Coin Lemmas with Random Variables

Katia Folegati and Roberto Segala

Department of Computer Science
University of Bologna - Italy

Abstract. Coin lemmas are a tool for decoupling probabilistic and nondeterministic arguments in the analysis of concurrent probabilistic systems. They have revealed to be fundamental in the analysis of randomized distributed algorithms, where the interplay between probability and nondeterminism has proved to be subtle and difficult to handle.
We reformulate coin lemmas in terms of random variables obtaining a new collection of coin lemmas that is independent of the underlying computational model and of more general applicability to the study of concurrent nondeterministic probabilistic systems.

1 Introduction

Coin lemmas are a tool for decoupling probabilistic and nondeterministic arguments in the analysis of concurrent probabilistic systems [12]. They have revealed to be fundamental in the analysis of randomized distributed algorithms [7, 8], where the interplay between probability and nondeterminism has proved to be subtle and difficult to handle [10].

Coin lemmas are formulated in the framework of probabilistic automata [11], a probabilistic extension of labeled transition systems (automata) where the notion of transition is extended so that a transition from some state s leads to a discrete probability distribution over states rather than to a single state. Each state enables several transitions, and the choice of which transitions to schedule is left unspecified. Nondeterminism is resolved by an entity called a *scheduler*, and the result of resolving nondeterminism can be described as an acyclic Markov process, called a *probabilistic execution*. Properties that involve probability can be studied on probabilistic executions, while typical arguments about probabilistic automata involve proving upper and lower bounds on the probabilities of events under the action of any scheduler, the events being sets of paths in the Markov process that represents a probabilistic execution.

A coin lemma is a tool to prove upper and lower bounds on probabilities of events within any probabilistic execution. Specifically, a coin lemma provides us with a rule to associate an event with each probabilistic execution together with a lower bound on the probability of the associated events. Typically the events associated with a probabilistic execution are derived from events in a known stochastic process, and similarly, the lower bounds on probabilities are derived from the known stochastic process. The analysis of a concurrent probabilistic system is then reduced to the analysis of the events returned by the rules, which, being just ordinary sets of paths, do not include any probability.

1.1 Coin Lemmas in the Setting of Algorithms

To understand better the role of coin lemmas we illustrate what happens in the area of distributed algorithms, which is the area where coin lemmas were proposed first. When studying a distributed algorithm the objective is to state that for every scheduler (the entity that decides the temporal ordering of the events at different processors) the probability of successful termination is at least some number p. That is, translated in terms of probabilistic automata, for each probabilistic execution the probability of the event that expresses successful termination is at least p.

In practice it is more convenient to show that a sub-event of successful termination has probability at least p, the sub-event expressing the fact that some random draws that occur in the algorithm give some specific results.

Example 1. The randomized algorithm of Lehmann and Rabin for the dining philosophers problem [6] works as follows: whenever a philosopher wants to eat he/she flips a coin to decide which fork to pick up first, waits for the fork to be free and picks it up, and finally checks whether the other fork is free. If the other fork is free, then the philosopher picks it up and eats, otherwise the philosopher puts down the first fork and starts again. It turns out that if two neighbor philosophers draw opposite coins at a certain stage, then the algorithm terminates successfully and one philosopher eats. However, the algorithm may terminate successfully even if the coins are not opposite. Studying the sub-event of successful termination that states that two coins are opposite is enough anyway to state that there is a high chance that some philosophers eat eventually.

In summary, the designer of an algorithm thinks of a stochastic process that supposedly occurs in any probabilistic execution, thinks of the fact that whenever the stochastic process gives some specific results R the algorithm terminates successfully, and concludes that the algorithm terminates successfully with a probability that is as high as the probability of R. Unfortunately, in most of the cases the stochastic process that the designer has in mind does not take place since, for example, the algorithm may terminate before the stochastic process completes.

Since designers rarely consider explicitly the fact that the stochastic process they have in mind may not complete, they often come to the wrong conclusion that an algorithm is correct whenever it is possible that the stochastic process does not complete and at the same time the algorithm does not terminate successfully.

Example 2. Several randomized algorithms in the literature are based on the following argument. The processors communicate with each other and at the end of the communication it turns out that some of them have participated in a game. If there is a unique winner in the game, then the algorithm terminates. The game consists of flipping fair coins until a head comes out and counting how many coins are flipped. Thus, the probability of drawing number i is $1/2^i$. The winner is the player who draws the highest number. We know that, given that there are k players, the probability of a unique winner is at least some constant c independent of k. Should we conclude that the algorithm terminates successfully with probability at least c?

Although in the literature we can find statements like the one above, the correctness of the statement depends on how the number of players is chosen. If the players are chosen before starting the game, then the lower bound c does hold. Otherwise, we may add players to the game until we obtain two winners, which would violate considerably the lower bound c.

Coin lemmas have the scope of formalizing the arguments that designers usually put forward about their algorithms and of avoiding the pitfalls given by incomplete stochastic processes. Thus, given a stochastic process, a coin lemma provides a mechanical rule to pick up an event in each probabilistic execution, the event representing the successful part of the stochastic process, and provides a lower bound on the probabilities of the chosen events that is the same as the probability of the successful part of the stochastic process. The main problem is to decide what event to return whenever the stochastic process does not complete in a probabilistic execution.

Viewing a stochastic process as a sequence of elementary experiments, the rule of a coin lemma identifies all the experiments that do take place in an execution (a path of the probabilistic execution) and their results. Each path in a probabilistic execution appears in the event returned by the rule if it is possible to fix the outcomes of the experiments that do not take place so that the underlying stochastic process is successful. This idea is called "the essence of a coin lemma" in [12], since it is the common denominator of all the coin lemmas that have been proposed so far.

