \in \mathcal{A}} \mathit{Prob}^A\{\omega \in \mathit{Path}^A_{\mathit{ful}}(s) \mid \omega \models_{\mathcal{A}} \phi_1 \mathcal{U} \phi_2\} \\
\mathit{Min}\mathcal{U}(\phi_1, \phi_2, \mathcal{A}, s) & = & \inf_{A \in \mathcal{A}} \mathit{Prob}^A\{\omega \in \mathit{Path}^A_{\mathit{ful}}(s) \mid \omega \models_{\mathcal{A}} \phi_1 \mathcal{U} \phi_2\}\ ,
\end{array}$$

with $\mathit{Max}\Box(\phi, \mathcal{A}, s)$ and $\mathit{Min}\Box(\phi, \mathcal{A}, s)$ defined in an analogous manner. The maximal and minimal until and globally probabilities can be obtained as solutions to linear programming problems in the case of finite systems [BdA95].

Note that dual of the until path formula $\phi_1 \mathcal{U} \phi_2$ is the *release* formula $(\neg \phi_1) \mathcal{V} (\neg \phi_2)$ (for the semantics of release, see for example [CGP99]). Next observe that, for any path formula φ, any state $s \in S$ and any adversary A:

$$\mathit{Prob}^A\{\omega \in \mathit{Path}^A_{\mathit{ful}}(s) \mid \omega \models_{\mathcal{A}} \varphi\} = 1 - \mathit{Prob}^A\{\omega \in \mathit{Path}^A_{\mathit{ful}}(s) \mid \omega \models_{\mathcal{A}} \neg \varphi\}\ .$$

Hence, we can rewrite the formula for the minimal until probability into a formula referring to the maximal probability of satisfying the dual release formula:

$$\begin{array}{rcl}
\mathit{Min}\mathcal{U}(\phi_1, \phi_2, \mathcal{A}, s) & = & \mathit{Max}\mathcal{V}(\neg \phi_1, \neg \phi_2, \mathcal{A}, s) \\
& = & \sup\limits_{A \in \mathcal{A}} \mathit{Prob}^A\{\omega \in \mathit{Path}^A_{\mathit{ful}}(s) \mid \omega \models_{\mathcal{A}} (\neg \phi_1) \mathcal{V} (\neg \phi_2)\}\ .
\end{array}$$

Finally, we note that $\Box \phi \equiv \mathtt{false} \mathcal{V} \phi$ and $\Box \phi \equiv \neg \Diamond \neg \phi$, and hence

$$\begin{array}{rcl}
\mathit{Max}\Box(\phi, \mathcal{A}, s) & = & \mathit{Max}\mathcal{V}(\mathtt{false}, \phi, \mathcal{A}, s) \\
\mathit{Min}\Box(\phi, \mathcal{A}, s) & = & \mathit{Max}\mathcal{U}(\mathtt{true}, \neg \phi, \mathcal{A}, s)\ .
\end{array}$$

To conclude, we can concentrate on algorithms for computing $\mathit{Max}\mathcal{U}$ and $\mathit{Max}\mathcal{V}$ in order to solve PCTL model checking.

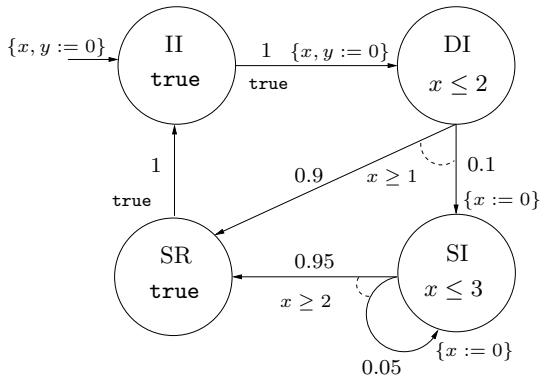

Figure 1: A probabilistic timed automaton modelling a probabilistic protocol.

2.3 Probabilistic timed automata

We now present probabilistic timed automata [KNSS02], which are classical timed automata [AD94, HNSY94] extended with probabilistic branching over the edges. Let $\mathcal{X}$ be a set of real-valued variables called *clocks*. Let $\mathit{Zones}(\mathcal{X})$ be the set of *zones* over $\mathcal{X}$, which are conjunctions of atomic constraints of the form $x \sim c$ and $x - y \sim c$, for $x, y \in \mathcal{X}$, $\sim \in \{<, \leq, \geq, >\}$, and $c \in \mathbb{N}$. A *probabilistic timed automaton* is a tuple $\mathsf{PTA} = (L, \mathcal{X}, \mathit{inv}, \mathit{prob}, \langle g_l \rangle_{l \in L})$, where:

- L is a finite set of *locations*;
- the function $\mathit{inv} : L \to \mathit{Zones}(\mathcal{X})$ is the *invariant condition*;
- the function $\mathit{prob} : L \to 2^{\mathsf{Dist}(L \times 2^{\mathcal{X}})}$ is the *probabilistic edge relation* such that $\mathit{prob}(l)$ is finite for all $l \in L$;
- for each $l \in L$, the function $g_l : \mathit{prob}(l) \to \mathit{Zones}(\mathcal{X})$ is the *enabling condition* for l.

A state of PTA is a pair (l, v) where $l \in L$ and $v \in \mathbb{R}^{|\mathcal{X}|}$. If the current state is (l, v), there is a nondeterministic choice of either letting *time pass* while satisfying the invariant condition $\mathit{inv}(l)$, or making a *discrete* transition according to any distribution in $\mathit{prob}(l)$ whose enabling condition $g_l(p)$ is satisfied. If the distribution $p \in \mathit{prob}(l)$ is chosen, then the probability of moving to l' and resetting all of the clocks in X to 0 is given by $p(l', X)$.

Example. Consider the PTA modelling a simple probabilistic communication protocol given in Figure 1. The nodes represent the locations: II (sender, receiver both idle); DI (sender has data, receiver idle); SI (sender sent data, receiver idle); and SR (sender sent data, receiver received). As soon as data has been received by the sender, the protocol moves to the location DI with probability 1. In DI, after between 1 and 2 time units, the protocol makes a transition either to SR with probability 0.9 (data received), or to SI with probability 0.1 (data lost). In SI, the protocol will attempt to resend the data after 2 to 3 time units, which again can be lost, this time with probability 0.05.

3 Symbolic probabilistic systems

3.1 NP$\bar{\text{N}}$ systems

We can also envisage different classes of system in which nondeterministic and probabilistic choice coexist; for example, the class of P$\bar{\text{N}}$ systems would make a transition by first making a probabilistic choice over a finite set of alternatives, and then making a choice over a possibly infinite set of nondeterministic alternatives. Indeed, we need not limit ourselves to transition comprising only two phases of choice; for this paper, we find it convenient to work with the class of NP$\bar{\text{N}}$ *systems*, in which transition are performed according to *three* phases of choice. An NP$\bar{\text{N}}$ system $\mathsf{NPN} = (S, \mathbf{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ comprises a set S of states, a set P of observations, and an observation function $\langle\!\langle \cdot \rangle\!\rangle$; however, in contrast with $\bar{\text{N}}$P systems, the NP$\bar{\text{N}}$ *transition function* $\mathbf{Steps} : S \to 2^{\mathsf{Dist}(2^S)}$ is defined such that, for each state $s \in S$, the set $\mathbf{Steps}(s)$ is *finite*, and, for each distribution $\nu \in \mathbf{Steps}(s)$, each $U \subseteq S$ such that $\nu(U) > 0$ is a possibly *infinite* set. That is, an NP$\bar{\text{N}}$ system makes an NP$\bar{\text{N}}$ *transition* $s \xrightarrow{\nu} t$ according to a three-phase choice:

1. the first phase comprises the nondeterministic selection of a distribution ν from the finite set $\mathbf{Steps}(s)$;
2. the second phase comprises a probabilistic choice of a state set $U \subseteq S$ according to ν (hence, we must have $\nu(U) > 0$);
3. the third phase comprises a nondeterministic choice of the target state $t \in U$.

3.1.1 From NP$\bar{\text{N}}$ systems to $\bar{\text{N}}$P systems.

As we regard $\bar{\text{N}}$P systems as a more natural, or "canonical" model for nondeterministic-probabilistic systems, we now show how an NP$\bar{\text{N}}$ system can be represented as an $\bar{\text{N}}$P system. Intuitively, the idea is that we push the third transition phase of NP$\bar{\text{N}}$ system to the first phase of choice. Hence, the possibly infinite nondeterministic choice in the third phase moves to the first phase, resulting in a first phase in which possibly infinitely many nondeterministic choices can be made, as in the definition of $\bar{\text{N}}$P systems. More formally, given an NP$\bar{\text{N}}$ system $\mathsf{NPN} = (S, \mathbf{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$, we construct its associated $\bar{\text{N}}$P system $\mathsf{NP} = (S, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$. The sets S and P of states and observables, and the observation function $\langle\!\langle \cdot \rangle\!\rangle$, are the same for NPN and NP. We now explain how $\mathbf{Steps}$ may be used to obtain Steps. For each state $s \in S$, let

$$\mathit{Steps}(s) = \bigcup_{\nu \in \mathbf{Steps}(s)} \mathit{Steps}^{\nu}(s) \ ,$$

where each $\mathit{Steps}^{\nu}(s)$ is defined in the following manner. Denote $\mathsf{support}(\nu) = \{U_1, ..., U_n\}$, and let $\mathsf{vectors}(s, \nu) = U_1 \times \cdots \times U_n$. Note that, if $(t_1, ..., t_n) \in \mathsf{vectors}(s, \nu)$, because it is possible that $U_i \cap U_j \neq \emptyset$ for some $1 \leq i, j \leq n$, we may also have $t_i = t_j$ for some $1 \leq i, j \leq n$. Then

$$\mathit{Steps}^{\nu}(s) = \{\mu_{\vec{t}} \in \mathsf{Dist}(S) \mid \vec{t} \in \mathsf{vectors}(s, \nu)\} \ ,$$

where, for each $\vec{t} = (t_1, ..., t_n)$ and for each state $t \in S$, we have:

$$\mu_{\vec{t}}(t) = \sum_{1 \leq i \leq n \ \& \ t = t_i} \nu(U_i) \ .$$

Now that we have obtained a $\bar{\text{N}}$P system from a NP$\bar{\text{N}}$ system, we can of course define the notions of its paths, adversaries, and satisfaction of PCTL formulae.

3.1.2 Finite-template NP$\bar{\text{N}}$ systems.

Let $\mathsf{NPN} = (S, \mathbf{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ be an NP$\bar{\text{N}}$ system, and let

$$\mathsf{Dist}(\mathsf{NPN}) = \bigcup_{s \in S} \mathbf{Steps}(s)$$

be the set of all distributions used in NPN. We say that two distributions $\nu, \nu' \in \mathsf{Dist}(2^S)$ are *isomorphic*, written $\nu \cong \nu'$, if and only if there exists a bijection $f : \mathsf{support}(\nu) \to \mathsf{support}(\nu')$ such that $\nu(U) = \nu'(f(U))$ for all $U \in \mathsf{support}(\nu)$ (clearly $\cong$ is an equivalence relation). We use $\mathsf{Dist}(\mathsf{NPN})/_{\cong}$ to denote the an $\cong$-quotient of $\mathsf{Dist}(\mathsf{NPN})$ such that, for each state $s \in S$, then each pair $\nu, \nu' \in \mathbf{Steps}(s)$ of distributions belong to different equivalence classes of $\mathsf{Dist}(\mathsf{NPN})/_{\cong}$. We then say that NPN is a *finite-template* NP$\bar{\text{N}}$ *system* if there exists a finite $\mathsf{Dist}(\mathsf{NPN})/_{\cong}$. The intuition is that an equivalence class $C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$ denotes a "template" for the distributions within C, in which the support sets of the distributions are immaterial, but the probabilities are paramount.

