Model Checking for a Probabilistic Branching Time Logic with Fairness

Christel Baier
Fakultät für Mathematik & Informatik
Universität Mannheim
68131 Mannheim, Germany
baier@pi1.informatik.uni-mannheim.de

Marta Kwiatkowska*
School of Computer Science
University of Birmingham, Edgbaston
Birmingham B15 2TT, UK
M.Z.Kwiatkowska@cs.bham.ac.uk

May 20, 1998

Abstract

We consider concurrent probabilistic systems, based on probabilistic automata of Segala & Lynch [55], which allow non-deterministic choice between probability distributions. These systems can be decomposed into a collection of "computation trees" which arise by resolving the non-deterministic, but not probabilistic, choices. The presence of non-determinism means that certain liveness properties cannot be established unless fairness is assumed. We introduce a probabilistic branching time logic $PBTL$, based on the logic $TPCTL$ of Hansson [30] and the logic $PCTL$ of [55], resp. $pCTL$ of [14]. The formulas of the logic express properties such as "every request is eventually granted with probability at least p". We give three interpretations for $PBTL$ on concurrent probabilistic processes: the first is standard, while in the remaining two interpretations the branching time quantifiers are taken to range over a certain kind of fair computation trees. We then present a model checking algorithm for verifying whether a concurrent probabilistic process satisfies a $PBTL$ formula assuming fairness constraints. We also propose adaptations of existing model checking algorithms for $pCTL^*$ [14, 4] to obtain procedures for $PBTL^*$ under fairness constraints. The techniques developed in this paper have applications in automatic verification of randomized distributed systems.

1 Introduction

Probabilistic techniques, and in particular probabilistic logics, have proved successful in the specification and verification of systems that exhibit uncertainty, for example, fault-tolerant systems, randomized algorithms, distributed systems, and communication protocols. However, as already observed in [45, 52, 56], concurrent probabilistic systems, for example randomized distributed algorithms, are notoriously difficult to verify: the proofs of their correctness are complex, and therefore argued informally, and thus appropriate

*Supported in part by EPSRC grant GR/K42028.

formal methods for their specification and verification are called for. This paper presents an automatic model checking method applicable, amongst others, to the verification of randomized distributed systems.

The particular difficulty in establishing correctness of randomized distributed algorithms is due to the fact that they exhibit both probabilistic choice (which comes from random assignment and is considered internal to the system) as well as non-determinism. This means that it is possible in a given state to *non-deterministically choose between two or more probability distributions on the successor states*; these distributions determine the probability with which a successor state is taken. Non-determinism may arise e.g. from the asynchronicity of certain subprocesses, or external intervention such as an action taken by environment. As an example of the former, consider the randomized dining philosophers: when two philosophers are simultaneously ready to flip a fair coin in order to decide which fork to pick up, one can think of this as two probability distributions, each respectively with probability $\frac{1}{2}$ of obtaining heads or tails, enabled in the same state. It is then convenient to think of a run of such a system as being the outcome of a *scheduler* (also called an *adversary*) who decides which of the two distributions to select first. As another example, consider a communication protocol which attempts to deliver a message to the recipient if one is received on the input channel from the environment, and loops back to the initial state otherwise. In a realistic scenario, the outcome of the delivery is probabilistic, and will result in a message being delivered successfully with some suitably high probability, say 0.999, or an error state being reached if a fault has occurred in the transmitting medium.

The models considered in this paper, called *concurrent probabilistic systems*, are based on Markov decision processes, see e.g. [58, 50, 20, 62, 51, 55, 14, 21, 39, 4], and exhibit probabilistic choice and non-determinism in the sense described above. The choices made by an adversary can use the knowledge of the past history, but are not, and should *not*, in general, be resolved probabilistically. An adversary resolves the non-deterministic choices (but not the probabilistic choices; these are resolved by the system itself) by selecting one of possibly many probability distributions. This yields a *computation tree* of the system. The computation trees are represented by (discrete-time) Markov chains which arise by ensuring that in every state there is at most one distribution, and so each such tree has a probability space on paths associated with it. Ranging over all schedulers, a concurrent probabilistic system can be decomposed into a collection of its computation trees.

Several probabilistic logics have been proposed which allow to specify properties of the form "the system satisfies property φ with probability at least p" where p is a real number in the interval [0,1], see e.g. [40, 26, 41, 42, 15, 30, 31, 55, 6, 14, 4]. Their models are variants of Markov chains which may or may not exhibit non-determinism as well as probabilistic choice. Typically, the verification aims to establish *qualitative properties*, i.e. properties that are fulfilled by almost all executions, which amounts to showing that the property is satisfied with probability 1, see e.g. [43, 33, 49, 32, 58, 59, 50, 19, 2, 3, 51, 21]. Although the above requirement of probability 1 is important in many cases, for some properties it is simply not the case that they are satisfied with probability 1, but instead with probability $1-\varepsilon$ for some suitable ε (an error). These *quantitative properties*, which are the focus of this paper, also play an important role in e.g. the analysis of the average-case behaviour of probabilistic systems, where one aims to show that a given property is satisfied with probability e.g. $\frac{1}{2}$.

Since in our models there are possibly several probability distributions to choose from in a state, similarly to the non-probabilistic case certain (qualitative or quantitative) liveness properties cannot be established unless *fairness of choice* is imposed. Consider, for instance, the protocol referred to above; then the property "the message is eventually delivered with probability 0.9" can only be established on condition that the protocol does not loop back to initial state forever. In the randomized dining philosophers example, if the scheduler never selects a given philosopher for execution even though he is ready to proceed (e.g. to flip the coin) the run thus produced would be unfair, and as a result one could not guarantee lack of starvation. In summary, fairness assumptions allow us to prove more properties, but at a cost of qualifying the correctness statement in the following sense: the system has been shown to satisfy a property *on condition that* the scheduler or user complies with the given fairness assumptions. We should point out that fairness can also be considered w.r.t. the probabilistic choices as in [49, 50, 51], but our approach is different as we impose *fairness of schedulers*.

Fairness for schedulers of concurrent probabilistic systems was first introduced by Hart, Sharir & Pnueli [33] and later considered by Vardi [58]. While [33] deals with concurrent probabilistic systems which arise by the interleaving of sequential probabilistic processes and defines an execution sequence π to be fair iff each sequential process is activated infinitely often in π, [58] deals with "concurrent Markov chains", which distinguish between non-deterministic and probabilistic states, and defines an execution sequence π of a concurrent Markov chain to be fair if all possible successor states of a non-deterministic state, in which fairness is required and which occur infinitely often in π, also occur infinitely often. We adapt Vardi's notion of fairness to our model for concurrent probabilistic processes – which does not distinguish between non-deterministic and probabilistic states – and define an execution sequence π to be fair if none of the non-deterministic alternatives in a state occurring infinitely often in π is refused continuously. Following [33] we define two types of fairness for schedulers: a scheduler is *strictly fair* iff each of its execution sequences is fair, and it is *fair* if almost all execution sequences are fair, i.e. if the measure of its fair execution sequences is 1.

We introduce a probabilistic branching time logic *PBTL*, based on the logics considered in [30, 31, 55, 14, 6, 4], which we interpret over concurrent probabilistic systems. *PBTL* contains atomic propositions, the usual boolean connectives, a next-step, a bounded and an unbounded until operator, and the usual branching time quantifiers $\exists$ ("there is a scheduler") and $\forall$ ("for all schedulers"). The next-step and the until operators are lifted to the probabilistic case, thus yielding formulas which express properties such as "the system terminates within k steps with probability $\geq p$". Depending on the interpretation, the branching time quantifiers $\exists$ and $\forall$ range over a certain type of schedulers. In contrast to the non-probabilistic case, where the schedulers that resolve the non-deterministic choices yield execution sequences of the system, the schedulers of a concurrent probabilistic system yield computation trees referred to above. We give three interpretations which differ in the range of schedulers. For each interpretation we fix a certain kind of a scheduler, and consider the computation trees which arise by schedulers of the given kind. The first interpretation does not make any restrictions on the schedulers, i.e. all schedulers are allowed. In the second and third interpretation we restrict our attention to the fair, resp. strictly fair, schedulers. It turns out that the interpretations based on fair or strictly fair schedulers allows to establish more (quantative or qualitative) liveness properties than the standard interpretation. The difference between the interpretations obtained by considering either the fair or the strictly fair schedulers is only marginal. This result is not surprising as it is already shown in [33] that each strictly fair scheduler can be "approximated" by fair schedulers.

A method for model checking w.r.t. the satisfaction relation induced by the first interpretation (the interpretation which does not make any fairness assumptions) is given in [14]; the same method also applies in our setting. For the fair and the strictly fair interpretation, we show why the standard procedure cannot be applied and give a model checking algorithm for $PBTL$: it takes a formula Φ and a concurrent probabilistic process as its input, and returns the set of states satisfying Φ. The algorithm works similarly to the model checking algorithm of [16] for CTL. Given a formula Φ we first construct the parse tree of Φ whose nodes represent the subformulas of Φ. Then, we successively compute the sets consisting of the states satisfying the subformula which the node v represents. It turns out that simple adaptations of existing techniques can be used for all operators except those containing unbounded until as their outermost operator. The latter is non-trivial, and we base the proposed solution on a series of technical results. In the non-probabilistic case the states satisfying a formula with an until operator can be obtained by an analysis of the strongly connected components of the underlying directed graph (cf. [16]). In the probabilistic case, this is not sufficient since for every computation tree the probability of the set of paths fulfilling the until condition is needed (instead of only the existence or non-existence of certain paths as in the non-probabilistic case). Until formulas assert something about the set of (strictly) fair computation trees, which in general is, as the (strictly) fair computation trees themselves, infinite. We show that – instead of ranging over *all* fair, resp. all strictly fair, schedulers – an investigation of the *simple* schedulers, a certain *finite* subclass of (in general unfair) schedulers whose computation trees can be represented by *finite-state* Markov chains suffices. While the system behaviour under such a simple scheduler can be described by a finite-state Markov chain where the probability of the set of paths fulfilling an until-property can be computed by solving a linear equation system (cf. [19, 31]), the minimal and maximal probabilities under all simple schedulers can be obtained by solving a linear optimization problem (cf. [20, 14]). We show that the problem of computing the minimal and maximal probabilities under all fair schedulers can be reduced to the computation of the minimal or maximal probabilities under all schedulers. Hence, as in the case of [14], our model checker uses standard methods of linear programming for the handling of formulas containing unbounded until as the outermost operator.

A short version of this paper appeared as [11].

Organization of the paper: Section 2 recalls the definition of (sequential) Markov chains (which we use to describe the computation trees of a concurrent probabilistic system) and the probability measure on their paths. Section 3 explains our model for concurrent probabilistic systems, defines adversaries for them and shows how concurrent probabilistic systems can be split into computation trees. Fairness and strict fairness of adversaries of concurrent probabilistic systems is introduced in Section 3.3. The syntax and the three semantics of our logic $PBTL$ are explained in Section 4. Our main results are contained in Section 5, where a method for testing whether a $PBTL$ formula is satisfied by a given concurrent probabilistic process under fairness assumptions is given. For readers' convenience, we omit the technical development of the results from this section; this is included in the Appendix (Section 12). In Section 6 we show an application of our algorithm to distributed systems: we discuss a simple protocol and demonstrate the need for fairness assumptions. The time complexity of our method is discussed in Section 7. Section 8 shows how to deal with fairness in the sense of [58] in our setting. In Section 9 we briefly explain how the model checking for the logic $pCTL^*$ presented in [4] can be modified to handle fairness. Section 10 discusses related work. Finally, Section 11 concludes the paper.

The reader is supposed to be familiar with basic notion of measure and probability theory (see e.g. [29]).

2 (Sequential) Markov chains

In this section we briefly recall the definition of Markov chains and the probability measure on their paths. We use Markov chains to describe the computation trees of a concurrent probabilistic system.

A *(sequential) Markov chain* is a tuple $MC = (S, \mathbf{P})$ where S is a countable set of states and $\mathbf{P} : S \times S \to [0, 1]$ is a function with $\sum_{t \in S} \mathbf{P}(s, t) = 1$ for all $s \in S$. A *path* in MC is a nonempty and finite or infinite sequence $x = s_0 s_1 \ldots$ consisting of states $s_i \in S$ s.t. $\mathbf{P}(s_i, s_{i+1}) > 0$. A *fulpath* is an infinite path in MC. $Path_{ful}(MC)$ denotes the set of fulpaths in MC, $Path_{ful}(s, MC)$ the set of fulpaths starting in s. $Path_{fin}(MC)$ denotes the set of finite paths in MC, $Path_{fin}(s, MC)$ the set of finite paths starting in s. Given a Markov chain $(S, \mathbf{P})$ and a state s, $\mathbf{P}$ induces a probabilistic space on $Path_{ful}(s, MC)$ as follows. We define the probability $\mathbf{P}(x, MC)$, abbrev. $\mathbf{P}(x)$, for finite paths x in MC by putting $\mathbf{P}(x) = 1$ if $x = s$, and $\mathbf{P}(x) = \mathbf{P}(s_0, s_1) \cdot \mathbf{P}(s_1, s_2) \cdot \ldots \cdot \mathbf{P}(s_{n-1}, s_n)$ otherwise, where $x = s_0 s_1 \ldots s_n$. Let $\Sigma(s, MC)$ be the smallest σ-algebra on $Path_{ful}(s, MC)$ which contains the sets $\{y \in Path_{ful}(s, MC) : x \text{ is a prefix of } y\}$, $x \in Path_{fin}(s, MC)$. The probability measure $Prob$ on $\Sigma(s, MC)$ is the unique measure with $Prob\,\{y \in Path_{ful}(s, MC) : x \text{ is a prefix of } y\} = \mathbf{P}(x)$.

3 Concurrent probabilistic systems

As pointed out in [58], certain states of a concurrent system whose components work asynchronously are inherently non-deterministic. The non-deterministic choices are beyond control of the process and can be supposed to be resolved by a scheduler, whereas the probabilistic choices are made by the system itself. In this section we introduce a model for concurrent probabilistic systems which is based on Markov decision processes (see e.g. [22, 53]). It generalizes the "concurrent Markov chains" considered e.g. in [58, 21] and essentially agrees with the "simple deterministic automata" of [55].

Notation 3.1 *For a finite set S, a* distribution *on S is a function $\mu : S \to [0, 1]$ such that $\sum_{t \in S} \mu(t) = 1$. If $s \in S$ then μ_s^1 denotes the unique distribution on S with $\mu_s^1(s) = 1$. $Supp(\mu)$ denotes the* support *of μ, i.e. the set of states $s \in S$ with $\mu(s) > 0$.*

State space: $S = \{s, t, u, v\}$
$Steps(s) = \{\mu, \mu_v^1\}$,
$Steps(t) = \{\mu_s^1\}$,
$Steps(x) = \{\mu_x^1\}$, $x \in \{u, v\}$

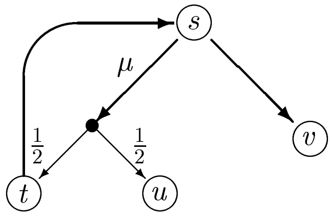


Figure 1: A concurrent probabilistic system

Definition 3.2 *A* concurrent probabilistic system *is a pair* $\mathcal{S} = (S, Steps)$ *where* S *is a finite set of states and Steps a function which assigns to each state* $s \in S$ *a finite, non-empty set* $Steps(s)$ *of distributions on* S.

Intuitively, *Steps* represents the non-deterministic alternatives in each state: given a state $s \in S$, a scheduler chooses some $\mu \in Steps(s)$. The process itself resolves the probabilistic choice, i.e. selects some state t with positive probability ($\mu(t) > 0$). We refer to the elements of $Steps(s)$ as the *transitions* of s. We model terminating behaviour by repeating the final state infinitely often, i.e. if s is a terminating state then we suppose that $Steps(s) = \{\mu_s^1\}$.

We depict concurrent probabilistic systems as follows. We use circles for the states. Thick lines stand for the outgoing transitions from a state. The thick line corresponding to a distribution $\mu \in Steps(s) \setminus \{\mu_t^1 : t \in S\}$ is directed and ends in a small filled circle that represents the probabilistic choice. We use directed thin lines leading from the circle of a probabilistic choice to the possible successor states (i.e. all states t where $\mu(t) > 0$). A distribution $\mu_t^1 \in Steps(s)$ is represented by a thick arrow leading from s to t. For the "terminal" states (i.e. all states $s \in S$ where $Steps(s) = \{\mu_s^1\}$) we omit the outgoing transition. For instance, for the system in Figure 1, non-deterministic choice is present only in state s. The other states are "deterministic" since there is only one distribution in $Steps(\cdot)$.

3.1 Paths in concurrent probabilistic systems

Execution sequences (which we call paths) arise by resolving both the non-deterministic and probabilistic choices. Formally, a *path* in a concurrent probabilistic system $\mathcal{S} = (S, Steps)$ is a nonempty (finite or infinite) "sequence"

$$\pi = s_0 \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} s_2 \ \ldots$$

where s_i are states and $\mu_i \in Steps(s_{i-1})$ and $\mu_i(s_i) > 0$, $i = 1, 2, \ldots$. (The case $\pi = s_0$ is allowed.) A path π is called a *fulpath* iff it is infinite. We use the following notation for paths. The first state of a path π is denoted by $first(\pi)$. If π is finite then the last state of π is denoted by $last(\pi)$. The length $|\pi|$ of a path is defined in the usual way as follows: if $\pi = s_0 \in S$ then $|\pi| = 0$; otherwise, $\pi = s_0 \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \ldots \xrightarrow{\mu_n} s_n$, in which case $|\pi| = n$. For infinite π we put $|\pi| = \infty$. If π is a fulpath then $inf(\pi)$ denotes the set of states $s \in S$ with $s = \pi(i)$ for infinitely many i. Let π be a finite or infinite path as above. If $k \leq |\pi|$ then $\pi(k)$ denotes the k-th state of π (i.e. $\pi(k) = s_k$). $\pi^{(k)}$ is the k-th prefix of π (i.e. if $k < |\pi|$ then $\pi^{(k)} = s_0 \xrightarrow{\mu_1} s_1 \xrightarrow{\mu_2} \dots \xrightarrow{\mu_k} s_k$, if $k \geq |\pi|$ then $\pi^{(k)} = \pi$.) If $i < |\pi|$ then we put $step(\pi, i) = \mu_{i+1}$. If ω is a finite path, π a path in $\mathcal{S}$ such that $last(\omega) = first(\pi)$ then $\omega\pi$ denotes the path with

$$(\omega\pi)(i) = \begin{cases} \omega(i) & : \text{ if } i < |\omega| \\ \pi(i - |\omega|) & : \text{ if } i \geq |\omega| \end{cases} \qquad step(\omega\pi, i) = \begin{cases} step(\omega, i) & : \text{ if } i < |\omega| \\ step(\pi, i - |\omega|) & : \text{ if } i \geq |\omega|. \end{cases}$$

In particular, $s\pi = \pi$ if $first(\pi) = s$. If ω is a finite path, $\mu \in Steps(last(\omega))$, $s \in S$ such that $\mu(s) > 0$ and π a path with $first(\pi) = s$ then $\omega \xrightarrow{\mu} \pi$ denotes the unique path γ with $\gamma^{(i)} = \omega$, $step(\gamma, i) = \mu$, $\gamma(i+1) = s$ and $\gamma(j+i+1) = \pi(j)$, $step(\gamma, j+i+1) = step(\pi, j)$, $j = 0, 1, \dots, |\pi|$ where $i = |\omega|$.

Example 3.3 For the system in Figure 1, $\omega = s \xrightarrow{\mu} t \xrightarrow{\mu_s^1} s \xrightarrow{\mu_v^1} v \xrightarrow{\mu_v^1} v$ is a finite path with $first(\omega) = \omega(0) = s$, $last(\omega) = \omega(4) = \omega(3) = v$, $\omega(1) = t$, $\omega(2) = s$, $step(\omega, 0) = \mu$, $step(\omega, 1) = \mu_s^1$, $step(\omega, 2) = step(\omega, 3) = \mu_v^1$ and $|\omega| = 4$. We have:

$$\omega^{(2)} = s \xrightarrow{\mu} t \xrightarrow{\mu_s^1} s, \quad \omega^{(2)} \left(s \xrightarrow{\mu} t\right) = s \xrightarrow{\mu} t \xrightarrow{\mu_s^1} s \xrightarrow{\mu} t, \quad \omega^{(2)} \xrightarrow{\mu_v^1} \left(v \xrightarrow{\mu_v^1} v\right) = \omega.$$

Moreover, $\omega v = \omega$ (where v stands for a finite path of length 0). ■

$Path_{ful}(\mathcal{S})$ denotes the set of all fulpaths in $\mathcal{S}$, $Path_{fin}(\mathcal{S})$ the set of all finite paths in $\mathcal{S}$, and $Path_{ful}(s, \mathcal{S})$ the set of fulpaths π with $first(\pi) = s$. A state t is called *reachable* from s if there exists a finite path π with $first(\pi) = s$ and $last(\pi) = t$. $Reach(s, \mathcal{S})$ denotes the set of states which are reachable from s. When it is clear from the context what $\mathcal{S}$ is we abbreviate $Path_{ful}(\mathcal{S})$ by $Path_{ful}$, and similarly $Path_{fin}(\mathcal{S})$ by $Path_{fin}$, $Path_{ful}(s, \mathcal{S})$ by $Path_{ful}(s)$, and $Reach(s, \mathcal{S})$ by $Reach(s)$. If Γ is a set of fulpaths in $\mathcal{S}$ and $s \in S$ then $\Gamma(s) = \Gamma \cap Path_{ful}(s)$. Similarly, if Ω is a set of finite paths then $\Omega(s) = \Omega \cap Path_{fin}(s)$.

3.2 Adversaries of concurrent probabilistic systems

We split a concurrent probabilistic system $\mathcal{S} = (S, Steps)$ into its computation trees (called "execution trees" in [33] and "maximal resolutions" in [38]), with each component described as a Markov chain. The computation trees arise by resolving the nondeterministic choices (but not the probabilistic choices). It is convenient to suppose that an adversary (called "policy" in the theory of Markov decision processes) decides – based on the past history of the system – which of the possible steps (probability distributions) to perform next. We only consider deterministic adversaries, i.e. those that schedule a unique next step. The notion of randomization of adversaries or probabilistic adversaries has been investigated in [33] and [55], where it is shown that the probability of a measurable set Γ w.r.t. a randomized adversary is a convex combination of the measure of Γ w.r.t. non-randomized adversaries, and hence lies between the minimal and maximal measure of Γ w.r.t. non-randomized adversaries. Since we are only interested in the maximal and minimal measures (cf. Section 4), we shall not need the randomized adversaries.

Definition 3.4 *An* adversary *(or* scheduler*) of a concurrent probabilistic system $\mathcal{S} = (S, Steps)$ is a function A mapping every finite path ω of $\mathcal{S}$ to a distribution $A(\omega)$ on S such that $A(\omega) \in Steps(last(\omega))$ is a transition in $\mathcal{S}$. An adversary A of $\mathcal{S}$ is called* simple *iff for every state $s \in S$ there exists a transition $\mu_s \in Steps(s)$ with $A(\omega) = \mu_{last(\omega)}$ for all $\omega \in Path_{fin}(\mathcal{S})$. $\mathcal{A}(\mathcal{S})$ denotes the set of all adversaries of $\mathcal{S}$ and $\mathcal{A}_{simple}(\mathcal{S})$ the set of simple adversaries.*

When clear from the context we write $\mathcal{A}$ and $\mathcal{A}_{simple}$ rather than $\mathcal{A}(\mathcal{S})$ and $\mathcal{A}_{simple}(\mathcal{S})$. An adversary (called "policy" in Markov decision processes) chooses for every finite path ω in $\mathcal{S}$ an outgoing transition from $last(\omega)$. Simple adversaries (corresponding to "stationary policies") resolve the non-determinism by selecting for every state a next step which is executed whenever the state s is reached – independent of the past history. In some sense, simple adversaries are extremely unfair and would be ruled out for practical purposes. We need them only for the sake of convenience. For example, the system of Figure 1 has exactly two simple adversaries A, B. These are given by $A(s) = \mu$, $B(s) = \mu_v^1$. (Note that the other states are "deterministic".)

With each adversary we associate a sequential (in general infinite-state) Markov chain which can be viewed as a computation tree of $\mathcal{S}$. Formally, if A is an adversary of a concurrent probabilistic system $\mathcal{S} = (S, Steps)$ then $MC^A = (Path_{fin}(\mathcal{S}), \mathbf{P}^A)$ is a Markov chain where $\mathbf{P}^A(\omega, \omega') = A(\omega)(s)$ if ω' is of the form $\omega \xrightarrow{A(\omega)} s$ and $\mathbf{P}^A(\omega, \omega') = 0$ in all other cases. If A is simple then MC^A can be identified with the finite-state Markov chain (S, A) where A is viewed as a function $S \times S \to [0, 1]$. For a simple adversary A we write $A(s, t)$ instead of $A(s)(t)$.

For an adversary A of a concurrent probabilistic system $\mathcal{S} = (S, Steps)$, $Path_{ful}^A(\mathcal{S})$, abbrev. $Path_{ful}^A$, denotes the set of all paths $\pi \in Path_{ful}(\mathcal{S})$ with $step(\pi, i) = A(\pi^{(i)})$ for all $i \geq 0$. Similarly, we define the set of paths $Path_{fin}^A(\mathcal{S})$ induced by the adversary A, abbrev. $Path_{fin}^A$, to be the set of all finite paths $\omega \in Path_{fin}$ with $step(\omega, i) = A(\omega^{(i)})$ for all $i < |\omega|$. If Γ is a set of fulpaths in $\mathcal{S}$ then $\Gamma^A = \Gamma \cap Path_{ful}^A$. In the notation of Section 3.1, $Path_{ful}^A(s) = Path_{ful}(s) \cap Path_{ful}^A$, $Path_{fin}^A(s) = Path_{fin}(s) \cap Path_{fin}^A$ and $\Gamma^A(s) = \{\pi \in \Gamma : first(\pi) = s, \pi \in Path_{ful}^A\}$. $Reach^A(s, \mathcal{S})$, abbrev. $Reach^A(s)$, denotes the set of states $t \in S$ such that there exists $\omega \in Path_{fin}^A(s)$ with $last(\omega) = t$. We identify each path $x = \omega_0\omega_1 \ldots$ in MC^A which starts in a state $s_0 \in S$ (i.e. $\omega_0 = s_0$ is a path of length 0) with the path

$$last(\omega_0) \xrightarrow{A(\omega_0)} last(\omega_1) \xrightarrow{A(\omega_1)} \ldots$$

in $\mathcal{S}$. Vice versa, if $\pi \in Path_{ful}^A$ then we identify π with the path $x = \pi^{(0)}\pi^{(1)}\pi^{(2)} \ldots$ in MC^A. This yields a one-to-one correspondence between $Path_{ful}^A(s)$ and $Path_{ful}(s, MC^A)$. Hence, for each $s \in S$ and adversary A, $Path_{ful}^A(s)$ is a probabilistic space (where the σ-algebra $\Sigma(s)$ and the measure $Prob$ is defined as in Section 2). If $\Gamma \subseteq Path_{ful}(s)$ and Γ^A is measurable then we refer to $Prob(\Gamma^A)$ as the measure of Γ w.r.t. A. For instance, for the system of Figure 1 and the finite path ω of Example 3.3, $\omega \in Path_{fin}^A(s)$ for each adversary A with $A(s) = \mu$, $A(s \xrightarrow{\mu} t \xrightarrow{\mu_s^1} s) = \mu_v^1$. For each such adversary A, the probability measure of the set of fulpaths $\pi \in Path_{ful}^A(s)$ which have ω as a prefix is $\frac{1}{2}$.

3.3 Fairness and strict fairness of adversaries

Our notion of fairness imposes (strong) fairness of the adversaries, or, in other words, fairness of non-deterministic choices between probability distributions as in [33, 58] (rather than the probabilistic choices as in [49, 50, 51, 12]). We adapt Vardi's notion of fair paths in concurrent Markov chains to our (more general) model for concurrent probabilistic systems; recall that we do not distinguish non-deterministic and probabilistic states. For simplicity, we require fairness in all states, which differs from the approach of [58] where the set of non-deterministic states is partitioned into the "fair" states (in which fairness is required) and possibly "unfair" states. This simplification is made for technical reasons only. In Section 8 we briefly explain how to extend our approach to cater for such sets of fair states. We define a fulpath π of a concurrent probabilistic system to be *fair* iff for each state s occurring infinitely often in π, each non-deterministic alternative which is enabled in s (i.e. each distribution $\mu \in \mathit{Steps}(s)$) is taken infinitely often in π.

Definition 3.5 *Let $\mathcal{S} = (S, \mathit{Steps})$ be a concurrent probabilistic system and π a fulpath in $\mathcal{S}$. π is called* fair *iff, for each $s \in \mathit{inf}(\pi)$ and each $\mu \in \mathit{Steps}(s)$, there are infinitely many indices i with $\pi(i) = s$ and $\mathit{step}(\pi, i) = \mu$. $\mathit{Fair}(\mathcal{S})$, abbrev. Fair, denotes the set of fair fulpaths in $\mathcal{S}$.*

Remark 3.6 Our notion of fairness of a fulpath is stronger than fairness of fulpaths in [33]. In [33] "process fairness" is considered, in the sense that for π to be fair all sequential processes (whose composition is the concurrent probabilistic system under consideration) are activated infinitely many times in π. If $\mathcal{S}$ is a concurrent probabilistic system which arises through the interleaving of sequential processes without shared variables then fairness in the sense of Definition 3.5 implies fairness in the sense of [33]; to see this suppose that there are k sequential probabilistic processes $\mathcal{P}_1, \ldots, \mathcal{P}_k$ where each of them is described by a Markov chain $MC_i = (S_i, \mathbf{P}_i)$, $i = 1, \ldots, k$, and that $\mathcal{S} = (S, \mathit{Steps})$ where $S = S_1 \times \ldots \times S_k$ and $\mathit{Steps}(s_1, \ldots, s_k) = \{\nu^i_{(s_1,\ldots,s_k)} : i = 1, \ldots, k\}$ where

$$\nu^i_{(s_1,\ldots,s_k)}(t_1, \ldots, t_k) \;=\; \begin{cases} \mathbf{P}_i(s_i, t_i) & : \ \text{if } t_j = s_j,\ j = 1, \ldots, k,\ i \neq j \\ 0 & : \ \text{otherwise.} \end{cases}$$

Then, whenever π is a fulpath in $\mathcal{S}$ that is fair in the sense of Definition 3.5 then π is fair in the sense of [33], which requires that for each $i \in \{1, \ldots, k\}$ there are infinitely many indices $j \geq 0$ with $\mathit{step}(\pi, j) = \nu^i_{\pi(j)}$. ■

As in [33] we consider two kinds of fairness for adversaries: *strictly fair* adversaries, where each fulpath is fair, and *fair* adversaries, where the set of fair paths has probability 1.

Definition 3.7 *Let $\mathcal{S} = (S, \mathit{Steps})$ be a concurrent probabilistic system and F an adversary for $\mathcal{S}$. F is called* strictly fair *iff $\mathit{Path}^F_{\mathit{ful}} \subseteq \mathit{Fair}$. F is called* fair *iff $\mathit{Prob}(\mathit{Fair}^F(s)) = 1$ for all $s \in S$. $\mathcal{A}_{\mathit{sfair}}(\mathcal{S})$ denotes the set of strictly fair adversaries, $\mathcal{A}_{\mathit{fair}}(\mathcal{S})$ the set of fair adversaries.*

When clear from the context, we write $\mathcal{A}_{\mathit{sfair}}$ and $\mathcal{A}_{\mathit{fair}}$ rather than $\mathcal{A}_{\mathit{sfair}}(\mathcal{S})$ and $\mathcal{A}_{\mathit{fair}}(\mathcal{S})$. Clearly, strictly fair adversaries are fair. If F is a fair adversary then for each $\omega \in \mathit{Path}^F_{\mathit{fin}}$ there exists $\pi \in \mathit{Fair}^F$ where ω is a prefix of π. This reflects "liveness" in the sense of [1] which states that every finite computation can be extended to an infinite (fair) computation.

Example 3.8 For the system of Figure 1, the fulpath $\pi_0 = s \xrightarrow{\mu} t \xrightarrow{\mu_s^1} s \xrightarrow{\mu} t \xrightarrow{\mu_s^1} s \xrightarrow{\mu} \ldots$ is not fair since $s \in \mathit{inf}(\pi_0)$ and $\mu_v^1 \notin \{\mathit{step}(\pi_0, i) : i \geq 0\}$. Every other fulpath $\pi \in \mathit{Path}_{\mathit{ful}}(s)$ "ends" in v or u (i.e. $\pi(i) \in \{u, v\}$ for almost all i). Thus, $\mathit{Fair}(s) = \mathit{Path}_{\mathit{ful}}(s) \setminus \{\pi_0\}$. The simple adversary B with $B(s) = \mu_v^1$ is strictly fair since $\pi_0 \notin \mathit{Path}_{\mathit{ful}}^B$. The simple adversary A with $A(s) = \mu$ is not strictly fair since $\pi_0 \in \mathit{Path}_{\mathit{ful}}^A(s)$. Nevertheless, A is fair. To see this, consider the set Γ of all fulpaths $\pi \in \mathit{Path}_{\mathit{ful}}$ where $\pi(i) \in \{u, v\}$ for almost all (or, in this case, for some) i. Then, $\Gamma^A(x) = \mathit{Fair}^A(x)$ for all states x since $\mathit{Prob}(\mathit{Fair}^A(u)) = \mathit{Prob}(\mathit{Fair}^A(v)) = 1$,

$$\mathit{Prob}\left(\mathit{Fair}^A(s)\right) = \sum_{i=0}^{\infty} \frac{1}{2} \cdot \left(\frac{1}{2}\right)^i = 1$$

and $\mathit{Prob}\left(\mathit{Fair}^A(t)\right) = \mathit{Prob}\left\{t \xrightarrow{\mu_s^1} \pi : \pi \in \mathit{Fair}^A(s)\right\} = 1$. Hence, A is fair. ■

4 Probabilistic branching time temporal logic

In this section we introduce the syntax of the logic *PBTL* (probabilistic branching time logic) interpreted over concurrent probabilistic systems and give three semantics for it. *PBTL* is a probabilistic extension of *CTL*[1] which allows to express quantitative properties such as "the system terminates within 3 steps with probability at least $\frac{2}{3}$". In essence, the syntax of *PBTL* agrees with the logic *PCTL* considered by Segala & Lynch [55] and the logic *pCTL* considered by Bianco & de Alfaro [14]. As in [14, 55], *PBTL* formulas are interpreted over the states of a concurrent probabilistic system. Note, however, that the models allow non-determinism in the sense of selecting one of possibly many distributions, and so one cannot establish the probability of an event unless non-determinism has been resolved (by means of adversaries). Since each adversary induces a probability space on paths, it is thus natural to allow *quantification over adversaries*: we replace the *CTL* formulas of the form $\exists\varphi$ (where φ is a path formula) by formulas of the form $[\,\exists\varphi\,]_{\geq p}$ (or $[\,\exists\varphi\,]_{>p}$) which state that there exists an adversary such that the probability for φ is $\geq p$ (or $> p$). In other words, the quantifiers $\exists$ and $\forall$ in *PBTL* range over the adversaries but yield Markov chains and not paths as in the non-probabilistic case.[2]

PBTL contains atomic propositions and operators of: next-step X, bounded until $\mathcal{U}^{\leq k}$ and unbounded until $\mathcal{U}$. The operators X, $\mathcal{U}^{\leq k}$ and $\mathcal{U}$ are used in connection with the branching time quantifiers $\exists$ and $\forall$ and an interval of probabilities. The bounded until operator can be used to describe constraints such as "with probability p, within k steps

[1] We call our logic *PBTL* in order to prevent confusion with other probabilistic extensions of *CTL*, e.g. *PCTL* [55], *TPCTL* [30] or *pCTL* [14].