The advantage of this formulation of coin lemmas is that the designer of an algorithm is given a set of paths (ordinary executions) that must be shown to lead to successful termination if he/she wants to conclude that the algorithm is successful with some minimum probability. The executions can be analyzed without resorting to probabilities. Furthermore, the rule would say something like "if a certain coin is not flipped then the resulting execution must be shown to lead to successful termination". Thus, the designer of the algorithm will be forced to prove either that the coin is always flipped or that the algorithm is indeed successful whenever the coin is not flipped.

Example 3. The coin lemma that we should use for the argument of Example 2 would be associated with the stochastic process where k numbers are drawn. Assuming that we have some way to decide which k to use, the scheduler that adds players until there are two winners would be discovered immediately since in the event returned by the rule we would have several executions where less than k numbers are drawn.

1.2 Our Contribution

A weak point in the current formulations of coin lemmas [12] is that the rules are formulated in a very restrictive language. As a result, many times simple and obvious variations to the rules require the proof of a new coin lemma. In this paper we propose a new formulation of coin lemmas based on random variables that gives us much more flexibility in the formulation of a rule without forcing us to prove new coin lemmas each time. Another advantage of the new formulation is that the concept of a random variable is independent of the structure of the probability space that represents a probabilistic execution. Thus, the new coin lemmas can be used easily on several models. On the contrary, the current formulation is heavily based on the structure of the transitions of a probabilistic automaton.

If we express a stochastic process as a sequence of elementary random experiments, the main objective for the formulation of a rule is to identify the places in an execution where each experiment takes place and its corresponding outcome. For this reason, we start with the concept of an *experiment*, which is a pair (D, U) where D is a $0/1$-valued random variable that identifies the places where the experiment takes place, and U is a random variable that identifies the successful outcomes. An experiment has taken place in an execution α whenever $D(\alpha) = 1$. We define the notion of a p-successful experiment, which means that the probability of success is at least p whenever the experiment takes place. Then, we prove that in each probabilistic execution the probability that either a p-successful experiment does not take place or that the experiment takes place and gives successful results is at least p.

The result about p-successful experiments allows us to capture immediately the coin lemmas of [12] that deal with single occurrences of actions; however, many more general properties can be captured easily like the intricate coin lemma necessary for the proof of correctness of the consensus algorithm of Ben-Or [1] (see [11] for a proof of correctness based on coin lemmas). The identification of the intermediate concept of p-successful experiment and the use of random variables turned out to be a considerable improvement over the formalization of [12].

We elaborate further on the concept of p-successful experiments by showing how several experiments can be combined into a single experiment. In this way we can capture easily the coin lemmas of [12] that look at the outcome of the first experiment that takes place among many experiments.

Finally, we show how to handle more general stochastic processes composed of finite or countable sequences of experiments. We derive two different coin lemmas. In both coin lemmas we consider a collection of experiments $\{(D_i, X_i)\}_{i \in I}$ and a measurable function $f(X_1, X_2, \ldots)$ with values in $\{0, 1\}$. Then we identify the probability distribution of the X_i's under the assumption that all the experiments take place, and we prove a lower bound on the probability of the event that considers all those cases where it is possible to fix the outcomes of the experiments that do not take place so that f evaluates to 1. Once again, if we assume to be successful whenever f evaluates to 1, the principle is that if some experiment does not take place and one of its possible outcomes is considered as successful, then we should make sure that the system under examination behaves correctly.

The rest of the paper is structured as follows. Section 2 gives the necessary background on probability theory and introduces some notation; Section 3 gives a description of probabilistic automata; Section 4 introduces the coin lemmas for single experiments and compares them with the formulation of [12]; Section 5 proves properties of single experiments that allow us to combine several experiments into a unique new experiment; Section 6 introduces the coin lemmas for general stochastic processes; Section 7 gives some concluding remarks.

2 Preliminaries

A σ-field over a set X is a set $\mathcal{F} \subseteq 2^X$ that includes X and is closed under complement and countable union. Observe that 2^X is a σ-field over X. A *measurable space* is a pair $(X, \mathcal{F})$ where X is a set and $\mathcal{F}$ is a σ-field over X. The set X is also called the *sample space*. A measurable space $(X, \mathcal{F})$ is called *discrete* if $\mathcal{F} = 2^X$. In the paper we work mostly with discrete measurable spaces except when we deal with probabilistic executions.

Given a measurable space $(X, \mathcal{F})$, a finite *measure* over $(X, \mathcal{F})$ is a function $\mu : \mathcal{F} \to \mathbb{R}^{\geq 0}$ such that, for each countable collection $\{X_i\}_{i \in I}$ of pairwise disjoint elements of $\mathcal{F}$, $\mu(\cup_I X_i) = \sum_I \mu(X_i)$. For the purpose of this paper we are interested in measures μ where $\mu(X) \leq 1$. A *probability measure* over a measurable space $(X, \mathcal{F})$ is a measure μ over $(X, \mathcal{F})$ such that $\mu(X) = 1$; a *sub-probability measure* over $(X, \mathcal{F})$ is a measure μ over $\mathcal{F}$ such that $\mu(X) \leq 1$. A measure over a discrete measurable space is called a *discrete measure*. We also say that a discrete measure over $(X, 2^X)$ is a discrete measure over X. Sometimes we refer to probability measures as *distributions*.