3.1.3 Example: probabilistic timed automata.

We show how NP$\bar{\text{N}}$ systems may be use to represent probabilistic timed automata. Note that the semantics of probabilistic timed automata are traditionally represented in terms of $\bar{\text{N}}$P systems [KNSS02].

Let $\mathsf{PTA} = (L, \mathcal{X}, \mathit{inv}, \mathit{prob}, \langle g_l \rangle_{l \in L})$ be a probabilistic timed automaton. A point $v \in \mathbb{R}^{|\mathcal{X}|}$ is referred to as a *clock valuation*. For $v \in \mathbb{R}^{|\mathcal{X}|}$ and $\eta \in \mathbb{R}_{\geq 0}$, the clock valuation $v + \eta$ is obtained from v by adding η to the value of each clock; and, for any $X \subseteq \mathcal{X}$, the clock valuation $v[X := 0]$ is obtained from v by resetting all clocks in X to 0. The clock valuation v *satisfies* the zone ζ, written $v \models^{\mathsf{zone}} \zeta$, if and only if ζ resolves to true after substituting each $x \in \mathcal{X}$ with the corresponding value v_x from v.

The NP$\bar{\text{N}}$ system $\mathsf{NPN} = (S, \mathbf{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ associated with PTA is defined in the following way.

- Let $S = \{(l, v) \mid l \in L \text{ and } v \models^{\mathsf{zone}} \mathit{inv}(l)\}$.
- For each state $(l, v) \in S$, let

  $$\mathbf{Steps}(l, v) = \{\nu_{\mathit{time}}\} \cup \{\nu_p \mid p \in \mathit{prob}(l) \wedge v \models^{\mathsf{zone}} g_l(p)\} \cup \{\nu_{\mathit{time}}\} \ ,$$

  where, $\nu_{\mathit{time}}(U) = 1$ for

  $$U = \{(l, v + \eta) \mid \eta \geq 0 \wedge \forall\, 0 \leq \eta' \leq \eta \ . \ v + \eta' \models^{\mathsf{zone}} \mathit{inv}(l)\} \ .$$

and for each $p \in prob(l)$:

$$\nu_p(l', v') = \sum_{X \subseteq \mathcal{X} \wedge v' = v[X:=0]} p(l', X) \ .$$

- Let $P \subseteq 2^{L \times Zones(\mathcal{X})}$.
- Let $\langle\!\langle l, \zeta \rangle\!\rangle = \{(l', v) \in S \mid l = l' \wedge v \models^{\mathsf{zone}} \zeta\}$ for each $(l, \zeta) \in P$.

It can be verified that NPN corresponds to an $\bar{\text{N}}$P system which defines the semantics of probabilistic timed automata, as presented in [KNSS02]. Also note that NPN is a finite-template NP$\bar{\text{N}}$ system. We note that hybrid automata with probabilistic branching over edges can also be represented as NP$\bar{\text{N}}$ systems; indeed, the notion of resetting continuous variables within intervals upon traversal of an edge, as seen in polyhedral hybrid automata [AHH96], uses the full generality of NP$\bar{\text{N}}$ systems.

3.2 Symbolic bi-labelled structures

3.2.1 Bi-labelled structures.

A *bi-labelled structure* $\mathsf{B} = (S, L_1, L_2, \Gamma_1, \Gamma_2, \delta, P, {}^{\sqcap}\cdot{}^{\sqcup})$ is a tuple comprising:

- a possibly infinite set S of *states*,
- two finite sets L_1, L_2 of *transition labels*,
- two *label assignments* $\Gamma_1 : S \to 2^{L_1} \setminus \emptyset$, $\Gamma_2 : S \to 2^{L_2} \setminus \emptyset$ defining the set of labels permissible in each state,
- a partial *transition function* $\delta : S \times L_1 \times L_2 \to 2^S$ assigning to each state $s \in S$ and labels $a \in L_1(s), b \in L_2(s)$ a possibly infinite set of successor states $\delta(s, a, b)$,
- a finite set P of *observables*,
- an *observation function* ${}^{\sqcap}\cdot{}^{\sqcup} : P \to 2^S$ which maps every observable to the set of states in which it is observed.

For each observable $p \in P$, we require that there exists a complementary observable $\bar{p} \in P$ such that ${}^{\sqcap}\bar{p}{}^{\sqcup} = S \setminus {}^{\sqcap}p{}^{\sqcup}$. [1]

[1] The reader may notice that bi-labelled structures and (infinite-state, 2-player) *game structures* [HHM99, dAHM01] are essentially equivalent. That is, in any state $s \in S$, player 1 makes a choice of its move by choosing a label $a \in \Gamma_1(s)$, and similarly player 2 makes a choice of its move by choosing a label from $b \in \Gamma_2(s)$; then the game moves to a state $t \in \delta(s, a, b)$. Game structures are used in the context of adversarial relationships between system components and their environment, which differs from our aim of studying probabilistic behaviour, and therefore we have changed the name to avoid semantic confusion.

3.2.2 From NP$\bar{\text{N}}$ systems to bi-labelled structures.

In this subsection and the next, we show how bi-labelled structures relate to finite-template NP$\bar{\text{N}}$ systems. The construction is defined such that L_1-labels refer to the first phase of nondeterministic choice in an NP$\bar{\text{N}}$ transition, whereas L_2-labels refer to the second phase of probabilistic choice. For a finite-template NP$\bar{\text{N}}$ system $\mathsf{NPN} = (S, \mathbf{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$, we define an associated bi-labelled structure $\mathsf{B}(\mathsf{NPN}) = (S, L_1, L_2, \Gamma_1, \Gamma_2, \delta, P, \ulcorner\!\urcorner \cdot \ulcorner\!\urcorner)$ in the following way.

- The sets of states and observables are the same in NPN and B, and we let $\ulcorner\!\urcorner \cdot \ulcorner\!\urcorner = \langle\!\langle \cdot \rangle\!\rangle$.
- Let $L_1 = \{a_C \mid C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}\}$ be a set of labels; that is, there is a distinct label in L_1 for each of the equivalence classes distributions of $\mathsf{Dist}(\mathsf{NPN})/_{\cong}$. The label set L_2 is defined as any finite set $\{b_1, b_2, ...\}$ of labels with cardinality greater than $\max_{\nu \in \mathsf{Dist}(\mathsf{NPN})} |\mathsf{support}(\nu)|$, the maximum branching degree of the distributions of NPN.
- For each state $s \in S$, we define the label assignments by

  $$\Gamma_1(s) = \{a_C \mid C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong} \text{ s.t. } \exists \nu \in \mathbf{Steps}(s) \text{ for which } \nu \in C\}$$

  and $\Gamma_2(s) = \{b_1, ..., b_n\}$ where $n = \max_{\nu \in \mathbf{Steps}(s)} |\mathsf{support}(\nu)|$.
- For each state $s \in S$, and each pair of distributions ν, ν' in the same $\cong$-class $C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$, we index the support sets $\mathsf{support}(\nu) = \{U_1, ..., U_m\}$ and $\mathsf{support}(\nu') = \{U'_1, ..., U'_m\}$ such that $\nu(U_i) = \nu'(U'_i)$ for all $1 \leq i \leq m$ (which is possible because ν and ν' are isomorphic). Then, for each distribution $\nu \in \mathbf{Steps}(s)$, where $\nu \in C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$, we let $\delta(s, a_C, b_i) = U_i$ for each $1 \leq i \leq m$, and $\delta(s, a_C, b_i) = \emptyset$ for each $m < i \leq n$.

Finally, by abuse of notation, for any $\cong$-class $C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$, we denote by $C(b_i)$ the probability that a distribution belonging to the class C assigns to the ith element in its support (the ith element will correspond to label b_i by the above construction of $\mathsf{B}(\mathsf{NPN})$).

3.2.3 Symbolic theories.

We proceed to define the notion of symbolic theory of bi-labelled structures, following closely the precedent of symbolic theories for non-probabilistic systems [dAHM01, HMR03]. A *symbolic theory* $(R, \ulcorner \cdot \urcorner)$ for a bi-labelled structure B consists of a possibly infinite set of region R paired with an *extension function* $\ulcorner \cdot \urcorner : R \to 2^S$ mapping each region $\sigma \in R$ to a possibly infinite set of states $\ulcorner \sigma \urcorner$, such that the following four conditions hold:

1. we have $P \subseteq R$ (every observable is a region), and $\ulcorner\!\urcorner p \ulcorner\!\urcorner = \ulcorner p \urcorner$ for all observables $p \in P$ ($\ulcorner\!\urcorner \cdot \ulcorner\!\urcorner$ and $\ulcorner \cdot \urcorner$ agree on all observables). We also include the regions $\mathtt{true}, \mathtt{false} \in R$ where $\ulcorner \mathtt{true} \urcorner = S$ and $\ulcorner \mathtt{false} \urcorner = \emptyset$.

2. For each pair $\sigma, \tau \in R$ of regions, we have regions $\mathsf{And}(\sigma,\tau) \in R, \mathsf{Or}(\sigma,\tau) \in R$, and $\mathsf{Diff}(\sigma,\tau) \in R$, such that $\mathsf{And}(\sigma,\tau) = \ulcorner\sigma\urcorner \cap \ulcorner\tau\urcorner$, $\mathsf{Or}(\sigma,\tau) = \ulcorner\sigma\urcorner \cup \ulcorner\tau\urcorner$, and $\mathsf{Diff}(\sigma,\tau) = \ulcorner\sigma\urcorner \setminus \ulcorner\tau\urcorner$. Furthermore, the functions $\mathsf{And} : R \times R \to R$, $\mathsf{Or} : R \times R \to R$ and $\mathsf{Diff} : R \times R \to R$ are computable.

3. For each region $\sigma \in R$ and each pair $a \in L_1$, $b \in L_2$, there is a region $\mathsf{Pre}^{a,b}(\sigma) \in R$ such that $\ulcorner\mathsf{Pre}^{a,b}(\sigma)\urcorner =$

   $$\{s \in S \mid a \in \Gamma_1(s) \text{ and } b \in \Gamma_2(s) \text{ and } \exists t \in \delta(s,a,b) \text{ such that } t \in \ulcorner\sigma\urcorner\} .$$

   Furthermore, the function $\mathsf{Pre} : R \times L_1 \times L_2 \to R$ is computable.

4. There exist computable functions $\mathsf{Empty} : R \to \mathbb{B}$ and $\mathsf{Member} : S \times R \to \mathbb{B}$ such that $\mathsf{Empty}(\sigma)$ if and only if $\ulcorner\sigma\urcorner = \emptyset$ and $\mathsf{Member}(s,\sigma)$ if and only if $s \in \ulcorner\sigma\urcorner$ (all emptiness and membership questions about regions can be decided).

The tuple $(R, P, \mathsf{And}, \mathsf{Or}, \mathsf{Diff}, \mathsf{Pre}, \mathsf{Empty})$ is called a *region algebra* for B.

3.2.4 Example: probabilistic timed automata.

It is not difficult to obtain a bi-labelled structure representation of a timed or hybrid automaton (indeed, this is made explicit in the context of timed and hybrid games in [HHM99, dAHM01]). We note briefly that the finite-template NP$\bar{\text{N}}$ system of a probabilistic timed or hybrid automaton may be used to obtain a bi-labelled structure using the technique presented in the previous subsection. Furthermore, symbolic theories for probabilistic timed automata, using the classical *zone*-based representation of regions [HNSY94], and for probabilistic *polyhedral* hybrid automata, using the classical polyhedra-based representation of regions [AHH96], are available for the resulting bi-labelled structures, as made explicit in [dAHM01, HMR03].

3.3 Symbolic probabilistic systems

A *symbolic probabilistic system* $\mathsf{SPS} = (\mathsf{NPN}, R, \ulcorner\cdot\urcorner)$ comprises a finite-template NP$\bar{\text{N}}$ system NPN, and a symbolic theory $(R, \ulcorner\cdot\urcorner)$ for a bi-labelled structure $\mathsf{B}(\mathsf{NPN})$ corresponding to NPN.