[2] This should be contrasted with the logics *PCTL*/*PCTL** considered in [31, 6, 37] which are probabilistic extensions of *CTL*/*CTL** that are interpreted over (sequential) Markov chains. *PCTL* and *PCTL** use the probabilistic operator $\mathbb{P}_*$ instead of the *CTL* path quantifiers $\exists$ and $\forall$. The *CTL*/*CTL** formulas $\exists\varphi$ and $\forall\varphi$ are replaced in *PCTL*/*PCTL** by formulas of the form $\mathbb{P}_{\geq p}(\varphi)$ which is true iff the probability measure of the paths fulfilling φ is $\geq p$.

something will happen" where the interpretation of a "step" depends on the underlying system. If the components of a concurrent system are asynchronous and proceed at a different pace, one can think of a step as the time taken by the slowest component to perform an atomic action.

In what follows we suppose a fixed set of atomic propositions. The syntax of $PBTL$ is:

$$\begin{array}{rcl}\Phi & ::= & tt \mid a \mid \Phi_1 \wedge \Phi_2 \mid \neg\Phi \mid [\ \exists X\ \Phi\ ]_{\sqsupseteq p} \mid [\ \forall X\ \Phi\ ]_{\sqsupseteq p} \mid [\ \Phi_1\ \exists\mathcal{U}^{\leq k}\ \Phi_2\ ]_{\sqsupseteq p} \mid \\ & & [\ \Phi_1\ \forall\mathcal{U}^{\leq k}\ \Phi_2\ ]_{\sqsupseteq p} \mid [\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p} \mid [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}\end{array}$$

where a is an atomic proposition, $p \in [0,1]$, $\sqsupseteq$ is either $\geq$ or $>$, and k a non-negative integer. Formulas of the form $X\Phi$, $\Phi_1\mathcal{U}^{\leq k}\Phi_2$ or $\Phi_1\mathcal{U}\Phi_2$, where Φ, Φ_1, Φ_2 are $PBTL$ formulas, are called *path formulas*.

$PBTL$ formulas are interpreted over states of concurrent probabilistic processes, whereas path formulas over paths. Informally, $X\Phi$ asserts that Φ holds in the next state. Formulas of the form $\Phi_1\mathcal{U}^{\leq k}\Phi_2$ state that there is some l with $0 \leq l \leq k$ such that Φ_1 holds from now on and including the $(l-1)$-th step and Φ_2 holds in the l-th step. $\Phi_1\mathcal{U}\Phi_2$ is true iff from now on Φ_1 is fulfilled until Φ_2 holds and Φ_2 holds sometime in the future. Depending on the interpretation, the branching time quantifiers $\exists$ and $\forall$ denote quantification over the adversaries of a certain type: $\exists$ "there exists an adversary of this type", and $\forall$ "for all adversaries of this type". The subscript $\sqsupseteq p$ denotes that the probability for paths (in an adversary) fulfilling the path formula is $\sqsupseteq p$. The usual derived constants and operators are: $ff = \neg tt$, $\Phi_1 \vee \Phi_2 = \neg(\neg\Phi_1 \wedge \neg\Phi_2)$, $\Phi_1 \rightarrow \Phi_2 = \neg\Phi_1 \vee \Phi_2$. Operators for modelling "eventually" $\Diamond$ or "always" $\Box$ can be derived by: $[\ \exists\Diamond\ \Phi\ ]_{\sqsupseteq p} = [\ tt\ \exists\mathcal{U}\ \Phi\ ]_{\sqsupseteq p}$, $[\ \forall\Diamond\ \Phi\ ]_{\sqsupseteq p} = [\ tt\ \forall\mathcal{U}\ \Phi\ ]_{\sqsupseteq p}$, $[\ \exists\Box\ \Phi\ ]_{\sqsupseteq p} = \neg[\ \forall\Diamond\ \neg\Phi\ ]_{\overline{\sqsupseteq}1-p}$, $[\ \forall\Box\ \Phi\ ]_{\sqsupseteq p} = \neg\ [\ \exists\Diamond\ \neg\Phi\ ]_{\overline{\sqsupseteq}1-p}$ where $\overline{\geq} = >$ and $\overline{>} = \geq$. For instance, $[\ \exists\Diamond\ \Phi\ ]_{\geq p}$ states the existence of an adversary where the probability for a path in which Φ eventually holds is $\geq p$. Formulas of the form $[\ \forall\Box\ \Phi\ ]_{\geq p}$ express safety properties asserting that for all adversaries the probability for paths where Φ holds continuously is $\geq p$.

The main difference between our logic $PBTL$ and the logic $PCTL$ of [55] is that the latter deals with action-labelled concurrent probabilistic systems, while we label the states with atomic propositions. The logic $pCTL$ of [14] (and [4]) essentially agrees with our logic $PBTL$ except that [14] uses a probabilistic operator $I\!\!P_*(\cdot)$ for formulas containing the until operator. The $PBTL$ formula $[\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ can be identified with the $pCTL$ formula $I\!\!P_{\sqsupseteq p}(\Phi_1\mathcal{U}\Phi_2)$, while $[\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{>p}$ corresponds to $\neg I\!\!P_{\leq p}(\Phi_1\mathcal{U}\Phi_2)$. It should be pointed out that [4] uses an extension of $pCTL$ that contains an operator to express bounds on the average time between events which does not have a counterpart in $PBTL$. More minor differences between $pCTL$ and $PBTL$ are that $PBTL$ contains the next step operator X and the bounded until operator $\mathcal{U}^{\leq k}$, whereas $pCTL$ does not (but these operators could easily be added). Vice versa, $pCTL$ contains the usual CTL quantifiers A and E that range over all paths: A meaning "for all paths" and E "there is a path". Formulas of the form $A\varphi$ and $E\varphi$ (where φ is a path formula) could be added to our logic $PBTL$, but we omit them for the sake of simplicity. For the semantics of $A\varphi$ and $E\varphi$ we can either use the standard interpretation (where $A\varphi$ asserts that φ holds for all fulpaths) or an interpretation that requires "path fairness" (i.e. where $A\varphi$ asserts that all fair fulpaths satisfy φ).[3]

[3]In the latter case our model checker of Section 5 would have to be extended, e.g. using the method

$s \models_{Adv} tt$ for all $s \in S$ and $s \models_{Adv} a$ iff $a \in L(s)$
$s \models_{Adv} \Phi_1 \wedge \Phi_2$ iff $s \models_{Adv} \Phi_i$, $i = 1, 2$
$s \models_{Adv} \neg\Phi$ iff $s \not\models_{Adv} \Phi$
$s \models_{Adv} [\exists X\Phi]_{\sqsupseteq p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} X\Phi\} \sqsupseteq p$ for some $A \in Adv$.
$s \models_{Adv} [\forall X\Phi]_{\sqsupseteq p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} X\Phi\} \sqsupseteq p$ for all $A \in Adv$.
$s \models_{Adv} [\Phi_1 \exists \mathcal{U}^{\leq k} \Phi_2]_{\sqsupseteq p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} \Phi_1 \mathcal{U}^{\leq k} \Phi_2\} \sqsupseteq p$ for some $A \in Adv$.
$s \models_{Adv} [\Phi_1 \forall \mathcal{U}^{\leq k} \Phi_2]_{\sqsupseteq p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} \Phi_1 \mathcal{U}^{\leq k} \Phi_2\} \sqsupseteq p$ for all $A \in Adv$.
$s \models_{Adv} [\Phi_1 \exists \mathcal{U} \Phi_2]_{\sqsupseteq p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} \Phi_1 \mathcal{U} \Phi_2\} \sqsupseteq p$ for some $A \in Adv$.
$s \models_{Adv} [\Phi_1 \forall \mathcal{U} \Phi_2]_{\sqsupseteq p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} \Phi_1 \mathcal{U} \Phi_2\} \sqsupseteq p$ for all $A \in Adv$.
$\pi \models_{Adv} X\Phi$ iff $\pi(1) \models_{Adv} \Phi$
$\pi \models_{Adv} \Phi_1 \mathcal{U}^{\leq k} \Phi_2$ iff $\pi(l) \models_{Adv} \Phi_2$ and $\pi(i) \models_{Adv} \Phi_1$, $i = 0, \ldots, l-1$, for some $l \leq k$
$\pi \models_{Adv} \Phi_1 \mathcal{U} \Phi_2$ iff $\pi \models_{Adv} \Phi_1 \mathcal{U}^{\leq k} \Phi_2$ for some $k \geq 0$.

Figure 2: The satisfaction relation $\models_{Adv}$

Definition 4.1 *A PBTL*-structure *is a tuple $M = (\mathcal{S}, L)$ where $\mathcal{S} = (S, Steps)$ is a concurrent probabilistic process and L a labelling function for the states, i.e. L is a mapping which assigns to each state s of S a set $L(s)$ of atomic propositions that are true in s.*

For a given *PBTL*-structure $M = (\mathcal{S}, L)$ and a set Adv of adversaries of $\mathcal{S}$, the satisfaction relation $\models_{Adv} \subseteq S \times PBTL$ is shown in Figure 2 (where we write $s \models_{Adv} \Phi$ instead of $(s, \Phi) \in \models_{Adv}$). When referring to the truth value of a path formula w.r.t. a path $\pi \in Path^A_{ful}(s)$ we consider π as a path of $\mathcal{S}$, but we take the probability $Prob\{\ldots\}$ w.r.t. the Markov chain MC^A. Recall that we identify each path $\pi \in Path^A_{ful}$ with the path $x = \pi^{(0)}\pi^{(1)}\pi^{(2)}\ldots$ in MC^A. (The fact that for a path formula φ the set $\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} \varphi\}$ is measurable is an easy verification, see e.g. [58].)

If $s \models_{Adv} \Phi$ then we say s fulfills Φ (or s satisfies Φ or Φ holds in s) w.r.t. Adv. The truth value of formulas involving the (linear time) quantifiers $\Diamond$ and $\Box$ can be derived. For example, $s \models_{Adv} [\ \exists\Diamond\ \Phi\ ]_{\geq p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi(k) \models_{Adv} \Phi$ for some $k \geq 0\} \geq p$ for some $A \in Adv$. By this we obtain: $s \models_{Adv} [\ \forall\Box\ \Phi\ ]_{>p}$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi(k) \models_{Adv} \neg\Phi$ for some $k \geq 0\} < 1 - p$ for all $A \in Adv$ iff $Prob\{\pi \in Path^A_{ful}(s) : \pi(k) \models_{Adv} \Phi$ for all $k \geq 0\} > p$ for all $A \in Adv$.

Given a probabilistic process $\mathcal{P}$, described by a concurrent probabilistic system $\mathcal{S}$ with an initial state s and a labelling function L which assigns to each state s a set $L(s)$ of atomic propositions, we say $\mathcal{P}$ satisfies a *PBTL* formula Φ (w.r.t. Adv) iff $s \models_{Adv} \Phi$ where $M = (\mathcal{S}, L)$. For instance, if *stop* is an atomic proposition which stands for termination and $\mathcal{P}$ satisfies $[\ \forall\Diamond\ stop\ ]_{\geq p}$ then $\mathcal{P}$ terminates with probability at least p; more precisely, for every computation tree of $\mathcal{P}$ that arises from an adversary $A \in Adv$, the probability for a terminating execution is $\geq p$. If $crit_i$, $i = 1, 2$, are atomic propositions stating that subprocesses $\mathcal{P}_i$ of $\mathcal{P}$ are in their critical sections and $\mathcal{P}$ satisfies $[\ \forall\Box\ (\neg crit_1 \vee \neg crit_2)\ ]_{\geq p}$

proposed in [16] for the standard interpretation or the method of [25] for checking whether a path formula holds for all (some) fair paths.

then $\mathcal{P}$ fulfills mutual exclusion with probability p; more formally, for each computation tree which arises from an adversary $A \in Adv$, the probability of an execution in which $\mathcal{P}_1$ and $\mathcal{P}_2$ are never simultanously in their critical sections is at least p.

We focus on three interpretations for $PBTL$ which are as follows. The first interpretation is standard and assumes the whole class of adversaries, i.e. $Adv = \mathcal{A}(\mathcal{S})$. In the second interpretation we allow only the fair adversaries, i.e. $Adv = \mathcal{A}_{fair}(\mathcal{S})$, while in the third the strictly fair adversaries, i.e. $Adv = \mathcal{A}_{sfair}(\mathcal{S})$. We write $\models$, $\models_{fair}$ and $\models_{sfair}$ for the induced satisfaction relations. In Section 5 and 6 we give examples that show the differences between $\models$, $\models_{fair}$ and $\models_{sfair}$.

5 Model checking for $PBTL$

In [14] a model checking algorithm for $PBTL$ w.r.t. $\models$ (called $pCTL$ in [14]) can be found, which is time polynomial in the size of the underlying concurrent probabilistic system. In addition, [14] contains a model checking algorithm for a much richer logic $pCTL^*$ which allows arbitrary combinations of path formulas by the boolean connectives and path operators such as next step and until (see Section 9). In the non-probabilistic case, e.g. when using CTL, fairness of fulpaths can be expressed by path formulas of the full branching time logic, e.g. CTL^*. Typically, this is achieved by means of formulas of the form $\varphi_{fair} = \bigvee_i \bigwedge_j (\Diamond\Box\varphi_{i,j} \vee \Box\Diamond\psi_{i,j})$, and the model checking for CTL under fairness assumptions (e.g. w.r.t. a non-probabilistic version of our satisfaction relation $\models_{sfair}$) can be reduced to the model checking problem for CTL^* since one has an equivalence of the form $s \models_{fair} \forall\varphi$ iff $s \models \forall(\varphi_{fair} \to \varphi)$ (cf. [25]). Unfortunately, this equivalence does *not* hold in the probabilistic case. The problem is that formulas of the form $[\ \forall(\varphi_{fair} \to \varphi)\ ]_{\sqsupseteq p}$ interpreted over $\models$ state that in all adversaries the measure of all fulpaths, whether fair or unfair, that satisfy φ is $\sqsupseteq p$, whereas the interpretation w.r.t. $\models_{fair}$ quantifies over the fair adversaries. Hence, the model checking algorithm of [14] for $PBTL^*$ cannot be used to handle fairness (at least not in a straightforward manner).

In this section we present a model checking algorithm for $PBTL$ for each of the satisfaction relations $\models_{fair}$ and $\models_{sfair}$. The starting point is a $PBTL$-structure $M = (\mathcal{S}, L)$ and a $PBTL$ formula Φ. The algorithm is similar to the model checking algorithms of [16] for CTL and that of [14] for $pCTL$. First, it builds the parse tree of Φ whose nodes stand for subformulas of Φ; more precisely, the leaves are labelled by tt or an atomic proposition, and the internal nodes are labelled by one of the operators $\wedge$, $\neg$, $[\ \exists X\ \_\ ]_{\sqsupseteq p}$, $[\ \forall X\ \_\ ]_{\sqsupseteq p}$, $[\ \_\ \exists\mathcal{U}^*\ \_\ ]_{\sqsupseteq p}$ or $[\ \_\ \forall\mathcal{U}^*\ \_\ ]_{\sqsupseteq p}$ (where $\mathcal{U}^*$ is either $\mathcal{U}$ or $\mathcal{U}^{\leq k}$ for some $k \geq 0$). Nodes labelled by $\neg$ or a next-step operator have exactly one son, representing the argument of the negation, resp. next step operator, in the corresponding subformula. Nodes labelled by $\wedge$ or an until operator have exactly two sons (their arguments). For each node v we calculate the set $Sat(\Psi)$ of states consisting of the states where the corresponding subformula Ψ holds. The cases where the associated formula of a node v is tt, a, $\neg\Phi'$ or $\Phi = \Phi_1 \wedge \Phi_2$ is clear as we have: $Sat(tt) = S$, $Sat(a) = \{s \in S : a \in L(s)\}$, $Sat(\neg\Phi') = S \setminus Sat(\Phi')$ and $Sat(\Phi_1 \wedge \Phi_2) \ = \ Sat(\Phi_1) \cap Sat(\Phi_2)$.

The next-step and the bounded-until operator are dealt with in the same way for all three interpretations. This is due to the fact that each mapping $A : \{\omega \in Path_{fin} : |\omega| \leq k\} \to$

$\bigcup_s Steps(s)$ with $A(\omega) \in Steps(last(\omega))$ can be extended in a fair (or a strictly fair) way. The following lemma shows how to deal with subformulas whose outermost operator is the next-step operator.

Lemma 5.1 *Let $Adv \in \{\mathcal{A}, \mathcal{A}_{fair}, \mathcal{A}_{sfair}\}$, Φ be a PBTL formula and $Sat(\Phi)$ the set of states $t \in S$ with $t \models_{Adv} \Phi$. Then, for all $s \in S$:*

$$s \models_{Adv} [\, \exists X\ \Phi\,]_{\sqsupseteq p} \quad \text{iff} \quad \text{there exists } \mu \in Steps(s) \text{ with } \sum_{t \in Sat(\Phi)} \mu(t) \sqsupseteq p,$$

$$s \models_{Adv} [\, \forall X\ \Phi\,]_{\sqsupseteq p} \quad \text{iff} \quad \sum_{t \in Sat(\Phi)} \mu(t) \sqsupseteq p \text{ for all } \mu \in Steps(s).$$

The lemma below characterises satisfaction relations for the bounded-until operator.

Lemma 5.2 *Let $Adv \in \{\mathcal{A}, \mathcal{A}_{fair}, \mathcal{A}_{sfair}\}$ and Φ_1, Φ_2 be PBTL formulas. Let $p_{s,l}^{max}$ and $p_{s,l}^{min}$, $s \in S$, $l \geq 0$, be given as follows. If $s \models_{Adv} \neg\Phi_1 \wedge \neg\Phi_2$ then $p_{s,l}^{max} = p_{s,l}^{min} = 0$ for all $l \geq 0$. If $s \models_{Adv} \Phi_2$ then $p_{s,l}^{max} = p_{s,l}^{min} = 1$ for all $l \geq 0$. If $s \models_{Adv} \Phi_1$ then $p_{s,0}^{max} = p_{s,0}^{min} = 0$ and*

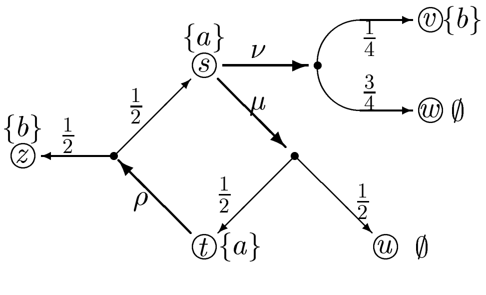
$$\begin{aligned} p_{s,l+1}^{max} &= \max \left\{ \textstyle\sum_{t \in S} \mu(t) \cdot p_{t,l}^{max} : \mu \in Steps(s) \right\}, \\ p_{s,l+1}^{min} &= \min \left\{ \textstyle\sum_{t \in S} \mu(t) \cdot p_{t,l}^{min} : \mu \in Steps(s) \right\}. \end{aligned}$$

Then, for all $s \in S$:

$$\begin{aligned} s &\models_{Adv} [\, \Phi_1\ \exists\mathcal{U}^{\leq k}\ \Phi_2\,]_{\sqsupseteq p} \quad \text{iff} \quad p_{s,k}^{max} \sqsupseteq p, \\ s &\models_{Adv} [\, \Phi_1\ \forall\mathcal{U}^{\leq k}\ \Phi_2\,]_{\sqsupseteq p} \quad \text{iff} \quad p_{s,k}^{min} \sqsupseteq p. \end{aligned}$$

Proof: For $A \in Adv$ let $p_{s,l}^A = Prob\left\{\pi \in Path_{ful}^A(s) : \pi \models_{Adv} \Phi_1 \mathcal{U}^{\leq l} \Phi_2\right\}$. By induction on l it can be shown that $p_{s,l}^{max} = \max\{p_{s,l}^A : A \in Adv\}$, $p_{s,l}^{min} = \min\{p_{s,l}^A : A \in Adv\}$ which yields the claim. ■

Example 5.3 Let M be the following $PBTL$-structure and $\Phi = [a \exists\mathcal{U}^{\leq 3} b]_{>1/4}$.



Using the notation of Lemma 5.2 we have $p_{v,3}^{max} = p_{z,3}^{max} = 1$, $p_{w,3}^{max} = p_{u,3}^{max} = 0$ and the recursive formulas $p_{s,i+1}^{max} = \max\{p_{t,i}^{max}/2, 1/4\}$, $p_{t,i+1}^{max} = 1/2 + p_{s,i}^{max}/2$ where $p_{s,0}^{max} = p_{t,0}^{max} = 0$. We obtain $p_{s,1}^{max} = 1/4$, $p_{t,1}^{max} = 1/2$, $p_{s,2}^{max} = 1/4$, $p_{t,2}^{max} = 5/8$, $p_{s,3}^{max} = 5/16$ and $p_{t,3}^{max} = 21/32$. Hence $s, t, v, z \models \Phi$, $w, u \not\models \Phi$. ■

The remainder of this section is concerned with the unbounded until operator. We develop a series of technical results (Theorems 1-7) which essentially allow us to obtain in our view a surprising result: to deal with the *unbounded* until operator an examination of the simple adversaries suffices. For readers' convenience we state the main theorems in this section without proof (those are included in Section 12). Instead, we include justification for the technical results in the form of examples.

First, the following example shows that the above-mentioned reduction to simple adversaries fails for the unbounded-until operator.

Example 5.4 Let M be as in Example 5.3. We saw that $s \models_{Adv} [a \exists \mathcal{U}^{\leq 3} b]_{>1/4}$. On the other hand, $Prob\{\pi \in Path_{ful}^{A}(s) : \pi \models a\mathcal{U}^{\leq 3} b\} = 1/4$ for each simple adversary A. (Note that there are exactly two simple adversaries A_μ and A_ν. These are given by $A_\mu(s) = \mu$ and $A_\nu(s) = \nu$ respectively. In both adversaries, there is exactly one fulpath π starting in s and fulfilling $a\mathcal{U}^{\leq 3}b$, namely the fulpath $s \xrightarrow{\mu} t \xrightarrow{\rho} z \xrightarrow{\mu_z^1} \ldots$ in A_μ and the fulpath $s \xrightarrow{\nu} v \xrightarrow{\mu_v^1} \ldots$ in A_ν.) ■

For the rest of this section we fix a *PBTL*-model $M = (\mathcal{S}, L)$ where $\mathcal{S} = (S, Steps)$ and two *PBTL* formulas Φ_1, Φ_2. We suppose that the sets of states $s \in S$ with $s \models_{Adv} \Phi_i$ are already computed, where Adv is either $\mathcal{A}$, $\mathcal{A}_{fair}$ or $\mathcal{A}_{sfair}$. We may suppose that Φ_1, Φ_2 are atomic propositions with $\Phi_i \in L(s_i)$ if and only if $s \models_{Adv} \Phi_i$, $i = 1, 2$. This simplifying assumption allows us to use the same notation for all three interpretations (since $s \models_{Adv} \Phi_i$ iff $s \models \Phi_i$), and is made for this reason alone.

Notation 5.5 *Let $Sat(\Phi_i) = \{s \in S : s \models \Phi_i\}$, $i = 1, 2$. For $A \in \mathcal{A}$, $\omega \in Path_{fin}$, $s \in S$,*

$$\begin{aligned}
p_\omega^A(\Phi_1\mathcal{U}\Phi_2) &= Prob\left\{x \in Path_{ful}(\omega, MC^A) \ : \ x \models \ \Phi_1\mathcal{U}\Phi_2 \right\}, \\
p_s^{max}(\Phi_1\mathcal{U}\Phi_2) &= \max \left\{ \ p_s^A(\Phi_1\mathcal{U}\Phi_2) \ : \ A \in \mathcal{A}_{simple} \right\}, \\
p_s^{min}(\Phi_1\mathcal{U}\Phi_2) &= \min \left\{ \ p_s^A(\Phi_1\mathcal{U}\Phi_2) \ : \ A \in \mathcal{A}_{simple} \right\}.
\end{aligned}$$

Here, for an adversary A and a fulpath x in MC^A, $x \models \Phi_1\mathcal{U}\Phi_2$ iff $\pi \models \Phi_1\mathcal{U}\Phi_2$ where π is the unique fulpath in $\mathcal{S}$ with $x = \pi^{(0)}\pi^{(1)}\ldots$. Note that

$$p_s^A(\Phi_1\mathcal{U}\Phi_2) \ = \ Prob\left\{\pi \in Path_{ful}^A(s) : \pi \models \Phi_1\mathcal{U}\Phi_2\right\}$$

and that $p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ and $p_s^{min}(\Phi_1\mathcal{U}\Phi_2)$ are well-defined since $\mathcal{A}_{simple}$ is finite.

Theorem 1 below can be derived from Corollary 20 (part 1) of [14], which uses the results of [20]. The reason we state it here is that part (a) of this theorem carries over to the satisfaction relation $\models_{fair}$ (Theorem 2), while part (b) does not (cf. Example 5.14).

Theorem 1 (cf. [20, 14]) *For all $s \in S$:*

(a) $s \models [\ \Phi_1 \ \exists\mathcal{U} \ \Phi_2 \ ]_{\sqsupseteq p}$ *iff* $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) \ \sqsupseteq \ p$.
(b) $s \models [\ \Phi_1 \ \forall\mathcal{U} \ \Phi_2 \ ]_{\sqsupseteq p}$ *iff* $p_s^{min}(\Phi_1\mathcal{U}\Phi_2) \ \sqsupseteq \ p$.

Theorem 1 ensures that $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) \geq p_s^F(\Phi_1\mathcal{U}\Phi_2)$ for all fair adversaries F. Vice versa, for each (simple) adversary A there is a fair adversary F with $\{\omega \in Path_{fin}^A : \omega \models \Phi_1\mathcal{U}\Phi_2\} \subseteq Path_{fin}^F$. (Here, $\omega \models \Phi_1\mathcal{U}\Phi_2$ iff $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \models \Phi_2$.) Thus, $p_s^F(\Phi_1\mathcal{U}\Phi_2) \geq p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ for all fair adversaries (see Section 12.4). Hence:

Theorem 2 *For all $s \in S$: $s \models_{fair} [\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ iff $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) \sqsupseteq p$.*

We defer the proof to Section 12.4. It turns out that the satisfaction relation $\models_{sfair}$ differs from $\models_{fair}$ and $\models$ in that only a stronger statement (whose proof is also given in Section 12.4) for formulas of the form $[\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{>p}$ can be shown:

Theorem 3 *For all $s \in S$: $s \models_{sfair} [\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{>p}$ iff $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) > p$.*

Example 5.6 shows that the inequality "$> p$" in Theorem 3 cannot be replaced by "$\geq p$" as $p_s^F(\Phi_1\mathcal{U}\Phi_2) < p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ for all $F \in \mathcal{A}_{sfair}$ is possible.

Example 5.6 Consider the *PBTL*-structure M of Example 1 where $L(s) = L(t) = \{a\}$, $L(v) = \emptyset$ and $L(u) = \{b\}$ and the path formula $a\mathcal{U}b$. Then, $p_s^F(a\mathcal{U}b) < 1$ for each strictly fair adversary F (and hence, $s \not\models_{sfair} [\ a\ \exists\mathcal{U}\ b\ ]_{\geq 1}$). On the other hand, $p_s^A(a\mathcal{U}b) = 1$ for the simple adversary A with $A(s) = \mu$. Hence, $p_s^F(a\mathcal{U}b) < 1$ but $p_s^{max}(a\mathcal{U}b) = 1$. (Note that A is fair, cf. Example 3.8). ■

In order to describe how the set of states fulfilling formulas of the form $[\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{\geq p}$ w.r.t. $\models_{sfair}$ can be computed we first introduce some notation. We define $Reach_{\Phi_1\wedge\neg\Phi_2}(s)$ to be the set of states which are reachable in $\mathcal{S}$ from s via a path where all states – possibly except the last one – fulfill the formula $\Phi_1 \wedge \neg\Phi_2$. $S^+(\Phi_1, \Phi_2)$ is the set of all states from which one can reach a Φ_2-state via a path through Φ_1-states. Formally:

Notation 5.7 *$Reach_{\Phi_1\wedge\neg\Phi_2}(s)$ is the set of states $t \in S$ such that there is some finite path ω in $\mathcal{S}$ with $first(\omega) = s$, $last(\omega) = t$ and $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i < |\omega|$. For a simple adversary A we define $Reach^A_{\Phi_1\wedge\neg\Phi_2}(s)$ to be set of states t such that there is some finite path $\omega \in Path^A_{fin}$ with $first(\omega) = s$, $last(\omega) = t$ and $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i < |\omega|$. Let $S^+(\Phi_1, \Phi_2) = \{s \in S : Reach_{\Phi_1\wedge\neg\Phi_2}(s) \cap Sat(\Phi_2) \neq \emptyset\}$.*

Then, $s \in S^+(\Phi_1, \Phi_2)$ iff $p_s^A(\Phi_1\mathcal{U}\Phi_2) > 0$ for some $A \in \mathcal{A}$ iff $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) > 0$.

Example 5.8 For the system of Example 5.6 (Figure 1 with $L(s) = L(t) = \{a\}$, $L(u) = \{b\}$ and $L(v) = \emptyset$) we have $Reach_{a\wedge\neg b}(s) = Reach_{a\wedge\neg b}(t) = \{s, t, u, v\}$ and $Reach_{a\wedge\neg b}(x) = \{x\}$ for $x \in \{u, v\}$. For the simple adversaries A, B with $A(s) = \mu$ and $B(s) = \mu_v^1$ we have:

$$Reach^A_{a\wedge\neg b}(s) = Reach^A_{a\wedge\neg b}(t) = \{s, t, u\}, \quad Reach^A_{a\wedge\neg b}(x) = \{x\}, \quad x \in \{u, v\},$$
$$Reach^B_{a\wedge\neg b}(s) = \{s, v\}, \quad Reach^B_{a\wedge\neg b}(t) = \{s, t, v\}, Reach^B_{a\wedge\neg b}(x) = \{x\}, \quad x \in \{u, v\}.$$

Moreover, $Sat(b) = \{u\}$, $Sat(a) = \{s, t\}$. Thus, $S^+(a, b) = \{s, t, u\}$. ■

We define a set $T^{max}(\Phi_1, \Phi_2)$ for which we show that it contains exactly those states s such that $p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ can be "reached" by a strictly fair adversary (i.e. $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) = p_s^F(\Phi_1\mathcal{U}\Phi_2)$ for some strictly fair adversary F).

Notation 5.9 *If $s \in S \setminus Sat(\Phi_1)$ then we define $MaxSteps(s) = Steps(s)$. For $s \in Sat(\Phi_1)$ let $MaxSteps(s, \Phi_1, \Phi_2)$ be the set of $\mu \in Steps(s)$ such that*

$$p_s^{max}(\Phi_1\mathcal{U}\Phi_2) = \sum_{t\in S} \mu(t) \cdot p_t^{max}(\Phi_1\mathcal{U}\Phi_2).$$

Let $T^{max}(\Phi_1,\Phi_2) = \bigcup_{i\geq 0} T_i^{max}(\Phi_1,\Phi_2)$ *where* $T_0^{max}(\Phi_1,\Phi_2) = Sat(\Phi_2) \cup (S \setminus S^+(\Phi_1,\Phi_2))$ *and* $T_j^{max}(\Phi_1,\Phi_2) = T_{j,1}^{max}(\Phi_1,\Phi_2) \cup T_{j,2}^{max}(\Phi_1,\Phi_2)$ *for* $j \geq 1$. *Here,*

- $T_{j,1}^{max}(\Phi_1,\Phi_2)$ *is the set of states* $t \in S \setminus \bigcup_{i<j} T_i^{max}(\Phi_1,\Phi_2)$ *such that* $Supp(\mu) \subseteq \bigcup_{i<j} T_i^{max}(\Phi_1,\Phi_2)$ *for some* $\mu \in MaxSteps(t,\Phi_1,\Phi_2)$,
- $T_{j,2}^{max}(\Phi_1,\Phi_2)$ *is the set of all states* $t \in S$ *which are contained in some subset* T *of* $S \setminus \left(\bigcup_{i<j} T_i^{max}(\Phi_1,\Phi_2) \cup T_{j,1}^{adm}(\Phi_1,\Phi_2)\right)$ *such that for all* $u \in T$*:*
  - *(i)* $MaxSteps(u,\Phi_1,\Phi_2) = Steps(u)$
  - *(ii) for all* $\mu \in Steps(u)$*:*

$$Supp(\mu) \subseteq T \cup \bigcup_{i<j} T_i^{max}(\Phi_1,\Phi_2) \cup T_{j,1}^{max}(\Phi_1,\Phi_2).$$

Intuitively, $MaxSteps(s,\Phi_1,\Phi_2)$ is the set of steps $\mu \in Steps(s)$ that are "optimal" in the sense that the maximal probability $p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ is obtained when in state s the distribution μ is chosen and when all further non-deterministic choices are made "optimal". Clearly, $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) = p^F(\Phi_1\mathcal{U}\Phi_2)$ for all states $s \in T_0^{max}(\Phi_1,\Phi_2)$ and strictly fair adversaries F. $T_{j,1}^{max}(\Phi_1,\Phi_2)$ denotes the set of states t for which there is such an "optimal" step μ_t such that, for all possible successor states (i.e. all states $u \in Supp(\mu_t)$), the maximal probability $p_u^{max}(\Phi_1\mathcal{U}\Phi_2)$ is obtained by a strictly fair adversary under which all fulpaths starting in u only pass those states that belong to $T_i^{max}(\Phi_1,\Phi_2)$ for some $i<j$. $T_{j,2}^{max}(\Phi_1,\Phi_2)$ is the set of states t where all possible steps are "optimal" in the above sense and where all possible successor states u either belong to $T_{j,2}^{max}(\Phi_1,\Phi_2)$ or are states for which the maximal probability $p_u^{max}(\Phi_1\mathcal{U}\Phi_2)$ is obtained by a strictly fair adversary F where all fulpaths $\pi \in Path_{ful}^F(u)$ only visit states from $T_{j,1}^{max}(\Phi_1,\Phi_2)$ or $\bigcup_{i<j} T_i^{max}(\Phi_1,\Phi_2)$.