Given a set X, denote by $Disc(X)$ the set of discrete probability measures on the measurable space $(X, 2^X)$, and denote by $SubDisc(X)$ the set of discrete sub-probability measures on the measurable space $(X, 2^X)$. We call a discrete probability measure a *Dirac* measure if it is concentrated at a single point, say x_0, by assigning measure 1 to any set containing x_0 and measure 0 to any other set. In the paper we use sub-probability measures to denote the fact that sometimes there is no progress from some state s. We view a discrete sub-probability measure μ over X as a discrete probability measure over $X \cup \{\perp\}$ where $\perp$ denotes the fact that there is no progress and $\mu(\perp) = 1 - \mu(X)$ denotes the probability of not making any progress. When $\mu(X) = 0$, then we can think of μ as concentrated at $\perp$. For this reason we abuse notation and say that a discrete sub-probability measure μ over X is Dirac if either μ is a Dirac probability measure or $\mu(\perp) = 1$. We drop the set notation whenever a set is singleton.

A function $f : X_1 \to X_2$ is said to be a *measurable function* from a measurable space $(X_1, \mathcal{F}_1)$ to a measurable space $(X_2, \mathcal{F}_2)$ if for each element C of $\mathcal{F}_2$, $f^{-1}(C) \in \mathcal{F}_1$. In such case, given a measure μ on $(X_1, \mathcal{F}_1)$, we can define the measure *induced* by f, denoted by $f(\mu)$, as $f(\mu)(C) = \mu(f^{-1}(C))$. A *random variable* defined on a measurable space $(X, \mathcal{F})$ is a measurable function from $(X, \mathcal{F})$ to $(\mathbb{R}, B)$, where B denotes the σ-field generated by the open intervals of the reals, also called the *Borel σ-field* over the reals.

3 The Model

In this section we give an overview of probabilistic automata. We omit the distinction between internal and external actions since it is irrelevant for the purpose of this paper. The reader interested in more details is referred to [11].

3.1 Probabilistic Automata

A *probabilistic automaton* $\mathcal{A}$ is a tuple $(S, \bar{s}, \Sigma, \mathcal{D})$ where S is a set of *states*, $\bar{s} \in S$ is a *start state*, Σ is a set of *actions*, and $\mathcal{D} \subseteq S \times \Sigma \times Disc(S)$ is a *transition relation*.

For convenience, states are ranged over by r, q, s; actions are ranged over by a, b, c; discrete distributions are ranged over by μ. We denote the elements of a probabilistic automaton $\mathcal{A}$ by $S, \bar{s}, \Sigma, \mathcal{D}$, and we propagate primes and indicies. Thus, the elements of a probabilistic automaton $\mathcal{A}'_i$ are denoted by $S'_i, \bar{s}'_i, \Sigma'_i, \mathcal{D}'_i$.

We call an element of $\mathcal{D}$ a *transition*. We say that a transition (s, a, μ) is *enabled* from s and is *labeled* by a. We also say that s *enables* (s, a, μ). We call a transition (s, a, μ) *Dirac* if μ is a Dirac measure. For a state s we denote by $\mathcal{D}(s)$ the set of transitions of $\mathcal{D}$ that are enabled from s.

We call a state s *Dirac* if all the transitions enabled from s are Dirac; we call s *deterministic* if it enables at most one transition for each action; we call s *singleton* if it enables at most one transition. We say that a probabilistic automaton $\mathcal{A}$ is *Dirac*, *deterministic*, *singleton*, if each state of $\mathcal{A}$ is Dirac, deterministic, singleton, respectively.

For comparison with other models, probabilistic automata can be seen as an extension of ordinary automata (also called labeled transition systems) since an ordinary automaton is essentially a Dirac probabilistic automaton. The reactive systems of [5] are deterministic probabilistic automata; thus, probabilistic automata could be called alternatively *nondeterministic reactive systems*. The probabilistic automata of [9] are equivalent to probabilistic automata. Markov Decision Processes [2] are equivalent to deterministic probabilistic automata. The alternating model of [3] can be seen as a probabilistic automaton where each state is either Dirac or singleton.

3.2 Executions

A *potential execution* of a probabilistic automaton $\mathcal{A}$ is a finite or infinite sequence of alternate states and actions, $\alpha = s_0 a_1 s_1 a_2 s_2 \cdots$, starting from a state and, if the sequence is finite, ending with a state. Define the length of α, denoted by $|\alpha|$, to be the number of occurrences of actions in α. If α is an infinite sequence, then $|\alpha| = \infty$. For a natural number $i \leq |\alpha|$, denote by $\alpha[i]$ the state s_i. In particular, $\alpha[0]$ is the start state of α. If α is finite, then denote by $\alpha[\perp]$ the last state of α.

Two potential executions $\alpha = s_0 a_1 s_1 \cdots a_n s_n$ and $\alpha' = s'_0 a'_1 s'_1 \cdots$ can be concatenated if $s_n = s'_0$. In such case the concatenation, denoted by $\alpha \frown \alpha'$, is the potential execution $s_0 a_1 s_1 \cdots a_n s_n a'_1 s'_1 \cdots$. If $\alpha = \alpha_1 \frown \alpha_2$, then we say that α_1 is a *prefix* of α, denoted by $\alpha_1 \leq \alpha$.

An *execution* of a probabilistic automaton $\mathcal{A}$ is a potential execution of $\mathcal{A}$, $\alpha = s_0 a_1 s_1 a_2 s_2 \cdots$, such that for each $i < |\alpha|$ there exists a transition (s_{i-1}, a_i, μ_i) of $\mathcal{D}$ where $\mu_i(s_i) > 0$. An execution α is said to be *initial* if $\alpha[0] = \bar{s}$. Denote by $execs(\mathcal{A})$ the set of executions of $\mathcal{A}$ and by $iexecs(\mathcal{A})$ the set of initial executions of $\mathcal{A}$. Similarly, denote by $execs^*(\mathcal{A})$ and $iexecs^*(\mathcal{A})$ the set of finite executions and finite initial executions, respectively, of $\mathcal{A}$.