4 PCTL model checking

In this section, we show how symbolic probabilistic systems may be model checked against PCTL formulae. In the manner standard for model checking, we progress up the parse tree of a PCTL formula, from the leaves to the root, recursively calling the symbolic semi-algorithm PCTLModelCheck, shown in Figure 2, for each sub-formula. (Note that we refer to PCTLModelCheck as a semi-algorithm because for finite-template NP$\bar{\text{N}}$ systems the model checking algorithm is semi-decidable.) Handling observables and Boolean operations is classical, and we therefore reduce our problem to computing the functions

Symbolic semi-algorithm PCTLModelCheck
input: $(R, P, \mathsf{And}, \mathsf{Or}, \mathsf{Diff}, \mathsf{Pre}, \mathsf{Empty})$
PCTL formula ϕ

output: $[\phi] :=$
if $\phi = p$ **then return** p;
if $\phi = \neg p$ **then return** $\bar{p}$;
if $\phi = \phi_1 \vee \phi_2$ **then return** $\mathsf{Or}([\phi_1], [\phi_2])$;
if $\phi = \phi_1 \wedge \phi_2$ **then return** $\mathsf{And}([\phi_1], [\phi_2])$;
if $\phi = \mathbb{P}_{\sim\lambda}(\phi_1 \mathcal{U} \phi_2)$ **then return** $\mathit{Until}(\phi_1, \phi_2, \sim, \lambda)$;
if $\phi = \mathbb{P}_{\sim\lambda}(\Box\phi)$ **then return** $\mathit{Globally}(\phi, \sim, \lambda)$;

Figure 2: PCTL model checking for symbolic probabilistic systems

$\mathit{Until}(\phi_1, \phi_2, \sim, \lambda)$ and $\mathit{Globally}(\phi, \sim, \lambda)$ which arise when we check an probabilistically quantified formula. The former function relies on the computation of maximal or minimal until probabilities, whereas the latter relies on the computation of maximal or minimal globally probabilities. We present a method for computing the maximal until probabilities in the next section, which, using the duality result mentioned in Section 2.2, also can be used for computing the minimal globally probabilities. Then, in Section 4.2, we present a method for computing the maximal release probabilities, which can be used for computing the maximal globally probabilities, and, again by duality, the minimal until probabilities.

4.1 The maximal probability of until

The semi-algorithm of [KNS01], which computes the maximal probability with which a certain state set of a symbolic probabilistic system can be reached, can be extended to deal with until formulae in the following way: first, the "target set" of states of the previous algorithm corresponds to the region $[\phi_2]$ which satisfies ϕ_2; secondly, the backwards search through the state space, which commences from $[\phi_2]$, is now restricted to the set of states which satisfy ϕ_1, as represented by the And operations which conjunct the generated predecessor regions with the region $[\phi_1]$. The termination condition $\ulcorner T_{i+1} \urcorner \subseteq \ulcorner T_i \urcorner$, which is shorthand for $\{\ulcorner \sigma \urcorner \mid \sigma \in T_i\} \subseteq \{\ulcorner \sigma \urcorner \mid \sigma \in T_{i+1}\}$, reflects the fact that the algorithm computes progressively larger sets of states (as in a classical *least fix-point* expression). If a fix-point is reached, then the graph (T_{i+1}, E_{i+1}) is returned. The edge set E_{i+1}, is then "extended" to generate the new edge set E such that, for every pair regions $\sigma, \sigma' \in T_{i+1}$, if $\ulcorner \sigma' \urcorner \subseteq \ulcorner \sigma \urcorner$ and $(\sigma, (a, b), \tau) \in E_{i+1}$, then $(\sigma', (a, b), \tau) \in E$ (see [KNS01] for details). For simplicity, we henceforth drop the subscript on T_{i+1}.

The graph (T, E) is then used to construct the finite-state $\bar{\text{N}}$P system $\mathsf{NP} = (T, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$. The construction is similar to the corresponding construction

Symbolic semi-algorithm MaxUntil
input: $(R, P, \mathsf{And}, \mathsf{Or}, \mathsf{Diff}, \mathsf{Pre}, \mathsf{Empty})$
until formula $\phi_1 \mathcal{U} \phi_2$
$T_0 := [\phi_2]$
$E_0 := \emptyset$
for $i = 0, 1, 2, \ldots$ **do**
$T_{i+1} := T_i$
for all $a \in L_1, b \in L_2 \wedge \sigma \in T_i$ **do**
$\sigma' := \mathsf{And}(\mathsf{Pre}^{a,b}(\sigma), [\phi_1])$
$T_{i+1} := \{\sigma'\} \cup T_{i+1}$
$E_{i+1} := \{(\sigma', (a, b), \sigma)\} \cup E_{i+1}$
$T_{i+1} := \{\mathsf{And}(\sigma', \tau) \mid \tau \in T_{i+1}\} \cup T_{i+1}$ $\quad (\star)$
end for all
until $\ulcorner T_{i+1} \urcorner \subseteq \ulcorner T_i \urcorner$
return (T_{i+1}, E_{i+1})

Figure 3: State-space exploration for until formulae

in [KNS01], and also to the release case presented below, so we proceed to the main result concerning until formulae.

Proposition 1 *For the symbolic probabilistic system* $\mathsf{SPS} = (\mathsf{NPN}, R, \ulcorner \cdot \urcorner)$*, the until formula* $\phi_1 \mathcal{U} \phi_2$*, and the finite-state* $\bar{\mathrm{N}}\mathrm{P}$ *system* NP *constructed from the semi-algorithm* MaxUntil*, then for any state* $s \in S$ *of* NPN*, we have:*

$$Max\mathcal{U}(\phi_1, \phi_2, Adv_{\mathsf{NPN}}, s) = \max_{\sigma \in T \wedge s \in \ulcorner \sigma \urcorner} Max\mathcal{U}(\phi_1, \phi_2, Adv_{\mathsf{NP}}, \sigma) \; .$$

Note that this proposition refers to the full adversary sets Adv_{NPN} and Adv_{NP} of NPN and NP.

Using these results, we are in a position to return the set of regions denoted by $Until(\phi_1, \phi_2, \sqsubseteq, \lambda)$ for $\sqsubseteq \in \{<, \leq\}$. That is, using the classical probabilistic model checking methods of [BdA95], we first compute $Max\mathcal{U}(\phi_1, \phi_2, Adv, \sigma)$; next, we compute the set of regions $T_{\sqsubseteq\lambda} \subseteq T$ such that $\sigma \in T_{\sqsubseteq\lambda}$ if and only if $Max\mathcal{U}(\phi_1, \phi_2, Adv, \sigma) \sqsubseteq \lambda$; finally, we let $Until(\phi_1, \phi_2, \sqsubseteq, \lambda) = T_{\sqsubseteq\lambda}$.

Similarly, we can return the set of regions denoted by $Globally(\phi, \sqsupseteq, \lambda)$ for $\sqsupseteq \in \{\geq, >\}$. We first compute $Max\mathcal{U}(\mathtt{true}, \neg\phi, Adv, \sigma)$ for each $\sigma \in T$; next, we compute the set of regions $T_{\sqsupseteq\lambda} \subseteq T$ such that $\sigma \in T_{\sqsupseteq\lambda}$ if and only if $1 - Max\mathcal{U}(\mathtt{true}, \neg\phi, Adv, \sigma) \sqsupseteq \lambda$; finally, we let $Globally(\phi, \sqsupseteq, \lambda) = T_{\sqsupseteq\lambda}$.

4.2 The maximal probability of release

We now present a method for computing the maximal probability with which a symbolic probabilistic system satisfies a release property. An algorithm for the analysis of the bi-labelled structure $\mathsf{B}(\mathsf{NPN})$ corresponding to a symbolic

Symbolic semi-algorithm MaxRelease
input: $(R, P, \mathsf{And}, \mathsf{Or}, \mathsf{Diff}, \mathsf{Pre}, \mathsf{Empty})$
release formula $\phi_1 \mathcal{V} \phi_2$
$T_0 := [\phi_2]$
$E_0 := \emptyset$
for $i = 0, 1, 2, \ldots$ **do**
$T_{i+1} := [\phi_1 \wedge \phi_2]$
for all $a \in L_1, b \in L_2 \wedge \sigma \in T_i$ **do**
$\sigma' := \mathsf{And}(\mathsf{Pre}^{a,b}(\sigma), [\phi_2])$
$T_{i+1} := \{\sigma'\} \cup T_{i+1}$
$E_{i+1} := \{(\sigma', (a, b), \sigma)\} \cup E_{i+1}$
$T_{i+1} := \{\mathsf{And}(\sigma', \tau) \mid \tau \in T_{i+1}\} \cup T_{i+1}$ $\quad (\star)$
end for all
until $\ulcorner T_{i+1} \urcorner \supseteq \ulcorner T_i \urcorner$
return (T_{i+1}, E_{i+1})

Figure 4: State-space exploration for release formulae

probabilistic system is shown in Figure 4. Like the semi-algorithm MaxUntil, of Figure 3, the semi-algorithm MaxRelease iterates successively conjunction and predecessor operations. The region $[\phi_2]$ is taken as the initial region; to see why, consider the fact that $\phi_1 \mathcal{V} \phi_2 \equiv \phi_2 \wedge (\phi_1 \vee \mathsf{X}(\phi_1 \mathcal{V} \phi_2))$. Hence, the semi-algorithm MaxRelease proceeds by iterating predecessor and intersection operations from the initial region $[\phi_2]$, at each stage taking the region $[\phi_2 \wedge \phi_1]$ and the intersection of the predecessor regions of the previous stage with $[\phi_2]$. The termination condition $\ulcorner T_{i+1} \urcorner \supseteq \ulcorner T_i \urcorner$ reflects the fact that the algorithm computes progressively smaller sets of states (as in a classical *greatest fix-point* expression). If a fix-point is reached, then the graph (T_{i+1}, E_{i+1}) is returned, and the set of edges E_{i+1} is extended to the set E using the same methodology as presented in Section 4.1. We drop the subscript also on T and henceforth use (T, E) to refer to the graph generated by MaxRelease.

Next, we construct a finite-state $\bar{\text{N}}$P system $\mathsf{NP} = (T, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ from (T, E). The state set T is the set of generated regions, the set P of observables is $\{\phi_1, \bar{\phi_1}, \phi_2, \bar{\phi_2}\}$, and $\langle\!\langle \phi_i \rangle\!\rangle = [\phi_i]$ and $\langle\!\langle \bar{\phi_i} \rangle\!\rangle = T \setminus [\phi_i]$ for $i \in \{1, 2\}$. In contrast to our usual presentation of $\bar{\text{N}}$P systems, the transition relation $\mathit{Steps} : T \to 2^{\mathsf{SubDist}}(T)$ uses *sub-distributions*, which are distributions which need not sum to 1; formally, a sub-distribution π is a function $\pi : T \to [0, 1]$ such that $\sum_{\sigma \in T} \pi(\sigma) \leq 1$. Then, for any region $\sigma \in T$, let $\pi \in \mathit{Steps}(\sigma)$ if and only if there exists an equivalence class $C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$ of distributions of NPN, and there exists a subset $E_\pi \subseteq E$ of edges such that:

- all edges of E_π have the same source regions (that is, $(\sigma', (a, b), \tau) \in E_\pi$ implies $\sigma' = \sigma$);
- all edges of E_π have the same L_1-label, which is a_C (that is, $(\sigma', (a, b), \tau) \in$

E_π implies $a = a_C$);

- all edges of E_π have distinct L_2-labels (that is, if $(\sigma', (a, b), \tau'), (\sigma, (a, b'), \tau')$ are distinct edges, then $b \neq b'$);
- the set E_π is maximal;
- for all regions $\tau \in T$, we have

$$\pi(\tau) = \sum_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) \ .$$

Proposition 2 *For the symbolic probabilistic system* $\mathsf{SPS} = (\mathsf{NPN}, R, \ulcorner \cdot \urcorner)$*, the release formula* $\phi_1 \mathcal{V} \phi_2$*, and the finite-state* $\bar{\mathrm{N}}\mathrm{P}$ *system* NP *constructed from the semi-algorithm* MaxRelease*, then for any state* $s \in S$ *of* NPN*, we have:*

$$Max\mathcal{V}(\phi_1, \phi_2, Adv_{\mathsf{NPN}}, s) = \max_{\sigma \in T \wedge s \in \ulcorner \sigma \urcorner} Max\mathcal{V}(\phi_1, \phi_2, Adv_{\mathsf{NP}}, \sigma) \ .$$

Using these results, we are in a position to return the set of regions denoted by $Until(\phi_1, \phi_2, \sqsupseteq, \lambda)$ for $\sqsubseteq \in \{\geq, >\}$. That is, using classical probabilistic model checking methods [BdA95, dA97], we first compute $Max\mathcal{V}(\neg\phi_1, \neg\phi_2, Adv_{\mathsf{NP}}, \sigma)$; next, we compute the set of regions $T_{\sqsubseteq\lambda} \subseteq T$ such that $\sigma \in T_{\sqsubseteq\lambda}$ if and only if $1 - Max\mathcal{V}(\neg\phi_1, \neg\phi_2, Adv_{\mathsf{NP}}, \sigma) \sqsupseteq \lambda$; finally, we let $Until(\phi_1, \phi_2, \sqsupseteq, \lambda) = T_{\sqsupseteq\lambda}$.