Example 5.10 We consider the *PBTL*-structure of Figure 3. We write T_*^{max} rather than $T_*^{max}(a,b)$. Then, $p_{s_5}^{max} = 1/2$, $p_{s_4}^{max} = p_{s_3}^{max} = p_{s_2}^{max} = 1/3$,

$$p_{s_6}^{max} = \frac{1}{9} \cdot \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i = \frac{2}{9}$$

and $p_{s_1}^{max} = \max\{2/3 \cdot 1/3, 2/9\} = 2/9$. Hence, $Steps(s_j) = MaxSteps(s_j)$, $j = 1,\ldots,5$, $\nu_2 = \mu_{u_6}^1 \notin MaxSteps(s_6)$. Thus, $T_0^{max} = S \setminus S^+(a,b) \cup Sat(b) = \{u_1,u_3,u_5,u_6,u_6',t_5,t_6\}$, $T_{1,1}^{max} = \{s_5\}$, $T_{1,2}^{max} = \{s_3,s_4\}$, $T_{2,1}^{max} = \{s_2\}$, $T_{2,2}^{max} = \emptyset$, $T_{3,1}^{max} = \{s_1\}$ and $T_*^{max} = \emptyset$ in all other cases. We get $T^{max}(a,b) = S \setminus \{s_6\}$. ■

For all $s \in S \setminus T^{max}(\Phi_1,\Phi_2)$ and strictly fair adversaries F, there is a finite path $\omega \in Path_{fin}^F(s)$ with $F(\omega) \notin MaxSteps(last(\omega),\Phi_1,\Phi_2)$ and $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0,1,\ldots,|\omega|$. Then, $p_\omega^F(\Phi_1\mathcal{U}\Phi_2) < p_{last(\omega)}^{max}(\Phi_1\mathcal{U}\Phi_2)$ which yields $p_s^F(\Phi_1\mathcal{U}\Phi_2) < p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$. (See Section 12.6, Lemma 12.33.) For instance, for the state s_6 of the system in Example 5.10 and each $F \in \mathcal{A}_{sfair}$, there is some finite path ω in $Path_{fin}^F$ of the form $s_6 \stackrel{\nu_1}{\to} s_6 \stackrel{\nu_1}{\to} \ldots \stackrel{\nu_1}{\to} s_6$ with $F(\omega) = \nu_2 \notin MaxSteps(s_6,a,b)$ which yields $p_{s_6}^F(a\mathcal{U}b) < 2/9 = p_{s_6}^{max}(a\mathcal{U}b)$.

Vice versa, a strictly fair adversary F with $F(\omega) = \mu_t$ for all $\omega \in Path_{fin}^F$ with $last(\omega) = t \in T_{j,1}^{max}(\Phi_1,\Phi_2)$ (where $\mu_t \in MaxSteps(t)$ and $Supp(\mu_t) \subseteq T_i^{max}(\Phi_1,\Phi_2)$ for some $i<j$) can be defined. For this adversary F, $p_t^F(\Phi_1\mathcal{U}\Phi_2) = p_t^{max}(\Phi_1\mathcal{U}\Phi_2)$ for all $t \in T^{max}(\Phi_1,\Phi_2)$. (See Section 12.6, Lemma 12.32). For instance, for the system of Example 5.10 and each strictly fair adversary with

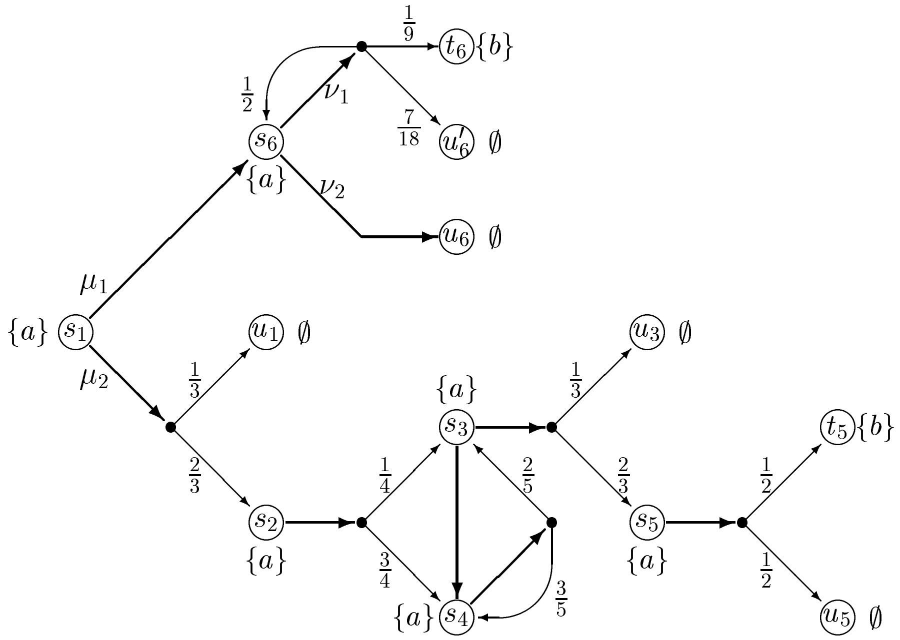


Figure 3:

$$F(\omega) = \mu_2 \text{ if } \omega \in Path^F_{fin},\ last(\omega) = s_1 \text{ and } \omega(i) \neq s_1,\ i = 0, 1, \ldots |\omega| - 1,$$

we have $p^F_t(a\mathcal{U}b) = p^{max}_t(a\mathcal{U}b)$ for all $t \in T^{max}(a, b)$. (Note that there is no fair fulpath where s_3 occurs infinitely often. Thus, $p^F_{s_3}(a\mathcal{U}b) = 1/3 = p^{max}_{s_3}(a\mathcal{U}b)$ for all $F \in \mathcal{A}_{sfair}$.) We obtain:

Theorem 4 *For all $s \in S$ and $p > 0$:*

$$s \models_{sfair} [\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{\geq p} \quad \Longleftrightarrow \quad \begin{cases} p^{max}_s(\Phi_1\mathcal{U}\Phi_2) \geq p & : \text{ if } s \in T^{max}(\Phi_1, \Phi_2) \\ p^{max}_s(\Phi_1\mathcal{U}\Phi_2) > p & : \text{ otherwise.} \end{cases}$$

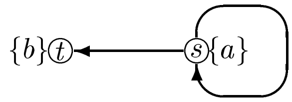
Example 5.11 For the system of Example 5.10 we have: $s_6 \not\models_{sfair} [\ a\ \exists\mathcal{U}\ b\ ]_{\geq 2/9}$ (but $s_6 \models_{fair} [\ a\ \exists\mathcal{U}\ b\ ]_{\geq 2/9}$) and $s_1 \models_{sfair} [\ a\ \exists\mathcal{U}\ b\ ]_{\geq 2/9}$. ■

Corollary 5.12 *If $s \notin T^{max}(\Phi_1, \Phi_2)$ then $s \not\models_{sfair} [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\geq 1}$.*

Example 5.13 For the system in Example 5.6 (Figure 1 with $L(s) = L(t) = \{a\}$, $L(u) = \{b\}$ and $L(v) = \emptyset$) we have $S^+(a, b) = \{s, t, u\}$ and $Sat(b) = \{u\}$ (cf. Example 5.8). Hence, $T^{max}_0(a, b) = \{u, v\}$. For the simple adversary A with $A(s) = \mu$ we get $p^A_s(a\mathcal{U}b) = p^A_t(a\mathcal{U}b)$ and $p^A_s(a\mathcal{U}b) = 1/2 + 1/2 \cdot p^A_t(a\mathcal{U}b)$. Hence, $p^A_s(a\mathcal{U}b) = 1$. Thus, $p^{max}_s(a\mathcal{U}b) = 1$. Since $\sum_x \mu^1_v(x) \cdot p^{max}_x(a\mathcal{U}b) = p^{max}_v(a, b) = 0$ we get $MaxSteps(s) = \{\mu\} \neq Steps(s)$. This yields $T^{max}_{1,1}(a, b) = T^{max}_{1,2}(a, b) = \emptyset$ and $T^{max}(a, b) = T^{max}_0(a, b) = \{u, v\}$. By Corollary 5.12 we obtain $s \not\models_{sfair} [a\ \exists\mathcal{U}\ b\ ]_{\geq 1}$ as $s \notin T^{max}(a, b)$. ■

Next we deal with formulas $[\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ and the satisfaction relations $\models_{fair}$ and $\models_{sfair}$. The following example shows that $p^{min}_s(\Phi_1\mathcal{U}\Phi_2) < \inf\{p^F_s(\Phi_1\mathcal{U}\Phi_2) : F \in \mathcal{A}_{fair}\}$ (thus, $s \models_{fair} [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ while $p^{min}_s(\Phi_1\mathcal{U}\Phi_2) < p$) is possible. In particular, this example shows the difference between $\models_{fair}$ and $\models$, and that in Theorem 1(b) the satisfaction relation $\models$ cannot be replaced by $\models_{fair}$ or $\models_{sfair}$).

Example 5.14 Consider the following *PBTL*-structure M and the path formula $a\,\mathcal{U}\,b$.



Then, $p_s^A(a\mathcal{U}b) = 0$ for the simple adversary A with $A(s) = \mu_s^1$, whereas $p_s^F(a\mathcal{U}b) = 1$ for each fair adversary F. Hence, $s \models_{fair} [\ a\ \forall\mathcal{U}\ b\ ]_{\geq 1}$ but $p_s^{min}(a\mathcal{U}b) = 0$. ■

In order to reduce the question of whether or not a formula of the form $[\Phi_1 \forall\mathcal{U}\Phi_2]_{\sqsupseteq p}$ is fulfilled w.r.t. $\models_{fair}$ to an investigation of certain simple adversaries we must find a subclass of simple adversaries A where the probabilities $p_s^A(\Phi_1\mathcal{U}\Phi_2)$ can be "approximated" by fair adversaries. In Example 5.14 we saw that the liveness property $[\ a\ \forall\mathcal{U}\ b\ ]_{\geq 1}$ cannot be established unless fairness is required. The problem with the simple adversary A is that it forces the system to stay forever in a "non-successful" state (s) from which a "successful" state (t) can be reached. In fair adversaries, with probability 1, all states that are reachable from a state that is visited infinitely often are also visited infinitely often (cf. Lemma 12.5(b) and Lemma 12.6). This explains why $p_s^A(a\mathcal{U}b)$ cannot be "approximated" by fair adversaries. Thus, in order to handle formulas of the form $[\Phi_1 \forall\mathcal{U}\Phi_2]_{\sqsupseteq p}$ we restrict our attention to those simple adversaries A in which almost all fulpaths π contain a state s that either fulfills Φ_2 (i.e. $s \in Sat(\Phi_2)$) or where $\Phi_1\mathcal{U}\Phi_2$ cannot be satisfied with a non-zero probability (i.e. $s \notin S^+(\Phi_1, \Phi_2)$).

Definition 5.15 *A simple adversary A is called* admissible *for (Φ_1, Φ_2) iff*

$$Reach^A_{\Phi_1 \wedge \neg\Phi_2}(s) \ \cap\ \Big(\ Sat(\Phi_2)\ \cup\ (S \setminus S^+(\Phi_1, \Phi_2))\Big)\ \neq\ \emptyset.$$

for all $s \in S^+(\Phi_1, \Phi_2)$. Let $\mathcal{A}_{adm}(\Phi_1, \Phi_2)$ be the set of simple adversaries that are admissible for (Φ_1, Φ_2).

Note that the definition of an admissible adversary depends on the underlying formula $\Phi_1\mathcal{U}\Phi_2$. For instance, for the system of Example 5.14, the simple adversary A with $A(s) = \mu_s^1$ is admissible for (b, a) (as $s \in Sat(a)$ and $t \notin S^+(b, a)$) but not for (a, b) (as $s \in S^+(a, b)$ while $t \notin Reach_{a\wedge\neg b}(s) = \{s\}$). If A is a simple adversary with $p_s^A(\Phi_1\mathcal{U}\Phi_2) = p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ for all $s \in S$ then $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$. (Since $Reach^A_{\Phi_1\wedge\neg\Phi_2}(s) \cap Sat(\Phi_2) \neq \emptyset$ for all $s \in S^+(\Phi_1, \Phi_2)$.) Hence, by Lemma 12.1 which ensures the existence of such a simple adversary, $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) \;=\; \max\{p_s^A(\Phi_1\mathcal{U}\Phi_2) : A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)\}$.

Notation 5.16 *For $s \in S$, let $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2) \;=\; \min\ \{p_s^A(\Phi_1\mathcal{U}\Phi_2) : A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)\}$.*

The following theorem states that to handle formulas of the type $\Phi_1\ \forall\mathcal{U}\ \Phi_2$ w.r.t. $\models_{fair}$ it suffices only to consider the simple adversaries that are admissible for (Φ_1, Φ_2); proof can be found in Section 12.5.

Theorem 5 *For all $s \in S$: $s \ \models_{fair}\ [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ iff $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2) \sqsupseteq p$.*

If $Reach_{\Phi_1\wedge\neg\Phi_2}(s) \subseteq S^+(\Phi_1, \Phi_2)$ and $s \in Sat(\Phi_1)$ then $p_s^F(\Phi_1\mathcal{U}\Phi_2) = 1$ for all $F \in \mathcal{A}_{fair}$ (see Remark 12.26). Hence, $s \ \models_{fair}\ [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\geq 1}$. Vice versa, if $Reach_{\Phi_1\wedge\neg\Phi_2}(s) \not\subseteq S^+(\Phi_1, \Phi_2)$ then there is a finite path $\omega \in Path_{fin}(s)$ with $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \notin S^+(\Phi_1, \Phi_2)$. Hence, for a fair adversary F with $\omega \in Path^F_{fin}(s)$, we have $p_s^F(\Phi_1\mathcal{U}\Phi_2) \leq 1 - \mathbf{P}(\omega) < 1$. Thus:

Corollary 5.17 *For all $s \in S$: $s \models_{fair} [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\geq 1}$ iff $Reach_{\Phi_1 \wedge \neg\Phi_2}(s) \subseteq S^+(\Phi_1, \Phi_2)$.*

In particular, a concurrent probabilistic system $\mathcal{S}$ with initial state s_{init} satisfies "qualitative progress properties" expressed by *PBTL* formulas of the form $[\ \forall\Diamond\Phi\ ]_{\geq 1}$ (satisfaction is understood to be w.r.t. $\models_{fair}$) if and only if the system is "safe" in the sense that no "deadlocked" state (a state t from which no Φ-state can be reached, i.e. $Reach(t) \cap Sat(\Phi) = \emptyset$) is reachable from the initial state s_{init}:

$$s_{init} \models_{fair} [\ \forall\Diamond\Phi\ ]_{\geq 1} \text{ iff } Reach(s_{init}) \subseteq \{s \in S : Reach(s) \cap Sat(\Phi) \neq \emptyset\}.$$

Thus, for verifying "qualitative progress properties" as explained above an analysis of the "topology" of the system suffices. This result was first established in [33].

Example 5.18 For the *PBTL*-structure of Example 5.14 we have $Reach(s) = \{s, t\}$ and $Sat(b) = \{t\}$. Hence, $s \models_{fair} [\ \forall\Diamond b\ ]_{\geq 1}$. ■

For the case of the $\models_{sfair}$ satisfaction relation we obtain:

Theorem 6 *For all $s \in S$: $s \models_{sfair} [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\geq p}$ iff $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2) \geq p$.*

A stronger version of Theorem 6 stating that $s \models_{sfair} [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{>p}$ iff $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2) > p$ is incorrect, as can be seen from Example 5.19 below. (This example again demonstrates the difference between $\models_{sfair}$ and $\models_{fair}$.)

Example 5.19 We consider the following *PBTL*-structure of Figure 1 with the interpretation $L(s) = L(t) = \{a\}$, $L(v) = \{b\}$ and $L(u) = \emptyset$. The simple adversary A with $A(s) = \mu$ is admissible for (a, b) since $Reach^A_{a\wedge\neg b}(s) = Reach^A_{a\wedge\neg b}(t) = \{s, t, u\}$ and $u \in S \setminus S^+(a, b)$. We have $p_s^A(a\mathcal{U}b) = 0$ but $p_s^F(a\mathcal{U}b) > 0$ for all $F \in \mathcal{A}_{sfair}$. Hence, $s \models_{sfair} [\ a\ \forall\mathcal{U}\ b\ ]_{>0}$ while $p_s^{adm}(a\mathcal{U}b) = 0$. ■

The next result is an analogue of Theorem 4, in which we show how to deal with formulas $[\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{>p}$ with respect to the satisfaction relation $\models_{sfair}$. (See Section 12.7 for the proof). Similarly to the definition of $T^{max}(\Phi_1, \Phi_2)$ we define a set $T^{adm}(\Phi_1, \Phi_2)$:

Notation 5.20 *If $s \in S \setminus Sat(\Phi_1)$ then we define $AdmSteps(s) = Steps(s)$. For $s \in Sat(\Phi_1)$ let $AdmSteps(s)$ be the set of $\mu \in Steps(s)$ such that*

$$p_s^{adm}(\Phi_1, \Phi_2) = \sum_{t \in S} \mu(t) \cdot p_t^{adm}(\Phi_1\mathcal{U}\Phi_2).$$

Let $T^{adm}(\Phi_1, \Phi_2) = \bigcup_{i \geq 0} T_i^{adm}(\Phi_1, \Phi_2)$ where $T_0^{adm}(\Phi_1, \Phi_2) = Sat(\Phi_2) \cup (S \setminus S^+(\Phi_1, \Phi_2))$ and $T_j^{adm}(\Phi_1, \Phi_2) = T_{j,1}^{adm}(\Phi_1, \Phi_2) \cup T_{j,2}^{adm}(\Phi_1, \Phi_2)$ for $j \geq 1$. Here,

- *$T_{j,1}^{adm}(\Phi_1, \Phi_2)$ is the set of states $t \in S \setminus \bigcup_{i<j} T_i^{adm}(\Phi_1, \Phi_2)$ such that $Supp(\mu) \subseteq \bigcup_{i<j} T_i^{adm}(\Phi_1, \Phi_2)$ for some $\mu \in AdmSteps(t, \Phi_1, \Phi_2)$,*
- *$T_{j,2}^{adm}(\Phi_1, \Phi_2)$ is the set of all states $t \in S$ which are contained in some subset T of $S \setminus \left(\bigcup_{i<j} T_i^{adm}(\Phi_1, \Phi_2) \cup T_{j,1}^{adm}(\Phi_1, \Phi_2)\right)$ such that for all $u \in T$:*
  - *(i) $AdmSteps(u, \Phi_1, \Phi_2) = Steps(u)$*

(ii) for all $\mu \in Steps(u, \Phi_1, \Phi_2)$*:*

$$Supp(\mu) \;\subseteq\; T \;\cup\; \bigcup_{i<j} T_i^{adm}(\Phi_1, \Phi_2) \;\cup\; T_{j,1}^{adm}(\Phi_1, \Phi_2).$$

Example 5.21 For the *PBTL*-structure of Example 5.10 (Figure 3) we get (with $T_*^{adm} = T_*^{adm}(a, b)$): $p_{s_5}^{adm} = 1/2$, $p_{s_2}^{adm} = p_{s_3}^{adm} = p_{s_4}^{adm} = 1/3$, $p_{s_6}^{adm} = 0$ and $p_{s_1}^{adm} = 0$. Thus, $\mu_2 \notin AdmSteps(s_1)$, $\nu_1 \notin AdmSteps(s_6)$ and $T_0^{adm} = S \setminus \{s_1, \ldots, s_6\}$, $T_{1,1}^{adm} = \{s_5, s_6\}$, $T_{1,2}^{adm} = \{s_3, s_4\}$, $T_{2,1}^{adm} = \{s_2\}$, $T_{2,2}^{adm} = \emptyset$ and $T_{3,1}^{adm} = \{s_1\}$. Hence, $T^{adm}(a, b) = S$. ■

In Section 12.7 we show that $T^{adm}(\Phi_1, \Phi_2)$ is the set of states $t \in S$ where $p_t^{adm}(\Phi_1 \mathcal{U} \Phi_2) = p_t^F(\Phi_1, \Phi_2)$ for some $F \in \mathcal{A}_{sfair}$. Finally, we obtain a result similar to Theorem 4 characterising the strictly fair satisfaction for formulas of the type $\Phi_1 \; \forall\mathcal{U} \; \Phi_2$:

Theorem 7 *For all* $s \in S$*:*

$$s \models_{sfair} [\, \Phi_1 \; \forall\mathcal{U} \; \Phi_2 \,]_{>p} \quad \Longleftrightarrow \quad \begin{cases} p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2) > p & : \;\; \textit{if } s \in T^{adm}(\Phi_1, \Phi_2) \\ p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2) \geq p & : \;\; \textit{otherwise.} \end{cases}$$

Corollary 5.22 *If* $s \notin T^{adm}(\Phi_1, \Phi_2)$ *then* $s \models_{sfair} [\, \Phi_1 \forall\mathcal{U} \; \Phi_2 \,]_{>0}$.

Example 5.23 In Example 5.19 we have $T^{adm}(a, b) = \{u, v\}$. Hence, $s \models_{sfair} [a \,\forall\mathcal{U}\, b\,]_{>0}$ since $s \notin T^{adm}(a, b)$. ■

6 Example

In this section we consider an application of the logic *PBTL* and our model checking algorithm to a simple distributed protocol. Our intention is to illustrate the necessity of fairness assumptions in the case of concurrent probabilistic systems. Other known protocols, such as the alternating bit protocol, can also be specified and verified.

Let $\mathcal{P}$ be a concurrent process which sends messages to another process $\mathcal{Q}$ along an uncertain medium $\mathcal{M}$, which possibly loses or destroys messages. In cases where the messages are lost, $\mathcal{M}$ sends a signal to $\mathcal{P}$ and $\mathcal{P}$ tries again to send the message. We suppose that neither $\mathcal{P}$ nor $\mathcal{M}$ nor $\mathcal{Q}$ are able to detect whether or not the message is destroyed in cases when a message is actually delivered to $\mathcal{Q}$. Apart from the sending of messages, $\mathcal{P}$ can also perform some internal actions which we do not specify. We describe the system from the point of view of an "observer" who does not have access to the local variables of $\mathcal{P}$. In all variants of the system which we consider we have the following states:

s_{init}	the initial state in which $\mathcal{P}$ can send messages or perform internal actions
s_{send}	the state in which the system is ready to send a message
s_{lost}	the state which the systems reaches when the message sent is lost
s_{ok}	the state which the systems reaches when the correct message is delivered
s_{error}	the state which the systems reaches when the message is destroyed

We assume that $\mathcal{M}$ delivers the correct message with probability $1 - \varepsilon$ (for some small ε) and that it loses (resp. destroys) the message with probability ε_l (resp. $\varepsilon_d = \varepsilon - \varepsilon_l$). We consider three variants of the system:

1. In the first variant (Figure 4), we suppose that the decision whether $\mathcal{P}$ attempts to send a message or performs an internal computations is made non-deterministically by a scheduler. The execution of internal actions always succeds and leads back to the initial state.
2. The second variant (Figure 5) extends the first one, where we assume that with probability δ the execution of an internal action of $\mathcal{P}$ causes deadlock and leads to the state $s_{deadlock}$ in which the system stays forever.
3. In the third variant (Figure 6) we suppose that, in the initial state, $\mathcal{P}$ decides non--deterministically either to abandon the delivery mode or to send a message. In the former case, the system reaches a state s_{exit} where only internal actions of $\mathcal{P}$ can be performed. The execution of an internal action in s_{exit} leads back to s_{exit}.

We consider the atomic propositions *send*, *lost* and *ok* and the labelling function L with $a \in L(s)$ iff $s = s_a$. Let μ_{send} be the unique distribution with $\mu_{send}(s_{send}) = 1$ and let A_{send} be the unique simple adversary with $A_{send}(s_{init}) = \mu_{send}$. $A^v_{\neg send}$ denotes the simple adversary with $A^v_{\neg send}(s_{init}) \neq \mu_{send}$ in the v-th variant ($v = 1, 2, 3$). $\models^v$, $\models^v_{fair}$, $\models^v_{sfair}$ denote the satisfaction relations in the v-th variant. The formula

$$\Phi_{\sqsupseteq p} \;=\; [\,\forall\Box\,(send \longrightarrow \Psi_{\sqsupseteq p})\,]_{\geq 1} \quad \text{where} \quad \Psi_{\sqsupseteq p} \;=\; [\,(send \vee lost)\;\forall\mathcal{U}\;ok\,]_{\sqsupseteq p}$$

asserts that whenever $\mathcal{P}$ tries to send a message (i.e. the system is in the state s_{send}) then $\mathcal{Q}$ will receive the message sometime in the future (i.e. the system reaches the state s_{ok}) with probability $\sqsupseteq p$. For all adversaries A, we have $p^A_s((send \vee lost)\;\mathcal{U}\;ok) = 0$ if $s \notin \{s_{send}, s_{lost}, s_{ok}\}$ and $p^A_{s_{ok}}((send \vee lost)\;\mathcal{U}\;ok) = 1$. For each adversary A, the probabilities $p^A_s((send \vee lost)\;\mathcal{U}\;ok)$, $s \in \{s_{send}, s_{lost}\}$, are obtained by solving the linear equation system

$$\begin{pmatrix} 1 & -\varepsilon_l \\ -1 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_{send} \\ p_{lost} \end{pmatrix} \;=\; \begin{pmatrix} 1-\varepsilon \\ 0 \end{pmatrix}$$

which has the unique solution $p_{send} = p_{lost} = (1-\varepsilon)/(1-\varepsilon_l)$. Hence, for all $Adv \in \{\mathcal{A}, \mathcal{A}_{fair}, \mathcal{A}_{sfair}\}$:

$$Sat(\Psi_{\sqsupseteq p}) \;=\; \begin{cases} \{s_{send}, s_{lost}, s_{ok}\} & : \text{ if } (1-\varepsilon)/(1-\varepsilon_l) \sqsupseteq p \\ \{s_{ok}\} & : \text{ otherwise.} \end{cases}$$

As $\Phi_{\sqsupseteq p}$ abbreviates the formula $\neg[\,\exists\Diamond(send \wedge \neg\Psi_{\sqsupseteq p})]_{>0}$, we have

$$Sat(\Phi_{\sqsupseteq p}) = S \setminus Sat([\,\exists\Diamond(send \wedge \neg\Psi_{\sqsupseteq p})]_{>0}).$$

We have:

- If $(1-\varepsilon)/(1-\varepsilon_l) \not\sqsupseteq p$ then $Sat\,([\exists\Diamond(send \wedge \neg\Psi_{\sqsupseteq p})]_{>0}) = Sat\,([\exists\Diamond send\,]_{>0}) = S$ since S_{send} can be reached from all states. We obtain $Sat(\Phi_{\sqsupseteq p}) = \emptyset$.
- If $(1-\varepsilon)/(1-\varepsilon_l) \sqsupseteq p$ then no state satisfies $send \wedge \neg\Psi_{\sqsupseteq p}$. Thus,

$$Sat\,([\exists\Diamond(send \wedge \neg\Psi_{\sqsupseteq p})]_{>0}) = \emptyset.$$

  Hence, all states fulfill $\Phi_{\sqsupseteq p}$. I.e., $Sat(\Phi_{\sqsupseteq p}) = S$.

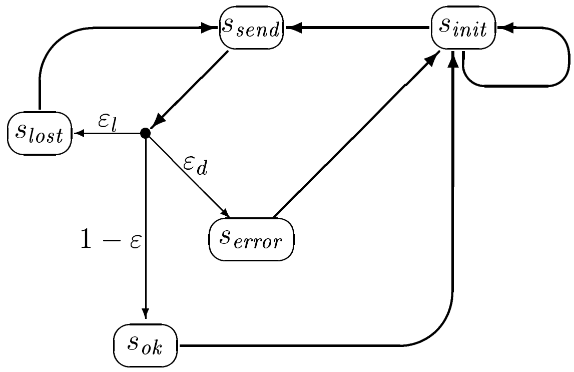


Figure 4: Variant 1

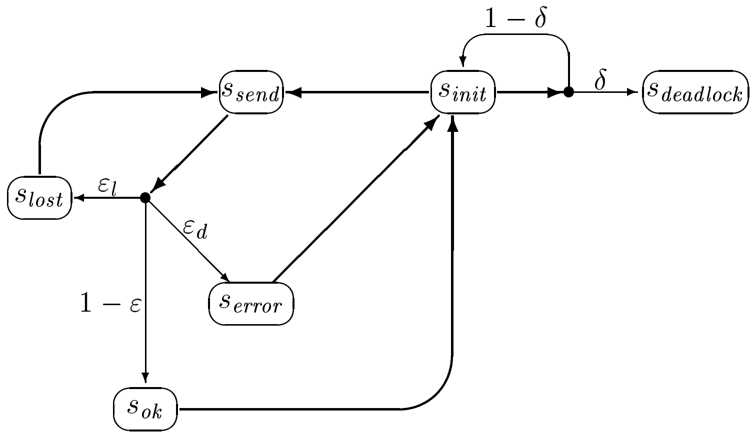


Figure 5: Variant 2

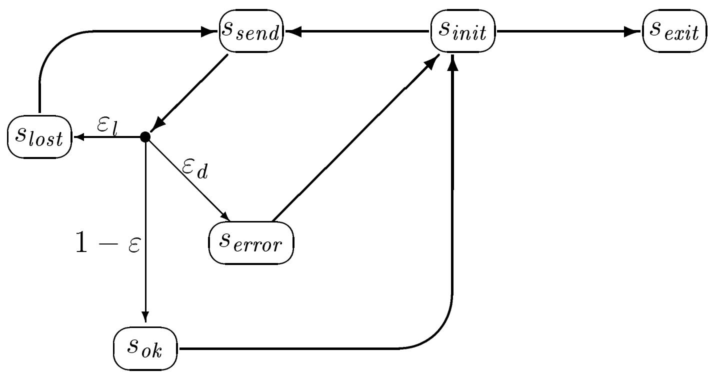


Figure 6: Variant 3

Thus, for $v = 1, 2, 3$, $Adv \in \{\mathcal{A}, \mathcal{A}_{fair}, \mathcal{A}_{sfair}\}$: $s_{init} \models^v_{Adv} \Phi_{\sqsupseteq p}$ iff $(1-\varepsilon)/(1-\varepsilon_l) \sqsupseteq p$.

As in [31], "soft deadlines" can be formulated by means of the bounded-until operator. The following formula $\Phi^k_{\sqsupseteq p}$ asserts that whenever $\mathcal{P}$ tries to send a message then $\mathcal{Q}$ will receive the message within at most k steps with probability $\sqsupseteq p$.

$$\Phi^k_{\sqsupseteq p} \;=\; \left[\, \forall\Box \left(send \longrightarrow \Psi^k_{\sqsupseteq p} \right) \,\right]_{\geq 1} \quad \text{where} \quad \Psi^k_{\sqsupseteq p} \;=\; [\ (send \vee lost)\ \forall\mathcal{U}\ ok\ ]_{\sqsupseteq p}$$

Since $\Phi^k_{\sqsupseteq p} = \neg[\ \exists\Diamond(send \wedge \neg\Psi^k_{\sqsupseteq p})]_{>0}$, the algorithm first computes the set of states satisfying $[\ \exists\Diamond(send \wedge \neg\Psi_{\sqsupseteq p})]_{>0}$. In all three variants we have: $p^0_{send} = p^0_{lost} = 0$, $p^1_{send} = 1-\varepsilon$, $p^{k+1}_{lost} = p^k_{send}$, $p^{k+1}_{send} = \varepsilon_l \cdot p^k_{lost} + 1 - \varepsilon$ where $p^k_* = p^A_{s_*}(\ (send \vee lost)\ \mathcal{U}^{\leq k}\ ok\ )$ and A is an arbitrary adversary. (Then, $p^k_* = p^{max}_{s_*}$ in the notation of Lemma 5.2.) Hence,

$$p^{2i+2}_{lost} \;=\; p^{2i+1}_{lost} \;=\; p^{2i}_{send} \;=\; p^{2i+1}_{send} \;=\; (1-\varepsilon) \cdot \sum_{l=0}^{i} \varepsilon_l^i \;=\; \frac{1-\varepsilon_l^{i+1}}{1-\varepsilon_l} \cdot (1-\varepsilon).$$

With $i = k$ div 2 we obtain in all three variants:

- If $(1-\varepsilon_l^{i+1})(1-\varepsilon)/(1-\varepsilon_l) \sqsupseteq p$ then $Sat(send \wedge \neg\Psi^k_{\sqsupseteq p}) = \{s_{send}\}$. Hence, $Sat(\Phi^k_{\sqsupseteq p})$ consists of exactly those states from which s_{send} is not reachable.
- If $(1-\varepsilon_l^{i+1})(1-\varepsilon)/(1-\varepsilon_l) \not\sqsupseteq p$ then $Sat(send \wedge \neg\Psi^k_{\sqsupseteq p}) = \emptyset$. Thus, all states satisfy $\Phi^k_{\sqsupseteq p}$.

In particular, $s_{init} \models^v_{Adv} \Phi^k_{\sqsupseteq p}$ iff $(1-\varepsilon_l^{i+1})(1-\varepsilon)/(1-\varepsilon_l) \sqsupseteq p$ where $Adv \in \{\mathcal{A}, \mathcal{A}_{fair}, \mathcal{A}_{sfair}\}$ and $v = 1, 2, 3$.

The liveness property $\Lambda_{\sqsupseteq p} = [\ \forall\Diamond send\ ]_{\sqsupseteq p}$ states that, in all computations, $\mathcal{P}$ will eventually try to send a message with probability $\sqsupseteq p$. In all three variants we have $s_{init} \not\models^v \Lambda_{\geq 1}$ since for the simple adversary $A = A^v_{\neg send}$ we have $p^A_{s_{init}}(tt\ \mathcal{U}\ send) = 0$, i.e. without any fairness assumptions we cannot ensure that $\mathcal{P}$ will try to send a message.