An execution is the result of resolving both nondeterminism and probability in a probabilistic automaton and records the sequence of states that the system goes through, along with the sequence of actions it engages in. In a probabilistic setting it is useful to know the probability distributions over executions that arise after resolving the nondeterminism. This is the argument of the next section.

3.3 Schedulers

A *scheduler* for a probabilistic automaton $\mathcal{A}$ is a function $\sigma : execs^*(\mathcal{A}) \rightarrow SubDisc(\mathcal{D})$ such that for each finite execution α, $\sigma(\alpha) \in SubDisc(\mathcal{D}(\alpha[\bot]))$. We say that a scheduler is *Dirac* if it assigns a Dirac measure to each execution.

In other words, a scheduler is an entity that given a finite execution α chooses arbitrarily either to stop or to perform one of the transitions enabled from the last state of α, possibly using randomization in its choices. Observe that the choice of a scheduler is based on the full history. A scheduler is the entity that resolves the nondeterminism in a probabilistic automaton. In the context of algorithms a scheduler is usually called an *adversary* since it is seen as an entity that tries to degrade the performance of an algorithm as much as possible (the worst case scenario is what matters); in the context of Markov Decision Processes an adversary is called a *policy* since the objective of the area of Markov Decision Processes is to find the best possible strategy to improve performance. What matters from our point of view is that schedulers, adversaries and policies are the same entity, which is the entity that resolves nondeterminism.

In the literature there is a distinction between probabilistic and deterministic schedulers. In this paper we have chosen to use the word scheduler for a probabilistic scheduler and the word Dirac scheduler for a nondeterministic scheduler. The reason for our choice is to avoid overloading the term "nondeterministic".

Consider a scheduler σ and a finite execution α with last state q. The distribution $\sigma(\alpha)$ describes how to move from q. Specifically, one of the transitions enabled from q is selected randomly according to $\sigma(\alpha)$; then a target state is chosen according to the selected transition. The result of the action of σ is a pair $(q, \mu_{\sigma(\alpha)})$, which we call the *combined transition* according to $\sigma(\alpha)$, where $\mu_{\sigma(\alpha)}$ is a distribution of $SubDisc(\Sigma \times S)$ defined as follows: for each pair (a, s),

$$\mu_{\sigma(\alpha)}((a,s)) \triangleq \sum_{(q,a,\mu) \in \mathcal{D}} \sigma(\alpha)((q,a,\mu))\mu(s).$$

The result of the action of a scheduler from a state s can be represented as a Markov process whose states are executions of $\mathcal{A}$ with start state s. We call these objects *probabilistic executions*. Formally, the probabilistic execution of $\mathcal{A}$ induced by a scheduler σ and a state s is a process (Q,μ) where Q is the set of executions of $\mathcal{A}$ that start with s, and μ is 0 except for pairs of the form (q,qar) where it is defined as follows: $\mu(q,qar) = \mu_{\sigma(q)}((a,r))$.

Given a scheduler σ and a state s we can define a probability measure $\mu_{\sigma,s}$ on the paths of the induced probabilistic execution. The measure is defined according to the cone construction, where a cone is a set of the form $C_\alpha = \{\alpha' \mid \alpha \leq \alpha'\}$, α being a finite execution that starts with s, and $\mu_{\sigma,s}$ is defined on the σ-field generated by the cones. The definition of $\mu_{\sigma,s}$ on the cones is given inductively as follows:

1. $\mu_{\sigma,s}(C_s) \triangleq 1$;
2. $\mu_{\sigma,s}(C_{\alpha a q}) \triangleq \mu_{\sigma,s}(C_\alpha)\mu_{\sigma(\alpha)}((a,q))$.

It is a standard argument [4] to show that $\mu_{\sigma,s}$ is σ-additive and that it can be extended uniquely to the σ-field generated by the cones.

Throughout the paper we usually refer to a probabilistic execution as the corresponding measure over executions $\mu_{\sigma,s}$ and we omit either σ or s whenever they are not needed or clear from the context.

4 Coin Lemmas for a Single Experiment

In this section we give our new formulation of a coin lemma for a single experiment and we compare it with the formulation of [12]. We start with a simple example that illustrates the structure of a coin lemma.

Example 4. Consider the experiment of rolling a die with 6 faces. We know that each face has probability 1/6. If we have a probabilistic automaton where a transition encodes the action of rolling a die and where a beep signal is sent whenever the outcome of the die is an even number, then we could say that in each probabilistic execution the probability of observing a beep signal is 1/2. However, it may be the case that a die is not rolled in every probabilistic execution; thus, a correct statement would be that in each probabilistic execution the probability of either not rolling any die or observing a beep signal is at least 1/2. This simple observation is at the base of the formulation of a coin lemma. Although the statement seems to be obvious, when the number of experiments grows the result is not obvious any more, and indeed in several occasions the principle was not applied correctly.

4.1 The Old Formulation of a Coin Lemma

We formulate a coin lemma for a single experiment as it is formulated in [12].