Similarly, we can return the set of regions denoted by $Globally(\phi, \sqsubseteq, \lambda)$ for $\sqsubseteq \in \{<, \leq\}$. We first compute $Max\mathcal{V}(\mathtt{false}, \phi, Adv, \sigma)$ for each $\sigma \in T$; next, we compute the set of regions $T_{\sqsubseteq\lambda} \subseteq T$ such that $\sigma \in T_{\sqsubseteq\lambda}$ if and only if $Max\mathcal{V}(\mathtt{false}, \phi, Adv, \sigma) \sqsubseteq \lambda$; finally, we let $Globally(\phi, \sqsubseteq, \lambda) = T_{\sqsubseteq\lambda}$.

4.3 Decidability of PCTL model checking

The termination of the semi-algorithm depends on the termination of the fixpoint algorithms MaxUntil and MaxRelease presented in Figure 3 and Figure 4. As both of these algorithms iterate progressively conjunction and predecessor operations, if a bi-labelled structure of a symbolic probabilistic system is closed under such operations, starting from the set P of observables, then both MaxUntil and MaxRelease, and hence PCTLModelChecking, will terminate.

Consider a bi-labelled structure B(NPN) of a symbolic probabilistic system. Let $\preceq$ be a binary relation on the state space S of NPN such that $s \preceq t$ implies:

1. for all observables $p \in P$, we have $s \in \ulcorner p \urcorner$ if and only if $t \in \ulcorner p \urcorner$;
2. for all $a \in L_1$, $b \in L_2$ and $s' \in \delta(s, a, b)$, there exists $t' \in \delta(t, a, b)$ such that $s' \preceq t'$.

We call such a relation a *bi-labelled simulation*. Let $\approx$ be an equivalence relation on the state space S such that $s \approx t$ if there exist bi-labelled simulations $\preceq, \preceq'$ such that $s \preceq t$ and $t \preceq' s$. We call such an equivalence $\approx$ a *bi-labelled mutual simulation*, and write that $\approx$ has *finite index* if there are finitely many equivalence classes of $\approx$. A symbolic probabilistic system $(\mathsf{NPN}, R, \ulcorner \cdot \urcorner)$ has a finite

bi-labelled mutual simulation quotient if there exists a bi-labelled structure $\mathsf{B}(\mathsf{NPN})$ with a finite bi-labelled mutual simulation quotient. The following result follows from similar conclusions in the non-probabilistic setting [HMR03], which state that closure of P under conjunction, union and predecessor operations characterizes simulation on (symbolic) transition systems, and in the maximal probabilistic reachability setting [KNS01].

Theorem 3 PCTL *model checking is decidable for symbolic probabilistic systems with a finite bi-labelled mutual simulation quotient.*

Note that this result contrasts with the analogous results in the non-probabilistic context, in which CTL model checking is decidable for symbolic transition systems with a finite bisimulation quotient [HMR03].

4.4 Probabilistic timed automata and time-divergence

The application of the semi-algorithms MaxUntil and MaxRelease to the symbolic probabilistic system of a probabilistic timed automaton is clear; however, we would like to consider only *time-divergent* adversaries, which let the elapsed time on a path diverge with probability 1 (see, for example, [KNSS02]). In particular, we note that the distinction between adversaries which let time diverge and those which do not is critical for the computation of $Max\Box(\phi, \_, \_)$, because of the presence of adversaries which let time converge while staying in ϕ, therefore trivially making $Max\Box(\phi, \_, \_) = 1$. Indeed, for formula of the form $\mathbb{P}_{\sqsubseteq\lambda}(\Box\phi)$, we restrict ourselves to the cases when $\phi = p$ for some observable p which contains at most one pair $(l, \_)$ for each location $l \in L$, as our approach relies of the *convexity* of zones generated during the state-space exploration.

First, we assume the following condition: that for all states of a probabilistic timed automaton PTA, for any adversary which makes discrete (edge-traversal) transitions infinitely often, there exists an divergent adversary which makes the same discrete choices. We can then proceed to construct the finite-state $\bar{\text{N}}\text{P}$ system NP according to the methodology of Section 4.2. However, we *remove* all self-loops of regions generated by the *time* label, except for those regions which have zone components which are unbounded from above (that is, those regions $(\_, \zeta)$ for which, for every clock valuation $v \models^{\mathsf{zone}} \zeta$ and every $\eta \geq 0$, we have $v + \eta \in \zeta$).

Proposition 4 *For a probabilistic timed automaton subject to the assumption of the previous paragraph, with its associated symbolic probabilistic system* $(\mathsf{NPN}, R, \ulcorner\cdot\urcorner)$*, a formula* $\Box p$*, and the finite-state* $\bar{\text{N}}\text{P}$ *system* NP *constructed from the semi-algorithm* MaxRelease*, then for any state* $s \in S$ *of* NPN*, we have:*

$$Max\mathcal{V}(\mathtt{false}, p, Adv^{div}_{\mathsf{PTA}}, s) = \max_{\sigma \in T \wedge s \in \ulcorner\sigma\urcorner} Max\mathcal{V}(\mathtt{false}, p, Adv_{\mathsf{NP}^{red}}, \sigma) \ .$$

We now return to the probabilistic timed automaton in Figure 1 to find the minimal probability of a message being correctly delivered within 4 time units of the data arriving at the sender (reaching $\langle \text{SR}, y{<}4 \rangle\rangle$ from $\langle \text{DI}, x{=}y{=}1 \rangle\rangle$).

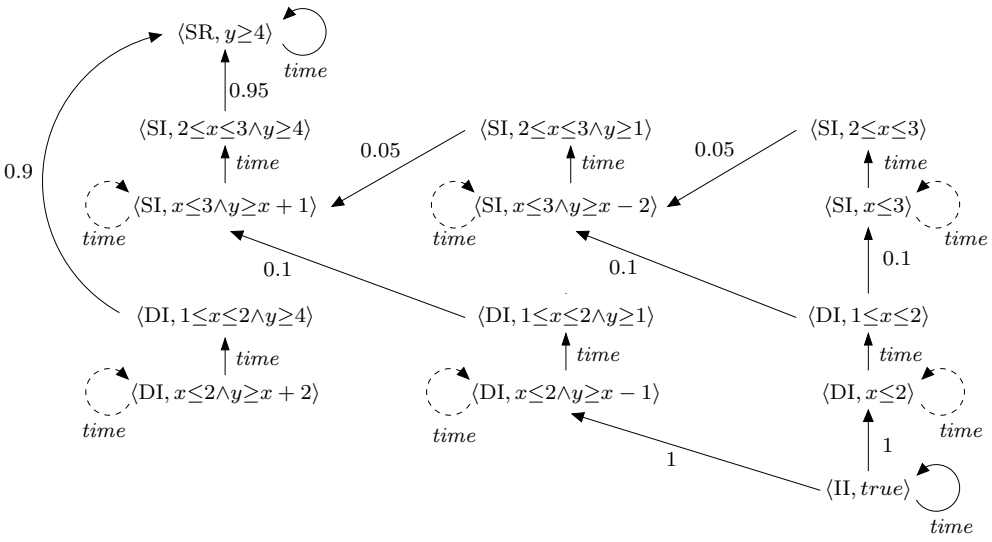

Figure 5: Graph generated by the algorithm MaxRelease

Following our methodolgy for PCTL model checking, we first calculate the maximum probability of remaining in the set of states

$$I = \{\langle \mathrm{SR}, y{\geq}4\rangle, \langle \mathrm{SI}, \mathtt{true}\rangle, \langle \mathrm{DI}, \mathtt{true}\rangle, \langle \mathrm{II}, \mathtt{true}\rangle\} \ ,$$

and derive the minimal probability of reaching $\langle \mathrm{SR}, y{<}4)\rangle$ as 1 minus this computed probability.

Therefore we apply the algorithm MaxRelease with ϕ_1 equal to `false` and ϕ_2 set to the formula which represents the set of states I. Applying this algorithms returns the graph given in Figure 5 from which we can then construct the probabilistic system on which we can calculate this maximum probability. Note that, as explained above, to limit our anaylsis to divergent adversaries we must remove the self-loops generated by the *time* label from those regions whose zone component is bounded. For this example such self-loops are represented with the dotted arrows, and hence these edges are ignored in the construction of the probabilistic system. By verifying the constructed probabilistic system, we find that the maximimum probability of remaining in this set of states after data arrives at the sender (that is, from a region containing the state $\langle \mathrm{DI}, x{=}y{=}0\rangle$), is 0.9. To illustrate this result, in Figure 6 we have represented the choices of an adversary which admits this maximal probability. Note that, since $\langle \mathrm{SI}, 2{\leq}x{\leq}3 \wedge y{\geq}4\rangle \subset \langle \mathrm{SI}, 2{\leq}x{\leq}3 \wedge y{\geq}1\rangle$, the transition from $\langle \mathrm{SI}, 2{\leq}x{\leq}3 \wedge y{\geq}4\rangle$ to $\langle \mathrm{SI}, x{\leq}3 \wedge y{\geq}x{+}1\rangle$ is generated from from the edge

$$(\langle \mathrm{SI}, 2{\leq}x{\leq}3 \wedge y{\geq}1\rangle, 0.05, \langle \mathrm{SI}, x{\leq}3 \wedge y{\geq}x{+}1\rangle)$$

of the graph in Figure 5.

Finally, we conclude that the minimal probability of a message being correctly delivered within 4 time units of the data arriving at the sender is $1{-}0.9 = 0.1$.

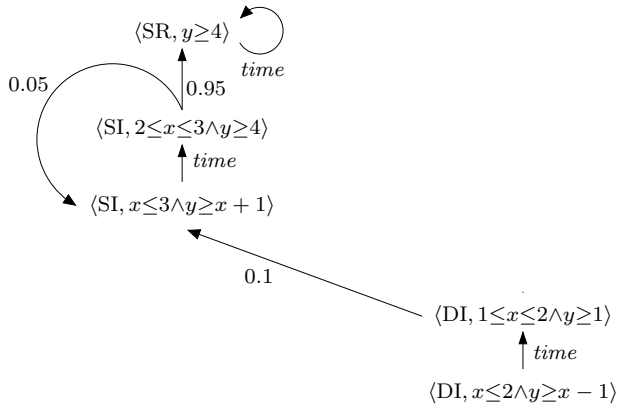

Figure 6: Adversary which admits the maximal probability

5 Conclusions

We have presented a method for model checking PCTL properties of symbolic probabilistic systems. The decidability result of Theorem 3 is of interest, and highlights differences between the probabilistic quantification over adversaries of PCTL and the quantification over paths in CTL. Note also that we reduce the problem of computing the minimum probability of satisfying an until formula to a PCTL model checking problem on a finite-state structure, which has a time complexity which is polynomial in the size of the system and linear in the size of the formula.