- In the first variant we have: $s_{init} \models^1_{fair} \Lambda_{\geq 1}$ and $s_{init} \models^1_{sfair} \Lambda_{\geq 1}$. This follows immediately by Theorem 5 and Theorem 6 and the fact that in the first variant all states can reach the state s_{send} (hence, $Reach_{tt \wedge \neg send}(s_{init}) \subseteq S^+(tt, send)$).
- In the second variant, A_{send} and $A^2_{\neg send}$ are admissible w.r.t. $(tt, send)$. Moreover, $A^2_{\neg send}$ is fair (but not strictly fair). With $A = A^2_{\neg send}$ we have $p^A_{s_{init}}(tt\ \mathcal{U}\ send) = 0$ and therefore, by Theorem 5: $p^{adm}_{s_{init}}(tt\ \mathcal{U}\ send) = 0$. Hence, $s_{init} \not\models^2_{fair} \Lambda_{>0}$. Since $AdmSteps(s_{init}, tt, send) = \{\mu_{send}\}$ we get $s_{init} \notin T^{adm}(tt, send)$. By Theorem 7, $s_{init} \models^2_{sfair} \Lambda_{>0}$.
- In the third variant, $A = A^3_{\neg send}$ is admissible for $(tt, send)$ and $p^A_{s_{init}}(tt\ \mathcal{U}\ send) = 0$. Hence, $s_{init} \not\models^3_{Adv} \Lambda_{>0}$ where $Adv \in \{\mathcal{A}, \mathcal{A}_{fair}, \mathcal{A}_{sfair}\}$.

7 Time complexity of model checking

The size of the parse tree (the number of nodes) is linear in the length $|\Phi|$. Let n be the set of states and m the number of transitions in the underlying *PBTL* structure. For every node v of the parse tree where the associated formula Φ is tt, an atomic proposition or of the form $\neg\Phi'$ or $\Phi_1 \wedge \Phi_2$, the costs for computing the set of states fulfilling Φ is $\mathcal{O}(n)$.

The nodes which represent formulas whose outermost operator is the next-step operator require $\mathcal{O}(n \cdot m)$ time, since for every transition $\mu \in \mathit{Steps}(s)$ we have to compute the sum $\sum \mu(t)$ (cf. Lemma 5.1). Computing the states that fulfill a formula whose outermost operator is the bounded until operator using the method of Lemma 5.2 takes $\mathcal{O}(k \cdot n \cdot m)$ time, where k is the superscript of the bounded until operator.

The main contribution of this paper is a method for dealing with nodes representing formulas of the form $[\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ or $[\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ w.r.t. the fair and strictly fair satisfaction relation. The technical results of the previous section (Theorems 2-7) establish that in order to compute the sets of states fulfilling formulas of the form $[\ \Phi_1\ \exists\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ or $[\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$, w.r.t. $\models_{fair}$ or $\models_{sfair}$ one has to calculate the probabilities $p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ or $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2)$, and possibly $T^{max}(\Phi_1, \Phi_2)$ or $T^{adm}(\Phi_1, \Phi_2)$. As in [20, 14], $p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$ can be computed by solving a linear programming problem by means of well-known methods. More precisely, the vector $(p_s^{max}(\Phi_1\mathcal{U}\Phi_2))_{s\in S}$ is the unique solution of the linear minimization problem

$$\begin{array}{rcl} x_s & = & 1 \quad \text{if } s \in \mathit{Sat}(\Phi_2) \\ x_s & = & 0 \quad \text{if } s \in S \setminus (\mathit{Sat}(\Phi_1) \cup \mathit{Sat}(\Phi_2)) \\ x_s & \geq & \sum\limits_{t \in S} \mu(t) \cdot x_t \quad \text{if } s \in \mathit{Sat}(\Phi_1) \setminus \mathit{Sat}(\Phi_2) \text{ and } \mu \in \mathit{Steps}(s) \end{array}$$

where $\sum_{s\in S}\ x_s$ is minimal. (Note the slight departure from [14] which proposes first to compute all states $s \in S$ where $p_s^{max}(\Phi_1\mathcal{U}\Phi_2) = 0$ by an analysis of the graph.) The algorithms with ellipsoid methods have time complexity which is polynomial in n and m.

Computation of $S^+(\Phi_1, \Phi_2)$: Let $G^+(\Phi_1, \Phi_2)$ be the directed graph (S, E) where $(s, t) \in E$ iff $t \models \Phi_1 \wedge \neg\Phi_2$ and $\mu(s) > 0$ for some $\mu \in \mathit{Steps}(t)$. Then, $S^+(\Phi_1, \Phi_2)$ is the set of states which are reachable in $G^+(\Phi_1, \Phi_2)$ from a state $s \in \mathit{Sat}(\Phi_2)$. Hence, $S^+(\Phi_1, \Phi_2)$ can be derived by a depth-first search in $G^+(\Phi_1, \Phi_2)$. The construction of $G^+(\Phi_1, \Phi_2)$ needs $\mathcal{O}(n \cdot m)$ steps. The time for performing a depth-first search in G is linear in n and the number of edges in $G^+(\Phi_1, \Phi_2)$. As the number of edges in $G^+(\Phi_1, \Phi_2)$ is bounded by $\min\{n^2, n \cdot m\}$ we get the time complexity $\mathcal{O}(n \cdot m)$ for the computation of $S^+(\Phi_1, \Phi_2)$.

Computation of $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2)$: We show how to compute the values $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2)$ by solving a linear optimization problem. The state space S is split into three parts:

$$S = S^1(\Phi_1, \Phi_2) \cup S^0(\Phi_1, \Phi_2) \cup S^?(\Phi_1, \Phi_2)$$

where

$$\begin{array}{rcl} S^0(\Phi_1, \Phi_2) & = & \left\{ s \in S : p_s^F(\Phi_1\mathcal{U}\Phi_2) = 0 \text{ for some } F \in \mathcal{A}_{fair} \right\}, \\ S^1(\Phi_1, \Phi_2) & = & \left\{ s \in S : \mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}(s) \cap S^0 = \emptyset \right\}, \\ S^?(\Phi_1, \Phi_2) & = & S \setminus \left(S^1(\Phi_1, \Phi_2) \cup S^0(\Phi_1, \Phi_2) \right). \end{array}$$

We assume that there are atomic propositions $a^?$ and a^0 with $a^? \in L(s)$ iff $s \in S^?(\Phi_1, \Phi_2)$ and $a^0 \in L(s)$ iff $s \in S^0(\Phi_1, \Phi_2)$. In Section 12.5, Corollary 12.27, we show:

Lemma 7.1 *For all states $s \in S$: $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2) \ = \ 1 - p_s^{max}(a^?\mathcal{U}a^0)$.*

Hence, the values $p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2)$ can be obtained by first computing the sets $S^0(\Phi_1, \Phi_2)$, $S^1(\Phi_1, \Phi_2)$ and $S^?(\Phi_1, \Phi_2)$ and then computing $p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2)$ by solving the following linear optimization problem. Using standard methods of linear programming we calculate the unique solution of the linear minimization problem

$$\begin{aligned} x_s &= 1 \;\text{ if } s \in S^0(\Phi_1, \Phi_2) \\ x_s &= 0 \;\text{ if } s \in S^1(\Phi_1, \Phi_2) \\ x_s &\geq \sum_{t \in S} \mu(t) \cdot x_t \;\text{ if } s \in S^?(\Phi_1, \Phi_2) \text{ and } \mu \in \mathit{Steps}(s) \end{aligned}$$

where

$$\sum_{s \in S} x_s \;\text{ is minimal}$$

and then put $p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2) = 1 - x_s$.

We describe a method to calculate the set $S^0(\Phi_1, \Phi_2)$. In Section 12.5, Lemma 12.28, we show that

$$S^0(\Phi_1, \Phi_2) = \bigcup_{i \geq 0} T_i$$

where $T_0 = S \setminus S^+(\Phi_1, \Phi_2)$ and T_{i+1} is the largest subset of $S \setminus (T_0 \cup \ldots \cup T_i \cup \mathit{Sat}(\Phi_2))$ such that for all $t \in T_{i+1}$ there is some $\mu_t \in \mathit{Steps}(t)$ with:

- $\mathit{Supp}(\mu_t) \subseteq T_0 \cup \ldots \cup T_i \cup T_{i+1}$
- there is a finite path $t = t_0 \xrightarrow{\mu_{t_0}} t_1 \xrightarrow{\mu_{t_1}} \ldots \xrightarrow{\mu_{t_{k-1}}} t_k$ where $t_0, \ldots, t_{k-1} \in T_{i+1}$ and $t_k \in T_0 \cup \ldots \cup T_i$.

We compute the sets $T_1, T_2, \ldots$ by the following graph analysis. For $s \in S$, let $\mathit{Steps}'(s)$ be the set of all $\mu \in \mathit{Steps}(s)$ where $\mu(s) = 0$ for all $s \in \mathit{Sat}(\Phi_2)$. We consider the directed graph $G^0(\Phi_1, \Phi_2)$ where the vertices are

$$V = \{\bot\} \cup \Big\{ (s, \mathit{Supp}(\mu)) : s \in S^+(\Phi_1, \Phi_2) \setminus \mathit{Sat}(\Phi_2), \mu \in \mathit{Steps}'(s) \Big\}$$

and where the edges are given by: $\bot \to (s, A)$ iff $A \cap S \setminus S^+(\Phi_1, \Phi_2) \neq \emptyset$ and $(s, A) \to (t, B)$ iff $s \in B$. For $v = (s, A) \in V \setminus \{\bot\}$, $v.\mathit{state}$ denotes the first component of v, i.e. $v.\mathit{state} = s$. Let $C_0 = \{\bot\}$, $C_1, \ldots, C_l$ be an enumeration of those strongly connected components of $G^0(\Phi_1, \Phi_2)$ that are reachable from $\bot$ (i.e. those strongly connected components C where $C_0 \to C$) such that, whenever $C_i \to C_j$ for some $i \in \{1, \ldots, l\}$ then $i \leq j$.[4] Let

$$C_0.\mathit{states} = S \setminus S^+(\Phi_1, \Phi_2), \quad C_i.\mathit{states} = \{v.\mathit{state} : v \in C_i\}, \; i = 1, \ldots, l.$$

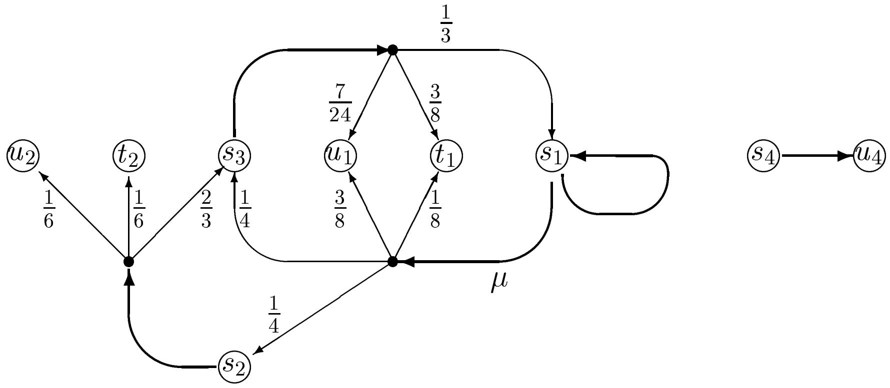
The set $S^0(\Phi_1, \Phi_2)$ can be computed with the following method.

(1) $I := \{0\}$
(2) For $j = 1, \ldots, l$ do
    If $\{i \in \{1, \ldots, j-1\} : C_i \to C_j\} \subseteq I$ then $I := I \cup \{j\}$
(3) Return $S^0 = \bigcup_{i \in I} C_i.\mathit{states}$.

[4] Here, we write $C_i \to C_j$ iff there exists $v \in C_i$ and an edge $v \to w$ for some $w \in C_j$.

Clearly, $G^0(\Phi_1, \Phi_2)$ has at most m vertices. Hence, the computation of $S^0(\Phi_1, \Phi_2)$ takes $\mathcal{O}(m^2)$ time. Thus, the sets $S^0(\Phi_1, \Phi_2)$, $S^1(\Phi_1, \Phi_2)$, $S^?(\Phi_1, \Phi_2)$ and the values $p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2)$ can be obtained in time polynomial in the size of the system.[5]

Example 7.2 We compute the probabilities $p_s^{adm}(a\mathcal{U}b)$ for all states s in the *PBTL*-structure that is shown in the following picture. Here, we suppose that $L(s_i) = \{a\}$, $L(t_j) = \{b\}$ and $L(u_k) = \emptyset$.



The outgoing transitions from the states t_j and u_k are omitted since they are irrelevant for the truth value of the path formula $a\mathcal{U}b$. Before we explain how our method for computing $p_*^{adm}(a\mathcal{U}b)$ works we observe that the desired result is $p_{u_k}^{adm}(a\mathcal{U}b) = 0$, $p_{t_j}^{adm}(a\mathcal{U}b) = 1$ and

$$p_{s_i}^{adm}(a\mathcal{U}b) = \begin{cases} 3/8 & : \text{ if } i = 1 \\ 1/2 & : \text{ if } i \in \{2,3\} \\ 0 & : \text{ if } i = 4. \end{cases}$$

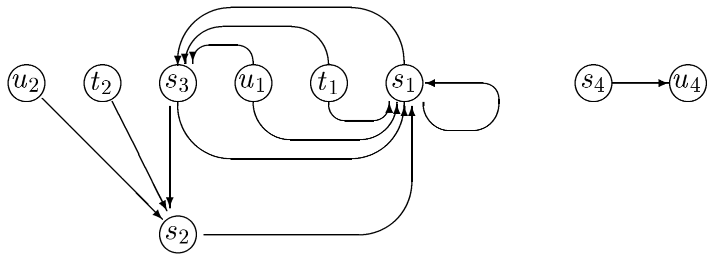
Note that there are two simple adversaries A_μ and A_ν. These are given by $A_\mu(s_1) = \mu$ and $A_\nu(s_1) = \nu$ where $\nu = \mu_{s_1}^1$. It is easy to see that A_μ is admissible for (a, b) while A_ν is not. Let $A = A_\mu$. Then, $p_*^{adm}(a\mathcal{U}b) = p_*^A(a\mathcal{U}b)$. We have $p_{t_i}^A(a\mathcal{U}b) = 1$, $p_{u_k}^A(a\mathcal{U}b) = p_{s_4}^A(a\mathcal{U}b) = 0$, $k = 1, 2, 4$. The probabilities $p_i = p_{s_i}^A(a\mathcal{U}b)$, $i = 1, 2, 3$, can be obtained by solving the linear equation system

$$\begin{pmatrix} 1 & -\frac{1}{4} & -\frac{1}{4} \\ 0 & 1 & -\frac{2}{3} \\ -\frac{1}{3} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} 1/8 \\ 1/6 \\ 3/8 \end{pmatrix}$$

whose unique solution is $p_1 = 3/8$, $p_2 = p_3 = 1/2$.

Now we show how the method for computing $p_*^{adm}(a\mathcal{U}b)$ described above works. First, we compute $S^+(a, b)$, $S^0(a, b)$, $S^1(a, b)$ and $S^?(a, b)$. Secondly, we have to calculate the probabilities $p_*^{max}(a^? \mathcal{U} a^0)$. For computing $S^+(a, b)$ we consider the directed graph $G^+(a, b)$ which is of the following form.

[5]Note that the set $S^1(\Phi_1, \Phi_2)$ can be derived from $S^0(\Phi_1, \Phi_2)$ by a reachability analysis in $G^+(\Phi_1, \Phi_2)$. More precisely, $S \setminus S^1(\Phi_1, \Phi_2)$ is the set of states that are reachable from a state $s \in S^0(\Phi_1, \Phi_2)$ in $G^+(\Phi_1, \Phi_2)$. Thus, $S^1(\Phi_1, \Phi_2)$ can be obtained in time $\mathcal{O}(n \cdot m)$.



We compute $S^+(a,b)$ as the set of states that are reachable from t_1 or t_2 (as they satisfy b) and obtain $S^+(a,b) = \{t_1, t_2, s_1, s_2, s_3\}$. For the computation of $S^0(a,b)$ we consider the following directed graph $G^0(a,b)$:

$$\begin{tikzcd} \perp \arrow[r] & s_4, \{u_4\} & & s_1, \{s_1\} \end{tikzcd}$$

(Note that $\mathit{Steps}'(s_2) = \mathit{Steps}'(s_3) = \emptyset$ and $\mathit{Steps}'(s_1) = \{\mu^1_{s_1}\}$.) We compute the strongly connected components of $G^0(a,b)$ that are reachable from $\perp$ and obtain $C_0 = \{\perp\}$ and $C_1 = \{(s_4, \{u_4\})\}$. Hence,

$$S^0(a,b) = \{u_1, u_2, u_4, s_4\}, \quad S^1(a,b) = \{t_1, t_2\}, \quad S^?(a,b) = \{s_1, s_2, s_3\}.$$

Thus, $p^{max}_{u_k}(a^? \mathcal{U} a^0) = p^{max}_{s_4}(a^? \mathcal{U} a^0) = 1$ and $p^{max}_{t_j}(a^? \mathcal{U} a^0) = 0$. For computing the values $p^{max}_{s_i}(a^? \mathcal{U} a^0)$, $i = 1,2,3$, we solve the following linear optimization problem:

$$\begin{array}{rcl} x_{s_1} & \geq & \frac{1}{4} \cdot x_{s_3} + \frac{1}{4} x_{s_2} + \frac{3}{8}, \qquad x_{s_1} \ \geq \ x_{s_1} \\ x_{s_2} & \geq & \frac{2}{3} \cdot x_{s_3} + \frac{1}{6} \\ x_{s_3} & \geq & \frac{1}{3} \cdot x_{s_1} + \frac{7}{24} \end{array}$$

where $x_{s_1} + x_{s_2} + x_{s_3}$ is minimal. We obtain

$$p^{max}_{s_i}(a^? \mathcal{U} a^0) = x_{s_i} = \left\{ \begin{array}{lcl} 5/8 & : & \text{if } i = 1 \\ 1/2 & : & \text{if } i \in \{2,3\}. \end{array} \right.$$

Thus, our method yields

$$p^{adm}_s(a \mathcal{U} b) \ = \ 1 - p^{max}_s(a^? \mathcal{U} a^0) \ = \ \left\{ \begin{array}{lcl} 1 & : & \text{if } s \in S^1(a,b) = \{t_1, t_2\} \\ 0 & : & \text{if } s \in S^0(a,b) = \{u_1, u_2, u_4, s_4\} \\ 3/8 & : & \text{if } s = s_1 \\ 1/2 & : & \text{if } s \in \{s_2, s_3\} \end{array} \right.$$

as desired. ■

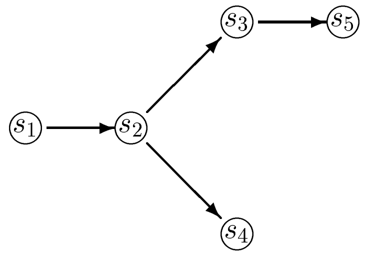
Computation of $T^{max}(\Phi_1, \Phi_2)$ and $T^{adm}(\Phi_1, \Phi_2)$: We give an algorithm for the computation of $T^{max}(\Phi_1, \Phi_2)$. (The computation of $T^{adm}(\Phi_1, \Phi_2)$ is similar; one only has to replace $\mathit{MaxSteps}(\cdot)$ by $\mathit{AdmSteps}(\cdot)$). First we compute $\mathit{MaxSteps}(s)$ for all $s \in S$ and set $T = \mathit{Sat}(\Phi_2) \cup (S \setminus S^+(\Phi_1, \Phi_2))$ and $U = \{v \in S \setminus T : \mathit{MaxSteps}(s) \neq \mathit{Steps}(s)\}$. We compute the strongly connected components in the directed graph $(S \setminus (T \cup U), E)$ where $(s,t) \in E$ iff $\mu(t) > 0$ for some $\mu \in \mathit{MaxSteps}(s) = \mathit{Steps}(s)$. Let $C_1, \ldots, C_k$ be an enumeration of the strongly connected components which satisfies: if $s \in C_j$, $s' \in C_l$ with $(s, s') \in E$ then $l \leq j$. For $i = 1, \ldots, k$ we compute the set W_i of states $w \in S \setminus T$

such that $\mu(w) > 0$ for some $\mu \in \mathit{Steps}(s)$ and $s \in C_i$. Let Z be the set of pairs (V, W) such that V, W are nonempty subsets of $S \setminus T$ and $V = \{v \in S \setminus T : \mu \in \mathit{MaxSteps}(v)\}$, $W = \{w \in S \setminus T : \mu(w) > 0\}$ for some distribution μ. For $z \in Z$ we denote the first component of z by $z.V$, the second component by $z.W$and we define $|z| = |z.W|$. Let S_0 be the set of states $s \in S \setminus T$ with $s \in z.V$ for some $z \in Z$ with $|z| = 0$. We successively modify S_0, T and $|z|$ by the following procedure:

For $i = 1, 2, \ldots, k+1$ do:
- (1) While $S_0 \neq \emptyset$ do:
  - (1.1) choose some $s \in S_0$
  - (1.2) $S_0 := S_0 \setminus \{s\}$, $T := T \cup \{s\}$
  - (1.3) For all $z \in Z$ do:
    - (1.3.1) If $s \in z.W$ then $|z| := |z| - 1$.
    - (1.3.2) If $|z| = 0$ then $S_0 := S_0 \cup (z.V \setminus T)$.
- (2) If $i \leq k$ and $W_i \subseteq C_i \cup T$ then $S_0 := S_0 \cup C_i \setminus T$.

Then, $T^{max}(\Phi_1, \Phi_2) = T$.

Example 7.3 We consider the *PBTL*-structure Example 5.10 (Figure 3). Then, $\mu_1, \mu_2 \in \mathit{MaxSteps}(s_1)$ and $\mathit{MaxSteps}(s_6) = \{\nu_1\}$. We obtain $U = \{s_6\}$ and $T = S \setminus \{s_1, \ldots, s_6\}$. We first compute the strongly connected components of the directed graph



and obtain $C_1 = \{s_5\}$, $C_2 = \{s_3, s_4\}$, $C_3 = \{s_2\}$, $C_4 = \{s_1\}$ and $W_1 = \emptyset$, $W_2 = \{s_3, s_4, s_5\}$, $W_3 = \{s_3, s_4\}$, $W_4 = \{s_2, s_6\}$. Initially, the set Z consists of the pairs

$$(\{s_5\}, \emptyset),\ (\{s_3\}, \{s_5\}),\ (\{s_3\}, \{s_4\}),\ (\{s_4\}, \{s_3, s_4\}),$$
$$(\{s_2\}, \{s_3, s_4\}),\ (\{s_1\}, \{s_2\}),\ (\{s_1\}, \{s_6\}),\ (\{s_6\}, \{s_6\}).$$

This yields $S_0 = \{s_5\}$. In the first iteration step ($i = 1$), we first remove s_5 from S_0 and obtain $S_0 = \{s_3\}$ and $s_5 \in T$. Then, we remove s_3 from S_0 and obtain $S_0 = \emptyset$, $s_3 \in T$. Thus, in the second iteration step ($i = 2$), step (1) is not applicable (since $S_0 = \emptyset$). In step (2) we have $W_2 = \{s_3, s_4, s_5\} \subseteq C_2 \cup T$ and obtain $S_0 = \{s_4\}$. The third iteration step ($i = 3$) removes s_4 from S_0 and yields $S_0 = \{s_2\}$, $s_4 \in T$. Then, we remove s_2 from S_0 and obtain $S_0 = \{s_1\}$, $s_2 \in T$. Finally, we remove s_1 from S_0 and get $S_0 = \emptyset$ and $s_1 \in T$. In the iteration steps $i = 4, 5$, only step (2) is applicable that yields $S_0 = \emptyset$. The algorithm returns $T^{max}(a, b) = S \setminus \{s_6\}$. ■

For the computation of $\mathit{MaxSteps}(\cdot)$, the set U, the components $C_1, \ldots, C_k$ and the sets $W_1, \ldots, W_k$ we need $\mathcal{O}(n \cdot m)$ time. (Note that for the computation of $\mathit{MaxSteps}(s)$ we have to calculate the sum $\sum_{t \in S} \mu(t) \cdot p_s^{max}(\Phi_1 \mathcal{U} \Phi_2)$ for each $\mu \in \mathit{Steps}(s)$. As G has at most $\min\{n^2, n \cdot m\}$ edges and as the strongly connected components of a directed graph can always be computed in time linear in the number of states and edges, the computation of $C_1, \ldots, C_k$ takes $\mathcal{O}(n \cdot m)$ time.) In what follows, T_0 denotes the initial value of T, i.e. the set $Sat(\Phi_1, \Phi_2) \cup S \setminus S^+(\Phi_1, \Phi_2)$. For the computation of Z and the function $|\cdot|$ we suggest the following method. Let $s_1, \ldots, s_l$ be an enumeration of the elements of $S \setminus T_0$. We construct a binary tree by successively inserting nodes and edges where each node y is labelled by a natural number $|y|$ and a subset $y.W$ of $S \setminus T_0$. The leaves of the resulting tree correspond to the elements of Z. Each leaf z has depth l and is labelled additionally by the set $z.V$. We start with the tree of depth 0, i.e. the tree consisting of its root y_0 which we label by $|y_0| = 0$ and $y_0.W = \emptyset$. Then, for each $s \in S \setminus T_0$ and each $\mu \in MaxSteps(s)$ we traverse the tree in the following way:

- If we have reached a node y of depth $k-1$, $k \leq l$, then:
  - If y has a left son z and $\mu(s_k) > 0$ then we go to z.
  - If y does not have a left son and $\mu(s_k) > 0$ then we create a new left son z of y where we set $|z| := |y| + 1$, $z.W := y.W \cup \{s_k\}$ and, if $k = l$, $z.V := \emptyset$. We go to z.
  - If y has a right son z and $\mu(s_k) = 0$ then we go to z.
  - If y does not have a right son and $\mu(s_k) = 0$ then we create a new right son z of y where we set $|z| := |y|$, $z.W := y.W$ and, if $k = l$, $z.V := \emptyset$. We go to z.
- If we have reached a node z of depth l then we set $z.V := z.V \cup \{s\}$.

This method takes $\mathcal{O}\left(\, l \cdot \sum_{s \in S \setminus T_0} |MaxSteps(s)| \,\right) = \mathcal{O}(n \cdot m)$ steps. (Note that $l = |S \setminus T_0| \leq n$.)

In what follows, we suppose the sets $T, C_1, \ldots, C_k$ and $z.W$ for $z \in Z$ to be represented as boolean vectors (one bit for each state $s \in S \setminus T_0$) and that each of the sets $Z, W_1, \ldots, W_k$, S_0 and $z.V$ for each $z \in Z$ is represented as a list consisting of pointers to their elements. Then, the test in (1) and steps (1.1), (1.2) can be performed in constant time. Step (1.3) can be performed in time linear in the size of Z. As $|Z| \leq m$ we get the time complexity $\mathcal{O}(m)$ for step (1.3). As each state $s \in S \setminus T_0$ can only be chosen once in step (1.1) the while-loop can be performed at most n-times. Hence, ranging over all $i \in \{1, 2, \ldots, k+1\}$ and all executions of the while loop we need $\mathcal{O}(n \cdot m)$ time to perform steps (1.1),(1.2) and (1.3). Ranging over all $i \in \{1, 2, \ldots, k\}$ we need

$$\sum_{i=1}^{k} \mathcal{O}(|W_i|) \;=\; \mathcal{O}(\, |S \setminus (T_0 \cup U)| \,) \;=\; \mathcal{O}(n)$$

time for step (2). We conclude that the time complexity of computing $T^{max}(\Phi_1, \Phi_2)$ by the method described above is $\mathcal{O}(n \cdot m)$. Summing up over all nodes in the parse tree we obtain the time complexity

$$\mathcal{O}\left(|\Phi| \cdot \left(k^\Phi \cdot n \cdot m + p(n, m)\right)\right).$$

Here, $p(n, m)$ is a function that is polynomial in n and m (the time for computing $p_*^{max}(\cdot)$ and $p_*^{adm}(\cdot)$ by solving a linear optimization problem) and k^Φ is either 1 (in the case where Φ does not contain the bounded until operator) or the maximal value k such that Φ contains a subformula of the form $[\ \Phi_1\ \exists\mathcal{U}^{\leq k}\ \Phi_2\ ]_{\sqsupseteq p}$ or $[\ \Phi_1\ \forall\mathcal{U}^{\leq k}\ \Phi_2\ ]_{\sqsupseteq p}$. I.e., the time complexity is polynomial in the size of the structure and linear in the size of the formula. The space complexity is $\mathcal{O}(|\Phi| \cdot n + n \cdot m)$. [6]

[6] The representation of the set associated with each node v of the parse tree requires $\mathcal{O}(n)$ space. For

8 Fairness à la Vardi

Our method can be easily modified to deal with the original definition of fair adversaries in the sense of Vardi [58] which requires fairness w.r.t. the non-deterministic choices in certain (but not all) states. For the rest of this section, $M = (\mathcal{S}, L)$ is a *PBTL*-structure, $\mathcal{S} = (S, Steps)$ and $W \subseteq S$.

Definition 8.1 *A fulpath π in $\mathcal{S}$ is* fair w.r.t. W *iff for all $s \in inf(\pi) \cap W$ and all $\mu \in Steps(s)$ there are infinitely many indices $j \geq 0$ with $step(\pi, j) = \mu$.*

Fairness w.r.t. $W = S$ (in the sense of Definition 8.1) is weaker than fairness of a fulpath in the sense of Definition 3.5. (Note that in Definition 8.1 we do not require that $step(\pi, j) = \mu$ and $\pi(j) = s$.) Vardi's notion of fairness adapted to our model for concurrent probabilistic systems is the following:

Definition 8.2 *An adversary F is called* fair w.r.t. W *iff, for all $s \in S$, the measure of the set of fulpaths $\pi \in Path^F_{ful}(s)$ which are fair w.r.t. W is 1.*

Let $\mathcal{A}^W_{fair}$ be the set of adversaries which are fair w.r.t. W and let $\models^W_{fair}$ be the induced satisfaction relation. As each fair adversary A (in the sense of Definition 3.7) is fair w.r.t. W, Theorem 2 carries over to the satisfaction relation $\models^W_{fair}$:

Theorem 8 *For all $s \in S$: $s \models^W_{fair} [\, \Phi_1 \, \exists\mathcal{U} \, \Phi_2 \,]_{\sqsupseteq p}$ iff $p^{max}_s(\Phi_1 \mathcal{U} \Phi_2) \sqsupseteq p$.*

Proof: follows by Theorem 1(a), (cf. Lemma 12.1), Theorem 2 and $\mathcal{A}_{fair} \subseteq \mathcal{A}^W_{fair}$. ■

Definition 8.3 *A simple adversary A is called* admissible for (Φ_1, Φ_2) w.r.t. W *iff for all $s \in S^+(\Phi_1, \Phi_2)$:*

$$Reach^A_{\Phi_1 \wedge \neg \Phi_2}(s) \cap \left(Sat(\Phi_2) \cup S^A_W \right) \neq \emptyset.$$

Here, $S^A_W = \bigcup_{i \geq 0} S^{A,i}_W$ with $S^{A,0}_W = S \setminus S^+(\Phi_1, \Phi_2)$ and, for $i \geq 1$, $S^{A,i}_W = S^{A,i,1}_W \cup S^{A,i,2}_W$ where $U^{A,i-1}_W = S^{A,0}_W \cup S^{A,1}_W \cup \ldots \cup S^{A,i-1}_W$ and

- $S^{A,i,1}_W = \left\{ t \in S \setminus (U^{A,i-1}_W \cup Sat(\Phi_2)) : A(t,u) > 0 \text{ implies } u \in U^{A,i-1}_W \right\}$
- *$S^{A,i,2}_W$ consists of all those states $t \in T$ where $T \subseteq S \setminus (U^{A,i-1}_W \cup S^{A,i,1}_W \cup Sat(\Phi_2))$ such that:*
  - *for all $t \in T \cap W$ and $\mu \in Steps(t)$: $Supp(\mu) \subseteq T \cup (U^{A,i-1}_W \cup S^{A,i,1}_W)$*
  - *for all $t \in T \setminus W$: $A(t,u) > 0$ implies $u \in T \cup (U^{A,i-1}_W \cup S^{A,i,1}_W)$.*

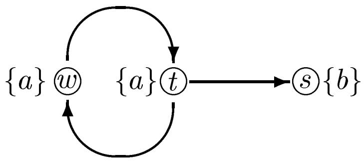
$\mathcal{A}^W_{adm}(\Phi_1, \Phi_2)$ (abbreviated $\mathcal{A}^W_{adm}$) denotes the set of adversaries which are admissible for (Φ_1, Φ_2) w.r.t. W. For $s \in S$ we define $p^{adm_W}_s(\Phi_1 \mathcal{U} \Phi_2) = \min\{p^A_s(\Phi_1 \mathcal{U} \Phi_2) : A \in \mathcal{A}^W_{adm}\}$.

Clearly, if $A \in \mathcal{A}^W_{adm}$, $s \in S^A_W$ then $Reach^A_{\Phi_1 \wedge \neg \Phi_2}(s) \subseteq S^A_W$ and $p^A_s(\Phi_1 \mathcal{U} \Phi_2) = 0$ (as $S^A_W \cap Sat(\Phi_2) = \emptyset$).

---

the *PBTL*-structure itself we need $\mathcal{O}(n \cdot m)$ space (where we neglect the space needed for the representation of the labelling function L). For the computation of $p^{max}_s(\Phi_1 \mathcal{U} \Phi_2)$, $p^{min}_s(\Phi_1 \mathcal{U} \Phi_2)$ or $p^{adm}_s(\Phi_1 \mathcal{U} \Phi_2)$ we need $\mathcal{O}(n^2)$ space while the computation of the sets $T^{max}(\Phi_1, \Phi_2)$ or $T^{adm}(\Phi_1, \Phi_2)$ needs $\mathcal{O}(n \cdot m)$ space. (Note that $n \leq m$.)

Remark 8.4 If $A \in \mathcal{A}_{simple}$ then A is admissible for (Φ_1, Φ_2) w.r.t. $W = S$ if and only if A is admissible for (Φ_1, Φ_2) in the sense of Definition 5.15. Note that when $W = S$ then $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(s) \cap (S \setminus S^+(\Phi_1, \Phi_2)) \neq \emptyset$ for all $s \in S^A_W$. In the case where W is a proper subset of S, $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(s) \cap (S \setminus S^+(\Phi_1, \Phi_2))$ might be empty for some state $s \in S^A_W$. For instance, consider the following *PBTL*-structure.