Lemma 1. *Let $\mathcal{A}$ be a probabilistic automaton, and let (a,U) be a pair consisting of an action a of $\mathcal{A}$ and a set of states U of $\mathcal{A}$. Let p be a real number between 0 and 1 such that for each transition (s,a,μ) of $\mathcal{A}$, $\mu(U) \geq p$.*

For each probabilistic execution μ_σ of $\mathcal{A}$ let $FIRST(a,U)(\mu_\sigma)$ be the set of executions α of $\mathcal{A}$ such that either a does not occur in α, or a occurs in α and the state reached after the first occurrence of a is a state of U.

Then, for each probabilistic execution μ_σ of $\mathcal{A}$, $\mu_\sigma(FIRST(a,U)(\mu_\sigma)) \geq p$.

The experiment described by the coin lemma above consists of performing a transition labeled by a, and checking whether a state from U is reached after performing a. If there are multiple occurrences of a-labeled transitions, then we observe the first occurrence. The hypothesis of the lemma requires that in each a-labeled transition the probability of reaching a state from U is at least p. The conclusion states that the probability of either not performing any a-labeled transition or reaching a state from U after performing the first a-labeled transition is at least p. It is easy to generalize Lemma 1 to a coin lemma where the i-th occurrence of an action a is observed.

Example 5. Returning to the example of rolling a die, we need a special action, say *roll*, to label all transitions where a die is rolled. The set U is the set of states where the die gives an even number, and p is $1/2$.

The rule of Lemma 1 that associates an event with each probabilistic execution considers all those executions where either a does not occur or the first occurrence of a is followed by a state from U. The coin lemma guarrantees that all the events returned by the rule have minimum probability p. If we show that some good property holds for each execution of the events returned by the rule, then we can conclude that the good property holds with probability at least p no matter how the nondeterminism is resolved. In particular we are forced to show that either a occurs always or that the good property holds whenever a does not occur.

4.2 The New Formulation of a Coin Lemma

The old formulation of coin lemmas relies strongly on the labels associated with the transitions of a probabilistic automaton. The rule to determine whether an experiment takes place (in the case of Lemma 1 the experiment takes place at the first occurrence of a) is fixed in the formulation of the coin lemma. Thus, for each kind of experiment a new coin lemma should be formulated. Furthermore, it is difficult to formulate coin lemmas based on non-trivial methods to determine when an experiment takes place (e.g., the first a after three consecutive a's lead to a state from U).

The fact that success means reaching a fixed set of states U gives us limited flexibility as well. An example of limited flexibility can be observed in the proof of correctness of the consensus algorithm of Ben Or [1, 11]. The experiment consists of flipping a coin and the successful result depends on the past history. Sometimes it is successful to obtain head and sometimes it is successful to obtain tail. A coin lemma formulated in terms of a unique set of states U does not suffice.

Rather than formulating ad-hoc coin lemmas each time, here we propose a new formulation of coin lemmas that relies on random variables both to identify the places where an experiment takes place and the successful results of an experiment. A coin lemma does not fix any more the rule to determine where an experiment takes place; rather, it takes the rule as an argument in the form of a random variable.

An experiment either takes place or does not take place. Thus, we can define a binary function that associates 0 to those executions where the experiment does not take place and 1 to those executions where the experiment takes place. We require the binary function to be measurable, thus a random variable. If an experiment takes place in a finite execution α, then the experiment takes place in any extension of α, since in any extension of α the prefix α has occurred. If an experiment takes place in α, then we can identify the exact point at which the experiment takes place by looking at the minimum prefix of α where the experiment takes place. The important concept for us is that an experiment must take place at some finite point. Thus, if an experiment takes place in an infinite execution α, then it must take place also at some finite prefix of α. The outcome of an experiment can be captured by some other random variable whose value is observed only after the experiment takes place.

We now start to capture formally the notion of an experiment, which consists of a random variable that identifies where the experiment takes place, called an experiment detector, and another random variable that describes the outcome of the experiment. We consider a generic probabilistic execution μ and we denote by $\mathcal{F}$ the σ-field on which μ is defined.

Definition 1. *Let X be a random variable on $\mathcal{F}$. We say that X is* finitely determined *if there exists a* root function ρ_X *such that, $\rho_X(\alpha)$ is a finite prefix of α for each execution α, and whenever $X(\alpha) \neq 0$ for some execution α, $X(\alpha') = X(\alpha)$ for each execution α' such that $\rho_X(\alpha) \leq \alpha' \leq \alpha$.*

We say that the random variable X is persistent *if, whenever $X(\alpha) \neq 0$ for some execution α, that $X(\alpha') = X(\alpha)$ for each execution α' such that $\alpha \leq \alpha'$.*

In other words, the random variable X is finitely determined if, whenever its value is not 0 in an infinite execution, we can identify a finite point after which its value is fixed; the random variable X is persistent if its value does not change once it is different from 0. The root function is not unique in general. Normally we can take $\rho_X(\alpha)$ to be the minimum prefix of α that satisfies the property of Definition 1. Although we do not state it explicitly in the rest of the paper, we always assume that a finitely determined random variable is equipped with its root function.

Definition 2. *A random variable is* binary *if it takes values in the set $\{0, 1\}$. An experiment detector is a finitely determined persistent binary random variable. An* experiment *is a pair (D, U) of persistent random variables where D is an experiment detector. An experiment is said to be* finitely determined *if U is finitely determined as well. We say that an experiment is a* binary experiment *if U is binary.*

Whenever the random variables D and U are binary, we can refer to D and U as sets and we can use the classical set notation when referring to D and U.

Observe that any measurable boolean condition on any random variable U gives rise to a binary variable. Thus, it is always possible to derive binary experiments from generic experiments.

Example 6. In the die example the experiment is a pair (D_d, U_d) where $D_d(\alpha) = 1$ whenever the action of rolling the die takes place in α, and $U_d(\alpha) = 1$ if the state reached after the die is rolled corresponds to an even outcome.