Acknowledgements

We would like to thank an anonymous referee of a previous version of this paper for helpful advice.

References

[AD94] R. Alur and D. L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.

[AHH96] R. Alur, T. A. Henzinger, and P.-H. Ho. Automatic symbolic verification of embedded systems. *IEEE Transactions on Software Engineering*, 22(3):181–201, 1996.

[AR03] P. A. Abdulla and A. Rabinovich. Verification of probabilistic systems with faulty communication. In A. Gordon, editor, *Proc. Foundations of Software Science and Computational Structures (FOSSACS 2003)*, volume 2620 of *LNCS*, pages 39–53. Springer, 2003.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In P. Thiagarajan, editor, *Proc. 15th Conference on Foundations of Software Technology and Theoretical*

Computer Science, volume 1026 of *LNCS*, pages 499–513. Springer, 1995.

[BHHK00] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen. Model checking continuous-time Markov chains by transient analysis. In A. Emerson and A. Sistla, editors, *Proc. 12th International Conference on Computer Aided Verification (CAV'00)*, volume 1855 of *LNCS*, pages 358–372. Springer, 2000.

[BK98] C. Baier and M. Z. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11(3):125–155, 1998.

[BS03] N. Bertrand and Ph. Schnoebelen. Model checking lossy channels systems is probably decidable. In A. Gordon, editor, *Proc. Foundations of Software Science and Computation Structures (FOSSACS'2003)*, volume 2620 of *LNCS*, pages 120–135. Springer, 2003.

[CGP99] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[dA97] L. de Alfaro. *Formal verification of probabilistic systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.

[dAHM01] L. de Alfaro, T. A. Henzinger, and R. Majumdar. Symbolic algorithms for infinite-state games. In K. Larsen and M. Nielsen, editors, *Proc. CONCUR 2001 - Concurrency Theory*, volume 2154 of *LNCS*, pages 536–550. Springer, 2001.

[dAM01] L. de Alfaro and R. Majumdar. Quantitative solution of omega-regular games. In *Proc. 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)*, pages 675–683. ACM Press, 2001.

[DGJP00] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Approximating labeled Markov processes. In *Proc. 15th Annual IEEE Symposium on Logic in Computer Science (LICS 2000)*, pages 95–106. IEEE Computer Society Press, 2000.

[DJJL01] P. D'Argenio, B. Jeannet, H. Jensen, and K. Larsen. Reachability analysis of probabilistic systems by successive refinements. In L. de Alfaro and S. Gilmore, editors, *Proc. 1st Joint International Workshop on Process Algebra and Probabilistic Methods, Performance Modeling and Verification (PAPM/PROBMIV'01)*, volume 2165 of *LNCS*, pages 39–56. Springer, 2001.

[HHM99] T. A. Henzinger, B. Horowitz, and R. Majumdar. Rectangular hybrid games. In S. Mauw J. Baeten, editor, *Proc. CONCUR '99: Concurrency Theory*, volume 1664 of *LNCS*, pages 320–335. Springer, 1999.

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

[HMP92] T. Henzinger, Z. Manna, and A. Puneli. What good are digital clocks? In W. Kuich, editor, *Proc. 19th International Colloquium on Automata, Languages and Programming (ICALP'92)*, volume 623 of *LNCS*, pages 545–558. Springer, 1992.

[HMR03] T. A. Henzinger, R. Majumdar, and J.-F. Raskin. A classification of symbolic transition systems, 2003. To appear. Preliminary version appeared in *Proc. STACS 2000*, volume 1770 of LNCS, pages 13–34, Springer, 2000.

[HNSY94] T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Information and Computation*, 111(2):193–244, 1994.

[KNS01] M. Kwiatkowska, G. Norman, and J. Sproston. Symbolic computation of maximal probabilistic reachability. In K. Larsen and M. Nielsen, editors, *Proc. Proc. CONCUR '01: Concurrency Theory*, volume 2154 of *Lecture Notes in Computer Science*, pages 169–183. Springer, 2001.

[KNS02] M. Kwiatkowska, G. Norman, and J. Sproston. Probabilistic model checking of deadline properties in the IEEE 1394 FireWire root contention protocol. *Special Issue of Formal Aspects of Computing*, 2002. To appear.

[KNSS00] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In C. Palamidessi, editor, *Proc. CONCUR 2000 - Concurrency Theory*, volume 1877 of *LNCS*, pages 123–137. Springer, 2000.

[KNSS02] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theoretical Computer Science*, 282:101–150, 2002.

[KSK76] J. G. Kemeny, J. L. Snell, and A. W Knapp. *Denumerable Markov Chains*. Graduate Texts in Mathematics. Springer, 2nd edition, 1976.

[Var85] M. Vardi. Automatic verification of probabilistic concurrent finite state programs. In *Proc. 26th Annual Symposium on Foundations of Computer Science (FOCS'85)*, pages 327–338. IEEE Computer Society Press, 1985.

A Appendix: optimizations

In order to ease presentation, we have presented our techniques with our optimizations and for a restricted class of symbolic probabilistic systems, where the

restrictions mainly concern conditions for the translation to bi-labelled structures. However, the techniques of this paper are applicable without such restrictions. We proceed to describe techniques which can both help to optimize the verification process and to apply the techniques to a wider class of symbolic probabilistic system.

A.1 Definition of finite-template $\mathrm{NP}\bar{\mathrm{N}}$ system.

Note that isomorphism of probability distributions in the definition of finite-template $\mathrm{NP}\bar{\mathrm{N}}$ system may mean that the number of equivalence classes of distributions is unnecessarily high, therefore resulting in an elevated number of labels in the set L_1. We could instead define a partial order $\trianglelefteq$ over the set $\mathsf{Dist}(\mathsf{NPN})$ by letting $\nu' \trianglelefteq \nu$ if and only if there exists a function $f : \mathsf{support}(\nu) \to 2^{\mathsf{support}(\nu')}$ such that (1) $f(U_1) \cap f(U_2) \neq \emptyset$ for all $U_1, U_2 \in \mathsf{support}(\nu)$, and (2) for every $U \in \mathsf{support}(\nu)$, we have $\nu(U) = \sum_{U' \in f(U)} \nu'(U')$. For example, if ν is such that $\nu(U_1) = 0.5$ and $\nu(U_2) = 0.5$, and ν' is such that $\nu'(U_3) = 0.2$, $\nu'(U_4) = 0.3$ and $\nu'(U_5) = 0.5$, then $\nu' \trianglelefteq \nu$ (using the function $f(U_1) = \{U_3, U_4\}$ and $f(U_2) = \{U_5\}$). Intuitively, we write $\nu' \trianglelefteq \nu$ if the probabilistic branching of ν can be obtained from the probabilistic branching ν', possibly by summing over some of the alternatives of ν'. Then, to every distribution $\nu \in \mathsf{Dist}(\mathsf{NPN})$, we can assign a unique minimal element $\nu_{\min}$ of $\mathsf{Dist}(\mathsf{NPN})$, according to the partial order $\trianglelefteq$, such that $\nu_{\min} \trianglelefteq \nu$. If we then regard distributions with the same assigned minimal elements as being equivalent, provided that they are not enabled within the same state, we have an equivalence over $\mathsf{Dist}(\mathsf{NPN})$. Note that, if this equivalence has a finite quotient, then $\mathsf{Dist}(\mathsf{NPN})/_{\cong}$ has a finite quotient. This equivalence can be used in a similar way as the equivalence presented in the main text, although the definition of a bi-labelled structure of an $\mathrm{NP}\bar{\mathrm{N}}$ system must be changed, in particular with regard to the probability distribution over labels of the set L_2.

A.2 Redundant conjunction operations.

As noted in [KNS01], the purpose of the conjunction operator And in the semi-algorithms $\mathsf{MaxUntil}$ and $\mathsf{MaxRelease}$ is to generate regions in which transitions resulting from distinct pairs (a, b) of L_1- and L_2-labels are available. However, there is no need to perform the conjunction of regions which are generated by predecessor operations which are not labelled with the same L_1-label. This can be seen on consideration that we uniquely identify each distribution template (that is, each equivalence class of $\mathsf{Dist}(\mathsf{NPN})/_{\cong}$) with a label in L_1. Hence, if two regions are generated by predecessor operations with different L_1-labels, then they correspond to pairs (a, b) and (a', b') of label-pairs which are *never both assigned positive probability* in the construction of the finite-state $\bar{\mathrm{N}}\mathrm{P}$ systems. Therefore, conjunction operations on regions generated by different L_1-labels are redundant. We can therefore alter the lines labelled by $(\star)$ to:

$$T_{i+1} := \{\mathsf{And}(\sigma', \tau) \mid \tau \in T_{i+1} \text{ such that } \exists(\tau, (a, b'), \tau') \in E_{i+1}\} \cup T_{i+1} \ .$$

A.3 Reconciliation with [KNS01].

The framework of [KNS01] for computing the maximal reachability probability presented a more general superclass of symbolic transition systems than that presented here, in which label pairs are represented by a single label which can be shared amongst multiple distribution templates, and in which distribution templates which do not have a correspondence with any system transition may be included. Our algorithms MaxUntil and MaxRelease can also be used with this more general model.

More formally we now give the definition of the symbolic probabilistic systems of [KNS01], and give a translation from the framework developed in this paper to such a system.

Definition of symbolic probabilistic systems. *A* symbolic probabilistic system $\mathbb{P} = (S, \mathit{Steps}, R, \ulcorner\cdot\urcorner, \mathcal{T}\mathit{ra}, \mathsf{D})$ *comprises: a probabilistic system* (S, Steps)*; a set of symbolic states* R*; an extension function* $\ulcorner\cdot\urcorner : R \to 2^S$*; a set of transition types* $\mathcal{T}\mathit{ra}$*, and, associated with each* $a \in \mathcal{T}\mathit{ra}$*, a transition function* $\delta_a : S \to 2^S$*; and a set of distribution templates* $\mathsf{D} \subseteq \mathsf{Dist}(\mathcal{T}\mathit{ra})$*, such that the following conditions are satisfied.*

1. *For all states* $s \in S$*, let* $\mathcal{T}\mathit{ra}(s) \subseteq \mathcal{T}\mathit{ra}$ *be such that for any* $a \in \mathcal{T}\mathit{ra}$*:* $a \in \mathcal{T}\mathit{ra}(s)$ *if and only if* $\delta_a(s) \neq \emptyset$*. Then, for all* $t \in S$*:*
   (a) *if* $a \in \mathcal{T}\mathit{ra}$ *and* $t \in \delta_a(s)$*, then there exists* $\mu \in \mathit{Steps}(s)$ *such that* $\mu(t) > 0$*;*
   (b) *if* $\mu \in \mathit{Steps}(s)$*, then there exists* $\nu \in \mathsf{D}$ *and a vector of states* $\langle t_a \rangle_{a \in \mathcal{T}\mathit{ra}(s)} \in \prod_{a \in \mathcal{T}\mathit{ra}(s)} \delta_a(s)$ *such that:*

$$\sum_{a \in \mathcal{T}\mathit{ra}(s) \wedge t = t_a} \nu(a) = \mu(t);$$

   (c) *if* $\nu \in \mathsf{D}$ *and* $\langle t_a \rangle_{a \in \mathcal{T}\mathit{ra}(s)}$ *is a vector of states in* $\prod_{a \in \mathcal{T}\mathit{ra}(s)} \delta_a(s)$*, then there exists* $\mu \in \mathit{Steps}(s)$ *such that:*

$$\mu(t) \geq \sum_{a \in \mathcal{T}\mathit{ra}(s) \wedge t = t_a} \nu(a).$$

2. *There exists a family of computable functions* $\{\mathsf{Pre}_a\}_{a \in \mathcal{T}\mathit{ra}}$ *of the form* $\mathsf{Pre}_a : R \to R$*, such that, for all* $a \in \mathcal{T}\mathit{ra}$ *and* $\sigma \in R$*:*

$$\ulcorner \mathsf{Pre}_a(\sigma) \urcorner = \{ s \in S \mid \exists t \in \delta_a(s) \,.\, t \in \ulcorner \sigma \urcorner \} \,.$$

3. *There is a computable function* $\mathsf{And} : R \times R \to R$ *such that* $\ulcorner \mathsf{And}(\sigma, \tau) \urcorner = \ulcorner \sigma \urcorner \cap \ulcorner \tau \urcorner$ *for each pair of symbolic states* $\sigma, \tau \in R$*.*

4. *There is a computable function* $\mathsf{Diff} : R \times R \to R$ *such that* $\ulcorner \mathsf{Diff}(\sigma, \tau) \urcorner = \ulcorner \sigma \urcorner \setminus \ulcorner \tau \urcorner$ *for each pair of symbolic states* $\sigma, \tau \in R$*.*

5. *There is a computable function* $\mathsf{Empty} : R \to \mathbb{B}$ *such that* $\mathsf{Empty}(\sigma)$ *if and only if* $\ulcorner \sigma \urcorner = \emptyset$ *for each symbolic state* $\sigma \in R$*.*

6. *There is a computable function* $\mathsf{Member} : S \times R \to \mathbb{B}$ *such that* $\mathsf{Member}(s, \sigma)$ *if and only if* $s \in \ulcorner \sigma \urcorner$ *for each state* $s \in S$ *and symbolic state* $\sigma \in R$.