Let $W = \{w\}$. We have $S^+(a, b) = \{s, t, w\}$. Hence, $S^{A,0}_W = \emptyset$. Then, $S^{A,1,1}_W = \emptyset$ and $S^{A,1,2}_W = \{t, w\}$. Thus, $S^A_W = \{t, w\}$ and $Reach^A_{a \wedge \neg b}(x) \cap S^A_W \neq \emptyset$, $x \in \{t, w\}$. Hence, the simple adversary A with $A(t) = \mu^1_w$ is admissible for (a, b) w.r.t. $W = \{w\}$. On the other hand, $Reach^A_{a \wedge \neg b}(w) \cap (Sat(\Phi_2) \cup S \setminus S^+(a, b)) \;=\; \{t, w\} \cap \{s\} = \emptyset$ which yields that A is not admissible for (a, b) in the sense of Definition 5.15. ■

Theorem 9 *For all $s \in S$: $s \models^W_{fair} [\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$ iff $p^{adm_W}_s(\Phi_1 \mathcal{U} \Phi_2) \sqsupseteq p$.*

The proof of Theorem 9 can be found in Section 12.8. Similarly to the results stated in Section 7 we have:

$$p^{adm_W}_s(\Phi_1 \mathcal{U} \Phi_2) \;=\; 1 - p^{max}_s(a^?_W \mathcal{U} a^0_W)$$

where $a^?_W$, a^0_W are atomic propositions that characterise the sets $S^?_W = S \setminus (S^1_W \cup S^0_W)$ and S^0_W. Here, S^0_W is the set of states $s \in S$ where $p^F_s(\Phi_1 \mathcal{U} \Phi_2) = 0$ for some $F \in \mathcal{A}^W_{fair}$ and $S^1_W = \{s \in S : Reach_{\Phi_1 \wedge \neg\Phi_2}(s) \cap S^0_W = \emptyset\}$. S^0_W and S^1_W can be computed with graph theoretic methods (cf. Remark 12.48 and see the proof of Lemma 12.40 for a graph theoretic characterization of S^0_W). Thus, the values $p^{adm_W}_s(\cdot)$ can be obtained by solving a linear optimization problem.

Theorems 2, 5, 8 and 9 show that the satisfaction relations $\models_{fair}$ and $\models^S_{fair}$ coincide, although fairness w.r.t. S (in the sense of Definition 8.1 with $W = S$) does *not* coincide with fairness in the sense of Definition 3.7. It is clear that fairness w.r.t. a proper subset W of S induces a satisfaction relation which differs from $\models_{fair}$ (e.g. in the example in Remark 8.4 we have $t \models_{fair} [\ a\ \forall\mathcal{U}\ b\ ]_{\geq 1}$ while $t \not\models^W_{fair} [\ a\ \forall\mathcal{U}\ b\ ]_{\geq 1}$).

Example 8.5 Consider the path formula $a\mathcal{U}b$ and the *PBTL*-structure of Figure 7 where $W = \{w_1, w_2\}$. Let A be a simple adversary with $A(w_1) = \mu^1_{v_1}$, $A(w_2) = \mu^1_{w_1}$ and $A(v_1) = \mu^1_{v_1}$. Then, $S^{A,0}_W = \emptyset$, $S^{A,1,1}_W = \{v_1\}$, $S^{A,1,2}_W = \emptyset$, $S^{A,2,1}_W = \{w_1\}$ and $S^{A,2,2}_W = \{w_2, v_2\}$. Thus, $S^A_W = \{v_1, v_2, w_1, w_2\}$. Hence, A is admissible w.r.t. W. We get $p^{adm_W}_x(a\mathcal{U}b) = 0$ for $x \in S^A_W$. Hence, $p^{adm_W}_t(a\mathcal{U}b) = 1/5$. By Theorem 9: $t \models^W_{fair} [\ a\ \forall\mathcal{U}\ b\ ]_{\geq 0.2}$. On the other hand, $t \not\models^W_{fair} [\ a\ \forall\mathcal{U}\ b\ ]_{\geq 1}$ while $t \not\models_{fair} [\ a\ \forall\mathcal{U}\ b\ ]_{\geq 1}$. (Note that $p^{adm}_x(a\mathcal{U}b) = 1$ for all $x \in S \setminus \{t\}$. Hence, $p^{adm}_t(a\mathcal{U}b) = 1$.) ■

9 Model checking for *PBTL**

Similarly to the way in which *CTL** extends *CTL* [24], our logic *PBTL* can be extended to a much richer logic *PBTL** which allows more complex path formulas, i.e. arbitrary

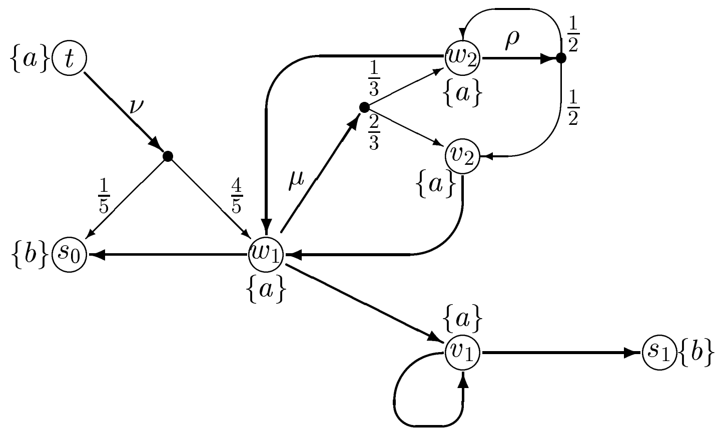


Figure 7:

combinations of path formulas by the boolean connectives and the operators X and $\mathcal{U}$. This logic (more precisely, the logic $pCTL^*$ which essentially coincides with $PBTL^*$) was already considered in [14]. In contrast to [14], where fairness is not treated, we also allow the interpretation of $PBTL^*$ over fair or strictly fair computations. [14] presents a model checking algorithm for $pCTL^*$ (corresponding to $PBTL^*$ with the standard interpretation $\models$) which uses the model checker for $pCTL$ and runs in time polynomial in the size of the concurrent probabilistic system and triply-exponential in the size of the formula. The main idea of [14] for reducing the complexity of the model checking problem for $pCTL^*$ to the model checking problem for $pCTL$ is the use of normal forms for path formulas. [4] proposes an alternative method, which is based on the representation of linear time formulas by ω-automata and runs in time polynomial in the size of the concurrent probabilistic system and doubly-exponential in the size of the formula. (Thus, the method of [4] is optimal by the results of [21].)

In this section, we briefly explain how the model checking procedure of [4] can be modified, thus yielding a model checking algorithm for $PBTL^*$ when interpreted by means of a satisfaction relation that involves fairness.

$PBTL^*$ consists of *state formulas* and *path formulas* which are recursively defined:

(1) tt and all atomic proposition are state formulas.
(2) If Φ, Φ_1, Φ_2 are state formulas then $\neg\Phi$ and $\Phi_1 \wedge \Phi_2$ are state formulas.
(3) If φ is a path formula, $p \in [0,1]$ then $[\ \exists\varphi\ ]_{\sqsupseteq p}$ and $[\ \forall\varphi\ ]_{\sqsupseteq p}$ are state formulas.
(4) If Φ is a state formula then Φ is a path formula.
(5) If φ, φ_1, φ_2 are path formulas then $\neg\varphi$, $\varphi_1 \wedge \varphi_2$, $X\varphi$ and $\varphi_1\mathcal{U}\varphi_2$ are path formulas.

Replacing (4) and (5) by

(6) If Φ, Φ_1, Φ_2 are state formulas then $X\Phi$ and $\Phi_1\mathcal{U}\Phi_2$ are path formulas.

we obtain the logic $PBTL$ (where e.g. the $PBTL^*$ formula $[\ \forall(\Phi_1\mathcal{U}\Phi_2)\ ]_{\sqsupseteq p}$ is identified with the $PBTL$ formula $[\ \Phi_1\ \forall\mathcal{U}\ \Phi_2\ ]_{\sqsupseteq p}$). As usual, $\Diamond\varphi\ =\ tt\,\mathcal{U}\,\varphi$ and $\Box\varphi\ =\ \neg\Diamond\neg\varphi$. Given a $PBTL$-structure $M = (\mathcal{S}, L)$ and a subset Adv of $\mathcal{A} = \mathcal{A}(\mathcal{S})$, the satisfaction relation $\models_{Adv}$ is defined in the obvious way. For instance, $s \models_{Adv} [\ \forall(\ X(\Phi_1\mathcal{U}\Phi_2)\ )\ ]_{\sqsupseteq p}$ iff $Prob\left\{\pi \in Path^A_{ful}(s) : \pi \models_{Adv} X(\Phi_1\mathcal{U}\Phi_2)\right\} \sqsupseteq p$ for all $A \in Adv$. Here, $\pi \models_{Adv} X(\Phi_1\mathcal{U}\Phi_2)$

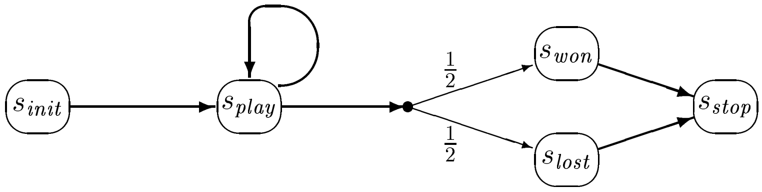


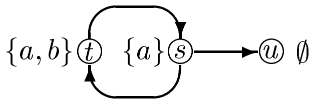
Figure 8: The roulette player

iff there is some $j \geq 1$ with $\pi(j) \models \Phi_2$ and $\pi(i) \models \Phi_1$ for all $1 \leq i < j$. As before, $\models$, $\models_{fair}$ and $\models_{sfair}$ denote the satisfaction relation w.r.t. $Adv = \mathcal{A}$, $\mathcal{A}_{fair}$, $\mathcal{A}_{sfair}$ respectively.

Example 9.1 Figure 8 shows a simplified roulette player who keeps placing bets until his wife comes to the casino (we assume that the player is infinitely rich). We use the atomic propositions *won* and *stop* together with the interpretation $L(s_{won}) = \{won\}$, $L(s_{stop}) = \{stop\}$ and $L(s_*) = \emptyset$ for the other states. The formula $\Phi = [\ \exists\varphi\ ]_{\geq 1/2}$ where $\varphi = \Diamond(won \wedge X stop)$ states that, with probability at least 1/2, the player leaves the casino having won the last game. Observe that the validity of Φ cannot be established unless fairness assumptions are made. Formally, $s_{init} \models_* \Phi$ where $*$ is *fair* or *sfair* while $s_{init} \not\models \Phi$. ■

Before we explain how the model checker of [4] for $pCTL^*$ w.r.t. $\models$ can be modified, thus yielding model checking procedures for $PBTL^*$ w.r.t. $\models_{fair}$ and $\models_{sfair}$, we give an example which shows that – in contrast to our results of Section 5 – an analysis of the simple adversaries is not sufficient when one uses $PBTL^*$.

Example 9.2 Let φ be the path formula $Xb \rightarrow \Box a$ and $\Phi = [\ \forall\varphi\ ]_{\geq 1}$. We consider the following *PBTL*-structure.



Then, $p_s^A(\varphi) = 1$ for all simple adversaries A, while $p_s^F(\varphi) = 0$ for the fair adversaries F with $F(s) = \mu_t$ and $F(\omega) = \mu_u$ for all paths ω with $last(\omega) = s$ and $|\omega| \geq 1$. This example also shows that, unlike in Theorem 2, a result stating that $\sup\{p_s^F(\varphi) : F \in \mathcal{A}_{fair}\} = \sup\{p_s^A(\varphi) : A \in \mathcal{A}\}$ cannot be established. ■

The model checker of [4] when applied to $PBTL^*$ w.r.t. $\models$ successively computes the set of states satisfying a state subformula of the given state formula Φ. Since $s \models_{Adv} [\ \forall\varphi\ ]_{\sqsupseteq p}$ iff $s \models_{Adv} \neg[\ \exists(\neg\varphi)\ ]_{\overline{\sqsupseteq} 1-p}$ (where $\overline{\geq} = >$ and $\overline{>} = \geq$) only the case where Φ has the form $\Phi = [\exists\varphi]_{\sqsupseteq p}$ is of interest. The state subformulas occurring in Φ can be viewed as atomic propositions. We briefly sketch how to modify the method of [4] in order to obtain a model checker for $PBTL^*$ w.r.t. $\models_{fair}$ and $\models_{sfair}$. This modification has been suggested by Luca de Alfaro [5].

We consider a state formula $\Phi = [\ \exists\varphi\ ]_{\sqsupseteq p}$ where we assume that the state subformulas $\Phi_1, \ldots, \Phi_k$ of φ are atomic propositions. As in [4], we use the results of [59, 54] for constructing a deterministic Rabin automaton $\mathsf{A} = (\mathsf{St}, \mathsf{q}_0, \mathsf{Alph}, \delta, \mathsf{AccCond})$ with state space St, initial state $\mathsf{q}_0 \in \mathsf{St}$, alphabet $\mathsf{Alph} = 2^{\{\Phi_1, \ldots, \Phi_k\}}$, transition relation $\delta : \mathsf{St} \times \mathsf{Alph} \rightarrow \mathsf{St}$ and acceptance condition $\mathsf{AccCond} = \{(\mathsf{H}_j, \mathsf{K}_j) : j = 1, \ldots, r\}$ such that the set $\mathsf{AccWords}(\mathsf{A})$ of accepted words over Alph is the set of words over Alph that satisfy φ (w.r.t. the usual satisfaction relation for linear time formulas built from the atomic propositions $\Phi_1, \ldots, \Phi_k$, the boolean connectives and the temporal operators X and $\mathcal{U}$). Formally, $\mathsf{AccWords}(\mathsf{A})$ is the set of words $\mathbf{a} = \mathsf{a}_0\mathsf{a}_1 \ldots$ over Alph such that for the induced word $\mathbf{q} = \mathsf{q}_0\mathsf{q}_1 \ldots$ over St (i.e. $\mathsf{q}_{i+1} = \delta(\mathsf{q}_i, \mathsf{a}_i)$ for all $i \geq 0$) and for some $j \in \{1, \ldots, r\}$: $\mathit{inf}(\mathbf{q}) \subseteq \mathsf{H}_j$ and $\mathit{inf}(\mathbf{q}) \cap \mathsf{K}_j \neq \emptyset$. ($\mathit{inf}(\mathbf{q})$ denotes the set of states $\mathsf{q} \in \mathsf{St}$ that occur infinitely often in $\mathbf{q}$.) The size of A is doubly-exponential in $|\varphi|$.

We construct a new *PBTL*-structure $M' = (\mathcal{S}', L')$, $\mathcal{S}' = (S', \mathit{Steps}')$, where $S' = S \times \mathsf{St}$, $\mu' \in \mathit{Steps}'(s, \mathsf{q})$ iff there exists $\mu \in \mathit{Steps}(s)$ with

$$\mu'(t, \mathsf{p}) \;=\; \begin{cases} \mu(t) & : \text{ if } \mathsf{p} = \delta(\mathsf{q}, L(s)) \\ 0 & : \text{ otherwise} \end{cases}$$

and $L'(s, \mathsf{q}) = L(s)$. We define $K'_j = S \times \mathsf{K}_j$, $H'_j = S \times \mathsf{H}_j$, $j = 1, \ldots, r$. We embed S into $S' = S \times \mathsf{St}$ as follows: for $s \in S$, let $\mathsf{q}_s = \delta(\mathsf{q}_0, L(s))$ and $s_{\mathsf{A}} = (s, \mathsf{q}_s)$. Each fulpath $\pi : s = s_0 \stackrel{\mu_1}{\to} s_1 \stackrel{\mu_2}{\to} \ldots$ in $\mathcal{S}$ induces a fulpath π' in $\mathcal{S}'$ which is given by:

$$s_{\mathsf{A}} \;=\; (s_0, \mathsf{p}_0) \stackrel{\mu'_1}{\to} (s_1, \mathsf{p}_1) \stackrel{\mu'_2}{\to} \ldots, \quad \mathsf{p}_0 = \mathsf{q}_s,\; \mathsf{p}_{i+1} = \delta(\mathsf{p}_i, L(s_i))$$

and where $\mu'_i(t, \mathsf{p}) = \mu_i(t)$ if $\mathsf{p} = \delta(\mathsf{p}_{i-1}, L(s_{i-1}))$, and otherwise $\mu'_i(t, \mathsf{p}) = 0$. The functions $\mathit{Path}_{\mathit{ful}}(s, \mathcal{S}) \to \mathit{Path}_{\mathit{ful}}(s_{\mathsf{A}}, \mathcal{S}')$, $\pi \mapsto \pi'$, yield a one-to-one correspondence between the fulpaths of $\mathcal{S}$ starting in s and the fulpaths of $\mathcal{S}'$ starting in s_{A} such that a fulpath π in $\mathcal{S}$ is fair (strictly fair) if and only if the corresponding path π' in $\mathcal{S}'$ is fair (strictly fair). Moreover, each fair (strictly fair) adversary F for $\mathcal{S}$ induces a fair (strictly fair) adversary F' for $\mathcal{S}'$ with $\mathit{Path}^{F'}_{\mathit{ful}}(s_{\mathsf{A}}) = \{\pi' : \pi \in \mathit{Path}^{F}_{\mathit{ful}}(s)\}$ and vice versa. Identifying each fulpath π in $\mathcal{S}$ with the corresponding fulpath π' in $\mathcal{S}'$, the associated probability spaces on $\mathit{Path}^{F}_{\mathit{ful}}(s)$ and $\mathit{Path}^{F'}_{\mathit{ful}}(s_{\mathsf{A}})$ coincide. For $j = 1, \ldots, r$, let U'_j be the union of all subsets T' of H'_j such that for all $t' \in T'$:

(1) $\mathit{Supp}(\mu') \subseteq T'$ for all $\mu' \in \mathit{Steps}'(t')$

(2) $\mathit{Reach}(t') \cap K'_j \neq \emptyset$

Let $U' = \bigcup_{1 \leq j \leq r} U'_j$ and let a'_U be an atomic propositions with $a'_U \in L'(s')$ iff $s' \in U'$. It is easy to see that for all $F' \in \mathcal{A}_{\mathit{fair}}(\mathcal{S}')$:

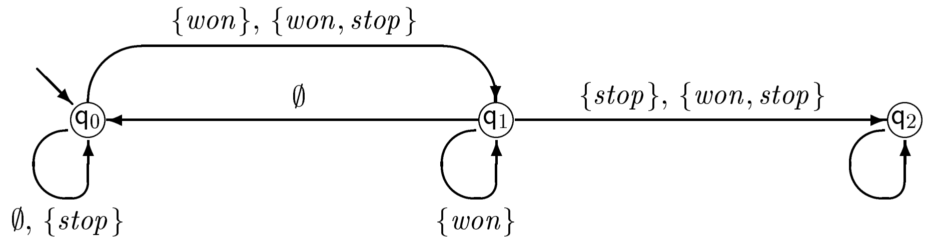
$$p^{F'}_{s'}(\Diamond a'_U) \;=\; \mathit{Prob}\{\pi' \in \mathit{Path}^{F'}_{\mathit{ful}}(s', \mathcal{S}') : \mathsf{word}(\pi') \in \mathsf{AccWords}(\mathsf{A})\}$$

where $\mathsf{word}(\pi') = L'(\pi'(0))L'(\pi'(1)) \ldots$. Hence, for all $s \in S$:

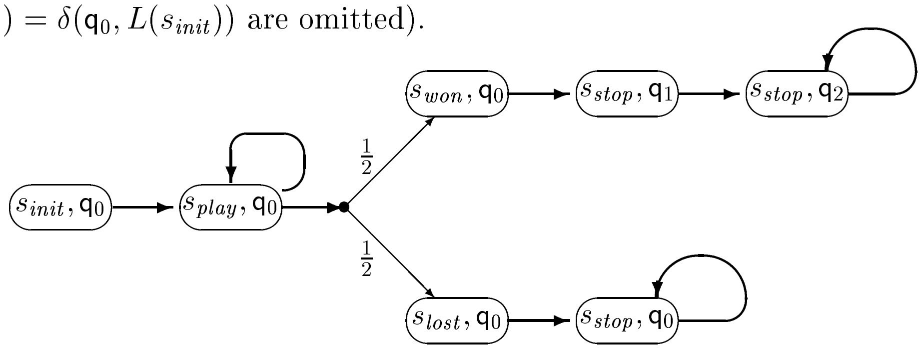
$$\begin{array}{lcl} s \;\models_{\mathit{fair}}\; [\,\exists\varphi\,]_{\sqsupseteq p} & \Longleftrightarrow & s_{\mathsf{A}} \;\models_{\mathit{fair}}\; [\,\exists\Diamond a'_U\,]_{\sqsupseteq p} \\ s \;\models_{\mathit{sfair}}\; [\,\exists\varphi\,]_{\sqsupseteq p} & \Longleftrightarrow & s_{\mathsf{A}} \;\models_{\mathit{sfair}}\; [\,\exists\Diamond a'_U\,]_{\sqsupseteq p} \end{array}$$

We obtain $\mathit{Sat}(\Phi)$ by computing the probabilities $p^{\mathit{max}}_{s_{\mathsf{A}}}(\Diamond a'_U)$ for the *PBTL*-structure M'. This can be done by means of the methods described in Section 5.

Example 9.3 We apply the method described above to the *PBTL*-structure of Example 9.1 and the formula $\Phi = [\forall\Diamond\varphi]_{\geq 1/2}$. The following Rabin automaton with acceptance condition $\mathsf{AccCond} = \{(\mathsf{St}, \{\mathsf{q}_0, \mathsf{q}_1\})\}$ accepts the words satisfying $\neg\Diamond\varphi = \neg\Diamond(\mathit{won} \wedge X\mathit{stop})$.



The *PBTL*-structure M' is of the following form (where states that are not reachable from $(s_{init}, \mathsf{q}_0) = \delta(\mathsf{q}_0, L(s_{init}))$ are omitted).



For the acceptance condition $(\mathsf{H}, \mathsf{K}) = (\mathsf{St}, \{\mathsf{q}_0, \mathsf{q}_1\})$ we get $H' = S'$ and $K' = \{(s, \mathsf{q}_i) : s \in S, i = 0, 1\}$. Thus, U' contains (s_{lost}, q_0) and (s_{stop}, q_0) but none of the other states shown in the picture above. We obtain $p^{F'}_{s_{init}}(\Diamond a'_U) = 1/2$ for each fair adversary F'. Hence, $s_{init} \not\models_{fair} [\exists \neg \Diamond \varphi]_{>1/2}$ which yields $s_{init} \models_{fair} [\forall \Diamond \varphi]_{\geq 1/2}$ (as stated in Example 9.1). ■

When dealing with the satisfaction relation $\models^W_{fair}$ the definition of U' has to be changed. We define $W' = W \times \mathsf{St}$ and $U' = \bigcup_{1 \leq j \leq r} U'_j$ where U'_j is the union of all subsets T' of H'_j such that for all $t' \in T'$:

(1) If $t' \in W'$, $\mu' \in \mathit{Steps}'(t')$ then $\mathit{Supp}(\mu') \subseteq T'$.

(2) If $t' \notin W'$ then there is some $\mu'_{t'} \in \mathit{Steps}'(t')$ with $\mathit{Supp}(\mu'_{t'}) \subseteq T'$.

(3) $\mathit{Reach}(t', \mathcal{S}'_0) \cap K'_j \neq \emptyset$ where $\mathcal{S}'_0 = (S', \mathit{Steps}'_0)$ is a concurrent probabilistic system such that:
   - If $t' \in (S' \setminus T') \cup (T' \cap W')$ then $\mathit{Steps}'_0(t') = \mathit{Steps}'(t')$.
   - If $t' \in T' \setminus W'$ then $\mathit{Steps}'_0(t') = \{\mu'_{t'}\}$ where $\mu'_{t'} \in \mathit{Steps}'(t')$ with $\mathit{Supp}(\mu'_{t'}) \subseteq T'$.

Let a'_U be an atomic proposition with $a'_U \in L'(s')$ iff $s' \in U'$. Then, $s \models^W_{fair} [\exists \varphi]_{\sqsupseteq p}$ iff $s_{\mathsf{A}} \models^W_{fair} [\exists \Diamond a'_U]_{\sqsupseteq p}$.

10 Related work

Apart from the temporal logic framework, probabilistic extensions of equivalence relations and preorders have been investigated and applied to the specification and verification of probabilistic processes; most are suitable for the formulation of qualitative and quantitative properties, see e.g. testing preorders [18, 62, 60, 39, 48, 47], bisimulation [41, 42, 36, 55, 10] and various kinds of simulations [38, 55, 61, 56, 57, 8].

Several authors presented verification methods for proving qualitative properties of concurrent probabilistic systems (e.g. [33, 49, 32, 58, 50, 59, 19, 2, 3, 51, 21]), but only a minority of them deal with fairness. [49, 50, 51] use fairness w.r.t. the probabilistic choices (extreme or α-fairness), rather than the non-deterministic choices, in order to verify qualitative properties of probabilistic proceses with non-probabilistic methods. [33] presents a decision procedure for checking that a given liveness property is fulfilled with probability 1 in all fair computation trees of a concurrent probabilistic system. [58] extends the results of [33] and presents a verification method for showing for a concurrent Markov chain whether or not a linear time formula holds with probability 1 for all fair computations. Our method is more general, as it allows to verify properties which hold with probability $\geq p$ for some $p \in [0, 1]$.

Verification methods for proving quantitative properties of probabilistic systems can be found in [20, 21, 31, 45, 52, 56, 6, 14, 4, 37, 34]. [21] investigates the complexity of model checking for (sequential and concurrent) probabilistic systems and presents an algorithm which tests whether a sequential Markov chain satisfies a linear time formula with probability 1. This algorithm computes the exact probabilities, and hence can also be adapted to the verification of quantitative properties of sequential probabilistic processes. [31, 6] deal with sequential Markov chains (and families of sequential Markov chains, called "generalized Markov chains" in [6]) as models for probabilistic systems and present model checking algorithms for probabilistic extensions of CTL. [37] gives a model checking algorithm for sequential Markov chains against probabilistic linear time specifications.

In contrast to the above *threshold-based* probabilistic extensions, which define a probabilistic formula as being true if the probability of the corresponding event is above the threshold level (e.g. is at least p for some p in the [0,1] interval), [34, 35] present a non-standard interpretation for the modal mu-calculus over sequential Markov chains with action labels. The operators $\mu x.\phi / \nu x.\phi$ are interpreted as least/greatest fixed points over the infinite lattice of maps from states to the unit interval, and a quantitative model checker for a fragment of the mu-calculus based on linear programming is given. Independently, [46] propose a similar interpretation of a temporal logic based on expectations and derive a suitable axiomatisation, but neither model checking nor fairness is considered.

The models for concurrent probabilistic processes used in [20] are based on concurrent Markov chains, and it is shown that the verification problem w.r.t. a specification given in the form of a set of ω-regular languages can be reduced to a linear programming problem (and thus yields a polynomial time method to verify concurrent probabilistic systems w.r.t. a specification expressed in some temporal logic). [14, 4] deal with a model similar to ours. The logic $pCTL$ considered in [14] agrees with our logic $PBTL$ (cf. Section 4), and is extended in [4] by an operator to express bounds on the average time between events. However, the issue of fairness is not treated in [14, 4]. [14] (and [4]) give model checking algorithms for $pCTL$ and $pCTL^*$ (and the extended versions). Although the logic $pCTL^*$ considered in [14] is capable of expressing fairness of execution sequences, formulas stating the existence or non-existence of (strictly) fair computations with certain properties cannot be expressed in $pCTL^*$. Hence, the model checking algorithm for $pCTL^*$ of [14] does not yield a method for proving quantitative properties of concurrent probabilistic systems assuming fairness of schedulers that resolve the non-deterministic choices.

[45, 52, 56] formulate proof rules for establishing quantitative (timed) progress properties for randomized distributed systems which can be combined with several notions of fairness, but model checking is not considered.

11 Concluding remarks and further directions

We have presented an algorithm for model checking of a probabilistic branching time logic $PBTL$ assuming fairness of choice, and also proposed modifications of existing model checking procedures for $PBTL^*$ [14, 4] to cater for fairness. The algorithms have applications in the specification and verification of concurrent probabilistic systems, for example, fault-tolerant systems or distributed systems with uncertainty. In many such systems it is essential to allow non-determinism, which arises through a scheduler or external intervention, as well as probabilistic choice. To the authors' knowledge, this is the first attempt to formulate an automatic method to verify quantitative properties of such systems which takes fairness into account.

Following [55] we consider a model of concurrent probabilistic systems which can be decomposed into a collection of "computation trees". These computation trees arise through selecting one of possibly many probability distributions available in a state, with the choices being made by the so called "adversaries", or "schedulers". We define fair and strictly fair adversaries by adapting Vardi's notion of fairness to our setting. The branching time quantifiers $\exists$ and $\forall$ range over adversaries. We consider three interpretations for $PBTL$: the first ($\models$) is standard (quantification is over all adversaries), while for the remaining two interpretations we restrict the class of adversaries to the fair adversaries ($\models_{fair}$) and the strictly fair adversaries ($\models_{sfair}$) respectively. While the difference between $\models_{fair}$ and $\models_{sfair}$ turns out to be only marginal (cf. Theorems 2-7), the difference between $\models$ and $\models_{fair}$ (or $\models_{sfair}$) reflects the familiar statement that certain liveness properties (e.g. the progress property in Section 6 which asserts that whenever the system is continuously able to send a message it will eventually send the message) can only be shown when appropriate fairness assumptions are made. We obtain, in our opinion, a surprising result: that to verify properties w.r.t. $\models_{fair}$ and $\models_{sfair}$ an examination of simple adversaries, known to be extremely unfair, suffices.

The algorithm presented here allows the verification of "quantitative" properties, i.e. those which state that something holds with probability $\geq$ or $>$ p for p in the interval $[0, 1]$. Our model checking method for $PBTL$ has the same time complexity as the one proposed in [14] (where fairness is not considered). Both run in time polynomial in the size of the system and linear in the size of the formula. The time complexity of the model checker for $PBTL^*$ depends on the size of the formula and the size of the model, and is in line with the expected complexity of verification for concurrent probabilistic processes [58, 21]. Model checking for $PBTL^*$ (with or without fairness assumptions) can be done in time doubly-exponential in the size of the formula and polynomial in the size of the system.

Alternatively, instead of computing the exact probability one could calculate lower and upper bounds on the probabilities of quantitative properties in the sense of [34, 46]. We expect this to be more efficient, although clearly at a cost of some loss of information. We investigate this issue in more detail separately [13]. Of course, these lower and upper bounds still apply when we range over fair (or strictly fair) adversaries, as opposed to all adversaries, but they cannot help in establishing (qualitative or quantitative) properties that are not satisfied unless fairness assumptions are made. Another direction worth pursuing is an investigation of whether the (multi-terminal) BDD method [17, 9] can be applied in this case to yield efficient procedures.

One possible use of our algorithm might be to test whether a probabilistic process meets its specification with a given probability p when the specification is given in the form of a collection of "canonical" formulas of a non-probabilistic logic such as CTL. For instance, given a canonical CTL safety formula $\forall\Box\varphi$ (where φ is a propositional formula) and a probabilistic process $\mathcal{P}$ – described by a concurrent probabilistic system $\mathcal{S}$ and an initial state – our method, applied to the formula $[\ \forall\Box\varphi\ ]_{\geq p}$ and one of the satisfaction relations $\models$, $\models_{fair}$ or $\models_{sfair}$, yields whether $\mathcal{P}$ satisfies $\forall\Box\varphi$ with probability p when the adversaries (or the environment in which $\mathcal{P}$ works) that resolve the non-deterministic choices in $\mathcal{P}$ obey the restrictions associated with the chosen satisfaction relation.

Finally, it is an open question whether satisfiability of $PBTL$ (or $PBTL^*$) is decidable w.r.t. any of the satisfaction relations, and – closely related to decidability – whether the synthesis of probabilistic processes fulfilling a given specification in the form of a satisfiable $PBTL$ formula can be performed automatically.

Acknowledgements

The authors are grateful to Luca de Alfaro, Roberto Segala and Ed Clarke for suggesting improvements to the paper.

12 Appendix: Proofs of Main Results

This section includes the proofs of Theorems 2-9 which we have used to derive the model checking procedure. We include them for the sake of completeness.

First, we explain some additional and simplified notations used throughout this section. We fix a $PBTL$-structure $M = (\mathcal{S}, L)$ where $\mathcal{S} = (S, Steps)$ and two $PBTL$ formulas Φ_1 and Φ_2 which we treat as atomic propositions. We use the following abbreviations in the proofs. We shortly write S^+ rather than $S^+(\Phi_1, \Phi_2)$, p_s^A instead of $p_s^A(\Phi_1\mathcal{U}\Phi_2)$, p_s^{max} instead of $p_s^{max}(\Phi_1\mathcal{U}\Phi_2)$, p_s^{adm} instead of $p_s^{adm}(\Phi_1\mathcal{U}\Phi_2)$, p_s^{min} instead of $p_s^{min}(\Phi_1\mathcal{U}\Phi_2)$. Similarly, we abbreviate $T^{max}(\Phi_1, \Phi_2)$ by T^{max}, $T_i^{max}(\Phi_1, \Phi_2)$ by T_i^{max}, $T^{adm}(\Phi_1, \Phi_2)$, resp. $T_i^{adm}(\Phi_1, \Phi_2)$, by T^{adm}, resp. T_i^{adm}, and $MaxSteps(s, \Phi_1, \Phi_2)$, resp. $AdmSteps(s, \Phi_1, \Phi_2)$, by $MaxSteps(s)$, resp. $AdmSteps(s)$.

We often use the following lemma which follows from the results of [14] (Corollary 20, part 1, in [14]) and yields Theorem 1.

Lemma 12.1 (cf. [14]) *There exist A^{max}, $A^{min} \in \mathcal{A}_{simple}$ with*

$$p_s^{A^{max}}(\Phi_1\mathcal{U}\Phi_2) \ \geq\ p_s^B(\Phi_1\mathcal{U}\Phi_2) \ \geq\ p_s^{A^{min}}(\Phi_1\mathcal{U}\Phi_2)$$

for all states $s \in S$ and all adversaries B.

Sometimes we shall need induction on the length of a shortest path through Φ_1-states leading to a Φ_2-state. In such cases we use the following notation:

Notation 12.2 *For $s \in S^+(\Phi_1, \Phi_2)$ we define $Success(s)$ to be the set of finite paths $\omega \in Path_{fin}(s, \mathcal{S})$ with $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \models \Phi_2$. We define $\|s\|$ to be the length of a shortest path in $Success(s)$.*

We extend the satisfaction relation $\models$ for $PBTL$ to path formulas of the form $\Box\Phi$ (where Φ is a $PBTL$ formula) by putting $\pi \models \Box\Phi$ iff $\pi(i) \models \Phi$ for all $i \geq 0$.

12.1 Fairness w.r.t. the probabilistic choices

We rely on a result established in [12] which states that when dealing with fairness w.r.t. the probabilistic choices (rather than the non-deterministic choices) then the set of fair paths has measure 1.

Definition 12.3 *Let π a fulpath in $\mathcal{S}$. π is called* state fair *iff for each $s \in S$ and $\mu \in Steps(s)$ such that $\pi(i) = s$, $step(\pi, i) = \mu$ for infinitely many i and each $t \in Supp(\mu)$, there are infinitely many indices j with $\pi(j) = s$, $step(\pi, j) = \mu$ and $\pi(j+1) = t$. π is called* total fair *iff π is fair and state fair.*

$StateFair$ denotes the set of state fair fulpaths in $\mathcal{S}$ and $TotalFair = Fair \cap StateFair$. State fairness can be viewed as a simplification of "extreme fairness" in the sense of [49] (cf. [12]).