In the example of Lemma 1, the experiment is the pair (D_a, U_a) where $D_a(\alpha) = 1$ if α contains an occurrence of action a, while $U_a(\alpha) = 1$ if a occurs in α and the state reached in α after the first occurrence of a is a state of the set U.

In the formulation of Lemma 1 there is an hypothesis stating that the probability of reaching a state from U in a transition labeled by a is at least p. In other words, once the experiment takes place, the probability of success must be p. We capture this idea with the notion of a p-successful experiment, which is expressed in terms of conditional distributions. The root function is used to determine the point where the experiment takes place.

Definition 3. *Given a binary experiment (D, U) and a number $p \in [0, 1]$, we say that the experiment (D, U) is* p-successful *in a probabilistic execution μ if for each execution $\alpha \in D$ it is the case that either $\mu(C_{\rho(\alpha)}) = 0$ or $\mu(U|C_{\rho(\alpha)}) \geq p$.*

The first result about p-successful experiments is that the local condition imposed in the definition implies a global condition as well on the probability of success whenever the experiment takes place.

Lemma 2. *Let $\mathcal{A}$ be a probabilistic automaton, μ be a probabilistic execution of $\mathcal{A}$, and (D, U) a p-successful experiment for μ. Then, either $\mu(D) = 0$ or $\mu(U|D) \geq p$.*

Proof. If $\mu(D) = 0$ then we are done. Otherwise, let θ be the set of minimal roots given by ρ_D, i.e., the minimal elements in the image under ρ_D of the event $\{D = 1\}$. By definition of θ, $\mu(D) = \sum_{\alpha\in\theta} \mu(C_\alpha)$. Then, $\mu(U|D) = \mu(D \cap U)/\mu(D) = \sum_{\alpha\in\theta} \mu(C_\alpha)\mu(U|C_\alpha)/\sum_{\alpha\in\theta} \mu(C_\alpha)$. By hypothesis, since the elements of θ are roots, for each $\alpha \in \theta$ we have $\mu(U|C_\alpha) \geq p$. Thus, $\mu(U|D) \geq p$.

The global condition proved in Lemma 2 leads directly to our new formulation of a coin lemma for single binary experiments.

Proposition 1. *Let $\mathcal{A}$ be a probabilistic automaton, μ be a probabilistic execution, and (D, U) a p-successful experiment for μ. Then, $\mu(\neg D \vee U) \geq p$.*

Proof. $\mu(\neg D \vee U) = \mu(\neg D) + \mu(D \cap U) = \mu(\neg D) + \mu(D)\mu(U|D) \geq \mu(\neg D) + \mu(D)p \geq \mu(\neg D)p + \mu(D)p = p$.

Example 7. Lemma 1 is just a special case of Proposition 1. To show that the experiment (D_a, U_a) is p-successful we extend the cone notation by considering cones of executions that end with an action as well. Then, the root of an execution

α that contains action a is α truncated at the first occurrence of a. The condition on the a-labelled transitions of Lemma 1 implies directly that (D_a, U_a) is p-successful. Observe that in any probabilistic execution μ the event $\neg D_a \vee U_a$ coincides with the event $FIRST(a, U)(\mu)$.

Using the structure of Proposition 1 we can formulate easily coin lemmas for the i^{th} occurrence of an action a or more elaborate coin lemmas that identify the occurrence of any generic condition. An example is the coin lemma needed in [11] for the analysis of the randomized consensus algorithm of Ben Or [1]. In this case the experiment consists of flipping a coin and the successful outcome depends on what happened before flipping the coin, which can be represented easily with the random variable U.

5 Properties of Single Experiments

There are several properties of p-successful experiments that can be studied separately and that lead to new coin lemmas. In particular, we can derive generalizations of the coin lemmas of [12] that identify the first experiment that takes place among many.

Definition 4. *A sub-experiment of an experiment (D, U) is an experiment (D', U') such that $D' \subseteq D$, $\rho_{D'} = \rho_D$, and $U \subseteq U'$.*

In a sub-experiment we perform the actual experiment in fewer places and impose fewer restrictions for its success. The condition on the root function ensures that we are not moving the places at which an experiment takes place.

Proposition 2. *Any sub-experiment of a p-successful experiment is p-successful.*

Binary experiments can be combined, leading to new binary experiments.

Proposition 3. *Let $\mathcal{A}$ be a probabilistic automaton and μ be a probabilistic execution of $\mathcal{A}$. Let (C, U) be a p-successful experiment for μ and (D, V) be a q-successful experiment for μ. Let $E = C \cup D$ and $Z = (C \cap U) \cup (D \cap V)$. Then (E, Z) is a $min(p, q)$-successful experiment for μ.*

The experiment (E, Z) of Proposition 3 is considered to be successful whenever at least one of the two experiments (C, U) and (D, V) occurs and is successful. By hypothesis we know that when one experiment takes place, then it is successful with probability either p or q. Thus, no matter what experiment takes place, the probability of success is at least $min(p, q)$.

Example 8. We can combine sub-experiments and Proposition 3 to observe the result of the first experiment among two as follows. Consider two binary experiments (C, U) and (D, V). Let $C \leq D$ denote the detector of experiment C occurring not after experiment D (experiment D may not occur at all); similarly, let $D < C$ denote the detector of experiment D occurring before experiment C.

Observe that $(C \leq D, U)$ is a sub-experiment of (C, U) and $(D < C, V)$ is a sub-experiment of (D, V). By applying Proposition 3 to $(C \leq D, U)$ and $(D < C, V)$, we obtain an experiment that is successful whenever the first experiment that take place among C, D is successful.