For a NP$\bar{\text{N}}$ system $\mathsf{NPN} = (S, \mathbf{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$, with a corresponding bi-labelled structure $\mathsf{B} = (S, L_1, L_2, \Gamma_1, \Gamma_2, \delta, P, \Vdash \cdot \dashv)$ and symbolic theory $(R, \ulcorner \cdot \urcorner)$ we now construct a symbolic probabilistic system $\mathbb{P}_\mathsf{B} = (S, \mathit{Steps}_\mathsf{B}, R, \ulcorner \cdot \urcorner, \mathcal{T}\mathit{ra}_\mathsf{B}, \mathsf{D}_\mathsf{B})$ as follows:

- $\mathit{Steps}_\mathsf{B}$ is the step function from the $\bar{\text{N}}$P system underlying NPN (see Section 3.1.1);
- $\mathcal{T}\mathit{ra}_\mathsf{B} = L_1 \times L_2$ where for any $s \in S$: $\delta_{(a,b)}(s) = \delta(s, a, b)$;
- $\mathsf{D}_\mathsf{B} = \{\nu_C \mid C \in \mathsf{Dist}(\mathsf{NPN})/\cong\}$ where for any $(a_{C'}, b) \in \mathcal{T}\mathit{ra}_\mathsf{B}$:

$$\nu_C(a_{C'}, b) = \begin{cases} C(b) & \text{if } C = C' \\ 0 & \text{otherwise.} \end{cases}$$

We now prove the correctness of this translation, that is, prove that $\mathbb{P}_\mathsf{B}$ is indeed a symbolic probabilistic system. First note that conditions *2–6* of a symbolic probabilistic system follow from the fact that $(R, \ulcorner \cdot \urcorner)$ is a symbolic theory and the fact that $\delta_{(a,b)}(s) = \delta(s, a, b)$. It therefore remains to prove *1(a)–(c)* which we consider in turn.

1(a) If $a \in \mathcal{T}\mathit{ra}$ and $t \in \delta_a(s)$, then $a = (a_C, b)$ and $t \in \delta(s, a_C, b)$ for some $a_C \in L_1$ and $b \in L_2$. Now, by construction of the bi-labelled system, there exists $\nu \in \mathbf{Steps}(s)$ such that $\nu(\delta(s, a_C, b)) > 0$. The result then follows from the construction of the underlying $\bar{\text{N}}$P system (see Section 3.1.1).

1(b) Consider any $s \in S$ and $\mu \in \mathit{Steps}_\mathsf{B}(s)$, now by construction of the underlying NP system, there exists a $\nu \in \mathbf{Steps}(s)$ with $\mathsf{support}(\nu) = \{U_1, \ldots, U_m\}$ and a vector of states $(t_1, \ldots, t_m) \in U_1 \times \cdots \times U_m$ such that $\mu(t) = \sum_{1 \leq i \leq m \wedge t_i = t} \nu(U_i)$. Now letting $\nu_C \in \mathsf{D}$ be such that ν is in the equivalence class C, it follows from the construction of the bi-labelled structure that $\delta(s, a_C, b_i) = U_i$ for all $1 \leq i \leq m$. Therefore taking the same vector and states we have:

$$\sum_{a \in \mathcal{T}\mathit{ra}(s) \wedge t = t_a} \nu_C(a) = \mu(t)$$

as required.

1(c) Consider any state s, distribution template $\nu_C \in \mathsf{D}$ and vector of states $\langle t_a \rangle_{a \in \mathcal{T}\mathit{ra}(s)} \in \prod_{a \in \mathcal{T}\mathit{ra}(s)} \delta_a(s)$. If there does not exist a $\nu \in \mathbf{Steps}(s)$ such that $\nu \in C$ we have $\delta(s, a_C, b) = \emptyset$ for all $b \in L_2$, and hence $\nu(a) = 0$ for all $a \in \mathcal{T}\mathit{ra}(s)$. Therefore for any $\nu \in \mathit{Steps}(s)$ and $t \in S$:

$$\mu(t) \geq 0 = \sum_{a \in \mathcal{T}\mathit{ra}(s) \wedge t = t_a} \nu(a) \; .$$

On the other hand, if there exists $\nu \in \mathbf{Steps}(s)$ such that $\nu \in C$, from the construction of the underlying NP system, supposing $\mathsf{support}(\nu) = \{U_1, \ldots, U_n\}$, then for any vector of states $(t_1, \ldots, t_n) \in U_1 \times \cdots \times U_n$ there exists $\mu \in \mathit{Steps}(s)$ such that for any $t \in S$: $\mu(t) = \sum_{1 \leq i \leq m \wedge t_i = t} \nu(U_i)$. Then since $\delta_{a_C, b_i}(s) = \delta(s, a_C, b_i) = U_i$ for all $1 \leq i \leq m$ and $C(b_i) = 0$ for all $i > m$, we can show that there exists $\mu \in \mathit{Steps}(s)$ such that for any $t \in S$:

$$\mu(t) = \sum_{a \in \mathcal{T}ra(s) \wedge t = t_a} \nu(a)$$

as required.

B Appendix: Proof of Proposition 2

Before we give the proof of Proposition 2, we require the following notation, definitions and lemmas. First, by abuse of notation, we say that a region σ satisfies a PCTL formula ϕ, written $\ulcorner \sigma \urcorner \models \phi$, if and only if $\ulcorner \sigma \urcorner \subseteq [\phi]$. Such notation allows us to occasionally avoid referring to observables and observation function explictly in the definition of $\bar{\mathrm{N}}\mathrm{P}$ systems.

Let $\mathsf{NP} = (S, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ be an $\bar{\mathrm{N}}\mathrm{P}$ system. For any adversary $A \in \mathit{Adv}_{\mathsf{NP}}$, let $\mathit{Path}^A_{ful} = \bigcup_{s \in S} \mathit{Path}^A_{ful}(s)$; similarly, for any state $s \in S$, let $\mathit{Path}_{ful}(s) = \bigcup_{A \in \mathit{Adv}_{\mathsf{NP}}} \mathit{Path}^A_{ful}(s)$. Then, for any adversary $A \in \mathit{Adv}_{\mathsf{NP}}$, we define a sequence of functions $(\mathbf{p}\mathcal{V}^A_n)_{n \in \mathbb{N}}$ such that for a state $s \in S$ and PCTL formulae ϕ_1, ϕ_2, $\mathbf{p}\mathcal{V}^A_n(\phi_1, \phi_2, s)$ equals:

$$\mathit{Prob}^A\{\omega \in \mathit{Path}^A_{ful}(s) \mid \text{for all } 0 \leq j \leq n \text{ if } \omega(i) \not\models \phi_1 \text{ for every } i < j \text{ then } \omega(j) \models \phi_2\} .$$

Definition 5 *Let $\mathsf{NP} = (S, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ be an $\bar{\mathrm{N}}\mathrm{P}$ system and ϕ_1, ϕ_2 be PCTL formulae. For any adversary $A \in \mathit{Adv}_{\mathsf{NP}}$ and finite path $\omega \in \mathit{Path}^A_{fin}$, let:*

$$\mathbf{p}\mathcal{V}^A_0(\phi_1, \phi_2, \omega) = \begin{cases} 1 & \text{if } \mathit{last}(\omega) \models \phi_2 \\ 0 & \text{otherwise,} \end{cases}$$

and for any $i \in \mathbb{N}$, if $A(\omega) = \mu$:

$$\mathbf{p}\mathcal{V}^A_{i+1}(\phi_1, \phi_2, \omega) = \begin{cases} 1 & \text{if } \mathit{last}(\omega) \models \phi_1 \wedge \phi_2 \\ \displaystyle\sum_{s' \in S} \mu(s') \cdot \mathbf{p}\mathcal{V}^A_i(\phi_1, \phi_2, \omega \xrightarrow{\mu} s') & \text{if } \mathit{last}(\omega) \models \neg\phi_1 \wedge \phi_2 \\ 0 & \text{otherwise.} \end{cases}$$

Lemma 6 *Let $\mathsf{NP} = (S, \mathit{Steps}, P, \langle\!\langle \cdot \rangle\!\rangle)$ be an $\bar{\mathrm{N}}\mathrm{P}$ system. For any state $s \in S$ and PCTL formulae ϕ_1, ϕ_2:*

$$\mathit{Max}\mathcal{V}(\phi_1, \phi_2, s) = \sup_{A \in \mathit{Adv}_{\mathsf{NP}}} \lim_{i \to \infty} \mathbf{p}\mathcal{V}^A_i(\phi_1, \phi_2, s) .$$

Lemma 7 *Let $\{(T_i, E_i)\}_{1 \leq i \leq k}$ be the sequence of graphs constructed in the algorithm* MaxRelease. *For any $i \in \mathbb{N}$, if $(\sigma, (a, b), \tau) \in E_i$, then $\ulcorner \sigma \urcorner \subseteq \ulcorner \mathsf{Pre}^{a,b}(\tau) \urcorner$.*

Definition 8 *If $\{(T_i, E_i)\}_{1\leq i\leq k}$, are the graphs constructed in the algorithm* MaxRelease*, then let* $\mathsf{NP}_\infty = (T_\infty, Steps_{\mathsf{NP}_\infty})$ *be the* $\bar{\text{N}}$P *system defined as follows:*

- $T_\infty = \bigcup_{i=0}^{\infty} T_i \times \{i\}$;
- *For any* $(\sigma, i) \in T_\infty$ *if* $i = 0$*, then* $Steps_{\mathsf{NP}_\infty}(\sigma, i) = \emptyset$*, and if* $i > 0$*, then* $\pi \in Steps_{\mathsf{NP}_\infty}(\sigma, i)$ *if and only if there exists a subset of edges* $E_\pi \subseteq E_i$ *and an equivalence class* $C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$ *of distributions of* NPN *such that:*
  1. *if* $(\sigma', (a', b'), \tau') \in E_\pi$*, then* $\ulcorner\sigma\urcorner \subseteq \ulcorner\sigma'\urcorner$ *and* $a = a_C$;
  2. *if* $(\sigma', (a_C, b'), \tau') \neq (\sigma'', (a_C, b''), \tau'') \in E_\pi$*, then* $b' \neq b''$;
  3. *the set* E_π *is maximal;*
  4. *for all* $(\tau, j) \in T$*:*

$$\pi(\tau, j) = \begin{cases} \sum\limits_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) & \text{if } j = i - 1 \\ 0 & \text{otherwise.} \end{cases}$$

  where $(T_i, E_i) = (T_k, E_k)$ *for all* $i > k$.