Lemma 12.4 (cf. [12]) *$Prob(StateFair^A(s)) = 1$ for all adversaries A and $s \in S$.*

Lemma 12.5 *Let π a fulpath in $\mathcal{S}$. For $t \in inf(\pi)$, let $Steps'(t)$ be the set of distributions $\mu \in Steps(t)$ where $step(\pi, i) = \mu$ for infinitely many i. Then, $Reach(t, \mathcal{S}') = inf(\pi)$ for all $t \in inf(\pi)$ where $\mathcal{S}' = (inf(\pi), Steps')$. In particular:*

(a) If A is a simple adversary of $\mathcal{S}$, $\pi \in StateFair^A$, $s \in inf(\pi)$ then $Reach^A(s) = inf(\pi)$.

(b) If $\pi \in TotalFair$ and $s \in inf(\pi)$ then $Reach(s) = inf(\pi)$.

Proof: easy verification. ■

Lemma 12.6 *$Prob(TotalFair^F(s)) = 1$ for all $F \in \mathcal{A}_{fair}$ and $s \in S$.*

Proof: follows by Lemma 12.4 and the fact that $Prob(\Gamma \cap \Gamma') = 1$ if $Prob(\Gamma) = Prob(\Gamma') = 1$ (which holds in every probabilistic space). ■

12.2 The prefix relation on paths

Notation 12.7 *Let $\prec$ be the "proper prefix ordering" on (finite or infinite) paths, i.e. if ω, γ are paths then $\omega \prec \gamma$ iff $\omega \in Path_{fin}$ and $\omega = \gamma^{(i)}$ for some $i < |\gamma|$. We write $\omega \preceq \omega'$ iff either $\omega = \omega'$ or $\omega \prec \omega'$. Let $\gamma \downarrow$ be the set of paths ω with $\omega \preceq \gamma$. For A to be an adversary and $\omega \in Path_{fin}^A$, we define $\omega \uparrow^A$ to be the set of fulpaths $\pi \in Path_{ful}^A$ with $\omega \prec \pi$ and $\omega \uparrow_{fin}^A$ to be the set of finite paths $\omega' \in Path_{fin}^A$ with $\omega \prec \omega'$. For $\Omega \subseteq Path_{fin}^A$, we define $\Omega \uparrow^A = \bigcup_{\omega \in \Omega} \omega \uparrow^A$.*

We often use the following facts. Let A be an adversary and $\Omega \subseteq Path_{fin}^A$ such that $\omega, \omega' \in \Omega$, $\omega \neq \omega'$ implies $\omega \not\prec \omega'$. Then, the sets $\omega \uparrow^A$, $\omega \in \Omega$, are pairwise disjoint. Hence, $Prob\left(\Omega \uparrow^A (s)\right) = \sum_{\omega \in \Omega(s)} \mathbf{P}(\omega)$. If, in addition, Ω is finite, $k = \min\{|\omega| : \omega \in \Omega\}$ and $\Delta = \{\omega^{(k)} : \omega \in \Omega\}$ then $\bigcup_{\omega \in \Omega(s)} \omega \uparrow^A \subseteq \bigcup_{\delta \in \Delta(s)} \delta \uparrow^A$. Thus, $\sum_{\omega \in \Omega(s)} \mathbf{P}(\omega) \leq \sum_{\delta \in \Delta(s)} \mathbf{P}(\delta)$ for all $s \in S$. If $\Omega \subseteq Path_{fin}^A$ such that

- if $\omega \in \Omega$ then $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$,
- if $\omega \in \Omega$ then $\omega \uparrow^A \cap \{\pi \in Path_{ful}^A : \pi \models \Phi_1 \mathcal{U} \Phi_2\} \neq \emptyset$,

- each fulpath $\pi \in Path^A_{ful}$ with $\pi \models \Phi_1 \mathcal{U} \Phi_2$ has a unique prefix in Ω,

then $\sum_{\omega \in \Omega(s)} \mathbf{P}(\omega) \cdot p^A_\omega(\Phi_1 \mathcal{U} \Phi_2) = p^A_s(\Phi_1 \mathcal{U} \Phi_2)$. In particular, if Ω is the set of finite paths with $\omega(i) \models \Phi_1 \wedge \neg \Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \models \Phi_2$ then $\sum_{\omega \in \Omega(s)} \mathbf{P}(\omega) = p^A_s(\Phi_1 \mathcal{U} \Phi_2)$.

Lemma 12.8 *Let A be an adversary, $\Omega \subseteq Path^A_{fin}$ and k a positive integer with*

(i) $\omega \not\prec \omega'$ for all ω, $\omega' \in \Omega$

(ii) for each $\gamma \in Path^A_{fin}$ either $\omega \prec \gamma$ for some $\omega \in \Omega$ or there exists a finite path λ with $first(\lambda) = last(\gamma)$, $|\lambda| \leq k$ and $\gamma\lambda \in \Omega$.

Then, for all $s \in S$: $Prob\left(\Omega \uparrow^A (s)\right) \;=\; Prob\left(\bigcup_{\omega \in \Omega(s)} \omega \uparrow^A \right) \;=\; 1$.

Proof: Let $s \in S$ and $\Gamma = Path_{ful}(s) \setminus \Omega \uparrow^A (s)$. We show that $Prob(\Gamma) = 0$. If the path $\omega = s$ belongs to Ω then $\Gamma = \emptyset$. Now we suppose that the path s does not belong to Ω. Let Δ be the set of paths $\delta \in Path^A_{fin}(s)$ with $\omega \not\prec \delta$ for all $\omega \in \Omega$. Then, $\Omega \subseteq \Delta$. (Recall that $\prec$ denotes the "proper" prefix relation.) Let $c = \min \{\mu(t) : \mu \in \bigcup_{s \in S} Steps(s), t \in Supp(\mu)\}$. Note that – since $\bigcup_s Steps(s)$ and S are finite – c is well-defined and $c > 0$. For each $\delta \in \Delta$ let $p_\delta = Prob\left(\Gamma \cap \delta \uparrow^A \right) / \mathbf{P}(\delta)$ and let $\kappa(\delta)$ be the length of a shortest path $\lambda \in Path_{fin}$ with $first(\lambda) = last(\delta)$ and $\delta\lambda \in \Omega$. Then, $\kappa(\delta) \leq k$ for all $\delta \in \Delta$. We show that $\sup\{p_\delta : \delta \in \Delta\} = 0$. (This yields $Prob(\Gamma) = p_s = 0$ (since $s \in \Delta$).)

Let $p = \sup\{p_\delta : \delta \in \Delta\}$. We show by induction on l that $p_\delta \;\leq\; (1 - c^l) \cdot p$ for all $\delta \in \Delta$ with $\kappa(\delta) = l$. In the basis of induction ($l = 0$) we get $p_\delta = 0$ since $\kappa(\delta) = 0$ implies $\delta \in \Omega$. Let $l \geq 1$ and $\delta \in \Delta$ with $\kappa(\delta) = l$ and let $\mu = A(\delta)$. For $u \in Supp(\mu)$, let $\delta_u = \delta \xrightarrow{\mu} u$. There exists $t \in Supp(\mu)$ with $\kappa(\delta_t) = l - 1$. Then, by induction hypothesis: $p_{\delta_t} \leq (1 - c^{l-1}) \cdot p$. With $U = Supp(\mu) \setminus \{t\}$ we have $\sum_{u \in U} \mu(u) \;=\; 1 - \mu(t)$. Since $\mu(t) \geq c$ we obtain:

$$\begin{aligned} p_\delta \;&=\; \sum_{u \in U \cup \{t\}} \mu(u) \cdot p_{\delta_u} \;\;\leq\; p \cdot \sum_{u \in U} \mu(u) \;+\; \mu(t) \cdot p_{\delta_t} \\ &\leq\; p \cdot \left(1 - \mu(t) \cdot c^{l-1}\right) \;\leq\; p \cdot (1 - c^l). \end{aligned}$$

Since $\kappa(\delta) \leq k$ for all $\delta \in \Delta$ we get $p \;\leq\; (1 - c^k) \cdot p$ and hence $p = 0$. ■

Remark 12.9 Let A be an adversary, $\Omega \subseteq Path^A_{fin}$ and k a positive integer with

(I) $\omega \not\prec \omega'$ for all ω, $\omega' \in \Omega$

(II) for each $\gamma \in Path_{fin}$ with $\gamma \prec \omega$ for some $\omega \in \Omega$ there exists a finite path λ with $first(\lambda) = last(\gamma)$, $|\lambda| \leq k$ and $\gamma\lambda \in \Omega$.

There exists a set Ω' with $\Omega \subseteq \Omega' \subseteq Path^A_{fin}$ and which satisfies the conditions (i) and (ii) of Lemma 12.8. For instance, $\Omega' \;=\; \Omega \cup \Lambda$ satisfies the conditions (i) and (ii) of Lemma 12.8 where Λ is the set of finite paths $\lambda \in Path^A_{fin}$ with $\lambda \notin \omega \downarrow$ and $\omega \not\prec \lambda$ for all $\omega \in \Omega$ and $\lambda^{(i)} \in \Omega \downarrow$ for all $i < |\lambda|$. (Here $\Omega \downarrow = \bigcup_{\omega \in \Omega} \omega \downarrow$.)

If B is simple, S_1, $S_2 \subseteq S$ and Ω the set of finite paths $\omega \in Path^B_{fin}$ such that $\omega(i) \in S_1 \setminus S_2$, $i < |\omega|$, and $last(\omega) \in S_2$ then Ω satisfies the conditions (I) and (II) of above. In Section 12.4 and 12.5 we use this fact with $S_1 = Sat(\Phi_1)$ and $S_2 = Sat(\Phi_2)$ resp. $S_2 = S \setminus S^+(\Phi_1, \Phi_2)$. ■

12.3 Extension of finite behaviour to (strictly) fair adversaries

In the proofs of Theorem 2, 3, 5 and 6 (Section 12.4 and 12.5) we show that the probabilities $p^{max}_s(\Phi_1 \mathcal{U} \Phi_2)$, resp. $p^{adm}_s(\Phi_1 \mathcal{U} \Phi_2)$, can be approximated by certain (strictly) fair adversaries.

We define these (strictly) fair adversaries with the help of the following lemma:

Lemma 12.10 *Let $A \in \mathcal{A}$, $\Omega \subseteq Path_{fin}^{A}$ and $k \geq 1$.*

(a) If Ω satisfies the conditions (I) and (II) of Remark 12.9 then there exists $F \in \mathcal{A}_{fair}$ with $\Omega \subseteq Path_{fin}^{F}$.

(b) If Ω is finite then there exists $F \in \mathcal{A}_{sfair}$ such that $\Omega \subseteq Path_{fin}^{F}$.

Proof: We may suppose that in both cases both conditions in (a) are satisfied. (Note that in case (b) we may deal with $\Omega' = \{\omega \in \Omega : \omega \not< \omega' \text{ for all } \omega' \in \Omega\}$ rather than Ω). For each state state $s \in S$ we choose an enumeration $\nu_0^s, \ldots, \nu_{k_s-1}^s$ of $Steps(s)$. We define a fair adversary F as follows. If $\sigma \in Path_{fin}$ is a prefix of some $\omega \in \Omega$ then we define $F(\sigma) = A(\sigma)$. For every path $\sigma \in Path_{fin}$ which has no prefix in Ω we define $F(\sigma) = \nu_j^s$ where $s = last(\sigma)$, $j = r \bmod k_s$ and r the number of indices $i < |\sigma|$ with $\sigma(i) = s$. (Here mod denotes the "modulo-division" function.) As $\Omega \in Path_{fin}^{A}$ we have if $\omega \in \Omega$ then $step(\omega, i) = A(\omega^{(i)}) = F(\omega^{(i)})$ for all $i < |\omega|$. Hence, $\omega \in Path_{fin}^{F}$. It is easy to see that if Ω is finite then F is strictly fair. Using Remark 12.9 and Lemma 12.8 it can be shown that F is fair. ■

Remark 12.11 In Lemma 12.10(a) we cannot ensure the existence of a strictly fair adversary F with $\Omega \subseteq Path_{fin}^{F}$. For instance, consider the fair adversary A of Example 5.6 and the set $\Omega = \{\omega \in Path_{fin}^{A}(s) : last(\omega) = u, \omega(i) \neq u,\ i = 0, 1, \ldots |\omega| - 1\}$. Then, for each adversary F with $\Omega \subseteq Path_{fin}^{F}$ we have if $last(\omega) = s$ then $F(\omega) = \mu$. Hence, F contains the unfair fulpath $s \xrightarrow{\mu} t \xrightarrow{\mu_s} s \xrightarrow{\mu} t \xrightarrow{\mu_s} \ldots$, i.e. F cannot be strictly fair. ■

12.4 Proof of Theorem 2 and 3

By Lemma 12.1 and 12.12(b), $\sup\left\{p_s^F(\Phi_1 \mathcal{U} \Phi_2) : F \in \mathcal{A}_{sfair}\right\} = p_s^{max}(\Phi_1, \Phi_2)$. This yields Theorem 3. Lemma 12.12(a) and 12.1 yield the existence of $A \in \mathcal{A}$, $F \in \mathcal{A}_{fair}$ with

$$\sup\left\{p_s^F(\Phi_1 \mathcal{U} \Phi_2) : F \in \mathcal{A}_{fair}\right\} = p_s^{max}(\Phi_1 \mathcal{U} \Phi_2) = p_s^A(\Phi_1 \mathcal{U} \Phi_2) \leq p_s^F(\Phi_1 \mathcal{U} \Phi_2).$$

Hence, $p_s^{max}(\Phi_1 \mathcal{U} \Phi_2) = \max\left\{p_s^F(\Phi_1 \mathcal{U} \Phi_2) : F \in \mathcal{A}_{fair}\right\}$ from which Theorem 2 follows.

Lemma 12.12 *For each $A \in \mathcal{A}_{simple}$ there exist*

(a) $F \in \mathcal{A}_{fair}$ with $p_s^A(\Phi_1 \mathcal{U} \Phi_2) \leq p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S$

(b) a sequence $(F_k)_{k \geq 1}$ of strictly fair adversaries F_k such that for all $s \in S$:

$$p_s^A(\Phi_1 \mathcal{U} \Phi_2) \leq \sup_{k \geq 1} p_s^{F_k}(\Phi_1 \mathcal{U} \Phi_2).$$

Proof: Let A be a simple adversary and let Ω be the set of finite paths $\omega \in Path_{fin}^{A}$ with $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, , \ldots, |\omega| - 1$, and $last(\omega) \models \Phi_2$. Let $\Omega_k = \{\, \omega \in \Omega : |\omega| \leq k \,\}$. Let F be a fair adversary with $\Omega \subseteq Path_{fin}^{F}$ (Lemma 12.10(a)). Then, $p_s^A = \sum_{\omega \in \Omega(s)} \mathbf{P}(\omega) \leq p_s^F$. Let F_k be a strictly fair adversary with $\Omega_k \subseteq Path_{fin}$ (Lemma 12.10(b)). Then, for all $s \in S$, $p_s^{F_k} \geq \sum_{\omega \in \Omega_k(s)} \mathbf{P}(\omega)$. Hence, $p_s^A = \sum_{\omega \in \Omega(s)} \mathbf{P}(\omega) = \sup_{k \geq 1} \sum_{\omega \in \Omega_k(s)} \mathbf{P}(\omega) \leq \sup_{k \geq 1} p_s^{F_k}$. ■

12.5 Proof of Theorem 5 and 6

By Lemma 12.15 and 12.25 we obtain the existence of $\min\{p_s^F : F \in \mathcal{A}_{fair}\}$ and that

$$\inf\left\{p_s^F(\Phi_1 \mathcal{U} \Phi_2) : F \in \mathcal{A}_{sfair}\right\} = \min\left\{p_s^F(\Phi_1 \mathcal{U} \Phi_2) : F \in \mathcal{A}_{fair}\right\} = p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2).$$

This yields Theorem 5 and 6.

Definition 12.13 *A fulpath π is called* critical *w.r.t. (Φ_1, Φ_2) iff $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$ and $inf(\pi) \cap S^+(\Phi_1, \Phi_2) \neq \emptyset$. Let $Critical_{(\Phi_1,\Phi_2)}(s)$ be the set of critical paths in $\mathcal{S}$ starting in s. For $A \in \mathcal{A}$, let $Critical^A_{(\Phi_1,\Phi_2)} = Critical_{(\Phi_1,\Phi_2)} \cap Path^A_{ful}$.*

$Critical^A_{(\Phi_1,\Phi_2)}(s)$ is measurable. Here, we use the fact that $Critical^A_{(\Phi_1,\Phi_2)}(s)$ is the set of paths $\pi \in Path^A_{ful}(s)$ with $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2) \wedge \Diamond\Box a^+$ where a^+ is a "new" atomic proposition with $a^+ \in L(s)$ iff $s \in S^+(\Phi_1, \Phi_2)$. See e.g. [51], where the measurability of sets of paths fulfilling a certain a formula of a linear time logic is shown.

Lemma 12.14 *Let $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$ and $s \in S$. Then, $Prob(\ Critical^A_{(\Phi_1,\Phi_2)}(s)\ ) = 0$.*

Proof: Since A is simple, MC^A can be identified with the finite-state Markov chain (S, A) (where A is identified with the function $S \times S \to [0,1]$, $(s,t) \mapsto A(s)(t)$). We show that every critical fulpath $\pi \in Critical^A_{(\Phi_1,\Phi_2)}$ is not state fair (w.r.t. state fairness in the sense of Section 12.1). Let $\pi \in Critical^A_{(\Phi_1,\Phi_2)}$ and $s \in inf(\pi) \cap S^+(\Phi_1, \Phi_2)$. As A is admissible there exists a state $u_s \in Reach^A_{\Phi_1 \wedge \neg\Phi_2}(s)$ with $u_s \in Sat(\Phi_2) \cup (S \setminus S^+(\Phi_1, \Phi_2))$. If we suppose π to be state fair then we have $Reach^A(s) \subseteq inf(\pi)$ (by Lemma 12.5(a)), and hence, $u_s \in inf(\pi)$. Since all states of π fulfill $\Phi_1 \wedge \neg\Phi_2$ the case $u_s \in Sat(\Phi_2)$ is impossible. Since s and u_s occur infinitely often on π there is a fragment ω of π leading from u_s to s. Hence, there is path $\omega \in Path_{fin}$ with $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i \leq |\omega|$ and $first(\omega) = u_s$, $last(\omega) = s$. Since $s \in S^+(\Phi_1, \Phi_2)$ we get $u_s \in S^+(\Phi_1, \Phi_2)$. Contradiction. We conclude $Critical^A_{(\Phi_1,\Phi_2)}(s) \subseteq Path^A_{ful}(s) \setminus StateFair^A(s)$ and we obtain $Prob(Critical^A_{(\Phi_1,\Phi_2)}(s)) = 0$ by Lemma 12.4. ■

Lemma 12.15 *Let $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$. Then, there exist*

(a) a fair adversary F with $p_s^A(\Phi_1 \mathcal{U} \Phi_2) \geq p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S$

(b) a sequence $(F_k)_{k \geq 1}$ of strictly fair adversaries such that for all $s \in S$:

$$p_s^A(\Phi_1 \mathcal{U} \Phi_2) \geq \inf_{k \geq 1} p_s^{F_k}(\Phi_1 \mathcal{U} \Phi_2)$$

Proof: We only use the fact that A is a simple adversary with $Prob(Critical^A_{(\Phi_1,\Phi_2)}(s)) = 0$ for all $s \in S$ (Lemma 12.14). We use the following notation for all adversaries B:

- $\Gamma^B = \{\ \pi \in Path^B_{ful} : \pi \not\models \Phi_1 \mathcal{U} \Phi_2\ \}$, $q_s^B = Prob(\Gamma^B(s)) = 1 - p_s^B$
- $\Omega^B = \{\omega \in Path^B_{fin} : \omega(i) \models \Phi_1 \wedge \neg\Phi_2, i = 0, 1, \ldots, |\omega| - 1, last(\omega) \in S \setminus S^+\}$
- $\Omega^B_k = \{\omega \in \Omega^B : |\omega| \leq k\}$, $\Gamma^B_1 = \bigcup_{\omega \in \Omega^B} \omega \uparrow^B$ and $\Gamma^B_2 = \Gamma^B \setminus \Gamma^B_1$

Then, Γ^B_2 is the set of fulpaths $\pi \in Path^B_{ful}$ with $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$. It is easy to see that $\Gamma^A_2 \subseteq Critical^A_{(\Phi_1,\Phi_2)}$. As Γ^A_2 is a measurable subset of $Critical^A_{(\Phi_1,\Phi_2)}$ we have $Prob(\Gamma^A_2) = 0$. Hence:

$$(*) \quad q_s^A = Prob(\Gamma^A(s)) = \sum_{\omega \in \Omega^A(s)} \mathbf{P}(\omega) = \sup_{k \geq 1} \sum_{\omega \in \Omega^A_k(s)} \mathbf{P}(\omega)$$

Let F be a fair adversary with $\Omega^A \subseteq \mathit{Path}_{\mathit{fin}}^F$ and let F_k be a strictly fair adversary with $\Omega_k^A \subseteq \mathit{Path}_{\mathit{fin}}^{F_k}$ (Lemma 12.10). We show that $p_s^F \le p_s^A$ and $\inf_{k\ge1} p_s^{F_k} \le p_s^A$ for all $s \in S$. $\Gamma^F(s)$ is a superset of the set of paths $\pi \in \mathit{Path}_{\mathit{ful}}^F(s)$ which have a prefix in Ω^A. By (*), $q_s^F \ \ge\ \sum_{\omega\in\Omega^A(s)} \mathbf{P}(\omega) \ =\ q_s^A$. We conclude $p_s^A \ =\ 1 - q_s^A \ \ge\ 1 - q_s^F \ =\ p_s^F$. Similarly, $\Gamma^{F_k}(s)$ is a superset of the set of paths $\pi \in \mathit{Path}_{\mathit{ful}}^{F_k}(s)$ which have a prefix in Ω_k^A. Hence, $q_s^{F_k} \ge \sum_{\omega\in\Omega_k^A(s)} \mathbf{P}(\omega)$. By (*),

$$q_s^A \ =\ \sup_{k\ge1} \sum_{\omega\in\Omega_k^A(s)} \mathbf{P}(\omega) \ \le\ \sup_{k\ge1} \mathit{Prob}(\Gamma^{F_k}(s)) \ =\ \sup_{k\ge1} q_s^{F_k}.$$

Therefore, $p_s^A \ =\ 1 - q_s^A \ \ge\ 1 - \sup_{k\ge1} q_s^{F_k} \ =\ \inf_{k\ge1} p_s^{F_k}$ for all $s \in S$. ■

Remark 12.16 [33] establishes a result stating that the measure induced by a strictly fair adversary can be approximated by the measures induced by fair adversaries (cf. Proposition 2.3 of [33]). Using this result (adapted to our notion of fairness), part (b) of Lemma 12.12 and Lemma 12.15 can also be derived from part (a) of Lemma 12.12, resp. 12.15. ■

Lemma 12.17 *Let $A \in \mathcal{A}_{simple}$ and $U \subseteq S$ with*

- $\mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^A(u) \ \subseteq\ \mathit{Sat}(\Phi_2) \cup U$
- $\mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^A(u) \cap \mathit{Sat}(\Phi_2) \ \ne\ \emptyset$

for all $u \in U$. Then, $p_u^A(\Phi_1\mathcal{U}\Phi_2) \ =\ 1$ for all $u \in U$.

Proof: W.l.o.g. $U \cap \mathit{Sat}(\Phi_2) = \emptyset$ (otherwise we deal with $U' = U \setminus \mathit{Sat}(\Phi_2)$). By the results of [31], the vector $(p_u^A)_{u\in U}$ is the unique solution of the equation system

$$(*) \ \ x_u \ =\ \sum_{v\in U} A(u,v)\cdot x_v \ +\ \sum_{s\in \mathit{Sat}(\Phi_2)} A(u,s), \quad u \in U.$$

(The uniqueness of the solution is guaranteed by the second assumption.) The first assumption implies $\sum_{v\in U\cup \mathit{Sat}(\Phi_2)} A(u,v) \ =\ 1$. Hence, the vector $(x_u)_{u\in U}$ with $x_u = 1$ for all $u \in U$ solves (*). By the uniqueness of the solution we get $p_u^A \ =\ 1$ for all $u \in U$. ■

Corollary 12.18 *Let $A \in \mathcal{A}_{simple}$, $s \in S$ with $p_s^A(\Phi_1\mathcal{U}\Phi_2) < 1$ and $p_u^A(\Phi_1\mathcal{U}\Phi_2) = p_u^{max}(\Phi_1\mathcal{U}\Phi_2)$ for all $u \in \mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^A(s)$. Then, $\mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^A \cap (S \setminus S^+(\Phi_1,\Phi_2)) \ne \emptyset$.*

Proof: Let $U = \mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^A(s)$. We suppose $U \subseteq S^+(\Phi_1,\Phi_2)$. Then, $p_u^A = p_u^{max} > 0$ for all $u \in U$. Thus, $\mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^A(u) \cap \mathit{Sat}(\Phi_2) \ne \emptyset$ and $p_s^A = 1$. (Lemma 12.17). Contradiction. ■

Lemma 12.19 *Let $F \in \mathcal{A}_{fair}$. Then, $\mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^F(t) \cap (S \setminus S^+(\Phi_1,\Phi_2)) \ne \emptyset$ for all $t \in S^+(\Phi_1,\Phi_2)$ with $p_t^F(\Phi_1\mathcal{U}\Phi_2) < 1$.*

Proof: We suppose $\mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^F(t) \cap (S \setminus S^+) = \emptyset$ for some $t \in S^+$ with $p_t^F < 1$. Let Γ be the set of all fulpaths $\pi \in \mathit{Path}_{\mathit{ful}}^F(t)$ with $\pi \not\models \Phi_1\mathcal{U}\Phi_2$. Then, $\mathit{Prob}(\Gamma) > 0$. By Lemma 12.6, $\mathit{Prob}(\mathit{TotalFair}^F(t)\cap\Gamma) = \mathit{Prob}(\Gamma) > 0$. Hence, $\mathit{TotalFair}^F(t)\cap\Gamma \ne \emptyset$. Let $\pi \in \mathit{TotalFair}^F(t)\cap\Gamma$ and $s \in \mathit{inf}(\pi)$. Since $\mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^F(t) \subseteq S^+$ we get by induction on i: $\pi(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i \ge 0$. By Lemma 12.5(b): $\mathit{Reach}(s) \subseteq \mathit{inf}(\pi)$. Hence, $\mathit{Reach}(s) \cap \mathit{Sat}(\Phi_2) = \emptyset$, and therefore $s \notin S^+$. Thus, $s \in \mathit{Reach}_{\Phi_1\wedge\neg\Phi_2}^F(t) \cap (S \setminus S^+)$. Contradiction. ■

Notation 12.20 *Let $S^0(\Phi_1,\Phi_2) \ =\ \{\ s \in S \ :\ p_s^F(\Phi_1\mathcal{U}\Phi_2) = 0$ for some $F \in \mathcal{A}_{fair}\ \}$.*

When Φ_1, Φ_2 are clear from the context, we often write S^0 instead of $S^0(\Phi_1, \Phi_2)$.

Lemma 12.21 *If $F \in \mathcal{A}_{fair}$ and $Reach_{\Phi_1 \wedge \neg\Phi_2}(s) \cap S^0(\Phi_1, \Phi_2) = \emptyset$ then $p_s^F(\Phi_1 \mathcal{U} \Phi_2) = 1$.*

Proof: We suppose $p_s^F < 1$. Then, $Reach^F_{\Phi_1 \wedge \neg\Phi_2}(s) \cap (S \setminus S^+) \neq \emptyset$ by Lemma 12.19. As $S \setminus S^+ \subseteq S^0$ we obtain a contradiction. ■

Lemma 12.22 *For each $F \in \mathcal{A}_{fair}$ there exists $F' \in \mathcal{A}_{fair}$ with*

- $p_s^F(\Phi_1 \mathcal{U} \Phi_2) \geq p_s^{F'}(\Phi_1 \mathcal{U} \Phi_2)$ *for all $s \in S$*
- $p_\omega^{F'}(\Phi_1 \mathcal{U} \Phi_2) = 0$ *for all $\omega \in Path^{F'}_{fin}$ with $\omega(i) \notin S^0(\Phi_1, \Phi_2)$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \in S^0(\Phi_1, \Phi_2)$.*

Proof: Let $F \in \mathcal{A}_{fair}$ and let Ω be the set of finite paths $\omega \in Path_{fin}$ with $\omega(i) \notin S^0$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \in S^0$. For each $s \in S^0$, let $F_s \in \mathcal{A}_{fair}$ such that $p_s^{F_s} = 0$. We define F' as follows: $F'(\gamma) = F_s(\lambda)$ if $\gamma = \omega\lambda$, $\omega \in \Omega$ and $F'(\gamma) = F(\gamma)$ in all other cases. It is easy to see that $F' \in \mathcal{A}_{fair}$, $p_s^{F'} \leq p_s^F$ for all $s \in S$ and $p_\omega^{F'} = p_{last(\omega)}^{F_{last(\omega)}} = 0$ for all $\omega \in \Omega$. ■

Lemma 12.23 *There exists $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$ with $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(t) \subseteq S^0(\Phi_1, \Phi_2)$ for all $t \in S^0(\Phi_1, \Phi_2)$. (Hence, $p_t^A(\Phi_1 \mathcal{U} \Phi_2) = 0$ for all $t \in S^0(\Phi_1, \Phi_2)$.)*

Proof: By Lemma 12.22, there exists $F \in \mathcal{A}_{fair}$ with $p_\omega^F = 0$ for all $\omega \in Path_{fin}$ with $\omega(i) \notin S^0$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \in S^0$. Let $\Omega^0_{\Phi_1 \wedge \neg\Phi_2}$ be the set of paths $\omega \in Path^F_{fin}$ such that $first(\omega) \in S^0$ and $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i < |\omega|$. For $s \in S \setminus (S^0 \cap Sat(\Phi_1))$, let $Steps^F(s) = Steps(s)$. For $s \in S^0 \cap Sat(\Phi_1)$, $Steps^F(s) = \{step(\omega, i) : i < |\omega|, \omega \in \Omega^0_{\Phi_1 \wedge \neg\Phi_2}, \omega(i) = s\}$. Let $\mathcal{S}^F = (S, Steps^F)$. Then, $Reach(t, \mathcal{S}^F) \subseteq S^0$ for all $t \in S^0$, In particular:

(*) If $A \in \mathcal{A}_{simple}$ with $A(t) \in Steps^F(t)$ for all $t \in S^0$ then $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(t) \subseteq S^0$ and $p_t^A = 0$ for all $t \in S^0$.

If $t \in S^0 \cap S^+$ then we define Ω_t to be the set of paths $\omega \in Path_{fin}(t, \mathcal{S}^F)$ with $last(\omega) \in S \setminus S^+$ and $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$. Lemma 12.19 yields $\Omega_t \neq \emptyset$ for all $t \in S^0 \cap S^+$. For $t \in S^0 \cap S^+$, let $\kappa(t)$ be the length of shortest path in Ω_t. Then, $\kappa(t) \geq 1$. We choose some $\omega_t \in \Omega_t$ with $\kappa(t) = |\omega_t|$ and put $\mu_t = step(\omega_t, 0)$. Then:

(**) If $A \in \mathcal{A}_{simple}$ with $A(t) = \mu_t$ for all $t \in S^0 \cap S^+$ then $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(t) \cap (S \setminus S^+) \neq \emptyset$ for all $t \in S^0 \cap S^+$.

If $s \in S \setminus S^+$ then we choose some $\mu_s \in Steps^F(s)$. For $s \in S^+ \setminus S^0$ we define a distribution $\mu_s \in Steps(s)$ by induction on $\|s\|$: If $\|s\| = 0$ then $s \in Sat(\Phi_2)$ and we choose an arbitrary $\mu_s \in Steps(s)$. If $\|s\| \geq 1$ then we choose some $\mu_s \in Steps(s)$ such that $\mu_s(w) > 0$ for some $w \in S^+$ with $\|w\| < \|s\|$. Let A be the simple adversary with $A(s) = \mu_s$ for all $s \in S$. Then, $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(s) \cap (Sat(\Phi_2) \cup (S^0 \cap S^+)) \neq \emptyset$ for all $s \in S^+ \setminus S^0$. By (**), $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(s) \cap (Sat(\Phi_2) \cup (S \setminus S^+)) \neq \emptyset$ for all $s \in S^+$. This yields $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$. By (*), it follows that $p_t^A = 0$ for all $t \in S^0$. ■

Notation 12.24 *Let $S^1(\Phi_1, \Phi_2) = \{s \in S : Reach_{\Phi_1 \wedge \neg\Phi_2}(s) \cap S^0(\Phi_1, \Phi_2) = \emptyset\}$ and $S^?(\Phi_1, \Phi_2) = S \setminus (S^1(\Phi_1, \Phi_2) \cup S^0(\Phi_1, \Phi_2))$.*

In what follows, we briefly write S^1 rather than $S^1(\Phi_1, \Phi_2)$ and $S^?$ rather than $S^?(\Phi_1, \Phi_2)$. We assume that there are atomic propositions $a^?$, a^0 and a^1 with $a^* \in L(s)$ iff $s \in S^*$, $* \in \{?, 0, 1\}$.

Lemma 12.25 *There exists $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$ with*

$$1 - p_s^{max}(a^? \mathcal{U} a^0) \;=\; p_s^A(\Phi_1 \mathcal{U} \Phi_2) \;\leq\; p_s^F(\Phi_1 \mathcal{U} \Phi_2)$$

for all $s \in S$ and $F \in \mathcal{A}_{fair}$.

Proof: Clearly, $S^1 \subseteq S^+$ and $Reach_{\Phi_1 \wedge \neg \Phi_2}(v) \subseteq S^1$ for all $v \in S^1$. Hence, $\{s \in S : \mu(s) > 0, \mu \in Steps(v)$ for some $v \in S^1\} \subseteq S^1$. By Lemma 12.21:

(1) $p_v^F = 1$ for all $v \in S^1$ and $F \in \mathcal{A}_{fair}$.

For $v \in S^1$ we define a distribution $\mu_v \in Steps(v)$ by induction on $\|v\|$: If $\|v\| = 0$ then we choose some $\mu_v \in Steps(v)$. If $\|v\| \geq 1$ then there exists $\mu_v \in Steps(v)$ with $\mu_v(w) > 0$ for some $w \in S^1$ with $\|v\| > \|w\|$. Then:

(2) If $A \in \mathcal{A}_{simple}$ with $A(v) = \mu_v$ for all $v \in S^1$ then $p_v^A = 1$ for all $v \in S^1$. In particular, $Reach^A_{\Phi_1 \wedge \neg \Phi_2}(v) \cap Sat(\Phi_2) \neq \emptyset$ for all $v \in S^1$.