Thus, the probability that either no experiment takes place or that the first experiment that takes place is successful is at least $min(p, q)$. By an inductive argument the same construction can be generalized to an arbitrary finite number of experiments.

Example 9. By keeping the setting of Example 8, if we use C to detect the first occurrence of an action a, D to detect the first occurrence of an action b, U to check whether a state from some set X is reached immediately after the occurrence of a, and V to check a state from some set Y is reached immediately after the occurrence of b, then, applying Proposition 3 to $(C \leq D, U)$ and $(D < C, V)$, together with Proposition 2, we derive the coin lemmas of [12] that deal with the outcome of the first action among two. Such coin lemmas are used in the analysis of the randomized dyning philosophers algorithm of Lehman and Rabin [6, 7].

One last result about binary experiments considers the union of two experiments with disjoint successful outcomes. In this case, the probabilities of success add up as expected.

Proposition 4. *Let $\mathcal{A}$ be a probabilistic automaton and μ be a probabilistic execution of $\mathcal{A}$. Let (D, U) be a p-successful experiment for μ and (D, V) be a q-successful experiment for μ such that $U \cap V = \emptyset$. Then $(D, U \cup V)$ is a $p + q$-successful experiment.*

6 Multiple Experiments

A stochastic process consists of a collection of experiments. In this section we describe several ways of combining experiments, leading to general formulation of coin lemmas. We first need some preliminary definitions to understand when two experiments can be treated as independent.

Definition 5. *We say that two experiments (C, U) and (D, V) are* separated *if for each execution $\alpha \in C \cap D$, $\rho_C(\alpha) \neq \rho_D(\alpha)$.*

We say that two finitely determined experiments (C, U) and (D, V) are ordered *if they are separated and for each $\alpha \in C \cap D$ either*

- *$\rho_C(\alpha) < \rho_D(\alpha)$ and $U(\alpha) \neq 0$ implies $\rho_U(\alpha) < \rho_D(\alpha)$, or*
- *$\rho_D(\alpha) < \rho_C(\alpha)$ and $V(\alpha) \neq 0$ implies $\rho_V(\alpha) < \rho_C(\alpha)$.*

Informally, two experiments are separated if they take place at different points in an execution, and two experiments are ordered if one experiment is completed (the outcome is observed) before the other experiment starts. Ordering of experiments is necessary to ensure that two experiments are independent.

6.1 Multiple Binary Experiments

Several times we are interested in observing the simultaneous success of a collection of experiments. The proposition below, which generalizes the coin lemmas about conjunction of [12], shows how to deal with two experiments. The generalization to any finite number of experiments follows by an inductive argument and can be derived from Proposition 6 in the next section.

Proposition 5. *Let $\mathcal{A}$ be a probabilistic automaton and μ be a probabilistic execution of $\mathcal{A}$. Let (C, U) be a p-successful experiment for μ and (D, V) be a q-successful experiment for μ such that (C, U) and (D, V) are ordered. Then $\mu((\neg C \cup U) \cap (\neg D \cup V)) \geq pq$*

In the coin lemma above we are interested in the outcome of two experiments. The ordering condition imposes that one experiment terminates before the other experiment starts. This condition avoids cases like those where the success of an experiment coincides with the failure of the other experiment, which would invalidate the lower bound on probabilities. In other words, since the experiments are performed at different points in time and their outcome is observed independently, the two experiments are guarranteed to be independent.

6.2 Multiple General Experiments

Our main objective is to derive a formulation of a coin lemma that is as general as possible and that does not depend too much on the underlying stochastic process as well as the underlying computational model. In this section we give two coin lemmas that are based on possibly countably many experiments whose outcomes are required to satisfy some specific property identified by a binary measurable function.

The first coin lemma that we propose deals with a scenario where the experiments may occur in any order within a probabilistic execution. To accept arbitrary orderings of experiments, each occurrence of an experiment is required to respect the same probability measure.

Proposition 6. *Let $\mathcal{A}$ be a probabilistic automaton and μ be a probabilistic execution of $\mathcal{A}$. Let $\{(D_i, X_i)\}_{i\in I}$, be a finite or countable collection of pairwise ordered experiments with root functions $\{\rho_i\}_{i\in I}$. Suppose that for each $i \in I$ there exists a measure μ_i such that for each execution $\alpha \in D_i$, $\mu_i = \mu(\cdot|\rho_i(\alpha))$, and let ν be the measure $\times_{i\in I}\mu_i$, the product of the μ_i's as independent measures.*

Let X be $f(X_1, X_2, \cdots)$, where f is a binary measurable function defined on the X_i's. Let Y be a binary function such that $Y = 1$ iff there exists a collection of real numbers $\{y_i\}_{i\in I}$ such that, for each $i \in I$, either $y_i = X_i$ or $D_i = 0$, and such that $f(y_1, y_2, \cdots) = 1$.

Then, $\mu(Y) \geq \nu(X)$.

Proposition 6 considers a collection of experiments $\{(D_i, X_i)\}_{i\in I}$. These experiments are assumed to be pairwise ordered, so that there is no place where two distinct experiments take place at the same time. This ensures independence. Each time an experiment (D_i, X_i) takes place there is always the same distribution μ_i that describes X_i conditional on the specific occurrence of the experiment. The distribution ν describes the composition of the μ_i measures as independent measures.

Function f is a boolean test on the variables X_i's. Function Y checks whether function f has a chance to be successful given the outcome of the experiments that take place. In other words, if there is a possibility to fix arbitrarily the outcome of the experiments that do not take place so that f evaluates to 1, then Y evaluates to 1 as well. For this purpose, observe that the y_i's must coincide with the X_i's whenever the corresponding experiments take place.