By abuse of notation, we say a state (σ, i) of T^∞ satisfies a PCTL formula ϕ if and only if σ satisfies ϕ.

Lemma 9 *For the* $\text{NP}\bar{\text{N}}$ *system* NPN *and* PCTL *formula* $\phi_1 \mathcal{V} \phi_2$*, if* NP *and* $\mathsf{NP}_\infty = (T_\infty, Steps_{\mathsf{NP}_\infty})$ *are the* $\bar{\text{N}}$P *systems constructed through the algorithm* MaxRelease *and by Definition 8 respectively, then for any* $\sigma \in T$*:*

$$Max\mathcal{V}(\phi_1, \phi_2, Adv_{\mathsf{NP}}, \sigma) = \sup_{B \in Adv_{\mathsf{NP}_\infty}} \lim_{i \to \infty} \mathbf{p}\mathcal{V}_i^B(\phi_1, \phi_2, (\sigma, i)) \,.$$

Proof. The proof follows from the fact that there exists $k \geq 0$ such that $(T_i, E_i) = (T, E)$ for all $i \geq k$ and the fact that the probabilistic transitions of NP and NP_∞ are constructed in the same way. □

Lemma 10 *Let* $\mathsf{NP}_\infty = (T_\infty, Steps_{\mathsf{NP}_\infty})$ *be the probabilistic system constructed through Definition 8. For any* $i \in \mathbb{N}$*,* $(\sigma, i{+}1) \in T_\infty$ *which satisfies* $\phi_2 \wedge \neg\phi_1$ *and* $\pi \in Steps(\sigma, i{+}1)$*, if* E_π *and* $C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$ *are the set of edges and equivalence class of distributions used to construct* π*, then*

$$\mathbf{p}\mathcal{V}_{i+1}^B(\phi_1, \phi_2, (\sigma, i{+}1)) = \sum_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) \cdot \mathbf{p}\mathcal{V}_i^B(\phi_1, \phi_2, \sigma \xrightarrow{\pi} (\tau, i)) \,.$$

Proof. Consider any $i \in \mathbb{N}$, $(\sigma, i{+}1) \in T_\infty$ and $B \in Adv_{T_\infty}$, if $B(\sigma, i{+}1) = \pi$ and π is constructed from is the set of edges E_π and class of distributions $C \in \mathsf{Dist}(\mathsf{NPN})/_{\cong}$, then by definition for any $(\tau, j) \in T_\infty$ we have:

$$\pi(\tau, j) = \begin{cases} \sum\limits_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) & \text{if } j = i - 1 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Since $(\sigma, i{+}1) \in T_\infty$ satisfies $\phi_2 \wedge \neg\phi_1$, from Definition 5 we have:

$$
\begin{aligned}
&\mathbf{p}\mathcal{V}^B_{i+1}(\phi_1, \phi_2, (\sigma, i{+}1)) \\
&\quad = \sum_{\tau' \in T_\infty} \pi(\tau') \cdot \mathbf{p}\mathcal{V}^B_i(\phi_1, \phi_2, (\sigma, i{+}1) \xrightarrow{\pi} \tau') \\
&\quad = \sum_{(\tau,i)\in T_\infty} \left(\sum_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) \right) \cdot \mathbf{p}\mathcal{V}^B_i(\phi_1, \phi_2, (\sigma, i{+}1) \xrightarrow{\pi} (\tau, i)) \quad \text{by (1)} \\
&\quad = \sum_{\tau \in T_i} \left(\sum_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) \cdot \mathbf{p}\mathcal{V}^B_i(\phi_1, \phi_2, (\sigma, i{+}1) \xrightarrow{\pi} (\tau, i)) \right) \quad \text{by Definition 8} \\
&\quad = \sum_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) \cdot \mathbf{p}\mathcal{V}^B_i(\phi_1, \phi_2, (\sigma, i{+}1) \xrightarrow{\pi} (\tau, i)) \quad \text{rearranging}
\end{aligned}
$$

as required. □

We now give the proof of Proposition 2, that is we show:

For the symbolic probabilistic system $\mathsf{SPS} = (\mathsf{NPN}, R, \ulcorner\cdot\urcorner)$*, the release formula* $\phi_1 \mathcal{V} \phi_2$*, and the finite-state* $\bar{\mathrm{N}}\mathrm{P}$ *system* NP *constructed from the semi-algorithm* $\mathsf{MaxRelease}$*, then for any state* $s \in S$ *of* NPN*, we have:*

$$Max\mathcal{V}(\phi_1, \phi_2, Adv_{\mathsf{NPN}}, s) = \max_{\sigma \in T \wedge s \in \ulcorner\sigma\urcorner} Max\mathcal{V}(\phi_1, \phi_2, Adv_{\mathsf{NP}}, \sigma) \; .$$

Proof of Proposition 2. Let $\{(T_i, E_i)\}_{i=0,1,\ldots}$ be the sequence of graphs constructed in the algorithm $\mathsf{MaxRelease}$, for the formula $\phi_1 \mathcal{V} \phi_2$. We split the proof into proving a sequence of properties: (a), (b) and (c). First consider the following:

(a) $\sigma \in T_i$ if and only if for all $s \in \ulcorner\sigma\urcorner$ there exists a path $\omega \in Path_{ful}(s)$ such that for all $0 \leq j \leq i$ if $\omega(k) \not\models \phi_1$ for every $k < j$, then $\omega(j) \models \phi_2$.

The proof is by induction on $i \in \mathbb{N}$. The case when $i = 0$ follows from the fact that $T_0 = [\phi_2]$. Now suppose that (a) holds from some $i \in \mathbb{N}$ and consider any $\sigma \in T_{i+1}$. From $\mathsf{MaxRelease}$ it follows that either $\ulcorner\sigma\urcorner \models \phi_1 \wedge \phi_2$ and the result is immediate, or $\ulcorner\sigma\urcorner \models \phi_2$ and $\sigma \subseteq \mathsf{Pre}^{a,b}(\tau)$ for some $a \in L_1$, $b \in L_2$ and $\tau \in T_i$. Now, by construction of the symbolic theory for NPN we have: $s \in \ulcorner\mathsf{Pre}^{a,b}(\tau)\urcorner$ for some $a \in L_1$, $b \in L_2$ if and only if there exists $\nu \in \mathbf{Steps}$ and $U \subseteq S$ such that $\nu(U) > 0$ and $U \cap \ulcorner\tau\urcorner \neq \emptyset$. Using these facts and induction property (a) follows.

It follows from (a) that $\sigma \in T_i$ if and only if for any $s \in \ulcorner\sigma\urcorner$, there exists an adversary A such that $\mathbf{p}\mathcal{V}^A_i(\phi_1, \phi_2, s) > 0$. Moreover, we have that

if $Max\mathcal{V}(\phi_1, \phi_2, s) > 0$, then there exists $\sigma \in T$ such that $s \in \ulcorner\sigma\urcorner$.

We now give the main step in the proof which involves showing a correspondence between the probability values of $\mathbf{p}\mathcal{V}_i^A$ for adversaries A of NPN and $\mathbf{p}\mathcal{V}_i^B$ for adversaries B of NP_∞. Formally we show that for any $i \in \mathbb{N}$ and $s \in S$ such that $\mathbf{p}\mathcal{V}_i^A(\phi_1, \phi_2, s) > 0$:

(b) if $B \in Adv_{\mathsf{NP}_\infty}$, $\sigma \in T_i$ and $s \in \ulcorner\sigma\urcorner$, then there exists $A \in Adv_{\mathsf{NPN}}$ such that $\mathbf{p}\mathcal{V}_i^A(\phi_1, \phi_2, s) \geq \mathbf{p}\mathcal{V}_i^B(\phi_1, \phi_2, (\sigma, i))$;

(c) if $A \in Adv_{\mathsf{NPN}}$, then there exists $\sigma \in T_i$ and $B \in Adv_{\mathsf{NP}_\infty}$ such that $s \in \ulcorner\sigma\urcorner$ and $\mathbf{p}\mathcal{V}_i^B(\phi_1, \phi_2, (\sigma, i)) \geq \mathbf{p}\mathcal{V}_i^A(\phi_1, \phi_2, s)$.

It follows from (a), Lemma 6 and Lemma 9 that to prove Proposition 2 it is sufficient to show that (b) and (c) hold. We now prove (b) and (c) by induction on $n \in \mathbb{N}$. The case for $i = 0$ for both (b) and (c) follow from Definition 5 and the fact that $T_0 = [\phi_2]$.

Next, suppose (b) and (c) hold for some $i \in \mathbb{N}$ and consider any $s \in S$ such that $\mathbf{p}\mathcal{V}_{i+1}^A(\phi_1, \phi_2, s) > 0$. If $s \models \phi_1 \wedge \phi_2$, then the result follows from Definition 5 and since $([\phi_1 \wedge \phi_2], i{+}1) \in T_\infty$. Therefore, from Definition 5 and (a) it remains to consider the case when $s \models \neg\phi_1 \wedge \phi_2$.

(b) Consider any adversary $B \in Adv_{\mathsf{NP}_\infty}$ and region $\sigma \in T_{i+1}$ such that $s \in \ulcorner\sigma\urcorner$; then $B(\sigma, i{+}1) = \pi$ for some distribution $\pi \in Steps_{\mathsf{NP}_\infty}(\sigma)$. By construction of NP_∞, there exists an equivalence class $C \in \mathsf{Dist}(\mathsf{NPN})/_\cong$ of distributions of NPN and non-empty set of edges:

$$E_\pi \subseteq E_{i+1} \cap (T_{i+1} \times (\{a_C\} \times L_2) \times T_i)$$

used to construct π. From the construction of the bi-labelled structure of NPN given in Section 3.2, there exists $\nu \in \mathbf{Steps}(s)$ such that $\nu \in C$. Furthermore, from this construction, the support set $\mathsf{support}(\nu)$ of ν can be written as $\{U_1, U_2, \ldots, U_m\}$ such that $\delta(s, a_C, b_j) = U_j$ for all $1 \leq j \leq m$ and $\delta(s, a_C, b_j) = \emptyset$ for all $m < j \leq n$. Note that, since $\delta(s, a_C, b_j) = \emptyset$ for all $m < j \leq n$, if $(\sigma', (a_C, b_j), \tau') \in E_{i+1}$ for any $m < j \leq n$, then $s \notin \ulcorner\sigma'\urcorner$, and hence $\ulcorner\sigma\urcorner \not\subseteq \ulcorner\sigma'\urcorner$. From Definition 8 it then follows that $(\sigma', (a_C, b_j), \tau') \notin E_\pi$ for any $m < j \leq n$.

Now, if we consider any $(\sigma', (a_C, b_j), \tau') \in E_\pi$, it follows from Definition 8 and Lemma 7 that $\ulcorner\sigma\urcorner \subseteq \ulcorner\sigma'\urcorner$ and $\ulcorner\sigma'\urcorner \subseteq \ulcorner\mathsf{Pre}^{a_C, b_j}(\tau')\urcorner$. Therefore, from the construction of the bi-labelled structure we have $\ulcorner\tau\urcorner \cap \delta(s, a_C, b_j) \neq \emptyset$. Using these results, for each $1 \leq i \leq m$, we define a state $t_j \in \delta(s, a_C, b_j)$ as follows:

- if $(\sigma', (a_C, b_j), \tau') \in E_\pi$ for some $\sigma' \in T_{i+1}$ and $\tau' \in T_i$, let $t_j \in \ulcorner\tau\urcorner \cap \delta(s, a_C, b_j)$;
- if $(\sigma', (a_C, b_j), \tau') \notin E_\pi$ for any $\sigma' \in T_{i+1}$ and $\tau \in T_i$, let t_j be arbitrary.