Let $\mathcal{S}' = (S, Steps')$ where $\mu \in Steps'(s)$ iff either $s \in S^?$ and $\mu \in Steps(s)$ or $s \notin S^?$ and $\mu = \mu_s^1$. We consider L as a labelling function for $\mathcal{S}$ and as a labelling function for $\mathcal{S}'$. Let $A' \in \mathcal{A}_{simple}(\mathcal{S}')$ with $p_s^{A'}(a^? \mathcal{U} a^0) \geq p_s^{B'}(a^? \mathcal{U} a^0)$ for all $B' \in \mathcal{A}(\mathcal{S}')$, $s \in S$ (Lemma 12.1).

<u>*Claim 1*</u>: $p_s^{A'}(a^? \mathcal{U} a^0) = 1 - p_s^{A'}(\Phi_1 \mathcal{U} a^1)$ for all $s \in S$.

<u>Proof</u>. Because of Lemma 12.4 it suffices to show that for each fulpath $\pi \in StateFair^{B'}$ either $\pi \models a^? \mathcal{U} a^0$ or $\pi \models \Phi_1 \mathcal{U} a^1$. (Note that π can fulfill at most one the path formulas $a^? \mathcal{U} a^0$ and $\Phi_1 \mathcal{U} a^1$.) Let $\pi \in StateFair^{A'}$. We suppose $\pi \not\models \Phi_1 \mathcal{U} a^1$ and $\pi \not\models a^? \mathcal{U} a^0$. Then, $\pi(i) \in S^?$ for all $i \geq 0$. Let $s \in inf(\pi)$. By Lemma 12.5(a) $Reach^{A'}(s) \subseteq inf(\pi)$. Hence, $Reach^{A'}(s) \subseteq S^?$. Thus, $Reach^{A'}(s) \cap S^0 = \emptyset$ and $p_s^{A'}(a^? \mathcal{U} a^0) = 0$. By the definition of A', we obtain $Reach_{\Phi_1 \wedge \neg \Phi_2}(s) \cap S^0 = \emptyset$. Hence, $s \in S^1$. Contradiction (as $S^? \subseteq S \setminus S^1$). ⌋

<u>*Claim 2*</u>: $p_s^{A'}(\Phi_1 \mathcal{U} a^1) \;\leq\; p_s^{F'}(\Phi_1 \mathcal{U} a^1)$ for all $s \in S^?$ and $F' \in \mathcal{A}_{fair}(\mathcal{S}')$.

<u>*Proof*</u>. Using Lemma 12.5(b) it can be shown that for all $\pi \in TotalFair^{F'}(s)$: $\pi \models a^? \mathcal{U} a^0$ or $\pi \models \Phi_1 \mathcal{U} a^1$. Thus, $p_s^{F'}(a^? \mathcal{U} a^0) = 1 - p_s^{F'}(\Phi_1 \mathcal{U} a^1)$ (by Lemma 12.6). Hence, by Claim 1: $p_s^{A'}(\Phi_1 \mathcal{U} a^1) = 1 - p_s^{A'}(a^? \mathcal{U} a^0) \leq 1 - p_s^{F'}(a^? \mathcal{U} a^0) = p_s^{F'}(\Phi_1 \mathcal{U} a^1)$. ⌋

<u>*Claim 3*</u>: Let $B \in \mathcal{A}(\mathcal{S})$ and $B' \in \mathcal{A}(\mathcal{S}')$ be an adversary with $B'(\omega) = B(\omega)$ for all $\omega \in Path^B_{fin}$ with $first(\omega) \in S^?$ and $last(\omega) \in S^?$. Then:

(i) $p_s^{B'}(a^? \mathcal{U} a^0) = p_s^B(a^? \mathcal{U} a^0)$ for all $s \in S$

(ii) If $p_\omega^B = 0$ for all $\omega \in Path^B_{fin}(s)$ with $last(\omega) \in S^0$ and $p_\omega^B = 1$ for all $\omega \in Path^B_{fin}(s)$ with $last(\omega) \in S^1$ then $p_s^{B'}(\Phi_1 \mathcal{U} a^1) = p_s^B(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S^?$.

<u>*Proof*</u>. (i) is an easy verification. We show (ii). Let Ω be the set of paths $\omega \in Path^B_{fin}(s)$ with $\omega(i) \in S^?$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \in S^1$. Then, $\Omega \subseteq Path^{B'}_{fin}(s)$ and $p_s^{B'}(\Phi_1 \mathcal{U} a^1) = \sum_{\omega \in \Omega} \mathbf{P}(\omega) = p_s^B(\Phi_1 \mathcal{U} \Phi_2)$. ⌋

Claim 3(i) yields: Whenever $A \in \mathcal{A}_{simple}$ with $A(s) = A'(s)$ for all $s \in S^?$ then

$$p_s^A(a^? \mathcal{U} a^0) \;=\; p_s^{max}(a^? \mathcal{U} a^0)$$

for all $s \in S$.

<u>Claim 4</u>: $p_s^{A'}(\Phi_1 \mathcal{U} a^1) \;\leq\; p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $F \in \mathcal{A}_{fair}(\mathcal{S})$ and $s \in S$.

<u>*Proof*</u>. Let $F \in \mathcal{A}_{fair}(\mathcal{S})$. We may assume w.l.o.g. that $p_\omega^F(\Phi_1 \mathcal{U} \Phi_2) = 0$ for all $\omega \in Path_{fin}$ with $\omega(i) \notin S^0$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \in S^0$ (Lemma 12.22). We define $F' \in \mathcal{A}(\mathcal{S}')$ as

follows: $F'(\omega) = F(\omega)$ if $last(\omega) \in S^?$ and $F'(\omega) = \mu_s^1$ if $last(\omega) \notin S^?$. Then, $F' \in \mathcal{A}_{fair}(\mathcal{S}')$. By Claim 2 and 3: $p_s^F(\Phi_1 \mathcal{U} \Phi_2) = p_s^{F'}(\Phi_1 \mathcal{U} a^1) \geq p_s^{A'}(\Phi_1 \mathcal{U} a^1)$. ⌋

Let $A'' \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$ be an adversary with $Reach^{A''}_{\Phi_1 \wedge \neg\Phi_2}(t) \subseteq S^0$ (and $p_t^{A''} = 0$) for all $t \in S^0$) (Lemma 12.23). Let $A \in \mathcal{A}_{simple}$ with

- $A(s) = A'(s)$ for all $s \in S^?$
- $A(v) = \mu_v$ for all $v \in S^1$
- $A(s) = A''(s)$ for all $s \in S^0$.

We show that A is admissible for (Φ_1, Φ_2). Clearly, $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(s) = Reach^{A''}_{\Phi_1 \wedge \neg\Phi_2}(s)$ for all $s \in S^0$. It suffices to show that $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(s) \cap S^0 \neq \emptyset$ for all $s \in S^? \setminus U$ and $p_u^A = 1$ for all $u \in U \cup S^1$ where $U = \{u \in S^? : p_u^{A'}(\Phi_1 \mathcal{U} a^1) = 1\}$. (This ensures that $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(s) \cap (Sat(\Phi_2) \cup (S \setminus S^+)) \neq \emptyset$ for all $s \in S^+$.) We suppose $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(s) \cap S^0 = \emptyset$ for some $s \in S^? \setminus U$. Then, $p_s^A(a^? \mathcal{U} a^0) = 0$. By Claim 3, $p_s^{A'}(a^? \mathcal{U} a^0) = 0$. Hence, by Claim 1, $p_s^{A'}(\Phi_1 \mathcal{U} a^1) = 1$. Thus, $s \in U$. Contradiction. For all $u \in U$, $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(u) \cap (V \cup Sat(\Phi_2)) = Reach^{A'}_{\Phi_1 \wedge \neg\Phi_2}(u) \cap (V \cup Sat(\Phi_2)) \neq \emptyset$. By (2), $p_v^A = 1$ for all $v \in S^1$. We get $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(u) \cap Sat(\Phi_2) \neq \emptyset$ for all $u \in U \cup S^1$. On the other hand, $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(u) \subseteq Sat(\Phi_2) \cup U \cup S^1$. By Lemma 12.17, $p_u^A = 1$ for all $u \in U \cup S^1$.

For all $F \in \mathcal{A}_{fair}(\mathcal{S})$ we have:

- $p_s^A(\Phi_1 \mathcal{U} \Phi_2) = p_s^{A'}(\Phi_1 \mathcal{U} a^1) \leq p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S^?$ (Claim 3 and 4)
- $p_s^A(\Phi_1 \mathcal{U} \Phi_2) = p_s^{A''}(\Phi_1 \mathcal{U} \Phi_2) = 0 \leq p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S^0$
- $p_v^A(\Phi_1 \mathcal{U} \Phi_2) = p_v^F(\Phi_1 \mathcal{U} \Phi_2) = 1$ for all $v \in S^1$ (by (1) and (2)).

Hence, $p_s^A(\Phi_1 \mathcal{U} \Phi_2) \leq p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S$. ■

Remark 12.26 If $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$ and $s \in S$ such that $Reach^{A}_{\Phi_1 \wedge \neg\Phi_2}(s) \subseteq S^+(\Phi_1, \Phi_2)$ then $p_s^A(\Phi_1 \mathcal{U} \Phi_2) = 1$ (Lemma 12.17). By Lemma 12.25: If $Reach_{\Phi_1 \wedge \neg\Phi_2}(s) \subseteq S^+(\Phi_1, \Phi_2)$ then $p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2) = p_s^F(\Phi_1 \mathcal{U} \Phi_2) = 1$ for all $F \in \mathcal{A}_{fair}$. Since $Reach_{\Phi_1 \wedge \neg\Phi_2}(t) \subseteq Reach_{\Phi_1 \wedge \neg\Phi_2}(s)$ for all $t \in Reach_{\Phi_1 \wedge \neg\Phi_2}(s)$ we get $p_t^F(\Phi_1 \mathcal{U} \Phi_2) = 1$ for all $F \in \mathcal{A}_{fair}$ and $t \in Reach_{\Phi_1 \wedge \neg\Phi_2}(s)$. This result is similar to the "Zero-One-Law" of [33] which states that $\min\{p_t^{fair} : t \in T\} \in \{0, 1\}$ where $p_t^{fair} = \inf\{p_t^F(tt\ \mathcal{U}\ \Phi) : F \text{ process fair }\}$ and $T = Reach_{tt \wedge \neg\Phi}(s)$ and where "process fairness" means fairness in the sense of [33]. ■

Corollary 12.27 (cf. Lemma 7.1) *For all $s \in S$: $p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2) = 1 - p_s^{max}(a^? \mathcal{U} a^0)$.*

Proof: follows immediately by Lemma 12.15(a) and 12.25. ■

Lemma 12.28 *Let $T_0 = S \setminus S^+(\Phi_1, \Phi_2)$ and, for $i = 0, 1, 2, \ldots$, T_{i+1} the largest subset of $S \setminus (T_0 \cup \ldots \cup T_i \cup Sat(\Phi_2))$ such that for all $t \in T_{i+1}$ there is some $\mu_t \in Steps(t)$ with:*

- *$Supp(\mu_t) \subseteq T_0 \cup \ldots \cup T_i \cup T_{i+1}$*
- *there is a finite path $t = t_0 \xrightarrow{\mu_{t_0}} t_1 \xrightarrow{\mu_{t_1}} \ldots \xrightarrow{\mu_{t_{k-1}}} t_k$ where $t_0, \ldots, t_{k-1} \in T_{i+1}$ and $t_k \in T_0 \cup \ldots \cup T_i$.*

Then, $S^0(\Phi_1, \Phi_2) = \bigcup_{i \geq 0} T_i$.

Proof: Let $T = \bigcup_{i \geq 0} T_i$. First we observe that whenever U is a subset of $S \setminus (T \cup Sat(\Phi_2))$ such that for all $u \in U$ there is some $\mu_u \in Steps(u)$ with:

- *$Supp(\mu_u) \subseteq T \cup U$*

- there is a finite path $u = u_0 \xrightarrow{\mu_{u_0}} u_1 \xrightarrow{\mu_{u_1}} \ldots \xrightarrow{\mu_{u_{k-1}}} u_k \xrightarrow{\mu_{u_k}} t$ where $u_0, \ldots, u_k \in U$ and $t \in T$

then $U = \emptyset$.

Let $s \in S^0$. Then, $p_s^A = 0$ for some $A \in \mathcal{A}_{adm}(\Phi_1, \Phi_2)$. Let $U = \{u \in S \setminus T : p_u^A = 0\}$. For $u \in U$, let $\mu_u = A(u)$. Clearly, $Supp(\mu_u) \subseteq \{u \in S : p_u^A = 0\} \subseteq T \cup U$. For $u \in S$, $p_u^A = 0$, let $\kappa(u)$ be the length of a shortest path $\omega \in Path_{fin}^A(u)$ with $last(\omega) \in S \setminus S^+$. By induction on $\kappa(u)$ we show that there exists a path

$$u = u_0 \xrightarrow{\mu_{u_0}} v_1 \xrightarrow{\mu_{u_1}} \ldots \xrightarrow{\mu_{u_{k-1}}} u_k \xrightarrow{\mu_{u_k}} t$$

where $u_1, \ldots, u_k \in U$ and $t \in T$. If $\kappa(u) = 0$ then there is nothing to show as $u \in S \setminus S^+ = T_0 \subseteq T$. Let $\kappa(u) \geq 1$ and let $\omega \in Path_{fin}^A(u)$ be a shortest path with $last(\omega) \in S \setminus S^+$. Let $v = \omega(1)$. Then, $p_v^A = 0$ and $\kappa(v) < \kappa(u)$. By induction hypothesis there is a finite path $\omega' \in Path_{fin}^A(v)$ where $last(\omega) \in T$ and $\omega(i) \in U$, $i = 0, 1, \ldots, |\omega| - 1$. Hence, $u \xrightarrow{\mu_u} \omega'$ is a path with the desired property. We conclude $U = \emptyset$. Hence, $s \in \{u \in S : p_u^A = 0\} \subseteq T$.

Next we show that $T \subseteq S^0$. For this, it suffices to show that there is an adversary A which is admissible for (Φ_1, Φ_2) and with $p_t^A = 0$ for all $t \in T$. For each $t \in \bigcup_{i \geq 1} T_i$, let $\mu_t \in Steps(t)$ be as above, i.e., if $t \in T_{i+1}$ then

- $Supp(\mu_t) \subseteq T_0 \cup \ldots \cup T_i \cup T_{i+1}$
- there is a finite path $t = t_0 \xrightarrow{\mu_{t_0}} t_1 \xrightarrow{\mu_{t_1}} \ldots \xrightarrow{\mu_{t_{k-1}}} t_k$ where $t_0, \ldots, t_{k-1} \in T_{i+1}$ and $t_k \in T_0 \cup \ldots \cup T_i$.

For all $s \in S \setminus (T \cup Sat(\Phi_2))$, we choose a path $\omega_s \in Path_{fin}(s)$ with $last(\omega_s) \models \Phi_2$, $\omega_s(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega_s| - 1$, and $|\omega_s| = \|s\|$. Let $\mu_s = step(\omega_s, 0)$. Let A be a simple adversary with $A(s) = \mu_s$ if $s \in S \setminus (Sat(\Phi_2) \cup S \setminus S^+)$. It is easy to see that A is admissible for (Φ_1, Φ_2) and that $p_t^A = 0$ for all $t \in T$. ■

12.6 Proof of Theorem 4

Theorem 4 follows by Lemma 12.32, Lemma 12.33 and Theorem 3.

Lemma 12.29 *Let $F \in \mathcal{A}_{sfair}$, $s \in S^+(\Phi_1, \Phi_2) \setminus Sat(\Phi_2)$ and Ω be the set of paths $\omega \in Path_{fin}^F(s)$ with $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \in S^+(\Phi_1, \Phi_2)$. If $F(\omega) \in MaxSteps(last(\omega), \Phi_1, \Phi_2)$ for all $\omega \in \Omega$ then $\{\pi \in \omega \uparrow^F : \pi \models \Phi_1 \mathcal{U} \Phi_2\} \neq \emptyset$ for all $\omega \in \Omega$.*

Proof: For $\omega \in \Omega$ with $last(\omega) \notin Sat(\Phi_2)$ we define $\omega \Uparrow$ to be the set of finite paths $\omega' \in \omega \uparrow_{fin}^F$ where $\omega'(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega'| - 1$. If $\omega \in \Omega$, $last(\omega) \notin Sat(\Phi_2)$ then $F(\omega) \in MaxSteps(last(\omega))$, and hence, $F(\omega)(t) > 0$ for some $t \in S^+$. Thus, the path $\omega \xrightarrow{F(\omega)} t$ belongs to $\omega \Uparrow$. Hence, $\omega \Uparrow \neq \emptyset$. We define $\min(\omega) = \min\{\|\omega'(i)\| : \omega' \in \omega \Uparrow, |\omega| < i \leq |\omega'|\}$. For $\omega \in \Omega$ with $last(\omega) \notin Sat(\Phi_2)$ let $\overline{\omega}$ be a path in $\omega \Uparrow$ with $\|last(\overline{\omega})\| = \min(\omega)$. We suppose that there is some $\omega_0 \in \Omega$ such that $\{\pi \in \omega_0 \uparrow^F : \pi \models \Phi_1 \mathcal{U} \Phi_2\} = \emptyset$. Then, $last(\omega_0) \notin Sat(\Phi_2)$. For $i \geq 0$ let $\omega_{i+1} = \overline{\omega_i}$. Note that $last(\omega_i) \in S^+ \setminus Sat(\Phi_2)$. As $\omega_0 \prec \omega_1 \prec \ldots$ we have $\min(\omega_0) \leq \min(\omega_1) \leq \ldots$. We choose some $k \geq 0$ with $\min(\omega_k) = \min(\omega_j)$ for all $j \geq k$ and define π to be the unique path with $\omega_i \prec \pi$ for all $i \geq 0$. Then, $\pi \in Path_{ful}^F(s)$, $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$ and there is some $u \in S^+ \cap inf(\pi)$ with $last(\omega_i) = u$ for infinitely many $i \geq 0$. Hence, $\|u\| = \min(\omega_j)$ for all $j \geq k$. Let $\mu \in Steps(u)$ such that $\|v\| < \|u\|$ for some $v \in S^+$ with $\mu(v) > 0$. As F is strictly fair there exists $j \geq k$ with $\pi(j) = u$ and $step(\pi, j) = \mu$.

Let ω' be the path $\pi^{(j)} \xrightarrow{\mu} v$. As $j \geq k$ we have $\omega' \in \omega_k \Uparrow$ and hence $\|v\| \geq \min(\omega_k) = \|u\|$. Contradiction. ■

Lemma 12.30 *Let $F \in \mathcal{A}_{fair}$, $\Gamma = \{\pi \in Path_{ful}^F : \pi \models \Phi_1 \mathcal{U} \Phi_2\}$ and $\Gamma_k = \bigcup_{\lambda \in \Lambda_k} \lambda \uparrow^F$ where $\Lambda_k = \{\lambda \in Path_{fin} : |\lambda| = k$ and $\lambda \prec \pi$ for some $\pi \in \Gamma\}$. Then, for all $s \in S$:*

$$p_s^F(\Phi_1 \mathcal{U} \Phi_2) \;=\; \lim_{k \to \infty} Prob(\Gamma_k)$$

Proof: We have $\Gamma_0 \supseteq \Gamma_1 \supseteq \ldots \supseteq \Gamma$. Let $\Gamma' = \bigcap_{k \geq 1} \Gamma_k$. Then, Γ' is measurable and $\Gamma' \supseteq \Gamma$. Hence, $p_s^F = Prob(\Gamma(s)) \leq Prob(\Gamma'(s)) = \lim_{k \to \infty} Prob(\Gamma_k(s))$. Using Lemma 12.5(b) it can be shown that $TotalFair \cap \Gamma' \subseteq \Gamma$. By Lemma 12.6, $Prob(\Gamma'(s)) = Prob(TotalFair^F(s) \cap \Gamma'(s)) \leq Prob(\Gamma(s)) = p_s^F$. Hence, $p_s^F = Prob(\Gamma'(s)) = \lim_{k \to \infty} Prob(\Gamma_k(s))$. ■

Lemma 12.31 *Let $F \in \mathcal{A}_{sfair}$, $s \in S^+(\Phi_1, \Phi_2)$ and let Ω be the set of paths $\omega \in Path_{fin}^A(s)$ with $\omega(i) \models \Phi_1 \wedge \neg \Phi_2$, $i = 0, 1, \ldots, |\omega|$. The following are equivalent:*

(i) $F(\omega) \in MaxSteps(last(\omega), \Phi_1, \Phi_2)$ for all $\omega \in \Omega$ with $last(\omega) \in S^+(\Phi_1, \Phi_2)$.

(ii) $p_s^{max}(\Phi_1 \mathcal{U} \Phi_2) \;=\; p_s^F(\Phi_1 \mathcal{U} \Phi_2)$.

(iii) $p_{last(\omega)}^{max}(\Phi_1 \mathcal{U} \Phi_2) \;=\; p_\omega^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $\omega \in \Omega$.

Proof: The implication (iii) $\Longrightarrow$ (ii) is obvious.

(ii) $\Longrightarrow$ (iii): We suppose $p_{\omega_0}^F < p_{last(\omega_0)}^{max}$ for some $\omega_0 \in \Omega$. Let A be the adversary given by: $A(\lambda) = F(\lambda)$ if $\omega_0 \not\preceq \lambda$ and $A(\lambda) = B(\gamma)$ if $\lambda = \omega_0 \gamma$ where B is an adversary with $p_{last(\omega_0)}^B = p_{last(\omega_0)}^{max}$ (Lemma 12.1). Then, $p_{\omega_0}^A = p_{last(\omega_0)}^B = p_{last(\omega_0)}^{max}$. Let $k = |\omega_0|$, Λ the set of paths $\lambda \in Path_{fin}^F(s)$ with $\lambda^{(l)} \in \Omega$ and $last(\lambda) \models \Phi_2$ where $l = |\lambda| - 1 < k$ and $\Omega' = \{\omega \in \Omega : |\omega| = k\}$. As $\Omega' \subseteq Path_{fin}^A(s)$ and $p_\omega^A = p_\omega^F$ for all $\omega \in \Omega' \setminus \{\omega_0\}$ and as $\Lambda \subseteq Path_{fin}^F(s)$ we obtain:

$$p_s^F \;=\; \sum_{\omega \in \Omega'} \mathbf{P}(\omega) \cdot p_\omega^F \;+\; \sum_{\lambda \in \Lambda} \mathbf{P}(\lambda) \;<\; \sum_{\omega \in \Omega'} \mathbf{P}(\omega) \cdot p_\omega^A \;+\; \sum_{\lambda \in \Lambda} \mathbf{P}(\lambda) \;=\; p_s^A \;\leq\; p_s^{max}$$

Contradiction.

(iii) $\Longrightarrow$ (i): If $\omega \in \Omega$, $s = last(\omega) \in S^+$ and $\mu = F(\omega)$ then $p_s^{max} = p_\omega^F = \sum_{t \in S} \mu(t) \cdot p_{\omega_t}^F = \sum_{t \in S} \mu(t) \cdot p_t^{max}$ where ω_t is the path $\omega \xrightarrow{\mu} t$. Hence, $\mu \in MaxSteps(s)$.

(i) $\Longrightarrow$ (iii): Let $\Lambda(\omega)$ be the set of paths $\lambda \in Path_{fin}^F$ with $last(\omega) = first(\lambda)$ and $\omega\lambda \prec \pi$ for some $\pi \in \omega \uparrow^F$ with $\pi \models \Phi_1 \mathcal{U} \Phi_2$. Let $\Lambda_k(\omega) = \{\lambda \in \Lambda(\omega) : |\lambda| = k\}$. Lemma 12.30 applied to the fair adversary F' with $F'(\lambda) = F(\omega\lambda)$ if $first(\lambda) = last(\omega)$ yields $p_\omega^F = \lim_{k \to \infty} p_\omega^k$ where $p_\omega^k = \sum_{\lambda \in \Lambda_k(\omega)} \mathbf{P}(\lambda)$. By induction on k it can be shown that $p_\omega^k \geq p_{last(\omega)}^{max}$ for all $\omega \in \Omega$ with $last(\omega) \in S^+$. This yields $p_\omega^F \geq p_{last(\omega)}^{max}$ for all $\omega \in \Omega$. Hence, $p_\omega^F = p_{last(\omega)}^{max}$ for all $\omega \in \Omega$. ■

Lemma 12.32 *There exists $F \in \mathcal{A}_{sfair}$ with $p_s^F(\Phi_1 \mathcal{U} \Phi_2) = p_s^{max}(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in T^{max}(\Phi_1, \Phi_2)$.*

Proof: For each $j \geq 1$, $t \in T_{j,1}^{max}$ we choose some $\mu_t \in MaxSteps(t)$ with $Supp(\mu_t) \subseteq \bigcup_{i<j} T_i^{max}$. We define a strictly fair adversary F as follows. Let $T = \bigcup_{j \geq 1} T_{j,1}^{max}$. For each $s \in S$, let $\nu_0^s, \ldots, \nu_{k_s - 1}^s$ be an enumeration of $Steps(s)$ (and k_s the cardinality of $Steps(s)$). For $\omega \in Path_{fin}$, let $\eta(\omega)$ be the number of indices $i > |\omega|$ with $\omega(i) = last(\omega)$. Let Ω be the set of finite paths $\omega \in Path_{fin}$ with $\omega(i) \models \Phi_1 \wedge \neg \Phi_2$ for all $i \leq |\omega|$. We define

$$F(\omega) \;=\; \begin{cases} \nu_j^{last(\omega)} & : \text{ if } (\omega \notin \Omega \text{ or } last(\omega) \in S \setminus T) \text{ and } j = \eta(\omega) \bmod k_{last(\omega)} \\ \mu_{last(\omega)} & : \text{ otherwise.} \end{cases}$$

The strict fairness of F follows by the fact that for each $\pi \in Path_{ful}^F$:

(i) If $\pi(i) \not\models \Phi_1 \wedge \neg\Phi_2$ then for each $s \in \mathit{inf}(\pi)$ and each $\mu \in \mathit{Steps}(s)$: $\mathit{step}(\pi, k) = \mu$ for infinitely many $k \geq i$ with $\pi(k) = s$.

(ii) If $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$ and $\pi(i) \in T^{max}_{j,2}$ then $\pi(k) \in T^{max}_{j,2} \cup \bigcup_{i<j} T^{max}_i$ for all $k > i$.

(iii) If $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$ and $\pi(i) \in T^{max}_{j,1}$ then $\pi(k) \in \bigcup_{i<j} T^{max}_i$ for all $k > i$.

(iv) If $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$, $t \in \mathit{inf}(\pi)$ for some $t \in T^{max}_{j,2}$ and $\mu \in \mathit{Steps}(t)$ then $\mathit{step}(\pi, k) = \mu$ for infinitely many k with $\pi(k) = t$.

By (iii), none of the states $t \in T$ can occur more than once in a path $\pi \in \mathit{Path}^F_{\mathit{ful}}$ with $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$. By (i), (ii) and (iii) it follows that for each path $\pi \in \mathit{Path}^F_{\mathit{ful}}$ and each $s \in \mathit{inf}(\pi)$, every $\mu \in \mathit{Steps}(s)$ is taken infinitely often in π. By definition of F it is immediately clear that $\mathit{Reach}^F_{\Phi_1 \wedge \neg\Phi_2}(t) \subseteq T^{max}$ for all $t \in T^{max}$ and that $F(\omega) \in \mathit{MaxSteps}(\mathit{last}(\omega))$ for all $\omega \in \Omega$ with $\mathit{last}(\omega) \in T^{max}$. By Lemma 12.29 and Lemma 12.31 we get $p^F_t = p^{max}_t$ for all $t \in T^{max}$. ■

Lemma 12.33 *If $s \notin T^{max}(\Phi_1, \Phi_2)$ then $p^F_s(\Phi_1 \mathcal{U} \Phi_2) < p^{max}_s(\Phi_1 \mathcal{U} \Phi_2)$ for all $F \in \mathcal{A}_{sfair}$.*

Proof: We suppose $p^F_s = p^{max}_s$ for some $F \in \mathcal{A}_{sfair}$ and $s \in S \setminus T^{max}$. Let Ω be the set of paths $\omega \in \mathit{Path}^F_{\mathit{fin}}(s)$ with $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i \leq |\omega|$. By Lemma 12.31:

(*) $F(\omega) \in \mathit{MaxSteps}(\mathit{last}(\omega))$ for all $\omega \in \Omega$.

By definition of T^{max} we have $S \setminus T^{max} \subseteq S^+ \setminus \mathit{Sat}(\Phi_2)$. Let U be the set states $u \in S \setminus T^{max}$ with $\mathit{MaxSteps}(u) \neq \mathit{Steps}(u)$.

<u>*Claim 1*</u>: For each $\omega \in \Omega$ with $\mathit{last}(\omega) \notin T^{max}$ there exists $\overline{\omega} \in \omega\uparrow^F \cap \Omega$ with $\mathit{last}(\overline{\omega}) \in U$.

<u>*Proof*</u>. We suppose that for some $\omega \in \Omega$ with $\mathit{last}(\omega) \in S \setminus T^{max}$ there does not exists a path $\overline{\omega} \in \omega\uparrow^F \cap \Omega$ with $\mathit{last}(\overline{\omega}) \in U$. Let T be the set of states $t \in S \setminus T^{max}$ such that $t = \mathit{last}(\omega_t)$ for some $\omega_t \in \omega\uparrow^F \cap \Omega$. For each $t \in T$, we define $\mu_t = F(\omega_t)$. By our assumption, $T \cap U = \emptyset$ and $\mathit{Supp}(\mu_t) \subseteq T \cup T^{max}$ (as otherwise, $\mu_t(u) > 0$ for some $u \in U$; hence, $\omega_t \xrightarrow{\mu_t} u \in \omega\uparrow^F \cap \Omega$). Since $\mathit{MaxSteps}(t) = \mathit{Steps}(t)$ for all $t \in T$ and by definition of T^{max} we have $T \subseteq T^{max}$ and therefore $T = \emptyset$ (as T is defined as a subset of $S \setminus T^{max}$). Let $\mu = F(\omega)$. Then, $\mathit{Supp}(\mu) \subseteq T^{max}$. We conclude $\mathit{last}(\omega) \in T^{max}_{j,1}$ for some j. Hence, $\mathit{last}(\omega) \in T^{max}$. Contradiction. ⌋

<u>*Claim 2*</u>: There exists $\pi \in \mathit{Path}^F_{\mathit{ful}}(s)$ with $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$ and $U \cap \mathit{inf}(\pi) \neq \emptyset$.

<u>*Proof*</u>. For each $\omega \in \Omega$ with $\mathit{last}(\omega) \in S \setminus T^{max}$ we choose some $\overline{\omega} \in \omega\uparrow^F \cap \Omega$ with $\mathit{last}(\overline{\omega}) \in U$ (Claim 1). Let $\omega_0 = s$ and $\omega_{i+1} = \overline{\omega_i}$. Let $\pi \in \mathit{Path}^F_{\mathit{ful}}(s)$ be the unique path with $\omega_i \prec \pi$ for all $i \geq 0$. Then, $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$ and $\mathit{inf}(\pi) \cap U \neq \emptyset$. ⌋

We choose some $\pi \in \mathit{Path}^F_{\mathit{ful}}(s)$ with $\pi \models \Box(\Phi_1 \wedge \neg\Phi_2)$ and $u \in \mathit{inf}(\pi) \cap U$. As F is strictly fair and $\mathit{MaxSteps}(u) \neq \mathit{Steps}(u)$ there exists $j \geq 0$ with $F(\pi^{(j)}) \notin \mathit{MaxSteps}(\pi(j))$. Contradiction (to (*)) as $\pi^{(j)} \in \Omega$. ■

12.7 Proof of Theorem 7

Theorem 7 follows by Theorem 6, Lemma 12.36 and Lemma 12.37. The proofs are similar to those in Section 12.6: Lemma 12.36 is the counterpart to Lemma 12.32 and Lemma 12.37 the counterpart to Lemma 12.33. We include the proofs for the sake of completeness.