In the coin lemma that follows the experiments are required to occur according to a predetermined order and no experiment may occur if the previous experiments have not occurred yet. In this last case we can simply look at the global outcome of an experiment without looking at each single occurrence. The main difference between the coin lemma below and Proposition 6 is in the conditions enforced on the measures μ_i's.

Proposition 7. *Let $\mathcal{A}$ be a probabilistic automaton and μ be a probabilistic execution of $\mathcal{A}$. Let $\{(D_i, X_i)\}_{i\in I}$, be a finite or countable collection of pairwise ordered experiments such that $D_i \leq D_j$ whenever $i > j$. For each $i \in I$ let μ_i be the measure $\mu(\cdot|D_i)$, i.e., the measure of X_i conditional on D_i. Let ν be the measure $\times_{i\in I}\mu_i$.*

Let X be $f(X_1, X_2, \cdots)$, where f is a binary measurable function defined on the X_i's. Let Y be a binary function such that $Y = 1$ iff there exists a collection of real numbers $\{y_i\}_{i\in I}$ such that, for each $i \in I$, either $y_i = X_i$ or $I_i = 0$, and such that $f(y_1, y_2, \cdots) = 1$.

Then, $\mu(Y) \geq \nu(X)$.

A simple consequence of Proposition 7 is that, for an experiment (D, X) and a real number q, $\mu(\neg I \vee (X \geq q)) \geq \mu(X \geq q|D)$. In particular, if $\mu(X \geq q|D) \geq p$, then $\mu(\neg D \vee (X \geq q)) \geq p$. The result is obtained by considering a function f that checks whether $X \geq q$.

Example 10. Consider a finite collection $(D_1, X_1), \cdots (D_k, X_k)$ of pairwise separated binary experiments, and suppose that each experiment (D_i, X_i) is p_i-successful for some probability p_i. Let f be the minimum of the X_i's. Then the random variable Y of Proposition 6 identifies all the executions of the set $(\neg D_1 \cup X_1) \cap \cdots \cap (\neg D_k \cup X_k)$. Furthermore, the lower bound given by Proposition 6 is $p_1 \cdots p_k$, thus proving the generalization of Proposition 5.

Example 11. If we consider a probabilistic automaton where a possibly infinite sequence of coin flips occurs and we identify each experiment detector D_i with the i^{th} coin flip and X_i with the i^{th} outcome, then Proposition 6 gives us the coin lemma for random walks that appears in [8].

7 Concluding Remarks

We have reformulated coin lemmas in terms of random variables and conditional distributions, thus generalizing all known coin lemmas and making them independent of the specific model of probabilistic automata. An important step is the formulation of the notion of p-successful experiments, which are an abstraction of the building blocks of the coin lemmas of [12]. The new formulation of coin lemmas highlights more precisely the fact that, in order for a system to function correctly with high probability, the adversarial scheduler should not be able to gain any advantage by avoiding to schedule any random draws.

The next step in the generalization of coin lemmas is to understand better what happens when we deal with expectations. So far the only result about expectations appears in [8], where an upper bound on the expected termination time for a random walk with barriers is studied within a framework with nondeterminism. The formulation of the coin lemma in [8] is specific to the problem under examination and it is not clear yet how to derive such bounds in general.

References

1. M. Ben-Or. Another advantage of free choice: completely asynchronous agreement protocols. In *Proceedings of the* 2^{nd} *Annual ACM Symposium on Principles of Distributed Computing*, Montreal, Quebec, Canada, August 1983.
2. C. Derman. *Finite State Markovian Decision Processes.* Academic Press, 1970.
3. H. Hansson. *Time and Probability in Formal Design of Distributed Systems*, volume 1 of *Real-Time Safety Critical Systems.* Elsevier, 1994.
4. J.G. Kemeny, J.L. Snell, and A.W. Knapp. *Denumerable Markov Chains.* Graduate Texts in Mathematics. Springer-Verlag, second edition, 1976.
5. K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. In *Conference Record of the* 16^{th} *ACM Symposium on Principles of Programming Languages*, Austin, Texas, pages 344–352, 1989.
6. D. Lehmann and M. Rabin. On the advantage of free choice: a symmetric and fully distributed solution to the dining philosophers problem. In *Proceedings of the* 8^{th} *Annual ACM Symposium on Principles of Programming Languages*, pages 133–138, January 1981.
7. A. Pogosyants and R. Segala. Formal verification of timed properties of randomized distributed algorithms. In *Proceedings of the* 14^{th} *Annual ACM Symposium on Principles of Distributed Computing*, pages 174–183, August 1995.
8. A. Pogosyants, R. Segala, and N. Lynch. Verification of the randomized consensus algorithm of Aspnes and Herlihy: a case study. *Distributed Computing*, 13:155–186, July 2000.
9. M.O. Rabin. Probabilistic automata. *Information and Control*, 6:230–245, 1963.
10. I. Saias. Proving probabilistic correctness: the case of Rabin's algorithm for mutual exclusion. In *Proceedings of the* 11^{th} *Annual ACM Symposium on Principles of Distributed Computing*, Quebec, Canada, August 1992.
11. R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, MIT, Dept. of Electrical Engineering and Computer Science, 1995. Also appears as technical report MIT/LCS/TR-676.
12. R. Segala. The essence of coin lemmas. In *Proceedings of PROBMIV'98*, volume 22 of *Electronic Notes in Theoretical Computer Science*, 1999.