Therefore, in the NP system underlying the NP$\bar{\mathrm{N}}$ system NPN (see Section 3.1.1 for this construction), there exists a distribution $\mu \in Steps(s)$ such that, for all

states $s' \in S$:

$$\mu(s') = \sum_{1 \leq j \leq m \wedge s' = t_j} \nu(U_j) = \sum_{1 \leq j \leq m \wedge s' = t_j} C(b_j). \tag{2}$$

By induction, for any $(\sigma', (a_C, b_j), \tau) \in E_\pi$, there exists an adversary A_j such that:

$$\mathbf{p}\mathcal{V}_i^{A_j}(\phi_1, \phi_2, t_j) \geq \mathbf{p}\mathcal{V}_i^{B'}(\phi_1, \phi_2, (\tau, i)) = \mathbf{p}\mathcal{V}_i^{B}(\phi_1, \phi_2, (\sigma, i{+}1) \xrightarrow{\pi} (\tau, i)) \tag{3}$$

where $B' \in Adv_{\mathsf{NP}_\infty}$ is the adversary such that $B'(\omega) = B((\sigma, i{+}1) \xrightarrow{\pi} \omega)$. Now suppose $A \in Adv_{\mathsf{NPN}}$ is the adversary that chooses μ in state s and then behaves like A_j once it reaches the state t_j (if $t_j = t_k$ for $j \neq k$, then let A behave like A_j if $\mathbf{p}\mathcal{V}_i^{A_j}(\phi_1, \phi_2, t_j) \geq \mathbf{p}\mathcal{V}_i^{A_k}(\phi_1, \phi_2, t_k)$ and A_k otherwise). By Definition 5 we have:

$$\begin{aligned}
\mathbf{p}\mathcal{V}_{i+1}^{A}(\phi_1, \phi_2, s) &= \sum_{t \in S} \mu(t) \cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1, \phi_2, s \xrightarrow{\mu} t) \\
&= \sum_{t \in S} \left(\sum_{\substack{1 \leq j \leq m \\ \wedge\, t_j = t}} C(b_j) \right) \cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1, \phi_2, s \xrightarrow{\mu} t) && \text{by (2)} \\
&\geq \sum_{t \in S} \left(\sum_{\substack{(\sigma', (a_C, b_j), \tau) \in E_\pi \\ \wedge\, t_j = t}} C(b_j) \right) \cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1, \phi_2, s \xrightarrow{\mu} t) && \text{by construction of } t_j \\
&= \sum_{(\sigma', (a_C, b_j), \tau) \in E_\pi} C(b_j) \cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1, \phi_2, s \xrightarrow{\mu} t_j) && \text{rearranging} \\
&= \sum_{(\sigma', (a_C, b_j), \tau) \in E_\pi} C(b) \cdot \mathbf{p}\mathcal{V}_i^{A_j}(\phi_1, \phi_2, t_j) && \text{by construction of } A \\
&\geq \sum_{(\sigma', (a_C, b_j), \tau) \in E_\pi} C(b) \cdot \mathbf{p}\mathcal{V}_i^{B}(\phi_1, \phi_2, (\sigma, i{+}1) \xrightarrow{\pi} (\tau, i)) && \text{by (3)} \\
&= \mathbf{p}\mathcal{V}_{i+1}^{B}(\phi_1, \phi_2, (\sigma, i{+}1)) && \text{by Lemma 10.}
\end{aligned}$$

Since $B \in Adv_{\mathsf{NP}_\infty}$ and $\sigma \in T$ are arbitrary, (b) holds by induction.

(c) Consider any adversary $A \in Adv_{\mathsf{NPN}}$ such that $\mathbf{p}\mathcal{V}_{i+1}^{A}(\phi_1, \phi_2, s) > 0$, then $A(s) = \mu$ for some $\mu \in Steps_{\mathsf{NPN}}(s)$. From the construction of the $\bar{\text{N}}$P system underlying NPN given in Section 3.1.1, μ is constructed from some $\nu \in \mathbf{Steps}(s)$. Letting C by the equivalence class of $\mathsf{Dist}(\mathsf{NPN})/_{\cong}$ of which ν belongs, from construction of the bi-labelled structure (see Section 3.2), $\mathsf{support}(\nu)$ is of the form $\{U_1, \ldots, U_n\}$ such that $\delta(s, a_C, b_j) = U_j$ and $\nu(U_j) = C(b_j)$ for all

$1 \leq j \leq m$. Now since μ is constructed from ν, there exists a vector of states $(t_1, \ldots, t_m) \in U_1 \times \cdots \times U_m$ such that for all $t \in S$:

$$\mu(t) = \sum_{1 \leq j \leq m \wedge t = t_j} \nu(U_j) = \sum_{1 \leq j \leq m \wedge t = t_j} C(b_j). \tag{4}$$

Now, for any $t \in S$ such that $\mu(t) > 0$ and $\mathbf{p}\mathcal{V}_i^A(\phi_1, \phi_2, s \xrightarrow{\mu} t) > 0$, by induction there exists $\tau_t \in T_i$ and an adversary $B_t \in Adv_{\mathsf{NP}_\infty}$ such that $t \in \ulcorner \tau_t \urcorner$ and

$$\mathbf{p}\mathcal{V}_i^{B_t}(\phi_1, \phi_2, (\tau_t, i)) \geq \mathbf{p}\mathcal{V}_i^{A'}(\phi_1, \phi_2, t) = \mathbf{p}\mathcal{V}_n^A(\phi_1, \phi_2, s \xrightarrow{\mu} t) \tag{5}$$

where $A' \in Adv_{\mathsf{NPN}}$ is the adversary such that $A'(\omega) = A(s \xrightarrow{\mu} \omega)$.

Next, let $L_t(s)$ be the set of labels of L_2 such that $b_j \in L_t(s)$ if and only if $\nu(b_j) > 0$ and $t_j = t$. Note that, for any distinct $t, t' \in S$: $L_t(s) \cap L_{t'}(s) = \emptyset$. Furthermore, let σ_t equal

$$\mathsf{And}\{\mathsf{And}(\mathsf{Pre}^{a_C,b}(\tau_t), [\phi_2]) \,|\, b \in L_t(s)\}$$

and E_t equal the set of edges

$$\{(\mathsf{And}(\mathsf{Pre}^{a_C,b}(\tau_t), [\phi_2]), (a_C, b), \tau_t) \,|\, b \in L_t(s)\}.$$

By construction $s \models \phi_2$ and $s \in \ulcorner \mathsf{Pre}^{a_C,b}(\tau_t) \urcorner$ for all $b \in L_t(s)$, and hence $s \in \ulcorner \sigma_t \urcorner$, $\sigma_t \in T_{i+1}$ and E_t is a subset of E_{i+1}.

Since this was for arbitrary $t \in S$ such that $\mu(t) > 0$ and $\mathbf{p}\mathcal{V}_i^A(\phi_1, \phi_2, s \xrightarrow{\mu} t) > 0$, letting σ equal

$$\mathsf{And}\{\sigma_t \,|\, t \in S \wedge \mu(t) > 0 \wedge \mathbf{p}\mathcal{V}_n^A(\phi_1, \phi_2, s \xrightarrow{\mu} t) > 0\},$$

and E_μ equal the union of the edges E_t for $t \in S$ such that $\mu(t) > 0$ and $\mathbf{p}\mathcal{V}_n^A(\phi_1, \phi_2, s \xrightarrow{\mu} t) > 0$, it follows that $\sigma \in T_{i+1}$, $s \in \ulcorner \sigma \urcorner$ and E_μ is a subset of E_{i+1} such that:

- if $(\sigma', (a, b), \tau') \in E_\mu$, then $\ulcorner \sigma \urcorner \subseteq \ulcorner \sigma' \urcorner$ and $a = a_C$;
- if $(\sigma', (a_C, b), \tau'), (\sigma'', (a_C, b'), \tau'') \in E_\mu$ are distinct, then $b \neq b'$ (since for any distinct $t, t' \in S$: $L_t(s) \cap L_{t'}(s) = \emptyset$).

Now, by construction of NP_∞ (Definition 8) there exists $\pi \in Steps_{\mathsf{NP}_\infty}(\sigma, i{+}1)$ and $E_\pi \supseteq E_\mu$ such that for all $\tau \in T$:

$$\pi(\tau, k) = \begin{cases} \sum\limits_{(\sigma', (a_C, b), \tau) \in E_\pi} C(b) & \text{if } k = i - 1 \\ 0 & \text{otherwise.} \end{cases}$$

Now suppose that B is the adversary which chooses π in $(\sigma, i{+}1)$ and for all $t \in S$ such that $\mu(t) > 0$ and $\mathbf{p}\mathcal{V}_{i+1}^A(\phi_1, \phi_2, s \xrightarrow{\mu} t) > 0$ behaves like B_t when it reaches the state τ_t (if $\tau_t = \tau_{t'}$ for $t \neq t'$, then let A behave like A_t if

$\mathbf{p}\mathcal{V}_i^{B_t}(\phi_1,\phi_2,(\tau_t,i)) \geq \mathbf{p}\mathcal{V}_i^{B_{t'}}(\phi_1,\phi_2,(\tau_{t'},i))$ and $B_{t'}$ otherwise). By Lemma 10 and construction of π we have:

$$\begin{aligned}
\mathbf{p}\mathcal{V}_{i+1}^{B}(\phi_1,\phi_2,(\sigma,i{+}1)) &= \sum_{(\sigma',(a_C,b),\tau)\in E_\pi} C(b)\cdot \mathbf{p}\mathcal{V}_i^{B}(\phi_1,\phi_2,(\sigma,i{+}1)\xrightarrow{\pi}(\tau,i)) \\
&\geq \sum_{(\sigma',(a_C,b),\tau)\in E_\mu} C(b)\cdot \mathbf{p}\mathcal{V}_i^{B}(\phi_1,\phi_2,(\sigma,i{+}1)\xrightarrow{\pi}(\tau,i)) \\
&\qquad \text{since } E_\mu \subseteq E_\pi \\
&= \sum_{\substack{t\in S\wedge\mu(t)>0\wedge \\ \mathbf{p}\mathcal{V}_i^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t)>0}} \left(\sum_{b\in L_t(s)} C(b)\cdot \mathbf{p}\mathcal{V}_i^{B}(\phi_1,\phi_2,(\sigma,i{+}1)\xrightarrow{\pi}(\tau_t,i))\right) \\
&\qquad \text{by construction of } E_\mu \\
&= \sum_{\substack{t\in S\wedge\mu(t)>0\wedge \\ \mathbf{p}\mathcal{V}_i^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t)>0}} \left(\sum_{b\in L_t(s)} C(b)\cdot \mathbf{p}\mathcal{V}_i^{B_t}(\phi_1,\phi_2,(\tau_t,i))\right) \\
&\qquad \text{by construction of } B \\
&\geq \sum_{\substack{t\in S\wedge\mu(t)>0\wedge \\ \mathbf{p}\mathcal{V}_n^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t)>0}} \left(\sum_{b\in L_t(s)} C(b)\cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t)\right) \\
&\qquad \text{by (5)} \\
&= \sum_{\substack{t\in S\wedge\mu(t)>0\wedge \\ \mathbf{p}\mathcal{V}_n^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t)>0}} \left(\sum_{b\in L_t(s)} C(b)\right)\cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t) \\
&\qquad \text{rearranging} \\
&= \sum_{\substack{t\in S\wedge\mu(t)>0\wedge \\ \mathbf{p}\mathcal{V}_n^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t)>0}} \left(\sum_{1\leq j\leq m\wedge t=t_j} C(b_j)\right)\cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t) \\
&\qquad \text{by construction of } L_t(s) \\
&= \sum_{\substack{t\in S\wedge\mu(t)>0\wedge \\ \mathbf{p}\mathcal{V}_n^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t)>0}} \mu(t)\cdot \mathbf{p}\mathcal{V}_i^{A}(\phi_1,\phi_2,s\xrightarrow{\mu}t) \\
&\qquad \text{by (4)} \\
&= \mathbf{p}\mathcal{V}_{i+1}^{A}(\phi_1,\phi_2,s) \\
&\qquad \text{by Definition 5}
\end{aligned}$$

as required. □