Lemma 12.34 *Let $(p_s)_{s \in S}$ be a vector of real numbers with $0 \leq p_s \leq 1$ and $p_s = 1$ for all $s \in \mathit{Sat}(\Phi_2)$ and $p_s = 0$ for all $s \in S \setminus S^+(\Phi_1, \Phi_2)$. For $s \in \mathit{Sat}(\Phi_2) \cup (S \setminus S^+(\Phi_1, \Phi_2))$ let $M(s) = \mathit{Steps}(s)$. For $s \in S^+(\Phi_1, \Phi_2) \setminus \mathit{Sat}(\Phi_2)$ let $M(s)$ be the set of distributions $\mu \in \mathit{Steps}(s)$ with $p_s = \sum_{t \in S} \mu(t) \cdot p_t$. Then: If $A \in \mathcal{A}_{simple}$ with $A(s) \in M(s)$ for all $s \in S$ then $p^A_s(\Phi_1 \mathcal{U} \Phi_2) \leq p_s$ for all $s \in S$.*

Proof: First we observe that the assumption $A(s) \in M(s)$ for all $s \in S$ implies $M(s) \neq \emptyset$ for all $s \in S$. Let $U = \{s \in S : p_s^A = 0\}$, $T = S \setminus U$ and $U' = \{s \in S : p_s = 0\}$. It is easy to see that $U' \subseteq U$. In particular, $p_s^A = 0 \leq p_s$ for all $s \in U$ (by Claim 1). Next we show $p_s^A \leq p_s$ for all $s \in T$. Let $\mathbf{A} = (A(s,t))_{s,t \in T}$ and $\mathbf{I}$ be the identity matrix with $|T|$ rows and columns where $|T|$ is the cardinality of T and let $\mathbf{b}^A = (b_s^A)_{s \in T}$ and $\mathbf{b} = (b_s)_{s \in T}$ where

$$b_s^A \ = \sum_{t \in Sat(\Phi_2)} A(s,t), \quad b_s \ = \ b_s^A \ + \sum_{t \in S \setminus (T \cup Sat(\Phi_2))} A(s,t) \cdot p_t.$$

Using the results of [31] we obtain that the matrix $\mathbf{I} - \mathbf{A}$ is regular and that the vector $\mathbf{p}^A = (p_t^A)_{t \in T}$ is the unique solution of the linear equation system $(\mathbf{I} - \mathbf{A}) \cdot \mathbf{x} = \mathbf{b}^A$, i.e. $(\mathbf{I} - \mathbf{A}) \cdot \mathbf{p}^A = \mathbf{b}^A$. It is easy to see that the vector $\mathbf{p} = (p_t)_{t \in T}$ solves the linear equation system $(\mathbf{I} - \mathbf{A}) \cdot \mathbf{x} = \mathbf{b}$. Let $\mathbf{Y} \ = \ (\mathbf{I} - \mathbf{A})^{-1}$, $\mathbf{Y} = (y_{s,t})_{s,t \in T}$. Then, $\mathbf{p}^A = \mathbf{Y} \cdot \mathbf{b}^A$ and $\mathbf{p} = \mathbf{Y} \cdot \mathbf{b}$. As $b_s^A \leq b_s$ for al $s \in T$ and $p_t = \sum_{s \in T} y_{s,t} \cdot b_s$, $p_t^A = \sum_{s \in T} y_{s,t} \cdot b_s^A$ it suffices to show that $y_{s,t} \geq 0$ for all s, $t \in T$. As $(\mathbf{I} - \mathbf{A}) \cdot \mathbf{Y} = \mathbf{I}$ we have $\mathbf{A} \cdot (-\mathbf{Y}) = \mathbf{I} - \mathbf{Y}$. Hence, for all s, $t \in T$:

$$\sum_{v \in T} A(s,v) \cdot (-y_{v,t}) \ = \ \begin{cases} 1 - y_{s,s} & : \ \text{if } s = t \\ -y_{s,t} & : \ \text{otherwise.} \end{cases}$$

Let $y = \min\{y_{s,t} : s, t \in T\}$. We suppose that $y < 0$. First we suppose that $y_{s,s} = y$ for some $s \in T$. As $\sum_{v \in T} \ A(s,v) \ \leq \ 1$ we get:

$$\begin{aligned} 1 \ &= \sum_{v \in T} A(s,v) \cdot (-y_{v,s}) \ + \ y_{s,s} \\ &\leq \sum_{v \in T \setminus \{s\}} A(s,v) \cdot (-y_{s,s}) \ - \ (1 - A(s,s)) \cdot (-y_{s,s}) \\ &\leq (1 - A(s,s)) \cdot (-y_{s,s}) \ - \ (1 - A(s,s)) \cdot (-y_{s,s}) \ = \ 0. \end{aligned}$$

Contradiction. Now we suppose that $y_{s,s} > y$ for all $s \in T$. Let V be the set of states $v \in T$ with $y_{v,t} = y$ for some $t \in T$. Let $s \in V$ and $t \in T$ such that $y_{s,t} = y$. As $\sum_{v \in T} \ A(s,v) \ \leq \ 1$ we get:

$$-y \ = \ -y_{s,t} \ = \sum_{v \in T} A(s,v) \cdot (-y_{v,t}) \ \leq \sum_{v \in T} A(s,v) \cdot (-y_{s,t}) \ \leq \ -y_{s,t} \ = \ -y$$

Hence, $\sum_{v \in T} A(s,t) = 1$ and $y_{v,t} = y$ for all $v \in T$ with $A(s,v) > 0$. In particular, if $A(s,v) > 0$ then $v \in V$. Hence, $Reach^A(v) \subseteq V$. Thus, $p_v^A = 0$ for all $v \in V$. (Note that $V \subseteq S \setminus Sat(\Phi_2)$.) In particular, $s \notin T$ (as $p_s^A = 0$). Contradiction (as $s \in V \subseteq T$). ■

Lemma 12.35 *Let $F \in \mathcal{A}_{sfair}$, $s \in S^+(\Phi_1, \Phi_2)$ and let Ω be the set of paths $\omega \in Path_{fin}^F(s)$ with $\omega(i) \ \models \ \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega|$. Then, the following are equivalent:*

(i) $F(\omega) \in AdmSteps(last(\omega), \Phi_1, \Phi_2)$ *for all* $\omega \in \Omega$.

(ii) $p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2) \ = \ p_s^F(\Phi_1 \mathcal{U} \Phi_2)$.

(iii) $p_{last(\omega)}^{adm}(\Phi_1 \mathcal{U} \Phi_2) \ = \ p_\omega^F(\Phi_1 \mathcal{U} \Phi_2)$ *for all* $\omega \in \Omega$.

Proof: (ii) $\Longrightarrow$ (iii): As F is strictly fair we have $p_\omega^F \ \geq \ p_{last(\omega)}^{adm}$ for all ω (Theorem 6). We suppose $p_{\omega_0}^F > p_{last(\omega_0)}^{adm}$ for some $\omega_0 \in \Omega$. Let $0 < \varepsilon \ < \ p_{\omega_0}^F - p_{last(\omega_0)}^{adm}$ and $G \in \mathcal{A}_{sfair}$ such that $p_{last(\omega_0)}^G < p_{last(\omega_0)}^{adm} + \varepsilon$ (Lemma 12.15). Let F' be the strictly fair adversary which given by: $F'(\lambda) = F(\lambda)$ if $\omega_0 \not\preceq \lambda$ and $F'(\lambda) = G(\gamma)$ if $\lambda = \omega_0\gamma$. Then, $p_{\omega_0}^{F'} \ = \ p_{last(\omega_0)}^G \ < \ p_{last(\omega_0)}^{adm} + \varepsilon \ < \ p_{\omega_0}^F$. Let $k = |\omega_0|$, Λ the set of paths $\lambda \in Path_{fin}^A(s)$ with $\lambda^{(l)} \in \Omega$ and $last(\lambda) \ \models \ \Phi_2$ where

$l = |\lambda| - 1 < k$ and let $\Omega' = \{\omega \in \Omega : |\omega| = k\}$. As $\Omega' \subseteq \mathit{Path}_{\mathit{fin}}^{F'}(s)$ and $p_\omega^{F'} = p_\omega^F$ for all $\omega \in \Omega' \setminus \{\omega_0\}$ and as $\Lambda \subseteq \mathit{Path}_{\mathit{fin}}^{F}(s)$ we obtain:

$$p_s^F = \sum_{\omega \in \Omega'} \mathbf{P}(\omega) \cdot p_\omega^A + \sum_{\lambda \in \Lambda} \mathbf{P}(\lambda) > \sum_{\omega \in \Omega'} \mathbf{P}(\omega) \cdot p_\omega^{A'} + \sum_{\lambda \in \Lambda} \mathbf{P}(\lambda) = p_s^{F'} \geq p_s^{adm}$$

Contradiction.

(iii) $\Longrightarrow$ (i): If $\omega \in \Omega$, $\mathit{last}(\omega) \in S^+$ and $\mu = F(\omega)$ then $p_s^{adm} = p_\omega^F = \sum_{t \in S} \mu(t) \cdot p_{\omega_t}^F = \sum_{t \in S} \mu(t) \cdot p_t^{adm}$ where ω_t is the path $\omega \xrightarrow{\mu} t$. Note that $p_{\omega_t}^F = p_t^{adm}$ for all $t \in \mathit{Sat}(\Phi_2) \cup (S \setminus \mathit{Sat}(\Phi_1))$. Hence, $\mu \in \mathit{AdmSteps}(s)$.

(i) $\Longrightarrow$ (ii): Let $\mathcal{S}^{adm} = (S, \mathit{AdmSteps})$. Let $\mathcal{A}_{simple}^{adm}$ be the set of simple adversaries for $\mathcal{S}^{adm}$. Let $p = \max\{p_s^A : A \in \mathcal{A}_{simple}^{adm}\}$. By Lemma 12.34 we get $p \leq p_s^{adm}$. As F is a strictly fair adversary of $\mathcal{S}^{adm}$, $p_s^F \leq p \leq p_s^{adm}$ (Lemma 12.25). ∎

Lemma 12.36 *There exists $F \in \mathcal{A}_{sfair}$ with $p_t^F(\Phi_1 \mathcal{U} \Phi_2) = p_t^{adm}(\Phi_1 \mathcal{U} \Phi_2)$ for all $t \in T^{adm}(\Phi_1, \Phi_2)$.*

Proof: similar to Lemma 12.32. ∎

Lemma 12.37 *If $s \notin T^{adm}(\Phi_1, \Phi_2)$ then $p_s^F(\Phi_1 \mathcal{U} \Phi_2) > p_s^{adm}(\Phi_1 \mathcal{U} \Phi_2)$ for all $F \in \mathcal{A}_{sfair}$.*

Proof: similar to Lemma 12.33. ∎

12.8 Proof of Theorem 9

This section gives a proof of Theorem 9, which follows by Lemma 12.38 and Lemma 12.47.

Lemma 12.38 *Let $A \in \mathcal{A}_{adm}^W$. Then, there exist an adversary F which is fair w.r.t. W and which satisfies $p_s^A(\Phi_1 \mathcal{U} \Phi_2) \geq p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S$.*

Proof: Similarly to Lemma 12.14 it can be shown that the measure of the set of critical fulpaths w.r.t. W in A is 0 where a fulpath $\pi \in \mathit{Path}_{\mathit{ful}}^A$ is called critical w.r.t. W iff $\pi \models \Box(\Phi_1 \wedge \neg \Phi_2)$ and $\mathit{inf}(\pi) \cap (S \setminus S_W^A) \neq \emptyset$. For an adversary B, let $\Gamma^B = \{\pi \in \mathit{Path}_{\mathit{ful}}^B : \pi \not\models \Phi_1 \mathcal{U} \Phi_2\}$. Let Ω^A be the set of all finite paths $\omega \in \mathit{Path}_{\mathit{fin}}^A$ with $\omega(i) \models \Phi_1 \wedge \neg \Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$, and $\mathit{last}(\omega) \in S_W^A$ and let $\Gamma_1^A = \bigcup_{\omega \in \Omega^A} \omega \uparrow^A$ and $\Gamma_2^A = \Gamma^A \setminus \Gamma_1^A$. Similarly to Claim 1 of Lemma 12.15 it can be shown that all paths $\pi \in \Gamma_2^A$ are critical w.r.t. W. Hence, $p_s^A = 1 - \mathit{Prob}(\Gamma^A(s)) = 1 - \mathit{Prob}(\Gamma_1^A(s)) = 1 - \sum_{\omega \in \Omega^A(s)} \mathbf{P}(\omega)$. Let F be a fair adversary with $\Omega^A \subseteq \mathit{Path}_{\mathit{fin}}^F$ (Lemma 12.10 and Remark 12.8). Then, F is fair w.r.t. W and $\Gamma^F(s)$ is a superset of the set of paths $\pi \in \mathit{Path}_{\mathit{ful}}^F(s)$ which have a prefix in Ω^A. Hence, $p_s^F = 1 - \mathit{Prob}(\Gamma^F(s)) \leq 1 - \sum_{\omega \in \Omega^A(s)} \mathbf{P}(\omega) = p_s^A$. ∎

Notation 12.39 *Let S_W^{adm} be the set of states $s \in S$ with $s \in S_W^A$ for some $A \in \mathcal{A}_{adm}^W$.*

Lemma 12.40 *There exists $A \in \mathcal{A}_{adm}^W$ with $S_W^A = S_W^{adm}$ and $p_s^A(\Phi_1 \mathcal{U} \Phi_2) = 0$ for all $s \in S_W^{adm}$.*

Proof: We define inductively subsets U^i and S^i of S. Let $U^0 = S^0$ and, for $i \geq 1$, $U^i = S^i \cup U^{i-1}$ where $S^0 = S \setminus S^+$ and, for $i \geq 1$, $S^i = S^{i,1} \cup S^{i,2}$. Here:

- $S^{i,1} = \{t \in S \setminus (U^{i-1} \cup \mathit{Sat}(\Phi_2)) : \mathit{Supp}(\mu_t) \subseteq U^{i-1}$ for some $\mu_t \in \mathit{Steps}(t)\}$.
- $S^{i,2}$ is the set of states $t \in T$ where $T \subseteq S \setminus (U^{i-1} \cup S^{i,1} \cup \mathit{Sat}(\Phi_2))$ such that:
  - $\mathit{Supp}(\mu) \subseteq T \cup U^{i-1} \cup S^{i,1}$ for all $t \in T \cap W$ and $\mu \in \mathit{Steps}(t)$

- for all $t \in T \setminus W$ there is some $\mu_t \in Steps(t)$ with $Supp(\mu_t) \subseteq T \cup U^{i-1} \cup S^{i,1}$.

Let $U = \bigcup_{i \geq 0} U^i$. Clearly, $S_W^{adm} \subseteq U$.

<u>*Claim*</u>: For all states $s \in S \setminus (U \cup Sat(\Phi_2))$ there exists a path $\omega \in Path_{fin}(s)$ with $\omega(i) \in S \setminus (U \cup Sat(\Phi_2))$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \models \Phi_2$.

<u>*Proof*</u>. It is clear that $S \setminus U \subseteq S^+$. We suppose that there is some $s \in S \setminus (U \cup Sat(\Phi_2))$ such that for each path $\omega \in Path_{fin}(s)$ with $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \models \Phi_2$ there is some $i < |\omega|$ with $\omega(i) \in U$. Let T be the set of states $t \in S$ with $t = last(\omega_t)$ for some $\omega_t \in Path_{fin}(s)$ with $\omega_t(i) \in S \setminus (U \cup Sat(\Phi_2))$ for all $i \leq |\omega_t|$. Then:

(1) $T \subseteq S \setminus (U \cup Sat(\Phi_2))$

(2) for all $t \in T$ and $\mu \in Steps(t)$: $Supp(\mu) \subseteq U \cup T$.[7]

By definition of U, (2) yields $T \subseteq U$. (More precisely, if $U = U^r$ then T is a subset of $S^{r+1,1}$.) Because of (1) we obtain $T = \emptyset$. Contradiction (as $s \in T$). ⌋

For $s \in S \setminus U$ let $\kappa(s)$ be the length of a shortest path $\omega \in Path_{fin}(s)$ with $\omega(i) \in S \setminus (U \cup Sat(\Phi_2))$, $i = 0, 1, \ldots, |\omega| - 1$, and $last(\omega) \models \Phi_2$. By induction on $\kappa(s)$ we define $\mu_s \in Steps(s)$. If $\kappa(s) = 1$ then we choose some $\mu_s \in Steps(s)$ with $\mu_s(t) > 0$ for some $t \in Sat(\Phi_2)$. If $\kappa(s) \geq 2$ then we choose some $\mu_s \in Steps(s)$ such that $\mu_s(u) > 0$ for some $u \in S \setminus U$ where $\kappa(u) = \kappa(s) - 1$. Let A be a simple adversary with

- $A(s) = \mu_s$ for all $s \in S \setminus U$
- $A(t) = \mu_t$ for all $t \in \bigcup_{i \geq 0} S^{i,1}$
  (where, for $t \in S^{i,1}$, $\mu_t \in Steps(t)$ such that $Supp(\mu_t) \subseteq U^{i-1}$).

Then, $S_W^A = U$ (by induction on i we get $S_W^{A,i} = S^i$). The claim yields that A is admissible for (Φ_1, Φ_2) w.r.t. W. Hence, $U = S_W^A \subseteq S_W^{adm}$. Thus, $S_W^A = U = S_W^{adm}$. Since $Reach^A_{\Phi_1 \wedge \neg\Phi_2}(u) \subseteq U$ for all $u \in U$ we get $p_u^A = 0$ for all $u \in U = S_W^{adm}$. ■

Notation 12.41 $S_W^0 = \{s \in S : p_s^F(\Phi_1 \mathcal{U} \Phi_2) = 0$ *for some* $F \in \mathcal{A}_{fair}^W\}$.

Lemma 12.42 $S_W^{adm} \subseteq S_W^0$

Proof: Let $A \in \mathcal{A}_{adm}^W$ with $S_W^A = S_W^{adm}$ (Lemma 12.40) and let $F \in \mathcal{A}_{fair}^W$ with $p_s^F \leq p_s^A$ for all $s \in S$ (Lemma 12.38). Then, for all $s \in S_W^{adm}$, $p_s^F = p_s^A = 0$. Hence, $s \in S_W^0$. ■

Remark 12.43 The proof of Lemma 12.40 (more precisely, the result $U = S_W^{adm}$) shows that whenever T is a subset of $S \setminus (S_W^{adm} \cup Sat(\Phi_2))$ such that

- for all $t \in T \cap W$, $\mu \in Steps(t)$: $Supp(\mu) \subseteq T \cup S_W^{adm}$.
- for all $t \in T \setminus W$ there exists $\mu \in Steps(t)$ such that $Supp(\mu) \subseteq T \cup S_W^{adm}$.

then $T = \emptyset$. ■

Lemma 12.44 *Let* $\pi \in Path_{ful}$ *such that* π *is fair w.r.t.* W*, state fair and* $\pi \not\models \Phi_1 \mathcal{U} \Phi_2$*. Then, there is some* $k \geq 0$ *with* $\pi(i) \models \Phi_1 \wedge \neg\Phi_2$*,* $i = 0, 1, \ldots, k-1$*, and* $\pi(k) \in S_W^{adm}$*.*

Proof: First we observe that, if $\pi(l) \in S_W^{adm}$ for some l then, with $k = \min\{l \geq 0 : \pi(l) \in S_W^{adm}\}$, $\pi(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i < k$. (Otherwise, for the smallest index $i < k$ with $\pi(i) \not\models \Phi_1 \wedge \neg\Phi_2$

[7] Note that $\mu(u) > 0$ for some $\mu \in Steps(t)$ and $u \notin U \cup T$ implies $u \in Sat(\Phi_2)$. Hence, $\omega_t \xrightarrow{\mu} u$ is a path leading from s to a Φ_2-state through states not belonging to U.

we have: either $\pi(i) \models \Phi_2$ which contradicts the assumption $\pi \not\models \Phi_1 \mathcal{U} \Phi_2$ or $\pi(i) \models \neg\Phi_1 \wedge \neg\Phi_2$ which implies $\pi(i) \in S_W^{adm}$ and contradicts the definition of k.)

We suppose that $\pi(k) \notin S_W^{adm}$ for all $k \geq 0$. Let $T = \mathit{inf}(\pi)$. Then, $T \subseteq S \setminus (S_W^{adm} \cup \mathit{Sat}(\Phi_2))$. For each $t \in T \cap W$ and $\mu \in \mathit{Steps}(t)$, we have $\mathit{Supp}(\mu) \subseteq T$. For each $t \in T \setminus W$, we choose some $\mu \in \mathit{Steps}(t)$ with $\mathit{step}(\pi, i) = \mu$ and $\pi(i) = t$ for infinitely many $i \geq 0$. Then, $\mathit{Supp}(\mu) \subseteq T$. Hence, $T = \emptyset$ by Remark 12.43. Contradiction. ■

Lemma 12.45 *If $F \in \mathcal{A}_{\mathit{fair}}^W$ and $p_s^F(\Phi_1 \mathcal{U} \Phi_2) < 1$ then $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^F(s) \cap S_W^{adm} \neq \emptyset$.*

Proof: We suppose $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^F(s) \cap S_W^{adm} = \emptyset$. Let Γ be the set of fulpaths $\pi \in \mathit{Path}_{\mathit{ful}}^F(s)$ such that π is fair w.r.t. W, state fair and $\pi \not\models \Phi_1 \mathcal{U} \Phi_2$. Then, $\mathit{Prob}(\Gamma) = 1 - p_s^F > 0$ (Lemma 12.6). Hence, $\Gamma \neq \emptyset$. By Lemma 12.44, $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^F(s) \cap S_W^{adm} \neq \emptyset$. ■

Lemma 12.46 *There exists $A \in \mathcal{A}_{adm}^W$ with $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^A(t) \subseteq S_W^0$ for all $t \in S_W^0$. Hence, $p_t^A(\Phi_1 \mathcal{U} \Phi_2) = 0$ for all $t \in S_W^0$.*

Proof: The argument is similar to that in the proof of Lemma 12.23. It can be shown that there exists $F \in \mathcal{A}_{\mathit{fair}}^W$ with $p_s^F = 0$ for all $s \in S_W^0$. Let $\Omega_{\Phi_1 \wedge \neg\Phi_2}^0$ be the set of paths $\omega \in \mathit{Path}_{\mathit{fin}}^F$ such that $\mathit{first}(\omega) \in S_W^0$ and $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$ for all $i < |\omega|$. Then, $p_\omega^F = 0$ and $\mathit{last}(\omega) \in S_W^0$ for all $\omega \in \Omega_{\Phi_1 \wedge \neg\Phi_2}^0$. For each $s \in S \setminus (S_W^0 \cap \mathit{Sat}(\Phi_1))$, let $\mathit{Steps}^F(s) = \mathit{Steps}(s)$. For each $s \in S_W^0 \cap \mathit{Sat}(\Phi_1)$ let $\mathit{Steps}^F(s)$ be the set of distributions $\mathit{step}(\omega, i)$ where $i < |\omega|$ such that $\omega(i) = s$ and $\omega \in \Omega_{\Phi_1 \wedge \neg\Phi_2}^0$. Then:

(*) $\mathit{Supp}(\mu) \subseteq S_W^0$ for all $t \in S_W^0$ and $\mu \in \mathit{Steps}^F(t)$.

Let A_0 be a simple adversary of $\mathcal{S}$ which is admissible w.r.t. W and $S_W^{A_0} = S_W^{adm}$ (Lemma 12.46). For $s \in S_W^{adm}$ we define $\mu_s = A_0(s)$. If $t \in S_W^0 \setminus S_W^{adm}$ then we define Ω_t to be the set of paths $\omega \in \mathit{Path}_{\mathit{fin}}(t, \mathcal{S}^F)$ with $\mathit{last}(\omega) \in S_W^{adm}$ and $\omega(i) \models \Phi_1 \wedge \neg\Phi_2$, $i = 0, 1, \ldots, |\omega| - 1$. Lemma 12.45 yields $\Omega_t \neq \emptyset$ for all $t \in S_W^0 \setminus S_W^{adm}$. For $t \in S_W^0 \cap S^+$ let $\kappa(t)$ be the length of shortest path in Ω_t. Then, $\kappa(t) \geq 1$. We choose some $\omega_t \in \Omega_t$ with $\kappa(t) = |\omega_t|$ and put $\mu_t = \mathit{step}(\omega_t, 0)$. For each $A \in \mathcal{A}_{\mathit{simple}}$ with $A(t) = \mu_t$ for all $t \in S_W^0$ we have:

(**) $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^A(t) \cap S_W^{adm} \neq \emptyset$

If $s \in S \setminus S^+$ then we choose an arbitrary distribution $\mu_s \in \mathit{Steps}(s)$. For $s \in S^+ \setminus S_W^0$ we define a distribution $\mu_s \in \mathit{Steps}(s)$ by induction on $\|s\|$. If $\|s\| = 0$ then $s \in \mathit{Sat}(\Phi_2)$ and we choose an arbitrary $\mu_s \in \mathit{Steps}(s)$. If $\|s\| \geq 1$ then we choose some $\mu_s \in \mathit{Steps}(s)$ such that $\mu_s(w) > 0$ for some $w \in S^+$ with $\|w\| < \|s\|$. Let A be the simple adversary with $A(s) = \mu_s$ for all $s \in S$. Then, $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^A(s) \cap (\mathit{Sat}(\Phi_2) \cup (S_W^0 \cap S^+)) \neq \emptyset$ for all $s \in S^+ \setminus S_W^0$. By (**), $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^A(s) \cap (\mathit{Sat}(\Phi_2) \cup S_W^{adm}) \neq \emptyset$ for all $s \in S^+$. Since $A(s) = A_0(s)$ for all $s \in S_W^{adm}$ we obtain $S_W^A \supseteq S_W^{A_0} = S_W^{adm}$. Hence, $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^A(s) \cap (\mathit{Sat}(\Phi_2) \cup S_W^A) \neq \emptyset$ for all $s \in S^+$. Hence, A is admissible for (Φ_1, Φ_2) w.r.t. W (and $S_W^A = S_W^{adm}$). By (*), $\mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}^A(t) \subseteq S_W^0$ for all $t \in S_W^0$. Thus, $p_t^A = 0$ for all $t \in S_W^0$. ■

Lemma 12.47 *There exists $A \in \mathcal{A}_{adm}^W$ with $p_s^A(\Phi_1 \mathcal{U} \Phi_2) \leq p_s^F(\Phi_1 \mathcal{U} \Phi_2)$ for all $s \in S$, $F \in \mathcal{A}_{\mathit{fair}}^W$.*

Proof: The argument is almost the same as in the proof of Lemma 12.25, the only essential difference being that we deal with $S_W^1 = \{v \in S^+ : \mathit{Reach}_{\Phi_1 \wedge \neg\Phi_2}(v) \cap S_W^0 = \emptyset\}$ rather than S^1 and S_W^0 rather than S^0.

Remark 12.48 By Lemma 12.42 and 12.47 we get: $S_W^{adm} = S_W^0$. ■

References

[1] B. Alpern, F. Schneider: Defining Liveness, Information Processing Letters, Vol. 21, pp 181-185, 1985.

[2] R. Alur, C. Courcoubetis, D. Dill: Model-Checking for Probabilistic Real-Time Systems, Proc. ICALP'91, LNCS 510, pp 115-127, 1991.

[3] R. Alur, C. Courcoubetis, D. Dill: Verifying Automata Specifications of Probabilistic Real-Time Systems, Proc. REX Workshop'91, LNCS 600, pp 27-44, 1991.

[4] L. de Alfaro: Temporal Logics for the Specification of Performance and Reliability, Proc. STACS'97, Lecture Notes in Computer Science 1200, pp 165-176, 1997.

[5] L. de Alfaro: private communication, 1996.

[6] A. Aziz, V. Singhal, F. Balarin, R. Brayton, A. Sangiovanni-Vincentelli: It usually works: The Temporal Logic of Stochastic Systems, Proc. CAV'95, LNCS 939, pp 155-165, 1995.

[7] A. Aho, J. Hopcroft, J. Ullman: The Design and Analysis of of Computer Algorithms, Addison-Wesley Publishing Company, 1974.

[8] C. Baier: Polynomial Time Algorithms for Testing Probabilistic Bisimulation and Simulation, Proc. CAV'96, LNCS 1102, pp 38-49, 1996.

[9] C. Baier, E. Clarke, V. Hartonas-Gramhausen, M. Kwiatkowska, M. Ryan: Symbolic Model Checking for Probabilistic Processes, Proc. ICALP'97, LNCS 1256, pp 430-440, 1997.

[10] C. Baier, H. Hermanns: Weak Bisimulation for Fully Probabilistic Processes, Proc. CAV'97, LNCS 1254, pp 119-130, 1997.

[11] C. Baier, M.Z. Kwiatkowska: Automatic Verification of Liveness Properties of Randomized Systems, Proc. PODC'97, ACM Press, 1997.

[12] C. Baier, M.Z. Kwiatkowska: On the Verification of Qualitative Properties of Probabilistic Processes under Fairness Constraints, to appear in Information Processing Letters.

[13] C. Baier, M.Z. Kwiatkowska, G. Norman: Computing Lower and Upper Bounds for *LTL* Formulae over Sequential and Concurrent Markov Chains, to appear in Proc. PROBMIV'98, Indianapolis, 1998.

[14] A. Bianco, L. de Alfaro: Model Checking of Probabilistic and Nondeterministic Systems, Proc. Foundations of Software Technology and Theoretical Computer Science, LNCS 1026, pp 499-513, 1995.

[15] L. Christoff, I. Christoff: Reasoning about Safety and Liveness Properties for Probabilistic Processes, Proc. 12th Conference on Foundations of Software Technology and Theoretical Computer Science, LNCS 652, pp 342-355, 1992.

[16] E.M. Clarke, E.A. Emerson, A.P. Sistla: Automatic Verification of Finite State Concurrent Systems Using Temporal Logic Specifications: A Pratical Approach, Proc. POPL'83, 1983.

[17] E. Clarke, M. Fujita, X. Zhao: Multi-Terminal Binary Decision Diagrams and Hybrid Decision Diagrams, Representations of Discrete Functions. In T. Sasao and M. Fujita, eds, Kluwer Academic Publishers, pp 93-108, 1996.

[18] R. Cleaveland, S. Smolka, A. Zwarico: Testing Preorders for Probabilistic Processes, Proc. ICALP'92, LNCS 623, pp 708-719, Springer-Verlag, 1992.

[19] C. Courcoubetis, M. Yannakakis: Verifying Temporal Properties of Finite-State Probabilistic Programs, Proc. FOCS'88, pp 338-345, 1988.

[20] C. Courcoubetis, M. Yannakakis: Markov Decision Processes and Regular Events, Proc. ICALP'90, LNCS 443, pp 336-349, 1990.

[21] C. Courcoubetis, M. Yannakakis: The Complexity of Probabilistic Verification, Journal of the ACM, Vol. 42, No. 4, pp 857-907, 1995.

[22] C. Derman: Finite-State Markovian Decision Processes, Academic Press, New York, 1970.

[23] E.A. Emerson, E.M. Clarke: Using Branching Time Logic to Synthesize Synchronization Skeletons, Sci. Comput. Programming 2, pp 241-266, 1982.

[24] E.A. Emerson, J. Halpern: Decision Procedures and Expressiveness in the Temporal Logic of Branching Time, Journal of Computer and System Science, Vol. 30, pp 1-24, 1985.

[25] E.A. Emerson, C. Lei: Modalities for Model Checking: Branching Time Strikes Back, Proc. POPL'85, pp 84-96, 1985.

[26] Y. Feldmann: A Decidable Propositional Dynamic Logic, Proc. 15th ACM Symp. on Theory of Computing, pp 298-309, 1983.

[27] A. Giacalone, C. Jou, S. Smolka: Algebraic Reasoning for Probabilistic Concurrent Systems, Proc. IFIP TC2 Working Conference on Programming Concepts and Methods, 1990.

[28] R. van Glabbeek, S. Smolka, B. Steffen, C. Tofts: Reactive, Generative, and Stratified Models for Probabilistic Processes, Proc. LICS'90, pp 130-141, 1990.

[29] P. Halmos: Measure Theory, Springer-Verlag, 1950.

[30] H. Hansson: Time and Probability in Formal Design of Distributed Systems, Real-Time Safety Critical Systems, Elsevier, 1994.

[31] H. Hansson, B. Jonsson: A Logic for Reasoning about Time and Probability, Formal Aspects of Computing, Vol. 6, pp 512-535, 1994.

[32] S. Hart, M. Sharir: Probabilistic Temporal Logic for Finite and Bounded Models, Proc. 16th ACM Symposium on Theory of Computing, pp 1-13, 1984.

[33] S. Hart, M. Sharir, A. Pnueli: Termination of Probabilistic Concurrent Programs, ACM Transactions on Programming Languages, Vol. 5, pp 356-380, 1983.

[34] M. Huth, M.Z. Kwiatkowska: Quantitative Analysis and Model Checking, Proc. LICS'97, pp 111-122, 1997.

[35] M. Huth and M.Z. Kwiatkowska. Comparing CTL and PCTL on Labeled Markov Chains. Proc. PROCOMET'98, IFIP, Chapman & Hall, 1998.

[36] T. Huynh, L. Tian: On some Equivalence Relations for Probabilistic Processes, Fundamenta Informaticae, Vol. 17, pp 211-234, 1992.

[37] P. Iyer, M. Narasima: "Almost always" and "definitely sometime" are not enough: Probabilistic quantifiers and probabilistic model-checking, Techn. Report, TR-96-16, North Carolina State University, 1996.

[38] B. Jonsson, K.G. Larsen: Specification and Refinement of Probabilistic Processes, Proc. LICS'91, pp 266-277, 1991.

[39] B. Jonsson, W. Yi: Compositional Testing Preorders for Probabilistic Processes, Proc. LICS'95, pp 431-443, 1995.

[40] D. Kozen: A Probabilistic PDL, Journal of Computer and System Sciences, Vol. 30, pp 291-297, 1985.

[41] K. Larsen, A. Skou: Bisimulation through Probabilistic Testing, Information and Computation, Vol. 94, pp 1-28, 1991.

[42] K. Larsen, A. Skou: Compositional Verification of Probabilistic Processes, Proc. CONCUR'92, LNCS 630, pp 456-471, 1992.

[43] D. Lehmann, S. Shelah: Reasoning with Time and Chance, Information and Control, Vol. 53, pp 165-198, 1982.

[44] O. Lichtenstein, A. Pnueli: Checking that Finite State Concurrent Programs Satisfy Their Linear Specification, Proc. POPL'85, pp 97-107, 1985.

[45] N. Lynch, I. Saias, R. Segala: Proving Time Bounds for Randomized Distributed Algorithms, Proc. PODC'94, pp 314-323, 1994.

[46] C. Morgan, A. McIver: A probabilistic temporal calculus based on expectations, Technical Report TR-13-97, University of Oxford, 1997.

[47] G.J. Norman. Metric semantics for reactive probabilistic processes. *Ph.D Thesis, School of Computer Science, The University of Birmingham, November 1997.*

[48] M. Núñez, D. de Frutos, L. Llana: Acceptance Trees for Probabilistic Processes, Proc. CONCUR'95, LNCS 962, pp 249-263, 1995.

[49] A. Pnueli: On the Extremely Fair Treatment of Probabilistic Algorithms, Proc. 15th ACM Symposium on Theory of Computing, pp 278-290, 1983.

[50] A. Pnueli, L. Zuck: Verification of Multiprocess Probabilistic Protocols, Distributed Computing, Vol. 1, No. 1, pp 53-72, 1986.

[51] A. Pnueli, L. Zuck: Probabilistic Verification, Information and Computation, Vol. 103, pp 1-29, 1993.

[52] A. Pogosyants, R. Segala: Formal Verification of Timed Properties of Randomized Distributed Algorithms, Proc. PODC, 1995.

[53] S. Ross: Introduction to Stochastic Dynamic Programming, Academic Press, New York, 1983.

[54] S. Safra: On the Complexity of ω-Automata, Proc. FOCS'88, pp 319-327, 1988.

[55] R. Segala, N. Lynch: Probabilistic Simulations for Probabilistic Processes, Proc. CONCUR'94, LNCS 836, pp 481-496, 1994.

[56] R. Segala: Modelling and Verification of Randomized Distributed Real-Time Systems, Ph.D. Thesis. Massachusetts Institute of Technology, 1995.

[57] R. Segala: A Compositional Trace-Based Semantics for Probabilistic Automata, Proc. CONCUR'95, LNCS 962, pp 234-248, 1995.

[58] M. Vardi: Automatic Verification of Probabilistic Concurrent Finite-State Programs, Proc. FOCS'85, pp 327-338, 1985.

[59] M. Vardi, P. Wolper: An Automata-Theoretic Approach to Automatic Program Verification, Proc. LICS'86, pp 332-344, 1986.

[60] S. Yuen, R. Cleaveland, Z. Dayar, S. Smolka: Fully Abstract Characterizations of Testing Preorders for Probabilistic Processes, Proc. CONCUR'94, pp 497-512, 1994.

[61] W. Yi: Algebraic Reasoning for Real-Time Probabilistic Processes with Uncertain Information, Proc. FTRTFT'94, LNCS 863, pp 680-693, 1994.

[62] W. Yi, K. Larsen: Testing Probabilistic and Nondeterministic Processes, Proc. Protocol Specification, Testing and Verification XII, Elsevier Science Publishers B.V. (North-Holland), pp 47-61, 1992.